

Schlussbericht

zu IGF-Vorhaben Nr. 18354N

Thema

Potential, Struktur und Schnittstellen von Cloud-basierten industriellen Steuerungsdiensten
(Cloud-based Industrial Control Services - CICS)

Berichtszeitraum

01.10.2014 - 30.09.2017

Forschungsvereinigung

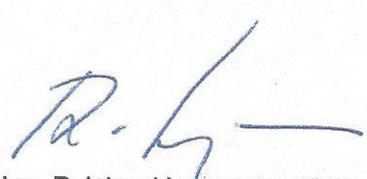
Eletrotechnik beim ZVEI e.V.

Forschungsstellen

Hochschule Düsseldorf, Düsseldorfer Teelabor (FST 1)

Universität Augsburg, Institut für Informatik, Lehrstuhl für Kommunikationssysteme (FST 2)

Düsseldorf, den 29.01.18


Prof. Dr.-Ing. Reinhard Langmann (FST 1)

München, den 29.1.18
Ort, Datum


Michael Stiller (FST 2)

Name und Unterschrift aller Projektleiter der Forschungsstelle(n)



Inhaltsverzeichnis:

1	Zusammenfassung.....	4
2	Problemstellung/Motivation	5
3	Durchgeführte Arbeiten und Ergebnisse, Gegenüberstellung mit den Zielen.....	7
3.1	Arbeitspaket 1 (FST 1 + FST 2).....	7
3.2	Arbeitspaket 2 (FST 1)	15
3.3	Arbeitspaket 3 (FST 1 + FST 2).....	20
3.4	Arbeitspaket 4 (FST 2)	29
3.5	Arbeitspaket 5 (FST 1)	37
3.6	Arbeitspaket 6 (FST 1)	39
3.7	Arbeitspaket 7 (FST 2)	44
3.8	Arbeitspaket 8 (FST 1)	47
3.9	Arbeitspaket 9 (FST 1 + FST 2).....	52
4	Verwendung der Zuwendung	55
5	Notwendigkeit und Angemessenheit der geleisteten Arbeit	57
6	Darstellung des wissenschaftlich-technischen und wirtschaftlichen Nutzens der erzielten Ergebnisse insbesondere für KMU	59
7	Plan zum Ergebnistransfer in die Wirtschaft	61
8	Zusammenstellung aller Arbeiten, die im Zusammenhang mit dem Vorhaben veröffentlicht wurden	65
9	Einschätzung zur Realisierbarkeit des vorgeschlagenen und aktualisierten Transferkonzepts.....	67

1 Zusammenfassung

Zielstellung des Forschungsvorhabens Cloud-based Industrial Control Services (CICS) war die Erforschung einer Referenzarchitektur einschließlich zugehöriger Schnittstellen auf Basis von Webtechnologien, die für eine Realisierung von industriellen Steuerungsprogrammen nach IEC 61131 als Steuerungsdienste geeignet ist. Die Steuerungsdienste können verfügbare öffentliche und/oder private Cloud-Strukturen für Management und Ausführung der CICS nach dem „Serverless Computing“-Prinzip nutzen. Als eine ergänzende Variante sollte auch die Ausführung der Steuerungsdienste auf Webclients (Webbrowser) möglich sein.

Im CICS-Projekt wurden die Rahmenbedingungen und Anforderungen ermittelt und analysiert, um CICS in automatisierungstechnischen Anwendungen einsetzen zu können. Dazu gehört u.a. eine Studie zum Anwendungs- und Nutzungspotential sowie die Bewertung neuer Webtechnologien für die Anwendung in einem CICS-System.

Schwerpunkt im Projekt war die Abbildung sowie der Transfer eines klassischen Steuerungssystems (IEC 61131) mittels Webtechnologien in eine Cloud-Struktur einschließlich der Definition zugehöriger schneller und zuverlässiger Prozessdaten-Schnittstellen (WebConnector).

Betrachtet man die historische Entwicklung der industriellen Steuerungstechnik ausgehend von der klassischen Hardware-SPS hin zur PC-basierten SoftSPS, so wurde mit dem CICS-Vorhaben die nächste Stufe der SPS-Entwicklung als im IP-Netz global verteilte Software-Instanzen (Steuerungsdienste) forciert.

Die Forschungsergebnisse ermöglichen u.a. die Entwicklung neuer, durchgängig weborientierter Steuerungssysteme, die sich extrem flexibel und mit wenig Aufwand an anwendungsspezifische Bedürfnisse anpassen lassen. Damit leistet CICS einen Beitrag für adaptive und wandlungsfähige Steuerungssysteme. Bisherige hardware-fixierte und zentralisierte Verfahren zur Steuerung automatisierter Geräte, Maschinen und Anlagen (z.B. SPS-Steuerungen) können für unkritische Echtzeitbedingungen (z.B. Umweltprozesse, Temperatur, Klima) durch vollständig im IP-Netz verteilte Software-Funktionen ersetzt werden. Damit ergeben sich u.a. folgende Vorteile für die Anwender:

- Kosteneinsparungen bei Inbetriebnahme und Betrieb durch ortsunabhängigen und Cloud-basierten Service.
- Flexible Bereitstellung von Steuerungsdiensten für wandlungsfähige Produktion.
- Verbessertes Herstellerservice (zentrales Backup, Reservesystem im Fehlerfall).
- Schaffung von neuen Miet-Geschäftsmodellen für Control as a Service.
- Darstellung von Informationen sind besser an den Benutzer anpassbar.
- Unterstützung von Simulation für bessere Planung und Prozessoptimierung.
- Produktionsoptimierung durch bedarfsgerechte Nutzung effizienter Algorithmen (App-Konzept).
- Systemschnittstellen innerhalb von Steuerungssystemen reduzieren sich erheblich und existieren nur noch als Softwareschnittstellen im IP-Netz.

Mit dem CICS-Konzept wurde ein neuer Typus einer Industriesteuerung entwickelt und erprobt, mit dem die vollständige Loslösung von Steuerfunktion und zugehöriger Anlage zu global verteilten, cloud-basierten Software-Steuerungsdiensten erfolgt. Eine CICS-Steuerung wird durch ein klassisches IEC61131-3-Steuerungsprogramm betrieben und sichert damit die Interoperabilität und Industrietauglichkeit der Steuerung. Die Anwendung des

Service-Paradigmas auf industrielle Steuerungsfunktionen steigert deutlich die Flexibilität, erfüllt Industrie 4.0-Anforderungen wie Wandelbarkeit, Rekonfiguration sowie Autonomie und ermöglicht neue Geschäftsmodelle zur Vermietung von Automatisierungsfunktionen.

Das Forschungsziel wurde in vollem Umfang erreicht.

2 Problemstellung/Motivation

Industrielle Steuerungen und dabei insbesondere SPS-Steuerungen bilden gegenwärtig eine wesentliche gerätetechnische Basis für die Automatisierung von industriellen Prozessen. Auch im Zeitalter von Industrie 4.0 (I40) und Industrial Internet kann davon ausgegangen werden, dass diese Steuerungen für die Produktionsautomatisierung von Morgen auch weiterhin benötigt werden. Die Steuerungen müssen aber zukünftig eine Reihe von zusätzlichen Anforderungen erfüllen, die aus den neuen Produktionsbedingungen resultieren.

Durch den zunehmenden Einfluss der IKT¹ und insbesondere der Vernetzung aller Komponenten eines Automatisierungssystems entwickelte sich in der Fachwelt die Auffassung, dass sich die klassische Automatisierungspyramide zunehmend in eine „flache“ Automatisierungs-„wolke“ auflöst. Bild 1 veranschaulicht dazu diesen Trend anhand eines Bildes aus dem Positionspapier der VDI/VDE-GMA von 2013 zu Cyber Physical Systems (CPS). Es besteht dazu die Auffassung, dass sich mittelfristig, beginnend in den höheren Ebenen der Automatisierungspyramide (prozessferne Ebene), zunehmend Funktionalität in eine vernetzte Cloud verlagern wird. Dies wird zuerst Automatisierungsfunktionen mit unkritischem Zeitverhalten betreffen. Es wird aber auch nicht ausgeschlossen, dass zukünftig Funktionen aus der prozessnahen Ebene unter Berücksichtigung der Reaktionszeit des zu automatisierenden technischen Prozesses in einer Cloud realisiert werden können.

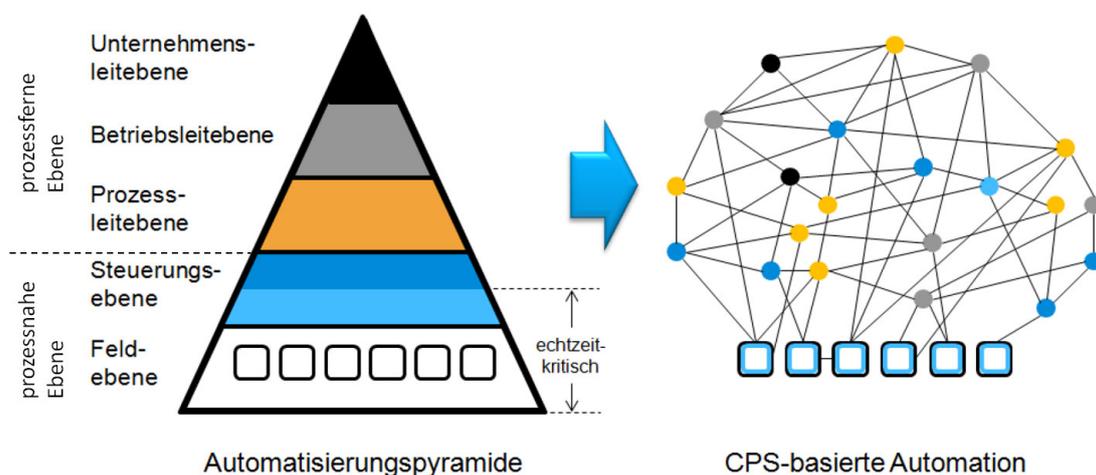


Bild 1: Auflösung der hierarchischen Automatisierungspyramide durch CPS mit verteilten Diensten

Bei Anwendung der Industrie 4.0-Prinzipien ergeben sich hochgradig vernetzte Produktionssysteme auf Basis von Cyber Physical Systems (CPS), die auch als Cyber Physical Production Systems (CPPS) bezeichnet werden. An die in diesen Systemen eingesetzten zukünftigen Steuerungen werden eine Reihe von I40-Anforderungen gestellt. Dazu gehören u.a.:

- Autonomie, Rekonfigurierbarkeit und Agilität (Plug&Work),

¹ IKT – Informations- und Kommunikationstechnologie

- Vernetzung in lokalen und globalen Netzwerken,
- Interoperabilität zwischen heterogenen Steuerungssystemen und unterschiedlichen Cloudlösungen,
- Abhängigkeiten sollen zur Laufzeit dynamisch veränderbar sein,
- Orchestrierung heterogener Steuerungen,
- Einführung des Dienst-Paradigmas in die Produktionsautomatisierung (Produktionsdienste).

Heutige SPS-Steuerungen können diese Anforderungen mehrheitlich noch nicht bzw. nur rudimentär oder mit extrem hohem Aufwand erfüllen. Eine der Basiseigenschaften zukünftiger SPS-Steuerungen ist dabei die effiziente Vernetzung in einem zumindest teilweise globalen Netzwerk sowie die Nutzung von Dienstprinzipien. Als globales Netzwerk fungiert dabei das IP-Netz (IP – Internet Protocol) in der Ausführung als Intranet oder Internet mit allen zugehörigen standardisierten IKT-Technologien.

Aus dem Stand der Technik zum Start des CICS-Projekts ließ sich zusammenfassend einschätzen, dass es verschiedene Lösungen und Bemühungen gab, SPS-Steuerungen mit zusätzlichen Funktionen auszurüsten, um die Steuerungen in einem IP-Netz im Sinne von Industrie 4.0 nutzen zu können. Die bekannten Arbeiten setzten dazu bereits Webtechnologien partiell, herstellerspezifisch und/oder begrenzt offen ein und nutzen dabei zunehmend auch das Dienstprinzip sowie Cloud-Strukturen als neues Paradigma für die Realisierung von Steuerungsfunktionen. Es gab aber noch Defizite, aus denen ein entsprechender Forschungsbedarf für das CICS-Projekt resultierte:

- Webtechnologien wurden zwar eingesetzt, eine flexible Verteilung von Struktur und Funktion der Steuerungsfunktionalität wurde aber nicht genutzt. Die Informationskapselung von industriellen Steuerungsprogrammen in lokalen oder virtualisierten Geräten (SPS-Steuerungen) wurde nicht in Frage gestellt.
- Für webbasierte Steuerungsdienste unter Nutzung von Cloud-Technologien als ein wesentliches Merkmal einer zukünftigen vernetzten Industrie fehlten systematische Untersuchungen, Architekturen, Schnittstellen und Testlabors.
- Verfügbare Standardtechnologien aus der Welt der IP-Netze für die Erhöhung von Flexibilität und Effizienz wurden in der Steuerungsebene nicht bzw. nur unzureichend genutzt.

Der Forschungsbedarf war weiterhin nach den Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 begründet, nach denen mittel- und langfristig in der Forschung u.a. folgende Themen für die Produktion von Morgen für Wesentlich gehalten werden:

- verstärkte Nutzung des Dienstprinzips,
- Sensor-Cloud und Cloud-Computing,
- Prozesssteuerung in der Cloud,
- Dezentrale Führungs- und Steuerungsformen,
- Online-Anpassung und Selbstkonfiguration von Anwendungen,
- Zusammenführung bisher heterogener Systemstrukturen.

3 Durchgeführte Arbeiten und Ergebnisse, Gegenüberstellung mit den Zielen

In diesem Abschnitt werden die wichtigsten Forschungsergebnisse aufgeführt. Die detaillierten Ergebnisse sowie Untersuchungen sind in den aufgelisteten Berichten niedergelegt. Die Berichte können bei Bedarf von den Forschungsstellen abgefordert werden.

3.1 Arbeitspaket 1 (FST 1 + FST 2)

Bewertung der Anforderungen und Eigenschaften klassischer Steuerungssysteme in Gegenüberstellung zu potentiellen Lösungen auf Basis von Webtechnologien

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 1.1: Herausarbeitung der Anforderungen an klassische Steuerungssysteme unter besonderer Berücksichtigung von Zeitverhalten, Deterministik, Zuverlässigkeit und Sicherheit sowie der Zuordnung zu Anwendungsklassen, -typen und -branchen (CICS-Forderungskatalog).

AP 1.2: Durchführung einer speziellen Recherche (einschl. Klärung der Patentsituation), zur Cloud-Thematik in Verbindung mit Automatisierungs- und Steuerungstechnik. Welches Nutzungspotential lässt sich aus vorhandenen Veröffentlichungen erkennen und mit welchen Umsetzungszeiträumen wird gerechnet, welche Probleme sind noch zu lösen (CICS-Potentialstudie).

AP 1.3: Herausarbeitung der Eigenschaften unzuverlässiger bzw. nicht deterministischer Kommunikationstechnologien (Internet, lokale drahtlose Kommunikation) für den Einsatz in heterogenen Kommunikationsinfrastrukturen.

AP 1.4: Herausarbeitung der potentiellen Eigenschaften von CICS, strukturiert nach Anwendungs- und Aufgabenklassen und Spiegelung dieser Eigenschaften an den Anforderungen des Zukunftsprojekts Industrie 4.0 insbesondere hinsichtlich des Forschungsbedarfs sowie der zu erwartenden Handlungsfelder (CICS-Industrie.4.0-Beitrag). Erstellung eines Ergebnisberichts für AP 1.

Die Forschungsstelle 1 war für die AP 1.1, AP 1.2 und AP 1.4 verantwortlich; die Forschungsstelle 2 für das AP 1.3.

▪ Durchgeführte Arbeiten

FST 1:

Auf Basis einer umfangreichen Literatur- und Patentrecherche wurden die potentiellen Eigenschaften von CICS herausgearbeitet. Dabei erfolgte eine Strukturierung nach Anwendungs- und Aufgabenklassen und Spiegelung dieser Eigenschaften an den Anforderungen des Zukunftsprojekts Industrie 4.0 (I40) insbesondere hinsichtlich des Forschungsbedarfs sowie der zu erwartenden Handlungsfelder (CICS-Industrie.4.0-Beitrag). Durch die FST 1 wurden insbesondere untersucht:

- Anforderungen an klassische Steuerungssysteme
- Potentielle Eigenschaften von CICS

FST 2:

Auf Basis einer Literaturrecherche und aufgrund der Kenntnisse bisher bearbeiteter Projekte wurden die Eigenschaften nicht deterministischer Kommunikationstechnologien für den Einsatz in heterogenen Kommunikationsinfrastrukturen herausgearbeitet. Dabei wurden vor allem lokale Netze (drahtgebundene als auch drahtlose) untersucht, die für den Einsatz in künftigen Industrie 4.0 Netzwerken von Bedeutung sind.

- Erzielte Ergebnisse

FST 1:

Als besondere Schwerpunkte für zukünftige Industrie 4.0-Steuerungen lassen sich erkennen:

1. Einführung des Dienst-Paradigmas in die Steuerungstechnik.
2. Schaffung von Voraussetzungen zur Optimierung/Anpassung und Selbstorganisation von Steuerungssystemen und -algorithmen.

Schwerpunkt 2 ist eng verbunden mit der fundamentalen Forschungs Herausforderung zu einem neuen Verständnis von Lokalität in vernetzten CPS-Systemen. Bisherige lokale Informationskapselung von z.B. SPS-Steuerungen wird aufgelöst und verteilt in Netze verlagert. Nur durch diese Auflösung der bisherigen Lokalität auf physischer Ebene sind der Einsatz von Optimierungs- und Selbstanpassungsalgorithmen in der neuen Netz-basierten Lokalität (verteilte und dienst-basierte Steuerungsalgorithmen) möglich.

Als I40-Fähigkeiten wurden basierend auf den beiden o.a. Schwerpunkten die beiden Eigenschaften *Dienstfähigkeit* (Service Ability) und *Steuerungslokalität* (Control Locality) eingeführt:

- Unter *Control Locality* wird für ein Steuerungssystem die Fähigkeit verstanden werden, die Steuerungsalgorithmen (Steuerungsprogramme) von der Steuerungsplattform (Runtime) zu separieren und diese auf räumlich und physisch andere Systeme zu verteilen. Als Steuerungsplattform kann dabei ein physikalisches Gerät (z.B. Hardware-SPS), eine Software-Plattform (z.B. Soft-SPS) oder auch eine virtualisierte Steuerungs-Plattform (z.B. SPS als Platform as a Service) fungieren.
- Unter *Service Ability* wird für ein Steuerungssystem die Fähigkeit verstanden werden, die Steuerungsalgorithmen (Steuerungsprogramme) als Dienste aus einer IT-Struktur (Netz, Cloud etc.) zu nutzen. In Anlehnung an das aus der IT-Welt bekannte Cloud-Nutzungsmodell Software as a Service (SaaS) ergibt sich damit bezogen auf Steuerungen das Nutzungsmodell Control as a Service (CaaS). CaaS können durch Dritte (Provider) angeboten werden.

Auf Basis der o.a. Eigenschaften wurde eine *CICS-Metrik* entwickelt (Tabelle 1) und davon Steuerungsklassen Cxx abgeleitet. So sind z.B. bei einer C33-Steuerung alle Steuerungsprogramme (grundlegender Programmteil, Programmteil für Verwaltungs-

funktionen, kritischer Programmteil hinsichtlich Echtzeit und Sicherheit) aus der Steuerung in ein IP-Netz (Cloud) verlagert und als Dienste verfügbar.

Tabelle 1: CICS-Metrik

Class	Service Ability	Control Locality
0	Keine Dienste.	Alle Steuerungsprogramme sind lokal in der Hardware gekapselt.
1	Dienste nur für unkritische Funktionalitäten.	Einige Steuerungsprogramme befinden sich verteilt auf anderen Systemen (z.B. im Netz).
2	Dienste für die meisten Funktionen verfügbar.	Die meisten Steuerungsprogramme befinden sich verteilt im Netz.
3	Alle Steuerungsprogramme als Dienste.	Alle Steuerungsprogramme befinden sich verteilt im Netz.

Aufbauend auf der CICS-Metrik wurden potentielle Eigenschaften exemplarischer CICS-Steuerungen gegenüber klassischen SPS-Steuerungen aufgezeigt. Bild 2 zeigt dazu die Bewertung der Eigenschaften von C3x-Steuerungen bezüglich der Anforderungen an eine Industriesteuerung.

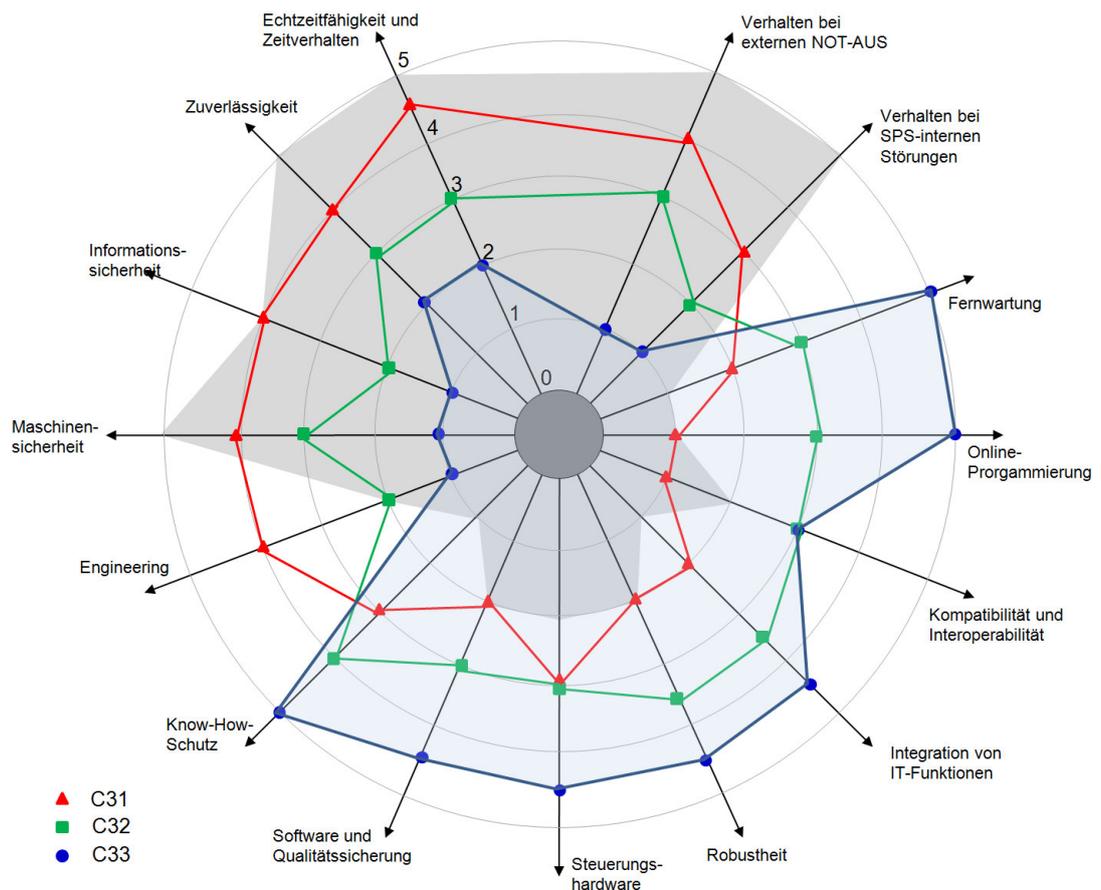


Bild 2: Bewertung der Eigenschaften von C3x-Steuerungen bezüglich der Anforderungen an eine Industriesteuerung

Die dunkelgrau markierte Fläche in Bild 2 charakterisiert eine klassische SPS. Mit zunehmender Verlagerung der Steuerungsfunktionalität als Dienste in eine Cloud wird deutlich, dass genau die Anforderungen, deren Realisierung für eine klassische SPS einen hohen Aufwand erfordern bzw. nur schwer zu realisieren sind (z.B. Integration von IT-Funktionen, Online-Programmierung), für eine CICS-Steuerung einfach erfüllt werden können. Der Aufwand verlagert sich dann zu den Anforderungen, die bei einer klassischen SPS aufgrund der Hardware-Struktur unproblematisch erfüllt werden können (z.B. Zeitverhalten, Sicherheitsprobleme).

Die besonderen Herausforderungen für eine CICS-Steuerung insbesondere der Klasse C33 als Schwerpunkt im CICS-Projekt lagen deshalb darin, die aus der SPS bekannten, eher Hardware-bezogenen und konventionellen Anforderungen, möglichst in ähnlicher Qualität zu erfüllen. Es galt dabei Methoden und Strukturen zu erforschen, die trotz z.T. unzuverlässiger Teilkomponenten (z.B. IP-Netz) ein zuverlässiges und belastbares CICS-Steuerungssystem ermöglicht

FST 2:

Bei zukünftigen Industrie 4.0 Steuerungen kann nicht davon ausgegangen werden dass das Kommunikationsnetz zwischen den Steuerungen und den zugeordneten Sensoren und Aktoren homogen bzw. ausschließlich aus einer Kommunikationstechnologie besteht. Die allg. Herausforderung liegt also darin einen möglichst robusten Ende zu Ende Kommunikationskanal bereitzustellen, auch wenn eine Verbindung aus einer Aneinanderreihung von (unterschiedlich) deterministischen und nicht deterministischen Teilnetzen besteht. Heterogene Netze, die z.B. aus den busbasierten Netzen innerhalb der Anlage und dem nicht-deterministischen Internet bestehen, besitzen eine Kombination der jeweiligen Eigenschaften. Dazu ist zunächst die Topologie der zugrundeliegenden Teilnetze zu betrachten. Diese besitzen jeweils, wie in Tabelle 2 dargestellt, strukturbedingt spezifische Vor- und Nachteile. Von den genannten Topologien existieren darüber hinaus hybride Derivate, welche meist auf einer Baumtopologie (für die globale Vernetzung) und einer anderen Struktur, z.B. P2P-Topologie, innerhalb der lokalen Vernetzung, basieren.

Tabelle 2: Topologievergleich

	Vorteile	Nachteile
P2P-Topologie	<ul style="list-style-type: none"> • Hohe Übertragungsrate • Störungssicher • Abhörsicher • Vorhersagbare, nutzbare Übertragungskapazität • Leicht erweiterbar • Leicht verständlich • Leichte Fehlersuche • Kein Routing benötigt 	<ul style="list-style-type: none"> • Eintragung jedes Nutzers auf jedem Rechner • keine zentrale Verwaltungsmöglichkeit • Im Worst Case vollvermaschtes Netz
Ring-Topologie	<ul style="list-style-type: none"> • Deterministische Rechnernetz-kommunikation ohne Paketkollisionen, da Vorgänger und Nachfolger definiert sind 	<ul style="list-style-type: none"> • Ausfall eines Endgerätes führt dazu, dass die gesamte Netzkommunikation unterbrochen wird • Hohe Latenzen zu entfernten

	<ul style="list-style-type: none"> • Alle Stationen arbeiten als Verstärker • Alle Rechner haben gleiche Zugriffsmöglichkeiten • Garantierte Übertragungsbandbreite • Sehr gute Skalierung 	<p>Knoten</p> <ul style="list-style-type: none"> • Datenübertragungen können leicht abgehört werden.
Stern-Topologie	<ul style="list-style-type: none"> • Ausfall eines Endgerätes hat keine Auswirkung auf den Rest des Netzes • hohe Übertragungsraten, wenn der Netzknoten ein Switch ist • Leicht erweiterbar • Leichte Fehlersuche • Sehr gute Eignung für Multicast-/Broadcastanwendungen 	<ul style="list-style-type: none"> • Durch Ausfall des Verteilers wird Netzverkehr unmöglich
Baum-Topologie	<ul style="list-style-type: none"> • Ausfall eines Endgeräts hat keine Konsequenzen • Strukturelle Erweiterbarkeit • Große Entfernungen realisierbar • Gute Eignung für Such- und Sortieralgorithmen 	<ul style="list-style-type: none"> • Bei (nicht behandelbaren) Ausfall eines Verteilers (Wurzel) ist der ausgehende Baum nicht mehr erreichbar • Etwaige Engpässe an den Wurzeln
Bus-Topologie	<ul style="list-style-type: none"> • Ausfall eines Gerätes hat für die Funktionalität des Netzwerkes keine Konsequenzen • Einfache Netzerweiterung • Keine aktiven Netzwerkkomponenten benötigt • Determinismus 	<ul style="list-style-type: none"> • Datenübertragungen können leicht abgehört werden • Störung des blockiert das gesamte Netzwerk • Zu jedem Zeitpunkt immer nur eine Station Daten senden. Währenddessen sind alle anderen Sender blockiert (müssen zu übertragende Daten intern zwischenpuffern). • Bei Bussen mit erlaubten Kollisionen erhöhte Bandbreite

Wird das lokale Kommunikationssystem der Maschine mit dem Internet und dahinterliegenden Cloud Diensten verbunden ergibt sich eine Topologie wie in Bild 3 dargestellt. Innerhalb von CICS ist eine Kopplung einer Baum-/Sternstruktur, wie der des globalen Internet und mit einem lokalen Netzwerk, z.B. mit einer Bus-Topologie der Automatisierungsanlage nötig. Dazu wird eine weitere Komponente (Netzwerkbrücke) benötigt, welche wiederum einer eigenen Topologie besitzen kann.

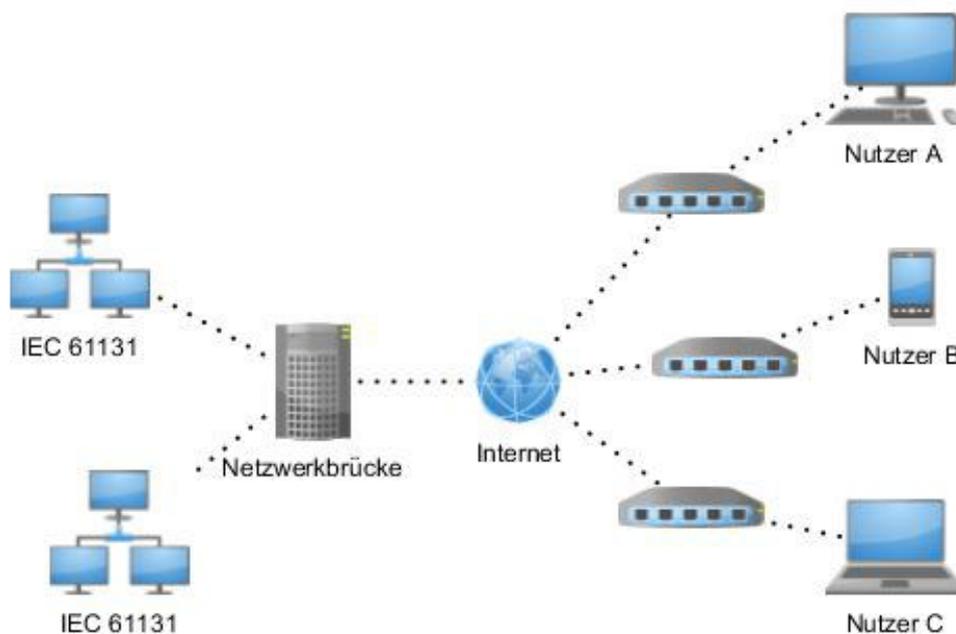


Bild 3: Kommunikationstopologie bei CICS

Neben den strukturellen Eigenschaften werden (heterogene) Kommunikationstechnologien anhand ihrer temporalen Eigenschaften auf dem Kanal bewertet. In heterogenen Netzwerken können dafür die Parameter jeweils für die Einzelnetze und das Gesamtnetz bestimmt werden. Dies umfasst die Parameter (Quelle/Ziel)-Bandbreite, die Verzögerungszeit, den Jitter und die Paketverlustrate. So zeigt Bild 4 ein beispielhaftes Netzwerk in Baumtopologie, welches die Verzögerungszeiten, Durchsatz und den Jitter für Einzelnetze (zwischen Host und Switch) und dem Gesamtnetz (Host A zu Host B) darstellt. Diese vier Parameter wurden in den Folgearbeitspaketen für verschiedene Protokoll- und Netzwerkstrukturen evaluiert und auf Basis dessen die geeignetste (heterogene) Kombination aus Netzwerktopologie und -kanal abgeleitet.

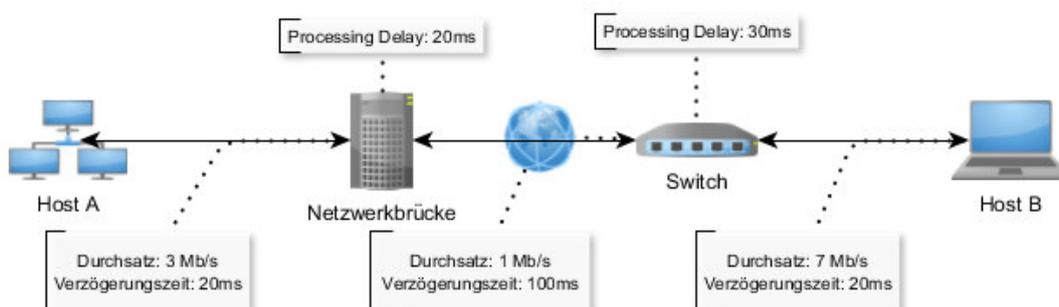


Bild 4: Beispielhaftes Netzwerk in Baumtopologie

Aufgrund der zunehmenden Akzeptanz drahtloser Netze im industriellen Umfeld wurde im Rahmen des Arbeitspaketes 1 auch die Heterogenität in diesen Netzwerken untersucht. Drahtlose Netze lassen sich in zwei Kategorien aufteilen:

1. **Infrastrukturbasierte drahtlose Netze:**

Infrastrukturbasierte drahtlose Netzwerke bestehen aus einem Backbone, der das drahtlose Zugangsnetz kontrolliert. Beispiele für drahtlose Zugangsnetze: GPRS, UMTS, LTE, 5G, WLAN, PAN

2. **Infrastrukturlose drahtlose Netze:**

Infrastrukturlose drahtlose Netzwerke haben keinen Backbone, der die Verbindung mit dem Endteilnehmer koordiniert. Daher spricht man auch von selbstorganisierenden Netzen. Beispiele sind hier MANETs und WSNs. Mobile Ad-hoc Networks (MANETs) werden von Geräten genutzt, die vor allem mit Menschen interagieren z.B. ad-hoc Netzwerke zwischen Fahrzeugen. Wireless Sensor Networks (WSNs) werden von Geräten verwendet, die mit ihrer Umgebung interagieren. WSN Knoten in einem Netzwerk haben Sensor und Aktor Fähigkeiten um die Umgebung zu messen oder zu beeinflussen. Beispiele sind hier Bluetooth oder ZigBee.

Drahtlose Netzwerke werden in „open loop“ (weiche Echtzeit) als auch zunehmend in „closed loop“ (Harte Echtzeit) Steuerungsanwendungen eingesetzt. Die größten Herausforderungen für aktuelle drahtlose Systeme in „closed loop“ Steuerungsanwendungen sind Latenz, Synchronisation und Zuverlässigkeit. In der Automatisierungsindustrie kommen WLAN und Bluetooth konforme standardisierte Übertragungskomponenten zum Einsatz, wobei teilweise zusätzliche proprietäre Protokollerweiterungen zum Einsatz kommen.

Zum Beispiel werden in der Fertigungsindustrie IWLAN, WISA, WSA-FA und in der Prozessindustrie Wireless HART oder ISA 100.11 verwendet. In manchen Industrieanwendungen wird auch Digital Enhanced Cordless Technologie (DECT) verwendet. In einigen Anwendungen (z.B. Transport, Produkttracking) kommt auch RFID zum Einsatz. Mobilkommunikation wird bis heute in industriellen Anwendungen nur für den Fernzugriff oder für Alarmierungssysteme eingesetzt. Narrow Band IoT (NB-IoT) und 5G sollen künftig aber auch innerhalb des Produktionsprozesses zum Einsatz kommen.

in Tabelle 3 wird ein Vergleich einiger drahtloser Technologien, die in der Fertigungsautomatisierung für „closed loop“ Steuerungsanwendungen verwendet werden, aufgezeigt. Es ist zu berücksichtigen, dass bei drahtlosen Systemen einzelne Werte nicht garantiert werden können: Schätzwerte in Klammern.

Tabelle 3: Vergleich von drahtlosen Technologien für die Fertigungsautomatisierung

Name	WLAN	Bluetooth	WSAN-FA & WISA
Max. System data rate	600 Mbps	3Mbps	1Mbps
Network topology	star	star	star
Nodes per network	(50)	7	120
Min.cycle time T_{cyc}	(100ms)	8.75ms	10ms

In drahtlosen Netzen müssen verschiedene Zugangstechnologien miteinander interoperieren:

- vertikales und horizontales Handover:
 - vertikales Handover tritt bei zwei unterschiedlichen drahtlosen Zugangstechnologien auf (z.B. Wechsel zw. 2G und 3G Zugangstechnologien),
 - horizontales Handover tritt innerhalb des selben Netzwerks zwischen verschiedenen Zellen auf (z.B. 2G zu 2G).
- Das Endgerät muß alle benötigten drahtlosen Zugangstechnologien beherrschen (z.B. GSM(TDMA, OFDM), 3G(WCDMA), 4G (OFDM)).

Für heterogene drahtlose Netzwerke ergeben sich aus den obigen Betrachtungen folgende Herausforderungen:

- Mobility Management
 - Da heterogene Netzwerke aus unterschiedlichen Zugangstechnologien bestehen können, kann es sein, dass unterschiedliche IP Versionen, unterschiedliche Multiplexing Verfahren und eine unterschiedliche Zahl an Routersprüngen aufeinandertreffen. Dies kann für eine Ende zu Ende Verbindung zu erheblichen Latenzen und Paketverlusten führen.
 - Mobile IP nimmt für alle Komponenten eine eindeutige routbare IP Adresse an. Aber wenn einige Netzwerkkomponenten NAT nutzen, ist deren IP mit der mobilen IP nicht identisch und eine Verbindung schlägt fehl.
 - Ein vertikales Handover für eine nahtlose Kommunikation ist immer kritisch, weswegen auch Vorhersagen zur Qualität einer solchen Verbindung nur schwer zu treffen sind.
- Herausforderung Ende zu Ende QoS
 - Wie in Tabelle 4 gezeigt, haben unterschiedliche drahtlose Zugangstechnologien unterschiedliche Latenzen. Diese Latenzen vergrößern sich durch Handover Ereignisse oder wechselnden Signalstärken. Daraus ergibt sich für LTE advanced insgesamt eine Round Trip Zeit zw. 70 und 100 ms.

Tabelle 4: Latenz in Zellulären Netzwerken

Network Latency	HSPA+	LTE	LTE Advanced
Idle to connected (ms)	<100	<100	<50
Dormant to active (ms)	<50	<50	<10
User plane one way latency (ms)	<10	<10	<5

- *Intra-/Inter Domänen Routing Management* bezogene Probleme: wurde bereits beim Mobility Management betrachtet.
- *Komplexität und Stabilität:* Die zunehmende Komplexität des Betriebs von hochgradig verteilten Netzen ist ein generelles Problem. Die Heterogenität der Netze fördert die Komplexität zusätzlich. Darüber hinaus

werden Netzwerke durch Netzwerkoperatoren betreut. Manuelle Kontrolle ist zeitraubend, teuer und fehleranfällig.

- *Skalierbarkeit, Robustheit, Reaktionsfähigkeit*: In sehr großen Netzwerken wie dem Internet ist es für das Routing wenig effizient alle Verkehrsflüsse zu messen, zu speichern und zu verarbeiten um eine garantierte QoS zu erreichen. Aus diesem Grund wäre es notwendig eher intelligente Inter Domain Routing Algorithmen zu entwerfen.
- *Heterogene, autonome Systeme (Ases)*: Es ist schwierig verschiedene Technologien, die in drahtlosen Multihop Netzwerken verwendet werden, in drahtgebundene Netzwerke, zu integrieren.
- *Unterscheidung von Datenströmen*: Es wäre sinnvoll wenn der individuelle Datenstrom unterschieden werden könnte. Dies ist jedoch sehr schwer zu erreichen, da eine Ende zu Ende QoS nur erreichbar ist, wenn sie auch skalierbar ist.
- *Sicherheit*: Dies kann ebenfalls für einzelne Dienste problematisch sein, wenn diese aus einer Domäne heraus für eine andere autorisiert werden sollen.

Gemeinsame Berichte der FST 1 + FST 2:

- Bericht „CICS project – Patent search“
Es wurden die Klassen G05B (12 Untergruppen), G06F (10 Untergruppen) und H04L (7 Untergruppen) durchsucht. Dabei konnten 12 naheliegende Patente/Patentanmeldungen gefunden und bezüglich der geplanten CICS-Struktur bewertet werden. Keines der gefundenen Patente steht jedoch mit der konzipierten CICS- Struktur in Konflikt.
- Bericht „Anwendungs- und Nutzungspotential von verteilten, Cloud-basierten Steuerungsdiensten“
- Gegenüberstellung Ziele / Ergebnisse

Mit der Entwicklung einer CICS-Metrik und der Bewertung des CICS-Potentials einschließlich einer ausführlichen Patentrecherche konnten die Ziele der Teilarbeitspakete AP 1.1, AP 1.2 und AP 1.4 durch die FST 1 vollständig realisiert werden.

Im Bericht „Anwendungs- und Nutzungspotential von verteilten, Cloud-basierten Steuerungsdiensten“ wurden von der FST 2 die Probleme heterogener Netzwerke analysiert, die beim Einsatz cloudbasierter Steuerungsdienste abhängig von der zugrundeliegenden Netzwerkinfrastruktur auftreten können. Die im Antrag beschriebenen Ziele konnten in der Studie vollständig erreicht werden. Da künftig neben lokalen drahtlosen Netzen auch Mobilfunktechnologien (LTE, 5G) für cloudbasierte Steuerungsdienste von Bedeutung sein können, wurden auch deren Eigenschaften betrachtet.

3.2 Arbeitspaket 2 (FST 1)

Bewertung der Anforderungen und Eigenschaften klassischer Steuerungssysteme in Gegenüberstellung zu potentiellen Lösungen auf Basis von Webtechnologien.

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 2.1: Recherche und Herausarbeitung der potentiell wichtigsten Webtechnologien, die zukünftig in der Automatisierungs- und Steuerungstechnik zum Einsatz kommen könnten.

AP 2.2: Aufstellung eines Bewertungsschemas und Durchführung einer SWOT²-Analyse der Technologien nach AP 2.1 hinsichtlich Ihrer Nutzbarkeit bezogen auf den in AP 1.1 entwickelten CICS-Forderungskatalog. Herausarbeitung wahrscheinlicher Einsatzszenarios und potentieller Eigenschaften. Erstellung eines Ergebnisberichts für AP 1.

- Durchgeführte Arbeiten

Basierend auf einer Internet- und Literaturrecherche wurden die potentiell wichtigsten Webtechnologien, die zukünftig in der Automatisierungs- und Steuerungstechnik zum Einsatz kommen könnten, herausgearbeitet und für den Einsatz in CICS bewertet. Die Technologien wurden strukturiert und bewertet sowie wahrscheinliche Einsatzszenarios für CICS wurden herausgearbeitet.

- Erzielte Ergebnisse

Die erzielten Ergebnisse behandeln die Schwerpunkte

- Lösungsvarianten für eine CICS-Steuerung,
- Webtechnologiene und Automatisierungstechnik sowie
- Webtechnologien in CICS.

Es wurde eine allgemeine Struktur für eine CICS-Steuerung entwickelt (Bild 5) und deren Modifikationen für server- und client-basierte Lösungen abgeleitet. Eine CICS-Steuerung besteht danach grundsätzlich aus folgenden Komponenten:

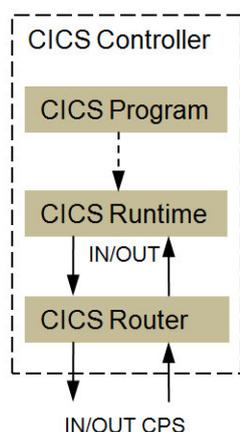


Bild 5: Allgemeine Struktur einer CICS-Steuerung

- *CISC-Programm (CICS-P):* Das CICS-P ist ein IEC61131-3-Steuerungsprogramm in der PCLopen XML-Notation. Es beinhaltet keine Steuerungskonfiguration sondern nur den Programm- und Variablenteil.
- *CICS-Runtime (CICS-RT):* Die CICS-RT ist die Ausführungsumgebung für das CICS-Programm. Sie kann zyklusgesteuert oder event-basiert arbeiten. Pro-

² SWOT - Strengths, Weaknesses, Opportunities, Threats

zessdateneingänge (CICS-IN) werden grundsätzlich event-basiert und asynchron verarbeitet.

- *CICS-Router (CICS-R)*: Der CICS-R beinhaltet die Gerätekonfiguration für eine CICS-Steuerung, d.h. darüber wird festgelegt, welche CPS-Komponenten (welche Automatisierungsgeräte) an die Steuerung angeschlossen sind. Da in der Web-Welt über die CPS-Komponente die realen Geräte bereits virtualisiert vorliegen, beinhaltet eine CICS-Konfiguration nur die Zuordnung zwischen den absoluten IN/OUT-Adressen im CICS-Programm und deren Zuordnung (Routing) zu den IN/OUTs der CPS-Komponenten. Eine CICS-Konfiguration wird deshalb auch als *CICS-Router* bezeichnet.

Unter Beachtung der grundsätzlichen CICS-Struktur ergeben sich insgesamt je eine Server-basierte (Server-based Mode – SM) und eine Client-basierte (Client-based Mode – CM) sowie zwei gemischte CICS-Lösungen (Server-based Mixed Mode – SMM und Client-based Mixed Mode - CMM). Damit ergeben sich vier prinzipielle Möglichkeiten für eine CICS-Lösung.

Server-basierte Lösungen

Bei einer Server-basierten CICS-Lösung befinden sich Runtime-Maschine (CICS-RT) und Steuerungsprogramm (CICS-P) auf dem Server und das Steuerungsprogramm wird auch auf dem Server ausgeführt. Das Automatisierungsgerät als CPS-Komponente muss während der Ausführung des Steuerungsprogramms mit der Runtime-Maschine im Server möglichst verzögerungsfrei und zuverlässig über das IP-Netz (Intranet/Internet) verbunden sein. Es ergeben sich dazu prinzipiell zwei Möglichkeiten:

- (1) Die CICS-Runtime wird in einem Projektierungsprozess statisch mit einer festgelegten CPS-Komponente verschaltet. CPS-Runtime und CPS-Komponente bilden ein vorprojektiertes einheitliches Funktionalsystem. Dies entspricht dem üblichen Konfigurationsprozess einer klassischen SPS, bei dem die Zuordnung der SPS-E/As zu den Prorgammvariablen fest projiziert wird. Nach Start der CICS-Steuerung über den Client verbindet sich die CICS-Steuerung automatisch mit der zugehörigen CPS-Komponente über das IP-Netz und führt das Steuerungsprogramm aus.
- (2) Vor Start der CICS-Runtime wird zuerst auf den Client ein CICS-Router vom Server geladen. Nach Start der CICS-Runtime verbindet dieser Router dynamisch die CPS-Komponente mit der CICS-Runtime im Server. Die Prozessdaten vom Automatisierungsgerät werden jetzt über den Client zum Server geleitet.

Bild 6 zeigt die beiden Kommunikationswege für die Prozessdaten zwischen CPS-Komponente und CICS-Steuerung bzw. CICS-Runtime für die server-basierten Lösungen.

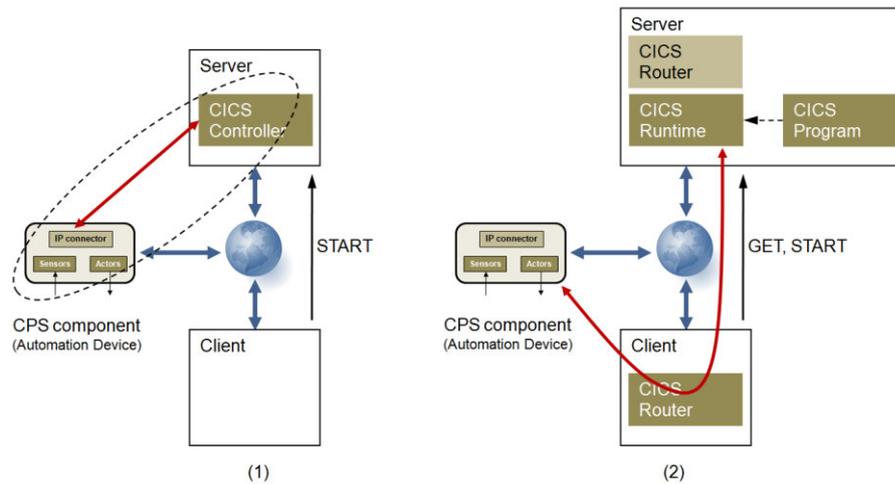


Bild 6: Komponentenstruktur und Kommunikationswege für server-basierte CICS-Lösungen

Client-basierte Lösungen

Auch bei der client-basierten CICS-Lösung ergeben sich zwei prinzipielle Varianten:

- (1) Das Steuerungsprogramm läuft zwar in der CICS-Runtime im Client, aber die Kommunikation zur CPS-Komponente läuft über einen dynamisch umkonfigurierbaren Router im Server. Hinsichtlich der Flexibilität und Online-Anpassbarkeit ergeben sich ähnliche Vorteile wie bei der server-basierten Mixed Mode-Lösung.
- (2) Die CICS-Runtime wird als Instanz im Client ausgeführt. Es muss eine offline-Projektierung der CICS-Runtime erfolgen, um diese mit den E/As der CPS-Komponente zur Laufzeit verbinden zu können. Es können aber beliebig viele CICS-Steuerungsprojekte erstellt und in einer Cloud abgelegt werden. Zur Laufzeit können diese dann dynamisch geladen und ohne Zeitverzug ausgeführt werden.

Bild 7 zeigt die beiden Kommunikationswege für die Prozessdaten zwischen CPS-Komponente und CICS-Steuerung bzw. CICS-Runtime für die server-basierten Lösungen.

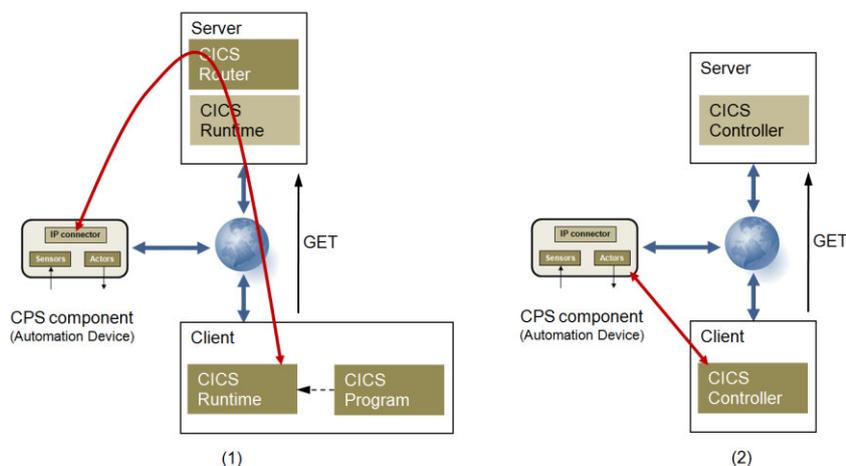


Bild 7: Komponentenstruktur und Kommunikationswege für client-basierte CICS-Lösungen

Die potentiell möglichen CICS-Lösungen wurden, bezogen auf den CICS-Forderungskatalog sowie bezogen auf verfügbare Webtechnologien, bewertet und es wurden Prioritäten für die prototypische Implementierung festgelegt. Unter Berücksichtigung der Zukunftsfähigkeit der unterschiedlichen Webtechnologien basiert die Festlegung der zum Einsatz kommenden Technologie für ein CICS-Steuerungssystem auf folgenden Prämissen:

- Technologien, die auf der Nutzung von PlugIn's und spezifischen Objekten im Webbrowser basieren (Java-Applets, ActiveX-Objekte, Flash usw.) werden wegen fehlender Zukunftsfähigkeit und Sicherheitsproblemen nicht eingesetzt.
- Zur Wahrung einer Konsistenz und Reduzierung des Entwicklungsaufwands sollten für Client-basierte und Server-basierte CICS-Lösungen möglichst gleiche oder kompatible Technologien zum Einsatz kommen.
- Zur Reduzierung der Komplexität und Vermeidung von Sicherheitsproblemen durch einen unnötigen Overhead, wurden schlanke und auf die wesentlichen Funktionen reduzierte Lösungen bevorzugt.

Im Ergebnis wurden folgende Prioritäten für die prototypischen Implementierungen festgelegt:

Priorität 1: CICS-Steuerung im Client-based Mode (CM)

Priorität 2: CICS-Steuerung im Server-based Mixed Mode (SMM)

Die Ergebnisse in AP 2 beinhalten darüber hinaus, ausgehend von allgemeinen Untersuchungen zur Nutzung von Diensten im Web, Schlussfolgerungen für die Dienstenutzung für die CICS-Komponentenstrukturen und deren Eigenschaften hinsichtlich Flexibilität und Wandelbarkeit.

Zur Einschätzung des Anwendungspotentials von CICS erfolgte die Durchführung einer SWOT-Analyse mit einer SWOT-Matrix zur Strategie für ein CICS-Steuerungssystem. Ableitend aus dieser Matrix sollten sich die Entwicklungsarbeiten für ein CICS-Steuerungssystem generell auf zwei Schwerpunkte konzentrieren:

1. Entwicklung eines CICS-Steuerungssystems, welches nachweislich die folgenden I40-Anforderungen erfüllen kann:
 - Autonomie und dezentrale Steuerungsformen,
 - Vernetzung von Steuerungssystemen,
 - Einführung des Dienst-Paradigmas in die Produktionsautomatisierung (Produktionsdienste),
 - Rekonfigurierbarkeit, Agilität,
 - Interoperabilität zwischen heterogenen Steuerungssystemen,
 - Abhängigkeiten sollen zur Laufzeit dynamisch veränderbar sein,
 - optimierte Steuerungsstrategien,
 - Nutzung von Modellen zur Entwicklung „höherwertiger“ Steuerungsansätze,
 - Orchestrierung heterogener Steuerungen,
 - Überwindung der starken Informationskapselung von Steuerungen,
 - Fernsteuerung,
 - Einsatz von Smartphones und Tablet-PCs für Steuerungszwecke.

2. Bei Einschränkung der I40-Möglichkeiten und Eingrenzung bestimmter Anwendungsgebiete (z.B. nur für mittlere und lange Reaktionszeiten) Sicherung der notwendigen klassischen SPS-Anforderungen, um ein CICS-Steuerungssystem in Relation zu neuen und erweiterten SPS-Systemen (SOA-SPS) auch unter üblichen Bedingungen einsetzen zu können.

Entsprechend der Projektbeschreibung konzentrierten sich die FuE-Arbeiten im CICS-Projekt auf den ersten Schwerpunkt aus der SWOT-Analyse.

Bericht:

Bericht „ Webtechnologien und CICS - Lösungsmöglichkeiten, Einsatzszenarios und potentielle Eigenschaften“

- Gegenüberstellung Ziele / Ergebnisse

Unter Berücksichtigung von Webtechnologien konnte die prinzipielle Struktur einer CICS-Steuerung aufgestellt und potentiell mögliche CICS-Lösungen ermittelt werden. Diese wurden in Gegenüberstellung mit klassischen SPS-Steuerungen bewertet und Prioritäten für die Arbeiten im Projekt abgeleitet. Die Ziele des Arbeitspakets wurden damit vollständig erreicht.

3.3 Arbeitspaket 3 (FST 1 + FST 2)

Modellierung einer Referenzarchitektur und Beschreibungsmethoden für ein CICS einschließlich zugehöriger Schnittstellen.

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 3.1: Erforschung einer Referenzarchitektur für ein CICS-System, geeignet für die Steuerung von automatisierungstechnischen Anlagen unter Berücksichtigung der zu automatisierenden Prozessklassen.

AP 3.2: Erarbeitung (bzw. Modifizierung verfügbarer) und Festlegung von syntaktisch und semantisch korrekten Schnittstellenbeschreibungen für die CICS-Komponenten in einer geeigneten Metasprache (z.B. XML, JSON). Die Beschreibung soll sich an den Prinzipien für bekannte XML-Protokoll- und XML-Dienstbeschreibungen anlehnen.

AP 3.3: Entwicklung von Vorschlägen zur Integration industrieüblicher Programmier- und Engineering-Werkzeuge in eine CICS-Architektur sowie Festlegung von evtl. erforderlichen Änderungen/Ergänzungen, um die üblicherweise als Stand-Alone arbeitenden Programmiersysteme auch Online für die Erstellung bzw. den Test (Debugging) von CS nutzen zu können.

AP 3.4: Aufbau einer Simulationsumgebung für die Evaluierung heterogener Kommunikations-Infrastrukturen mit deterministischem und nicht deterministischem bzw. stör anfälligem Verhalten.

AP 3.5: Abstrakte Modellierung, Analyse und Test eines Ende zu Ende Szenarios mit Fokus auf nichtfunktionale Eigenschaften. Dieser Arbeitspunkt bildet die Grundlage für die Auslegung des Webconnectors in AP7 hinsichtlich Robustheit und Performance.

Die Forschungsstelle 1 war für die AP 3.1 – 3.3. verantwortlich; die Forschungsstelle 2 für die AP 3.4 und AP 3.5.

▪ Durchgeführte Arbeiten

FST 1:

Es erfolgte die Erforschung einer Referenzarchitektur für ein CICS-System, geeignet für die Steuerung von Automatisierungstechnischen Anlagen unter Berücksichtigung der zu automatisierenden Prozessklassen. CICS-Komponenten und zugehörige Schnittstellen wurden mit software-modelltechnischen Methoden (ERD- und Sequenz-Diagrammen, Zustandsautomaten usw.) beschrieben und als syntaktisch und semantisch korrekte Beschreibungen fixiert. Die Integration industrieeüblicher Programmier- und Engineering-Werkzeuge in die entwickelte CICS-Architektur wurde untersucht und berücksichtigt und die Möglichkeiten zur Nutzung des PLCopen XML-Formats für CICS-Steuerungsdienste im Detail untersucht.

FST 2:

Es wurde eine Simulationsumgebung zur Evaluierung heterogener Kommunikations-Infrastrukturen mit deterministischem und nicht deterministischem Verhalten aufgebaut. Dazu wurden verschiedene Testaufbauten konzipiert um verschiedene Technologien getrennt und in Kombination testen zu können. So wurde für den Test eines TSN (Time Sensitive Network) basierten Ethernet Netzwerks das DE Starter KIT von TTTech evaluiert und in die Testumgebung eingebunden. Ausserdem wurden OPC UA Client Server Lösungen evaluiert und ebenfalls in die Umgebung mit eingebracht.

Für Analyse und Test von Ende zu Ende Szenarios mit Fokus auf nichtfunktionale Eigenschaften wurde ein WAN Emulator evaluiert. Dieser ermöglicht die in AP 1 dargestellten Charakteristika von heterogenen Kommunikationsinfrastrukturen vereinfacht nachzubilden. Die temporalen Eigenschaften wie Durchsatz, Latenz, Jitter, Paketverluste und Verbindungsabbrüche lassen sich hiermit für eine Ende zu Ende Verbindung einzeln und in Kombination zufallsgesteuert beeinflussen. Ebenfalls wurde ein Tool zur Aufzeichnung des Verhaltens einer realen Verbindung integriert. Ziel ist es hier ein aufgezeichnetes Verhalten nachstellen zu können. Die Ergebnisse und deren Verifikation sind weiter unten bei den erzielten Ergebnissen der FST 2 aufgeführt.

▪ Erzielte Ergebnisse

FST 1:

Im Rahmen der FuE-Arbeiten wurde eine CICS-Referenzarchitektur und ein Dienstinterface entwickelt. Es wurden folgende Komponenten identifiziert und beschrieben:

- allgemeine Struktur einer CICS-Steuerung,
- allgemeine Struktur eines CICS-Dienstes,
- Datenmodell eines CICS,
- CICS-Runtime-Dienst,
- CICS-Router-Dienst,

- Komponentenstruktur einer CICS-Steuerung,
- CICS-Betriebsfähigkeit und
- CICS-Engineering.

Die einzelnen CICS-Komponenten und Schnittstellen sind mit software-technischen Methoden modelliert und bilden die Grundlage für eine implementierungstechnische Umsetzung. Da die Komponenten einer CICS-Steuerung nicht mehr als Hardware vorliegen, sondern als Softwareobjekte in der IT bzw. Web-Welt und dort auch i.d.R. in geeigneten Datenbanken gespeichert werden, wurden für die CICS-Architektur ERD-Datenmodelle als eine der Modellierungsmethoden zusammen mit XML-Beschreibungen genutzt.

Die Steuerungseigenschaften einer CICS-Steuerung sollen nicht mehr als klassische Steuerungsfunktionen sondern nach dem Dienstparadigma als Steuerungsdienste zur Verfügung stehen. Ein CICS (im Wortsinn: Cloud-based Industrial Control Service) kann damit alle Eigenschaften eines Cloud-Computing nutzen und somit neue Business-Modelle zur z.B. Vermietung von Steuerungsdiensten schaffen. Eine CICS-Steuerung wird dazu mit zwei Diensten realisiert:

- Runtime-Dienst,
- Router-Dienst.

Eine CICS-Steuerung benötigt üblicherweise beide Dienste. Sie kann aber auch nur durch einen Runtime-Dienst realisiert werden, wenn die angeschlossene CPS- bzw. I40-Komponente (Automatisierungsgerät) bereits ein CICS-Runtime-Interface besitzt. Der Router-Dienst ist mit dem E/A-Konfigurationsteil einer klassischen SPS vergleichbar.

Beide CICS-Dienste sind entsprechend dem Prinzip der Web-orientierten Automatisierungsdienste aufgebaut.

CICS-RT: Runtime-Dienst

Entsprechend der Zustandsmaschine in einer klassischen SPS besitzt die CICS-RT als wichtigste Komponente gleichfalls ein festgelegtes Ablaufverhalten, um ein Steuerungsprogramm auszuführen. Eine CICS-RT realisiert in Anlehnung an eine SPS damit die in Tabelle 5 aufgeführten Betriebszustände.

Tabelle 5: Betriebsarten der CICS-Runtime (CICS-RT)

Betriebszustand	Beschreibung
EXISTENT	Der CICS-RT-Dienst wird instanziiert und steht zum Betrieb zur Verfügung.
READY	Es wird das PLCopen XML-Programm geladen, dessen URL in der CICS-RT-Instanz angegeben ist.
AUTOMATIC	Das Steuerungsprogramm wird abgearbeitet und das interne E/A-Abbild der Steuerung aktualisiert.
STEP	Der STEP-Zustand ermöglicht eine schrittweise Abarbeitung des SPS-Programms für Debugging-Zwecke.
SUSPEND	Die CICS-RT wird gestoppt und alle Ausgänge werden auf FALSE/0 gesetzt.

Eine CICS-RT kann sowohl in einer zyklischen Betriebsart als auch ereignisbasiert betrieben werden. Im Zyklusbetrieb erfolgt das Aktualisieren des E/A-Abbildes äquivalent zu einer klassischen SPS. Im ereignisbasierten Betrieb wird das Steuerungsprogramm nur dann abgearbeitet, wenn sich der Wert einer Eingangsvariablen ändert oder ein internes Ereignis auftritt (z.B. Ablauf eines Timers).

CICS-R: Router-Dienst

Der E/A-Konfigurationsdienst eines CICS-Steuerungssystems (CICS-R) ist von der CICS-Runtime separiert. Damit lässt sich eine dynamische Re-Konfigurierung einfach sichern, d.h. bei identischem Steuerungsprogramm kann die E/A-Konfiguration innerhalb eines Programmzyklusses geändert werden. Darüber hinaus können identische Maschinen/Anlagen trotz unterschiedlicher E/A-Baugruppen mit dem gleichen Steuerungsprogramm betrieben werden.

Letztendlich bildet ein verteilter separater Konfigurationsdienst auch die Grundlage für eine zukünftige automatische Gerätekonfiguration im Industrial Internet of Things (IIoT).

Das CICS-Routing arbeitet nach den folgenden beiden Prinzipien:

- Ein CICS-Programm (PLCopen XML-Programm) arbeitet mit absoluten I/O-Adressen.
- Der CICS-Router verbindet die absoluten E/A-Adressen mit den realen E/A-Adressen der Geräte (CPS-Komponenten).

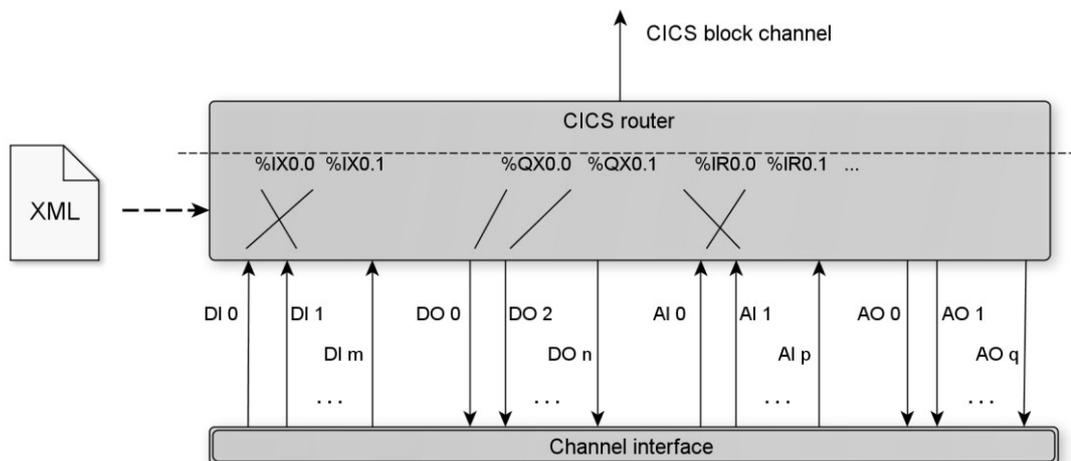


Bild 8: CICS-Router

Bild 8 veranschaulicht die Funktionsweise eines CICS-Routers. Die über das Channel-Interface angeschlossenen digitalen und analogen Ein- und Ausgänge eines Gerätes werden auf absolute E/A-Programmadressen geroutet und über einen *CICS block channel* an die CICS-RT übertragen. Die Routing-Regeln (Verschaltungsmatrix) werden über eine CICS-R-XML-Datei festgelegt.

Erstellung eines Steuerungsprogramms für CICS (CICS-Engineering)

Eine wesentliche Eigenschaft von CICS ist die Nutzung herkömmlicher und industrieeüblicher SPS-Programmiersysteme zur Erstellung von Steuerungsprogrammen. Nur der Einsatz von Industriestandards für CICS, wie die IEC 61131, gewährleistet ein Akzeptanz bei potenziellen CICS-Anwendern. Als einheitliches Format eines standardisierten IEC 61131-3 Steuerungsprogramms, welches auch mittels Web-Technologien ausgezeichnet verarbeitet werden kann, bietet sich das PLCopen XML-Format an. Die bisher nur als Austauschformat vorgesehene XML-Notation wird nun in CICS als ausführbare Runtime-Notation genutzt.

Eine spezielle CICS-Untersuchung analysierte für die Programmiersysteme *PC WORX*, *CoDeSys*, *STEP7 (TIA-Portal)* und *openPCS* die Konformität zum PLCopen-Standard sowie die Fähigkeiten dieser Werkzeuge SPS-Programme als PLCopen-konforme XML-Programme zu exportieren bzw. auch zu importieren. Die Untersuchung kommt zusammengefasst zu folgendem Ergebnis:

- Klassische SPS-Programme, programmiert in ST mit CoDeSys oder programmiert in AWL mit PC WORX und exportiert als PLCopen XML-Programme können die Basis für ein CICS-Steuerungsprogramm bilden.

openPCS eignet sich wegen Instabilitäten beim Export eines PLCopen XML-Programms nicht als Programmierumgebung für CICS. Das *TIA-Portal* kann die erstellten Steuerungsprogramme bisher nicht als konforme PLCopen XML-Programme exportieren (Stand November 2016).

Die exportierten PLCopen XML-Programme in der AWL oder ST-Notation müssen für die Ausführbarkeit in der CICS-RT in die interne CICS-RT-Programmiersprache (JavaScript) umgesetzt werden. Dazu wurden für die beiden CICS-Prototypen zwei Varianten entwickelt und getestet:

Variante 1 mit AWL-Interpreter:

Für die CICS-CM wird das AWL-Programm nach einer Übersetzung (Umformung des PLCopen XML-Programms in eine interne geeignete Form) beim Laden des Programms in die CICS-RT mittels eines Interpreters ausgeführt. Der Interpreter ist untrennbarer Bestandteil der CICS-RT.

Variante 2 mit ST-Compiler:

Für die CICS-SMM wurde ein separater ST-Compiler entwickelt, der die PLCopen XML-Programme zu direkt ausführbaren JavaScript-Programmen compiliert. Der ST-Compiler basiert auf serverseitigem JavaScript mit node.js und arbeitet als Webservice für beliebige CICS-Instanzen.

Unabhängig von der Sprachnotation im PLCopen XML-Programm ergeben sich aber für das CICS-Engineering einige Einschränkungen gegenüber der klassischen Erstellung eines SPS-Programms, da die SPS-Engineering-Welt keinen gemeinsamen Bereich mit der Web-Welt besitzt (zumindest gegenwärtig noch nicht). Dazu gehören u.a.:

- Ein Online-Debugging über das Programmier-Werkzeug ist nicht möglich. Es wurden deshalb zusätzliche Web-basierte Testmöglichkeiten (z.B. Step-Betrieb, Ausgabe der Step-Befehle in der Browser-Konsole) in die CICS-Steuerung eingeführt.
- Es müssen absolute Adressen für die E/A-Prozessdaten im Programm verwendet werden. Dies ist problemlos während der Erstellung des Steuerungsprogramms realisierbar.
- Die Verwendung von Steuerungsprogramm-Bibliotheken ist bisher nicht möglich.

Zusammengefasst ergibt sich zum CICS-Engineering, dass ein industrietaugliches Engineering mit einem etwas höheren Aufwand gegenüber klassischer SPS-Programmierung durchaus möglich ist und sicher in einer Übergangszeit für erste CICS-Anwendungslösungen eingesetzt werden kann. Auf längere Sicht gesehen wird es aber unumgänglich für Steuerungsdienste aus der Cloud auch entsprechende Cloud- und Web-basierte Engineering-Werkzeuge zu entwickeln, die nahtlos in die global vernetzte Welt von Industrie 4.0 und IoT passen.

Berichte:

- Bericht „CICS-Referenzarchitektur und Dienstinterface“ (AP3-Architekturbericht)
- Bericht „IEC 61131-3 programming in CICS - Problems and possible solutions“

FST 2:

In verschiedenen Testreihen wurde das Verhalten von OPC UA Client Server Verbindungen analysiert, wobei das schnellere binäre Protokoll zum Einsatz kam. Es hat sich herausgestellt dass sich bei mehrmaligem Request der gleichen Client-Funktion stark unterschiedliche Round Trip Zeiten ergeben haben, was darauf schliessen läßt dass der OPC Server z.B. beim ersten Aufruf Initialisierungen durchführt, die bei den nachfolgenden Aufrufen wegfallen. So zeigte sich dass der erste Aufruf bis zu 70 mal mehr Zeit benötigt als der Nachfolgende (270 ms \leftrightarrow 4 ms). Es wurden folgende Aufrufe getätigt: connectToOpcServer, createOpcSession, browseFolder und readVariable. Es wurden ausserdem der Mittelwert, die Standardabweichung und die Varianz von ca. 1000 Samples ermittelt. Tabelle 6 stellt die Untersuchung der Latenz von unterschiedlichen OPC UA Client Server Aufrufen im Vergleich dar.

Tabelle 6: Untersuchung der Latenz von unterschiedlichen OPC UA Client Server Aufrufen

name	method	os	arch	cpus	cpu_model	cpu_s peed	total_mem	free_mem	mean	stdev	variance	samples
Calling connectToOpcServer (First request...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0909E+10	61.507483	0	0	1
Calling connectToOpcServer (Following requests...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0896E+10	6.86176571	1.22406731	1.49834078	931
Calling createOpcSession (First request...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0896E+10	272.016782	0	0	1
Calling createOpcSession (Following requests...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0884E+10	4.86143441	0.83799795	0.70224057	981
Calling browseFolder "RootFolder" (First request...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0884E+10	15.146826	0	0	1
Calling browseFolder "RootFolder" (Follow- ing requests...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0863E+10	4.19213578	0.55836842	0.31177529	1923
Calling OPC UA read- VariableValue "ns=6;s=unix_call" (First request...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0863E+10	18.280407	0	0	1
Calling OPC UA read- VariableValue "ns=6;s=unix_call" (Following request...)	RTT	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0866E+10	2.46537717	0.35267874	0.12438229	956
Subscribing to changes on "ns=5;s=unix_time"	latency +jitter	win32	ia32	4	Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz	2594	1.7068E+10	1.0915E+10	2341.62238	11.6651268	136.075184	1001

Für die Latenzmessung von TSN basierten Netzwerken (Bild 9, Bild 10) wurden drei Switche miteinander über TSN enabled Ports verbunden.

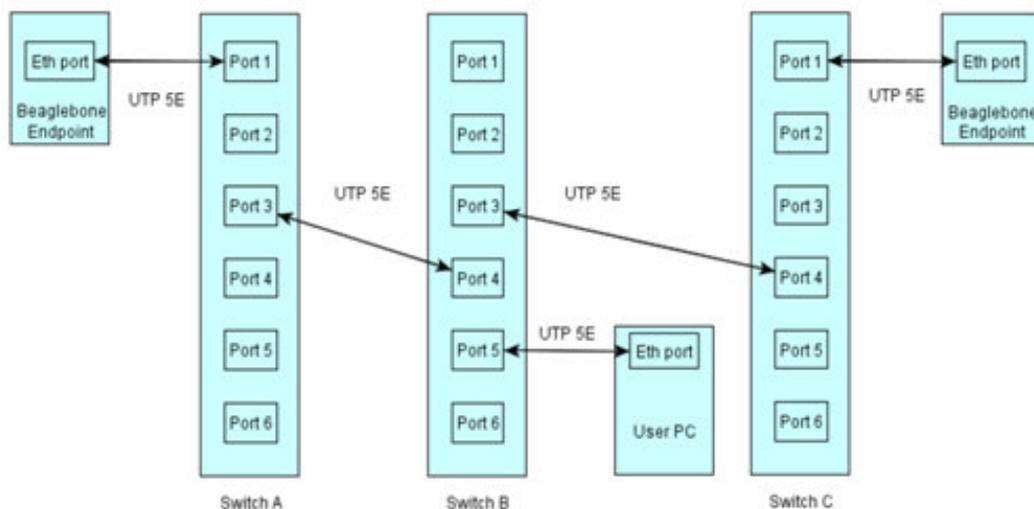


Bild 9: OPC UA Client Server Test auf Basis von TSN und Traffic Generator Ostinato

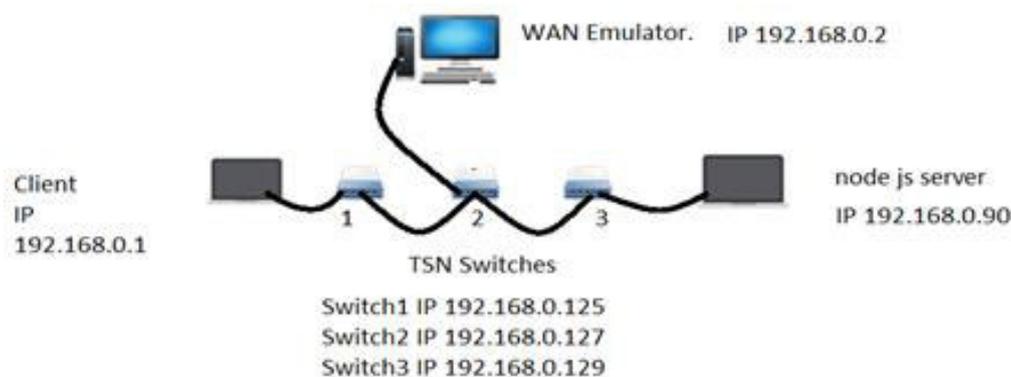


Bild 10: WebSocket Test auf Basis von TSN und WAN Emulator

An zwei der Switche wurden je ein OPC UA Server und ein OPC UA Client von der Fa. Unified Automation verwendet. Zur Traffic Generierung wurde ein PC am 3. Switch angeschlossen, auf dem PC wurde der Generator Ostinato installiert. Die OPC UA Server Aufrufe finden periodisch statt und können entweder durch eine TSN basierte Verbindung oder durch herkömmliche Ethernetverbindung geleitet werden. Es wurden jeweils Tests mit und ohne eingeleitete Störungen durchgeführt. Es konnten bei einer TSN basierten Verbindung und der Überlagerung von Störungen nur eine sehr begrenzte Veränderung festgestellt werden. Dagegen konnte man bei einer nicht TSN basierten Verbindung feststellen, dass sich bei eingeleiteten Störungen sehr starke Veränderungen im Bereich der Latenz und des Jitters ergaben.

Zum Zeitpunkt der Tests war TSN noch nicht standardisiert (wird für 2018 erwartet). Deshalb ist es möglich, dass sich nach Projektende bei künftigen Tests leicht veränderte Werte ergeben. In den Bildern 11 und 12 sind exemplarisch Messergebnisse für ein TSN basiertes Netz unter Einbeziehung des Trafficgenerators bzw. des WAN Emulators dargestellt. Bild 11 zeigt die Auswirkungen von zusätzlicher Netzwerklast, erzeugt

mit dem Generator Ostinato, Bild Abbildung 1 12 zeigt die Auswirkungen des WAN Emulators auf das oben dargestellte Testnetz.

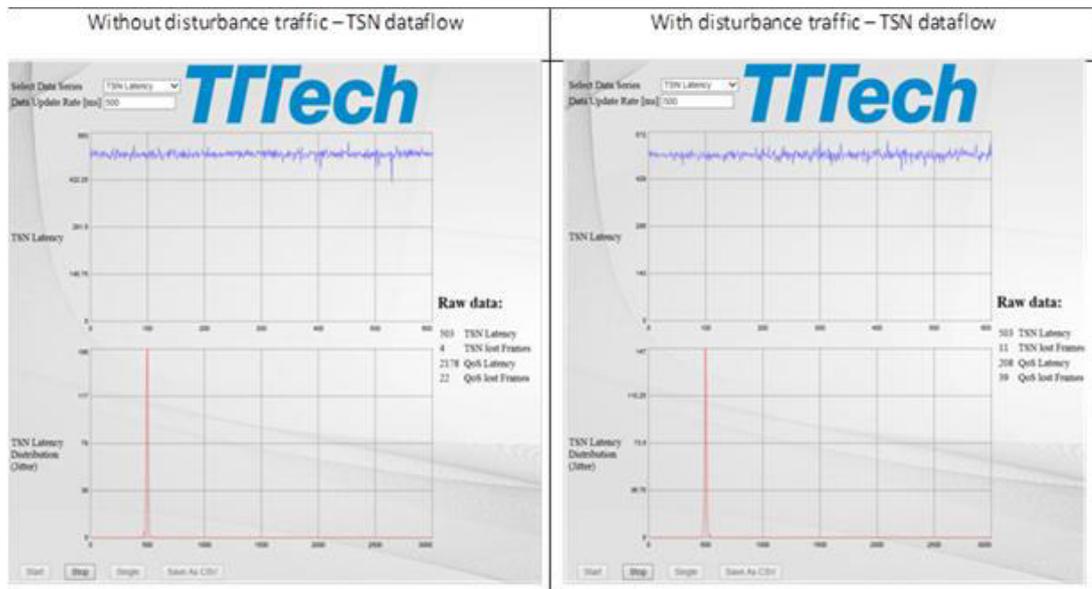


Bild 11: Auswirkungen von zusätzlicher Netzwerklast, erzeugt mit dem Generator Ostinato

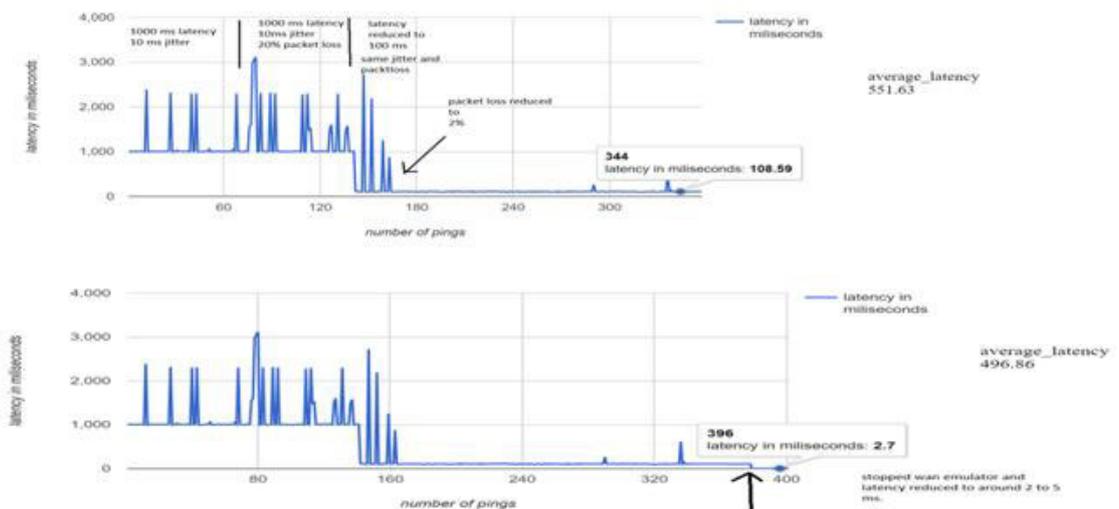


Bild 12: Auswirkungen des WAN Emulators auf das Testnetz

Im Rahmen des AP3 wurden zur Beeinflussung des Netzes der Traffic Generator Ostinato und der WAN Emulator der TATA Group evaluiert, die sich für die weiteren Untersuchungen in AP 4 und AP 7 als geeignet erwiesen haben. Während Ostinato die Generierung von UDP/TCP Paketen und hier die Simulation verschiedener Lastfälle ermöglicht, können mit Hilfe des WAN Emulator gezielt oder zufällig Übertragungsparameter wie Latenz, Paketverluste, Bandbreiteneinschränkungen oder Verbindungsabbrüche simuliert werden. Basierend auf diesem Testnetz und dessen Eigenschaften wurde exemplarisch das bei CICS verwendete Websocket-Protokoll analysiert. Dabei wurden erste folgende Ergebnisse erzielt:

- Die Latenz ist nicht nur von der Bandbreite sondern auch von dem verwendeten Protokoll abhängig.
- Beim Einsatz eines Browsers werden bei websocket Nutzdaten erst innerhalb des Browsers gepuffert bevor sie mit Javascript ausgelesen werden. Dies kann für ein zusätzliches Delay verantwortlich sein.
- Über die Websocket API kann man nur komplette Websocket-Nachrichten schicken, was bei hohen Paketverlusten zu weiteren Delays führen kann.

- Eine einzelne Websocket-Verbindung kann mehrere Nachrichten nicht verschachteln, es kann also immer nur eine Nachricht behandelt werden. Dies kann bei großen Nachrichten zu Problemen führen. Eine Lösung wird hier in Zukunft mit der Multiplexingfähigkeit von Websocket erwartet.

Weitergehende Ergebnisse im Bereich der Webprotokolle wurden in AP 4 erzielt

- Gegenüberstellung Ziele / Ergebnisse

FST 1:

Das Ziel der Modellierung einer Referenzarchitektur sowie einer Beschreibungsmethodik für CICS konnte durch die FST 1 erfolgreich umgesetzt werden. Es liegen syntaktische und semantische Schnittstellenbeschreibungen für die CICS-Komponenten als ERD-Diagramme, XML-Beschreibungen sowie als JSON-Objekte vor. Es wurde darüber hinaus eine Lösung entwickelt, um industrieübliche Programmier- und Engineering-Werkzeuge in eine CICS-Architektur zu integrieren. Dies ist aber aktuell noch nicht nahtlos möglich, da sich die Ausführungsumgebung einer CICS-Architektur (Webtechnologien) grundsätzlich von der einer klassischen SPS-Programmierungsumgebung (C, C++, Visual Basic) unterscheidet. Ein Online-Debugging eines CICS-Steuerungsprogramms kann deshalb gegenwärtig auch nicht realisiert werden. Dazu müsste zukünftig eine IEC61131-3-Programmierungsumgebung geschaffen werden, die vollständig auf Webtechnologien (JavaScript, node.js etc.) setzt und den Webbrowser als Projektierungsoberfläche nutzt.

FST 2:

Die im Antrag genannten Ziele, der Aufbau einer Simulationsumgebung zur Evaluierung heterogener Kommunikationsinfrastrukturen und die Analyse eines Ende zu Ende Szenarios konnten aufgrund der Verfügbarkeit erster TSN Prototypen vollständig und zukunftsorientiert umgesetzt werden. Speziell bei TSN zeigt sich, dass zum Zeitpunkt der Erstellung des Abschlußberichts seitens der Industrie eine breite Bereitschaft besteht künftig auf diese Technologie im Rahmen von Industrie 4.0 zu setzen. Die Tests haben gezeigt dass eine TSN basierte Ethernet-Infrastruktur aufgrund seines deterministischen Verhaltens und seiner geringen Latenz für den Einsatz von CICS in lokalen Netzen (private cloud) beste Voraussetzungen liefert. Das Ende zu Ende Szenario hat zudem noch einmal verdeutlicht, dass seitens der FST 2 der Schwerpunkt der Arbeiten in AP 4 und AP 7 vor allem in der Betrachtung von Optimierungsmöglichkeiten zur Anbindung von CICS als Public Cloud Dienst zu legen ist, da hier auch in Zukunft die größten Herausforderungen bzgl. Robustheit, Latenz und deterministischem Verhalten zu sehen sind.

3.4 Arbeitspaket 4 (FST 2)

Analyse und Bewertung des Web-basierten Zugriffs auf industrieübliche Schnittstellen der prozessnahen Ebene

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 4.1: Auswahl von mindestens drei typischen Industrieschnittstellen der prozessnahen Ebene in Abstimmung mit dem PA und Festlegung der Web-Connectoren, mit denen Beispielimplementierungen aufgebaut werden..

AP 4.2: Aufbau und Test der Beispielimplementierungen als Webclient- und -serveranwendungen. Als Automatisierungsgeräte wird verfügbare Gerätetechnik aus den Forschungsstellen genutzt.

AP 4.3: Konzipierung, Konstruktion und Test des Messaufbaus für die Zeitmessungen an den Beispielimplementierungen sowie Durchführung der Messungen. Im Ergebnis der Messungen werden fundierte Aussagen über das Echtzeitverhalten der ausgewählten Web-Zugangsstrukturen und damit auch über deren Eignung für ein CICS möglich sein.

AP 4.4: Erarbeitung von Schlussfolgerungen für die Entwicklung eines neuartigen CICS-Webconnectors, der industrieübliche Schnittstellen aufgreift und Geräte der Feldebene einfach, zuverlässig und sicher mit einem IP-Netz verbindet.

- Durchgeführte Arbeiten

In AP 4.1 wurden 3 typische Schnittstellen/Protokolle für den webbasierten Zugriff auf die prozessnahe Ebene analysiert, gegenübergestellt und in Abstimmung mit dem Projektausschuß ausgewählt. Es handelt sich dabei um http 2.0, WEBSOCKET, WEB RTC und OPC UA.

Im Rahmen der Beispielimplementierungen (AP 4.2) wurde ein Roboterarm (UR3 von Universal Robots) in die CICS Steuerungsarchitektur über Modbus TCP und die PureBox des Projektpartners W&T integriert. Weiterhin wurde eine Meß- und Prüfstation umgebaut und über Node-RED angebunden. Mit dieser auf Basis von Node.js quelloffenen Javascript Applikation wurde eine Verbindung zwischen Modbus-TCP und WebSocket geschaffen. Die Prüfstation beinhaltet sowohl verschiedene Sensoren und Aktoren, die über I/O Feldbuscontroller angebunden wurden als auch einen Servoantrieb mit zugehörigem Controller des Projektpartners ESR Pollmeier.

Entsprechend AP 4.3 wurde eine Testumgebung für die zuvor ausgewählten Webprotokolle http/2.0, WebSocket und WebRTC konzipiert und implementiert. Um vergleichbare Ergebnisse zu erzielen wurde eine Testumgebung erstellt, die es ermöglicht Testdaten zeitgleich über verschiedene Protokolle zu übertragen. Dabei wurden sowohl Latenzen als auch Verbindungsabbrüche mit Hilfe eines WAN Emulators simuliert und deren Auswirkungen auf die jeweiligen Protokolle untersucht.

AP 4.4 bestand schwerpunktmäßig aus der Erarbeitung der Schlußfolgerungen. Diese wurden in ein Paper für die KommA 2016 eingearbeitet und der Öffentlichkeit vorgestellt.

- Erzielte Ergebnisse

Es wurden sowohl theoretische als auch praktische Analysen der ausgewählten Schnittstellen HTTP2.0, WebSocket und WebRTC durchgeführt. Darüber hinaus wurde anhand einer Klassifizierung zwischen synchronen, asynchronen, unidirektionalen und bidirektionalen Schnittstellen unterschieden. Weiterhin wurden Möglichkeiten der Priorisierung der Datenströme und die Securityeigenschaften der Schnittstellen be-

trachtet. Die nachfolgende Tabelle 7 zeigt einen Vergleich der Eigenschaften der ausgewählten Webprotokolle. Während der Laufzeit des Projekts wurde auch regelmäßig der aktuelle Verbreitungsgrad der Protokolle anhand des Webportals <http://caniuse.com> beobachtet. Dabei zeigte sich während der Projektlaufzeit ein starker Zuwachs des WebRTC und eine hohe Durchdringung des WebSocket Protokolls.

Tabelle 7: Vergleich der Eigenschaften der ausgewählten Webprotokolle

Eigenschaften	HTTP2.0	WebSocket	WebRTC Data Channel
Encryption	Always	Configurable	Always
Reliability	Reliable	Reliable	Configurable
Delivery	ordered	Ordered	Configurable
Multiplexed	yes	No	Yes
Transmission	Stream Oriented	Message oriented	Message oriented
Binary and UTF-8 text Transfer	only Binary	yes	yes
Compression	Yes	no (extension)	No

COAP

mit einbezogen um diese mit den webbasierten Schnittstellen zu vergleichen, bzw. um deren Integrationsmöglichkeiten in den webbasierten Schnittstellen zu untersuchen. Tabelle 8 stellt diesen Vergleich dar.

Die in AP 4.2 durchgeführten Beispielimplementierungen konnten erfolgreich umgesetzt werden und sowohl für interne Tests als auch für die Präsentation auf der SPS IPC DRIVES 2016 als Demonstrator (Bild 13) verwendet werden. Im Rahmen der Implementierung hat sich herausgestellt, dass sowohl die Lösung mit der PureBox über PHP als auch Node-RED aufgrund der Anforderungen der Schnittstellen des Roboters als auch des Servo Controllers nicht uneingeschränkt verwendet werden können. Bei beiden Lösungen mussten von der FST 2 Erweiterungen entwickelt werden.

Tabelle 8: Eigenschaften der gebräuchlichsten IoT Schnittstellen im Vergleich zu HTTP

	PubSub	RPC	Web native	Open Standard	TTL / Interrupts	Connection Recovery	Transactions	Security Mechanisms
AJAX / HTTP	o	X	X	(O)	o	(O)	?	?
[...]								
AMQP / MQTT	X	(X)	o	X	X	?	?	X
SOAP / JSON/XML-RPC / ...	o	X	X	X	(X)	o	(X)	(X)
STOMP	X	o	X	X	(X)	X	X	o
COAP	(X)	X	(X)	X	X	X	X	X!
WAMP.WS	X	X	X	X	(X)	o	o	(O)
[...]								
MBWS	https://tools.ietf.org/id/draft-hapner-hybi-messagebroker-subprotocol-03.html#rfc.section.2.1.7.3 (expired...)							

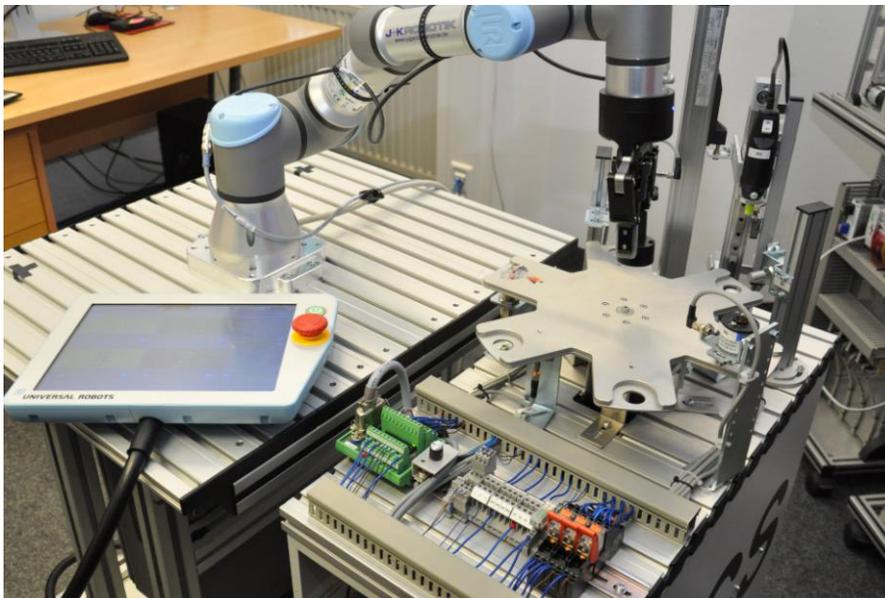


Bild 13: Beispielanwendung der FST 2

Das in AP 4.3 entwickelte Testbed zum Vergleich der verschiedenen Webprotokolle wurde als Webapplikation auf Basis von NodeJS realisiert und mit den in AP3 evaluierten Tools wie z.B. dem WAN Emulator kombiniert. Wie in Bild 14 dargestellt, besteht das Testbed aus mehreren simultan laufenden NodeJS Servern. Für jedes betrachtete Protokoll wurde ein eigener Server entwickelt. Dadurch konnten die Protokolle voneinander unabhängig betrachtet und verglichen werden. Jeder Protokollserver lauscht auf verschiedenen Ports auf von aussen auf die Server zugreifende Clients. Jeder dieser Server ist dann für die Verarbeitung der ankommenden Daten, für deren Protokollierung und dessen Ablage in einer Datenbank zuständig. Der Monitor Server ist für alle Client Anfragen und für das Management der Datenbank zuständig.

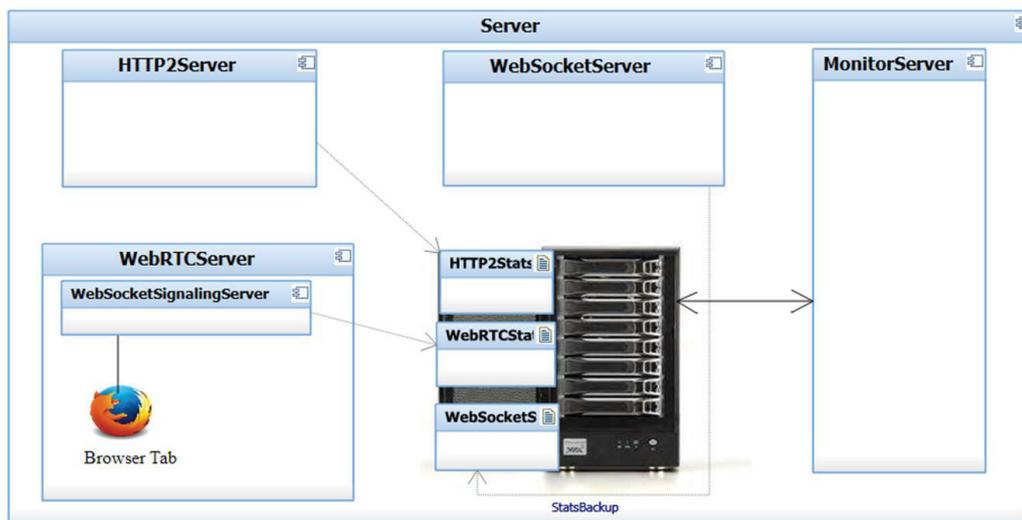


Bild 14: Testbed Architektur

Der Anwender greift auf den Server über einen Browser auf seinem Desktop oder Smartphone zu (siehe auch Bild 15). Der Server bietet dem Anwender dabei eine HTML5 Anwendung, mit der verschiedene Einstellungen vorgenommen werden können um jedes der Protokolle einzeln auf dem Browser und dem jeweiligen Server testen zu können. Die vom jeweiligen Server empfangenen Daten werden als Echo zurück an den Client gesendet. Der Datentransfer wird durch die Anwendung geloggt um die Qualität der laufenden Kommunikationsverbindung analysieren zu können. Am Ende eines Protokolltests werden die insgesamt erfassten Daten in Form einer Statistik an den Monitor Server geschickt und von diesem in der Datenbank abgelegt.

Innerhalb der Webanwendung kann der Anwender die Statistiken und die Qualität des Datentransfers beobachten. Es kann sowohl das zeitliche Verhalten des Netzwerks als auch das der Applikation betrachtet werden. Darüber hinaus kann der Anwender zeitgleich die Testergebnisse der verschiedenen Protokolle in einem Graphen überlagern um somit vergleichende Messungen anstellen zu können.

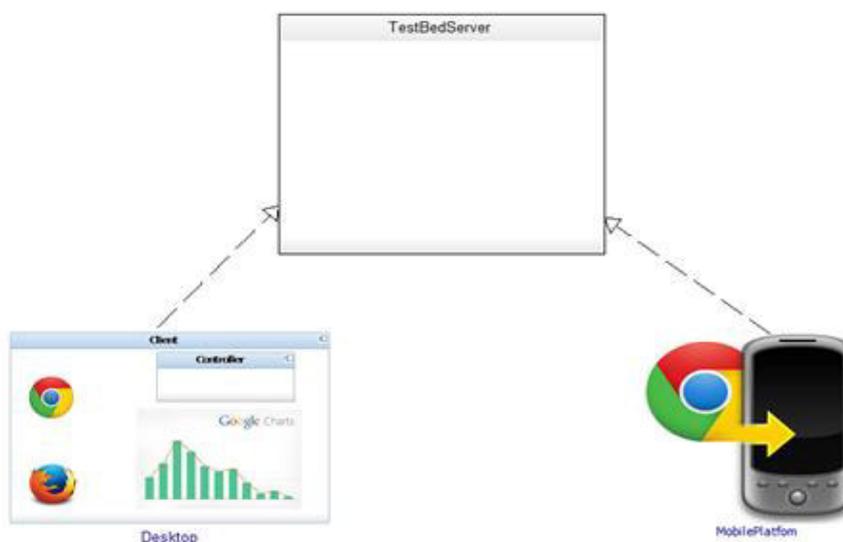


Bild 15: Client Server Test innerhalb des Testbeds

Auf Basis des beschriebenen Testbeds wurden eine Reihe von Analysen durchgeführt, die die Grundlage für die Konzeption und Implementierung des WebConnectors in AP 7 gelegt haben. Im Einzelnen wurden Benchmarks für WebRTC(Reliable), WebRTC(Unreliable), WebSocket und http/2.0 erstellt. Es wurde bei allen Protokollen die gleiche definierte Zeichenkette übertragen. Das Ergebnis der Performance Analyse zeigen Tabelle 9 bzw. Bild 16.

Tabelle 9: Statistik für 100 Pings

Protocol	Web Socket	WebRTC	WebRTC-R	HTTP/2.0
Average RTT (ms)	4.1	1.54	1.61	6.1

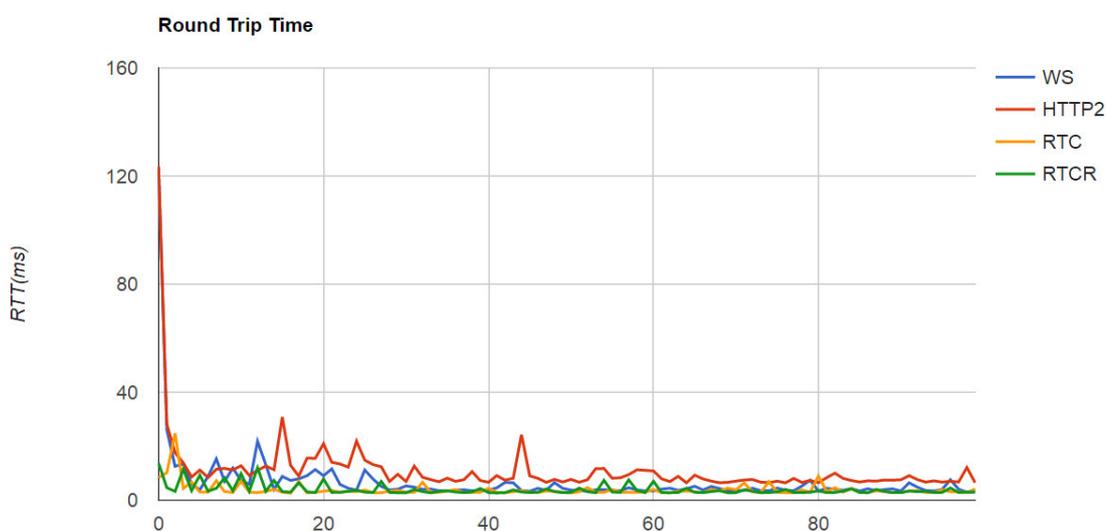


Bild 16: RTT für verschiedene WebProtokolle auf Basis des Chrome Browsers

Deutliche Unterschiede in der „Bearbeitung“ der Webprotokolle konnte bei einem Vergleich verschiedener Browser festgestellt werden. Aus diesem Grund wurden Testreihen sowohl für Chrome als auch Mozilla erstellt und anschließend miteinander verglichen. Bild 17 zeigt im Vergleich die Performance der beiden Browser abhängig von den betrachteten Protokollen.

Es läßt sich erkennen, dass unter gleichen Bedingungen der Chrome Browser eine bessere Performance liefert als der Mozilla Browser. Vor allem bei WebRTC besitzt der Chrome Browser deutlich bessere Performance.

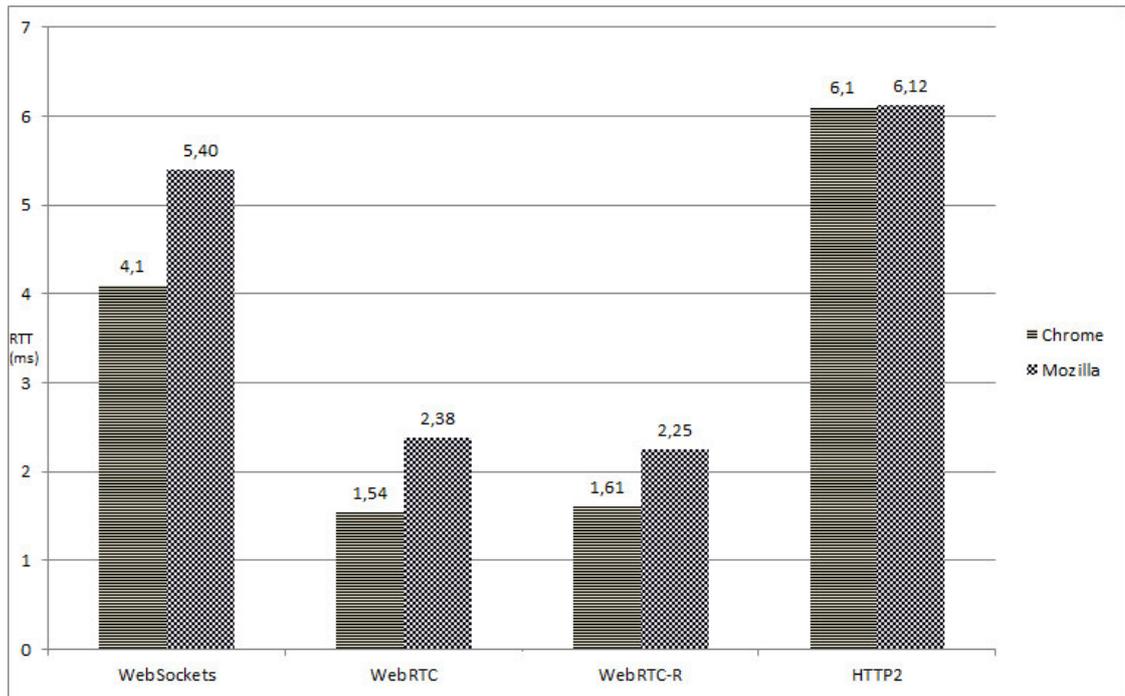


Bild 17: Browser Performance

Desweiteren wurden drei verschiedene Testfälle unter Einsatz des WAN Emulators aufgestellt:

- Delay=100ms, Pings = 20
- Bandwidth=2097kbps, Delay=10ms, PktLoss=2%, Jitter=10ms, , Pings=100
- Bandwidth=2097kbps, Delay=10ms, PktLoss=2%, Jitter=10ms, Pings=1000

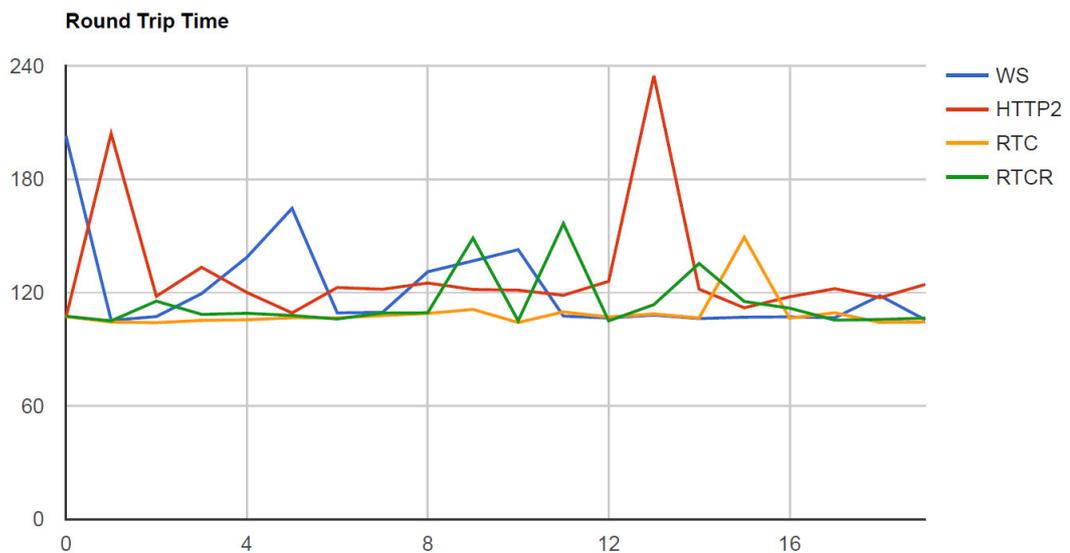


Bild 18: Testfall 1: Delay 100ms, Pings = 20

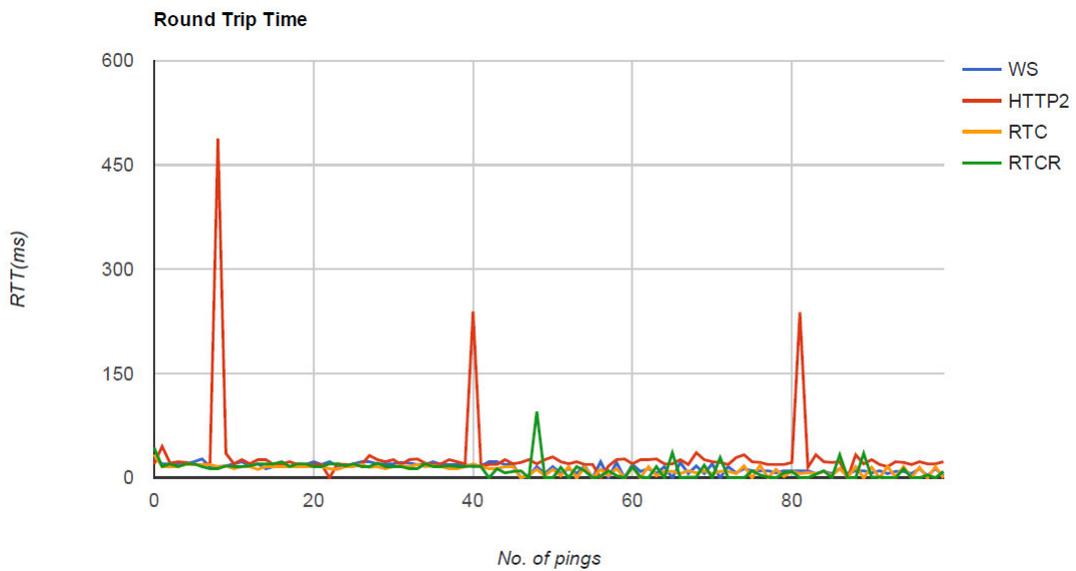


Bild 19: Testfall 2: Delay 100ms, Pings = 100

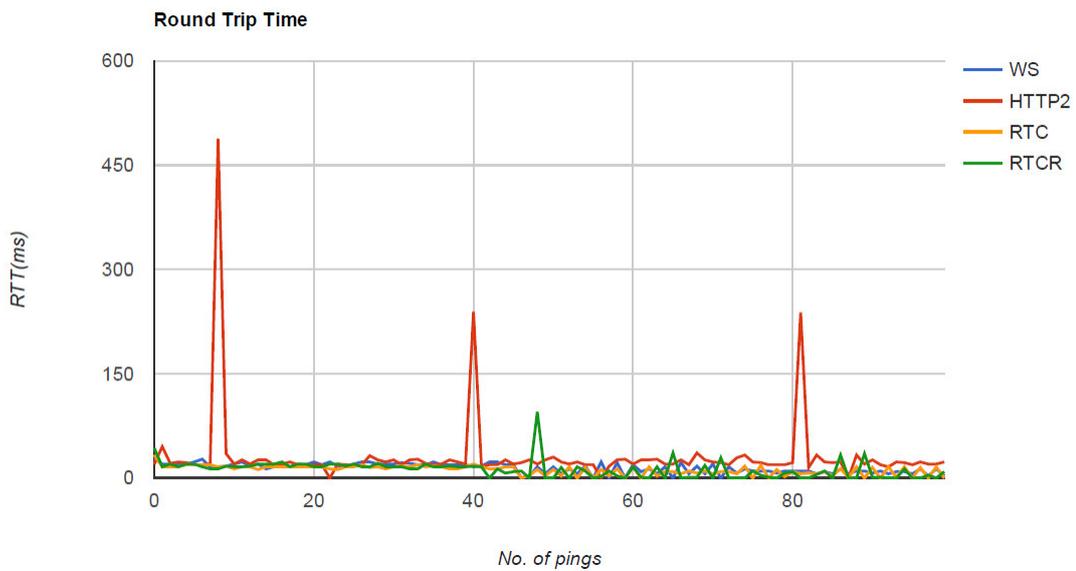


Bild 20: Testfall 3: Delay 100ms, Pings = 1000

Bild 20 stellt den 3. Testfall dar. Hier zeigt sich dass http/2.0 auf Basis der festgelegten Parameter des WAN Emulators deutliche Abweichungen zeigt. Auf der anderen Seite zeigt WebRTC stetig hohe Performance.

Das in AP 4.4 erstellte und veröffentlichte Paper beschreibt zum einen die ausgewählten Protokolle und die in AP4.3. durchgeführten Analysen. Im Detail wurden die den Protokollen zugrundeliegenden Mechanismen zum Verbindungsaufbau oder wie bei WebRTC zur Stabilisierung der Ende zu Ende Verbindung über ICE (Interactive Connectivity Establishment) untersucht und mit dem Verhalten aus den Testfällen in Bezug gesetzt. Außerdem wurde eine Bewertung der Protokolle hinsichtlich ihrer Eignung für

den industriellen Einsatz vorgenommen. So wird eine Kombination aus WebRTC und WebSocket empfohlen. Die sichere Datenkommunikation findet dabei über WebRTC statt, die Signalisierung über WebSocket. Scheitert die Verbindung über WebRTC kann WebSocket als Ersatzprotokoll genutzt werden.

Die Beispielimplementierungen und das Testbed haben deutlich mehr Zeit als geplant benötigt, so dass sich eine mehrmonatige Verzögerung gegenüber dem ursprünglichen Projektplan ergab.

- Gegenüberstellung Ziele / Ergebnisse

Trotz des Ausscheidens eines wissenschaftlichen Mitarbeiters in der Forschungsstelle 2 während der Projektlaufzeit im Juni 2015 und der verzögerten Nachbesetzung konnte das Arbeitspaket 4 vollständig bearbeitet werden. Das Ziel des Arbeitspakets 4, web-basierte Technologien für den Zugriff auf die prozessnahe Ebene, zu untersuchen, wurde erreicht und hat zu der erstaunlichen Schlußfolgerung geführt, dass durch UDP basierte Protokolle gegenüber TCP basierten Protokollen trotz zusätzlich nötiger Sicherungsschicht ein Performancegewinn und eine bessere Skalierung für Cloud Dienste gegenüber anderen Web-Protokollen erreicht werden kann. So konnten in vergleichenden Messungen die Vorteile des UDP basierten Protokolls Web RTC herausgearbeitet werden. Eine weitere Schlußfolgerung war, dass ein auf UDP basiertes Web Socket Protokoll zu weiteren Verbesserungen im Latenzverhalten führen könnte. Die während der Erstellung des Abschlussberichts laufenden Arbeiten der IETF zum Entwurf des UDP basierten Webprotokolls QUIC (Abschluß des „Core protocoll documents“ bis Nov. 2018) zeigen die Richtigkeit und Bedeutung dieser Schlußfolgerungen für künftige webbasierte Protokolle.

3.5 Arbeitspaket 5 (FST 1)

Analyse von IT-Sicherheitsproblemen.

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 5.1: Analyse und Festlegung der für CICS erforderlichen IT-Schutzmaßnahmen unter Berücksichtigung unterschiedlicher CS-Cloud-Strukturen.

AP 5.2: Untersuchung der erforderlichen Methoden und Maßnahmen zur Sicherung der Prozessdatenzugängen zu vernetzten Sensoren/Aktoren und Evaluierung der Wirksamkeit anhand ausgewählter Funktionsbeispiele.

- Durchgeführte Arbeiten

Im Rahmen der Untersuchung der Betriebsfähigkeit im AP3-Architekturbericht wurden die Aspekte Zuverlässigkeit, Datensicherheit und Maschinensicherheit betrachtet und Maßnahmen dazu vorgeschlagen. Es erfolgte eine Analyse und Festlegung der für CICS erforderlichen IT-Schutzmaßnahmen. Zusätzlich wurden die erforderlichen Methoden und Maßnahmen zur Sicherung der Prozessdatenzugängen zu vernetzten Sensoren/Aktoren untersucht und am Beispiel verschlüsselter Zugänge (HTTPS, WSS) getestet.

- **Erzielte Ergebnisse**

Es wurden die kritischen Stellen hinsichtlich Datensicherheit und Maschinensicherheit bei einer Client-basierten CICS-Lösung (CICS-CM) und einer Server-basierten Mixed-Mode CICS-Lösung (CICS-SMM) ermittelt und bewertet sowie mögliche Maßnahmen zur Verbesserung der Datensicherheit vorgeschlagen sowie teilweise getestet. Zu den kritischen Aspekten, die betrachtet wurde, gehören:

- Störung der Prozessdatenkommunikation,
- Verbindungsabbruch,
- Runtime-Fehler in der CICS-Steuerung,
- Verbindungsverzögerung und/oder Verbindungsunterbrechung,
- Einbruch in die Cloud und Modifizierung der Dateien für CICS-Dienste,
- Einbruch in den Client und Störung des CICS-Betriebs,
- Einbruch in die Cloud und Störung des CICS-Betriebs.

Weiterhin wurden umfangreiche Recherchen zum Thema HTTPS-Protokoll sowie dem SSL-Zertifikatsystem durchgeführt. Darauf aufbauend erfolgte im ersten Schritt die prototypische Umsetzung eines HTTP(S)-Echoservers zur Veranschaulichung der verschlüsselten Datenübertragung im Web, der Analyse, von denkbaren Angriffsmöglichkeiten sowie daraus abgeleitet Empfehlungen für IT-Schutzmaßnahmen.

Darüber hinaus wurden folgende Sicherheitsmaßnahmen untersucht und an CICS-Prototypen umgesetzt:

- Verschlüsselung des HTTP- und WebSocket-Protokolls (WS),
- Erstellung eines Zertifikatsystems für CICS,
- Verschlüsselte Anbindung von Automatisierungsgeräten über hardware-basierte VPN-Lösungen (mit Tosibox) an CICS,
- Nutzung von SSL-Proxies auf Basis von nginx zur Anbindung von Automatisierungsgeräten an CICS,
- Sicherung der Portalumgebung für eine CICS gegen Brute-Force-Angriffe,

Im zweiten Schritt wurde ein Demonstrationsmuster einer sicheren CICS-Plattformumgebung mit HTTPS/WSS aufgebaut, die je nach Bedarf für Testzwecke öffentlich zur Verfügung steht (<https://woass.ccad.eu>).

Bericht:

Bericht „Analyse von IT-Sicherheitsproblemen bei einer CICS-Steuerung“

- **Gegenüberstellung Ziele / Ergebnisse**

Entsprechend dem Ziel des Arbeitspakets erfolgte eine grundsätzliche Analyse der IT-Sicherheitsprobleme und es wurden auch zugehörige IT-Schutzmaßnahmen vorgeschlagen und diese für die Sicherung der Prozessdatenzugänge in den CICS-Prototyplösungen teilweise umgesetzt. Aufgrund der Komplexität der Problematik sowie der dazu erforderlichen Spezialkenntnisse konnten aber nur grundsätzliche Untersuchungen durchgeführt werden. Die dazu umgesetzten Maßnahmen wie z.B. die Nutzung von HTTPS und WSS für die Prozessdatenübertragung konnten bisher noch nicht in praktischen Sicherheitstest ausführlich evaluiert werden. Dazu würden Spezial-

kenntnisse aus der Hacker-Szene benötigt, um tatsächlich die Wirksamkeit der Maßnahmen zu überprüfen.

3.6 Arbeitspaket 6 (FST 1)

Funktionsmusters einer Web-geeigneten IEC 61131-Runtime-Umgebung.

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 6.1: Aufstellung eines Pflichtenhefts für eine CICS Runtime Environment (CICS-RE) unter Berücksichtigung einer Erweiterung auf IEC 6199-Notation und Auswahl der geeigneten Webtechnologien für die Implementierung.

AP 6.2: Analyse, Design und Implementierung des Funktionsmuster CICS-RE unter besonderer Berücksichtigung einer server- und clientseitigen Ausführung.

AP 6.3: Evaluierung und Test der CICS-RE mit ausgewählten IEC 61131-3-Programmen (erstellt mit einem klassischen SPS-Programmierungswerkzeug) in praxisnahen Beispielszenarios.

AP 6.4: Entwicklung von Vorschlägen, um klassische SPS-Werkzeuge auch für das Debugging von CICS nutzen zu können, evtl. auch Durchführung von Tests.

AP 6.5: Erarbeitung von Erweiterungsvorschlägen für die CICS-RE, um diese auch für IEC 6199-Systeme nutzen zu können. Dazu sollen auch bereits erste Tests mit verfügbaren IEC 61199-Werkzeugen durchgeführt werden.

▪ Durchgeführte Arbeiten

Der Schwerpunkt der Arbeiten in diesem AP lag in 2016. Bereits 2015 konnte aber ein erster Prototyp einer CICS-CM (client-basierte Lösung) implementiert werden. 2016 erfolgte Implementierung und Test einer CICS-SMM (server-basierte Lösung), bestehend aus einer CICS-Runtime, einem Compiler für IEC61131-Steuerungsprogramme in ST und einem CICS-Router. Alle Komponenten wurden als instanzierbare Dienste realisiert und beispielhaft in die IIoT-Plattform WOAS (<http://woas.ccad.eu>; Betreiber: Forschungsstelle 1) integriert. Es erfolgten umfangreiche Tests in praxisnahen Beispielszenarios. Zusätzlich wurden Vorschläge entwickelt, um klassische SPS-Werkzeuge für das Engineering von CICS-Steuerungen zu nutzen.

Für die Implementierung einer CICS-SMM erfolgte die Entwicklung eines Pflichtenheftes. Basis bildete dazu u.a. eine implementierte α -Version (noch keine Funktionstests) eines ST-Compilers mit Node.js und Virtualisierung mit Docker.

▪ Erzielte Ergebnisse

Im Rahmen des CICS-Projekts erfolgten zwei prototypische Implementierungen für eine CICS-Steuerung:

- CICS-Steuerung im Client Mode (CM),
- CICS-Steuerung im Server-based Mixed Mode (SMM).

CICS-Steuerung im Client Mode (CM)

Bei der CM-Lösung wird die CICS-Steuerung vollständig als Instanz im Client (Webbrowser) ausgeführt. CICS-RT und CICS-R können in diesem Fall als ein gemeinsamer Dienst realisiert werden und werden im Client (Webbrowser) ausgeführt. Bild 21 zeigt die CICS-Steuerung im Client Mode.

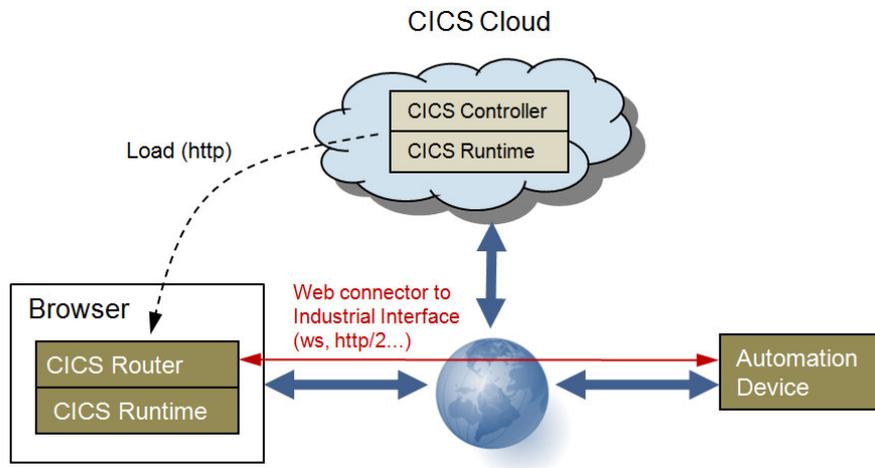


Bild 21: CICS-Steuerung im Client Mode (CM)

Die Implementierung der CICS-Steuerung erfolgte als JavaScript-Objekt, welches aus einer Cloud (CICS-Cloud) auf den Client geladen wird. Die Kommunikation zwischen der CICS-Steuerungs-Instanz und den Automatisierungsgeräten erfolgt über ein Device-Gateway als Webconnector mit z.B. Zugang zu Modbus TCP oder OPC über ein Webprotokoll (Websocket, WebRTC).

Mit der CICS-CM kann ein IEC 61131-Programm als PLCopen XML-Programm in AWL-Notation als Steuerungsprogramm genutzt werden. Dazu wurde ein Interpreter auf Basis JavaScript für das PLCopen XML-Programm konzipiert und implementiert.

Bericht:

Bericht „JavaScript Runtime-Umgebung für eine programmierbare Web-Steuerung“

Es erfolgte die Integration der CICS-CM als Steuerungsdienst in die IIoT-Plattform WOAS. Damit kann die CICS-CM vollständig als instanzierbarer Dienst genutzt werden. Es wurden verschiedene Beispielszenarios aufgebaut und evaluiert, die auch öffentlich bei Bedarf zur Verfügung stehen.

Bericht:

WOAS-Handbuch „Teil – Beschreibung des Dienstes CICS-CM für die IIoT-Plattform WOAS“

Zum Test der CICS-CM wurde eine im CCAD vorhandenen Bearbeitungs- und Prüfstation als CICS-Demonstrator teilweise neu aufgebaut bzw. modifiziert. Dazu wurde die vorhandene Station u.a. mit IoT-Gateways ausgerüstet, um einen Web-basierten Zugriff auf die Prozessdaten zu erhalten.

Bericht:

Schaltungs- und Umbaudokumentation für die Bearbeitungs- und Prüfstation

Entsprechend der Prioritätsvorgabe für die prototypischen CICS-Implementierungen aus AP 2 erfolgte die Entwicklung eines Pflichtenheftes für eine CICS-SMM, die für ein IEC 61131-Programm einen ST-Compiler aus der Cloud nutzen soll.

Bericht:

Pflichtenheft: CICS Server-based Mixed Mode

CICS-Steuerung im Server-based Mixed Mode (SMM)

Design und Implementierung des Prototyps einer CICS-SMM, die ein IEC 61131-Programm als PLCopen XML-Programm in ST-Notation mit einem ST-Compiler abarbeiten kann. Dazu wurden folgende Komponenten implementiert:

- CICS-Compiler für ein PLCopen XML-Programm in ST-Notation,
- CICS-Runtime zur zyklischen und ereignisbasierten Abarbeitung des Steuerungsprogramms,
- CICS-Router zur flexiblen Anbindung der physischen Automatisierungsgeräte an die CICS-Steuerung.

Bei der SMM-Lösung wird die CICS-Runtime-Instanz im Server (Cloud) und die CICS-Router-Instanz im Client (Webbrowser) ausgeführt.

Eine direkte Prozessdatenkommunikation erfolgt auch hier nur zwischen Client und Geräten. Es gibt zwischen CICS-R und CICS-RT einen speziellen Block-Kanal zur Übertragung der E/A-Abbilder. Die Prozessdaten werden über diesen Kanal als JSON-Strings mittels Websocket übertragen. Die CICS-Runtime wird über einen HMI-Proxy auf dem Client bedient. Bild 22 zeigt im Detail die Struktur einer CICS-SMM- Lösung.

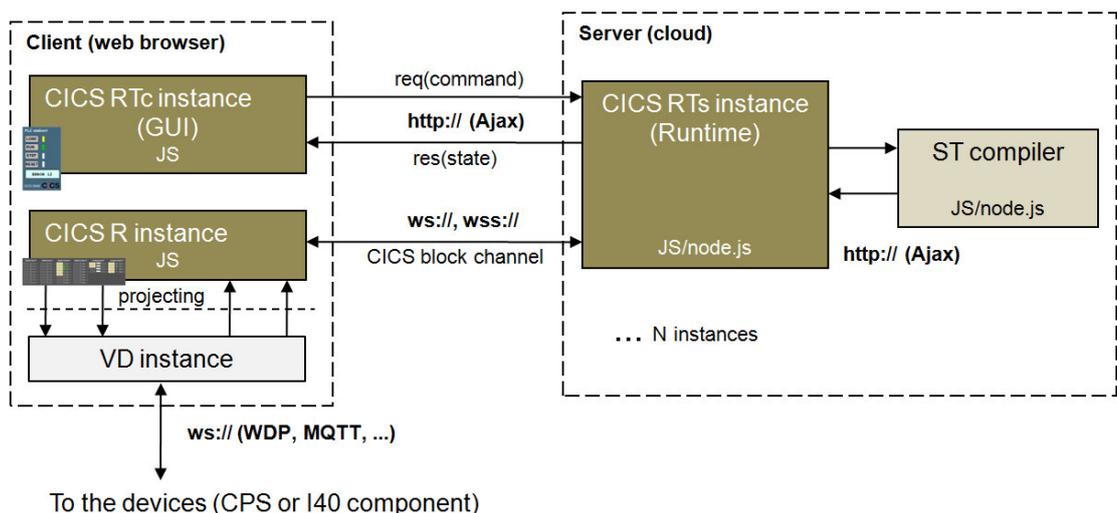


Bild 22: Detailstruktur einer CICS-SMM-Steuerung (RTs – server-basierter Teil, RTc – client-basierter Teil)

Implementierungstechnisch ist die CICS-SMM-Lösung eine verteilte aufwendige Lösung. Die Prozessdatenverbindung zu den Geräten kann gleichfalls lokal erfolgen und

trotzdem kann die CICS-RT die volle Leistung des Servers nutzen. Eine dynamische Re-Konfigurierung ist einfach möglich.

Nach Aufruf der entsprechenden Webseite im Client wird im Server eine CICS-RTs-Instanz als Runtime-Maschine dynamisch erzeugt und gleichzeitig das zugehörige Steuerungsprogramm durch den ST-Compiler im Server in ein ausführbares JS-Programm kompiliert. Dieses JS-Programm wird dann in die CICS-RTs-Instanz geladen. Die VD-Instanz im Client stellt als virtuelles Gerät (VD – Virtual Device) die Verbindung zu den Automatisierungsgeräten her.

Im Server können prinzipiell beliebig viele CICS-RTs-Instanzen laufen, die jeweils mit anderen Clients zusammenarbeiten können. Der ST-Compiler arbeitet für alle CICS-RTs-Instanzen gemeinsam und kann auch in einen weiteren Server (Cloud) verlagert werden.

Einige Vorteile der SMM-Lösung sind:

- Über unterschiedliche CICS-Router, die auch dynamisch erzeugt werden können, ist eine dynamische Umkonfigurierung des Steuerungssystems online möglich. Dies bildet eine Grundlage für adaptive und selbstanpassende Systeme.
- Die CICS Runtime kann die volle Leistungsfähigkeit und Stabilität des Servers nutzen.
- Die Leistungsfähigkeit des Clients spielt nur noch eine untergeordnete Rolle. Im Client erfolgt nur das Routing der E/A-Prozessdaten.
- Es können dienstfähige C3x-Steuerungen aufgebaut werden, da der Server als Service-Broker im System arbeiten kann. Das Cloud-Modell SaaS (Software as a Service) bzw. CaaS (Control as a Service) ist realisierbar.

Berichte:

- Bericht „Entwicklung eines Compilers als Cloud-Dienst mit Node.js und Virtualisierung mit Docker, für die industrielle Programmiersprache Strukturierter Text nach der Norm IEC 61131-3 für industrielle Websteuerungen“
- Bericht „Entwicklung und Analyse einer auf Cloud- und Webtechnologien basierenden Runtime-Umgebung zur dynamischen Steuerung einer Automatisierungsanlage“

Die CICS-SMM wurde als frei instanzierbarer Steuerungsdienst in die IIoT-Plattform WOAS integriert. Es wurden verschiedene Beispielszenarios aufgebaut und evaluiert, die auch öffentlich bei Bedarf zur Verfügung stehen.

Bericht:

WOAS-Handbuch „Teil - Beschreibung des Dienstes CICS-SMM für die IIoT-Plattform WOAS“

Es liegen umfangreiche Ergebnisse zu Performance- und Zeituntersuchungen für die beiden CICS-Prototypen vor. Dabei wurden verschiedene Anwendungsszenarios wie z.B. Private Cloud, Public Cloud sowie Betrieb im lokalen Netz und in Weitverkehrsnetzen (Internet) untersucht. Insgesamt kann basierend auf diesen Ergebnissen eingeschätzt werden dass eine CICS-Steuerung im lokalen Betrieb Reaktionszeiten von (30

... 50) ms und über das Internet Reaktionszeiten von (100...150) ms mit einer Wahrscheinlichkeit von ca. (90 .. 95)% realisieren kann

Berichte:

- Bericht „CICS-CM Demonstrator - Performance and reaction time“
- Bericht „CICS-SMM – Zeitverhalten“

CICS und IEC 61499

Entsprechende Untersuchungen zur prinzipielle Bedeutung der IEC 61499 für das CICS-Projekt ergaben folgende Aspekte/Probleme:

- Die IEC 61499 ist primär für die Erstellung von Steuerungsprogrammen für verteilte und unterschiedliche Hardwar-Steuerungen vorgesehen. Alle CICS-Steuerungen existieren nur als Software-Instanzen und sind damit im grundsätzlichen Aufbau identisch. Verteilte Hardware (Automatisierungsgeräte) kann problemlos mit einer oder verschiedenen CICS-Instanzen verbunden werden, so dass bereits mit einem IEC 61131-Programm ein im Netz verteiltes Steuerungssystem einfach aufgebaut werden kann. Die Bedeutung der IEC 61499 würde damit reduziert auf die ereignisbasiert Programmiermethodik.
- Bisher wird die IEC 61499 nur sehr eingeschränkt in der Industrie eingesetzt. Es gibt dazu auch nur wenige Programmierwerkzeuge und Runtime-Maschinen.

Das CICS-Projekt konzentrierte sich auf die weltweit akzeptierte IEC 61131, d.h. auf eine akzeptierte und verbreitete Programmiermethodik für Steuerungsprogramme, um dem potenziellen Anwender trotz der völlig neuen Ablaufumgebung (Steuerung als Dienst aus der Cloud) zumindest eine gewohnte Programmierumgebung bereitzustellen. Die IEC 61499 wurde im Projekt deshalb nur am Rande betrachtet. Die ursprünglich im AP 6.5 geplanten Personalressourcen für die Anwendung der IEC 61499 in CICS (HPA A: 1 PM) wurde deshalb zusätzlich für die umfangreichen CICS-Implementierungsarbeiten genutzt.

Sollte die IEC 61499 zukünftig auf eine höhere Industrieakzeptanz stoßen, könnte aber die Verbindung dieses Standards mit der CICS-Methodik sehr effizient sein, da für die Umsetzung eines IEC 61499-Programms auf verschiedene CICS-Instanzen nur noch eine einheitliche Runtime-Maschine benötigt wird.

▪ Gegenüberstellung Ziele / Ergebnisse

AP 6 war für die FST 1 das Schwerpunkt-AP, da hier entsprechende CICS-Funktionsmuster konzipiert, implementiert und evaluiert werden mussten. Alle Ziele der Arbeitspakete 6.1 – 6.3 konnten erfolgreich realisiert werden. Die Funktionsmuster von zwei CICS-Steuerungen in den Betriebsarten Client Mode und Server-based Mixed Mode stehen in unterschiedlichen Beispielszenarien zur Verfügung und wurden umfangreich evaluiert.

AP 6.4 (Debugging von CICS mit klassischen SPS-Werkzeugen) sowie AP 6.5 (Integration der IEC 61199) wurde aus den o.a. Gründen nur prinzipiell untersucht und behandelt. Insbesondere wurden dazu bisher keine Tests durchgeführt. Die dafür geplanten

ten Personalressourcen wurden für AP 6.1 – AP 6.3 genutzt, da die Realisierung von zwei unterschiedlichen Runtime-Maschinen für die CICS-Typen CM und SMM erhebliche Implementierungsressourcen erforderte (im Projektantrag war ursprünglich nur eine CICS Runtime-Maschine geplant).

3.7 Arbeitspaket 7 (FST 2)

Funktionsmuster Web-Connector und Evaluierung.

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 7.1: Analyse und Design eines „schlanken“ CICS-Web-Connectors zum ereignisbasierten bidirektionalen Zugriff auf Prozesssignale in der Feldebene eines technischen Prozesses bevorzugt über Websocket-Verbindungen.

AP 7.2: Implementierung eines Funktionsmusters für den CICS-Web-Connector auf Basis einer am Markt verfügbaren Low-Cost-Hardware (evtl. auch Eigenentwicklung).

AP 7.3: Test und Evaluierung des Web-Connectors an Funktionsbeispielen

▪ Durchgeführte Arbeiten

Zunächst wurden erste Anforderungen an das geplante Funktionsmuster des Web-Connector gesammelt und bewertet. Ausserdem wurden auf Basis der Beispielimplementierungen (AP 4) Analysen für das Design eines „schlanken“ CICS Web Connectors durchgeführt. Im Anschluß daran wurde die Architektur des Web-Connectors für die Integration der Automatisierungskomponenten entworfen. Dieser basiert auf dem Web-RTC Prokoll und besteht aus 3 Komponenten.

▪ VirtualDevice für WebRTC (VD4RTC)

Für die Integration einer WebRTC Schnittstelle wurde ein eigenes Virtual Device entworfen. Dieses ermöglicht die Kopplung von CICS mit WebRTC fähigen Geräten. Die dafür entworfene VD4RTC API schafft eine Abstraktion des WebRTC Protokolls und ermöglicht nicht nur Peer to Peer Kommunikation sondern auch Multipeer Kommunikation, wie sie z.B. in Mesh Topologien verwendet wird. Das Konzept der Abstraktion ermöglicht es neben WebRTC auch websocket zu nutzen falls es die Infrastruktur erfordert. Die API enthält Funktionen zur Initialisierung, zum Aufbau und zur Nutzung der Kommunikationsschnittstelle.

▪ Websocket Signaling Server

Auf Basis einer Websocket Verbindung wurde ein Signalisierungsserver entworfen, der bei WebRTC für den Verbindungsaufbau zwischen 2 und mehreren Teilnehmern notwendig ist. Dieser Server kann sowohl lokal als auch in einer beliebigen Cloud Umgebung betrieben werden.

▪ WebRTC Node-RED Modul

Da es während der Durchführung des Projekts keine industrielle Automatisierungskomponente mit WebRTC Schnittstelle gab war es notwendig eine Brücke zwischen existierenden Automatisierungskomponenten und WebRTC zu schaffen. Hierzu wurde auf die Open Source Plattform Node-RED zurückgegriffen, die zwar kein WebRTC aber eine Reihe von der Industrie genutzte Protokolle wie OPC UA

oder Modbus TCP unterstützt. Im Rahmen des AP 7 wurde nun ein WebRTC Node RED Module entworfen und implementiert, das ein Routing zwischen WebRTC und den Industrie-Protokollen ermöglicht. Damit konnte zwischen dem VD4RTC und Node RED eine WebRTC Verbindung aufgebaut werden, deren Inhalte dann an eine AUTomatisierungskomponente weitergereicht wurde.

Die Komponenten wurden jeweils prototypisch implementiert und getestet. Der Webconnector unterstützt aktuell folgende Funktionen:

- Gesicherte Peer to Peer Kommunikation auf Basis von UDP
- Gesicherte Peer to Multipeerkommunikation auf Basis von UDP
- Videostreaming in Verbindung mit einem Dienst zur Echtzeitbildauswertung (falls beide Peers Fähigkeit der Medienverarbeitung/Nutzung haben)
- Kommunikation zw. WebRTC und beliebigen von Node RED unterstützten Protokollen
- Analyse der Zustände aller beteiligten Peers

Für einen produktnahen Ausbau wurden folgende Funktionen konzipiert, aber im Rahmen des Projekts nicht umgesetzt:

- Wechselmöglichkeit zwischen gesicherten und ungesicherten Betrieb
- Hybride Unterstützung von WebRTC und WebSocket durch Erweiterung des Signalisierungsservers
- Anwenderfreundliche Visualisierung der Zustände der Verbindungen
- Funktionen zum automatischen Wiederaufbau der Verbindung.

▪ Erzielte Ergebnisse

Die Architektur des oben beschriebene Web-Connectors ist in Bild 23 dargestellt. Als Beispielapplikation wurde der in AP 4 umgesetzte Demonstrator gewählt.

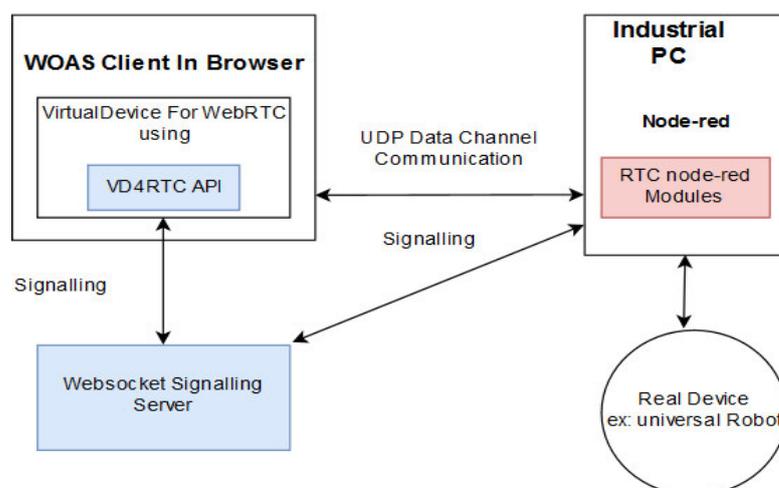


Bild 23: Architektur WebConnector

Hierbei kommt der Kommunikationsfluß nach Bild 24 zum Einsatz.

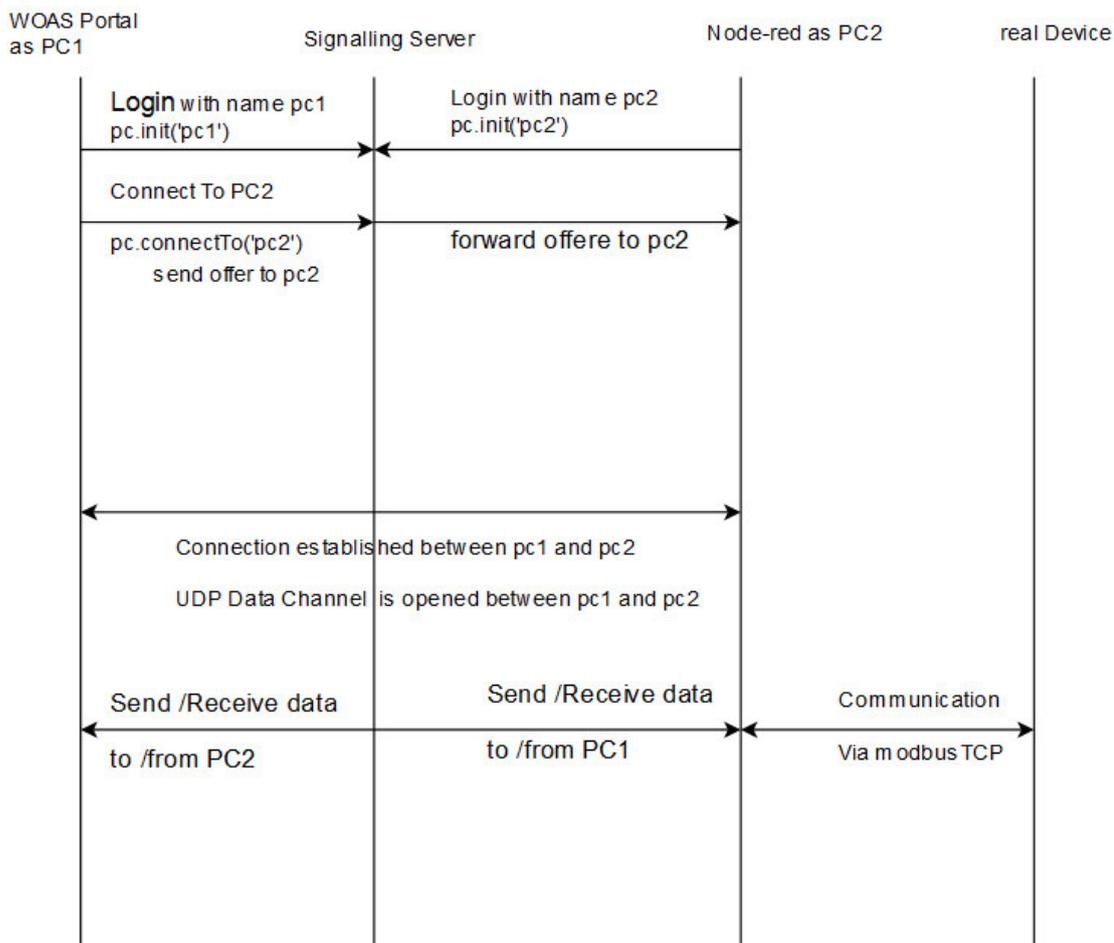


Bild 24: Kommunikationsfluß des WebConnectors

Beide Kommunikationspartner (hier das WOAS Protal als PC1 und Node-Red als PC2) müssen sich zunächst beim Signalisierungsserver einloggen. Danach stellt der PC1 eine Verbindungsanfrage, die an den PC2 weitergeleitet wird. Wenn der Signalisierungsserver die Verbindung zwischen den beiden Partnern aufgebaut hat sind diese direkt miteinander verbunden. Die VD4RTC API wurde als Javascript Bibliothek umgesetzt und sowohl für die Implementierung des VD's als auch des Node RED Moduls verwendet. Eine Übersicht der innerhalb der API implementierten Funktionen und Event Listener ist in Tabelle 10 aufgeführt.

Tabelle 10: Übersicht der Funktionen und Event Listener des Webconnectors

init (string)	This function takes string parameter for login process.
connectTo (string)	This function takes string parameter to connect to another peer.
getLocalStream ()	This function takes no parameter . It will get the local stream url of the peer if peer has any video media sources.
getRemoteStream (string)	This function takes string parameter . It will get the Remote stream url of the remote peer if peer has already logged in to signaling server and it has any video media sources.

getPeerIds()	This function takes no parameter . It will get the all connected Remote login names if remote peer has already logged in to signaling server.
getMyId()	This function takes no parameter . It will return peer name itself.
getRTC(string)	This function takes string parameter . It will return the whole remote peer object. It is useful to observe the channel connection states of remote peer objects.
sendData(string, string)	This function takes two string parameter . First string is the remote peer name and second string is the message. This function is used to send data to remote peers.
onConnect(function(loginname) { })	Event dispatched when pc.init('pc1') executes.
onCameraAccess(function(localStreamURL) { })	Event dispatched if peer has video media source.
onUserConnect(function(remote peer name) { })	Event dispatched if remote peer get connected to local peer.
onData(function(data) { })	Event dispatched when data is arrived on data channel.
onError(function(error) { })	Event dispatched when error is occurred.

Aufgrund der im Web-Connector eingeführten Abstraktionschicht kann dieser für künftige Protokolle wie z.B. QUIC erweitert werden.

- Gegenüberstellung Ziele / Ergebnisse

Die im Antrag geplante direkte Erweiterung der Sensoren, Aktoren zu Cyber Physical Systemen konnte aufgrund des begrenzten Umfangs des AP's nicht wie geplant umgesetzt werden. Anstatt einen neuen Sensor / Aktor zu entwerfen und mit einem Web-Connector auszustatten, wurde das im Bereich der IoT Gateways zunehmend eingesetzte Node RED verwendet um mit einer zusätzlichen Brücke die Verbindung zwischen dem Web-Connector und existierenden Aktoren und Sensoren mit industriellen Kommunikationsschnittstellen zu schaffen. Damit konnte eine Ende zu Ende Kommunikation erfolgreich aufgebaut werden, allerdings mussten durch die zusätzliche Komponente weitere Latenzen in Kauf genommen werden.

3.8 Arbeitspaket 8 (FST 1)

CICS-Demonstrator.

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 8.1: Entwicklung eines Konzepts sowie Vorbereitung des Aufbaus des CICS-Demonstrators als Multivendor-Referenzanlage aus dem Bereich Fertigungs-/Montageautomatisierung und/oder Prozessautomatisierung zur Verifizierung und Validie-

rung eines CICS-Steuerungssystems. Die Prozesssignale (digitale und analoge) sollen im Demonstrator von unterschiedlichen Sensor/Aktor/Geräteschnittstellen geliefert werden.

AP 8.2: Aufbau, Inbetriebnahme und Test des CICS-Demonstrators.

AP 8.3: Die CICS-Cloud wird mit unterschiedlicher Netzinfrastruktur getestet (Private Cloud im Intranet, Public Cloud im Internet). Als Privat Cloud kommt das WOAS-Cloud-System aus dem IGF-Projekt WOAS zum Einsatz. Für den Test in der Public Cloud ist ein im Markt verfügbares Cloud-System vorgesehen.

AP 8.4: Erstellung eines Evaluierungsberichts über die Anwendbarkeit von CICS sowie der zugehörigen Architektur eines CICS-Steuerungssystems unter Berücksichtigung von Zeitverhalten, Deterministik, Zuverlässigkeit und Sicherheit sowie Engineering-Aufwand, Nutzerfreundlichkeit, Wartungs- und Betriebsverhalten der damit realisierten Steuerungssysteme in unterschiedlichen IP-Netz-Typen.

- Durchgeführte Arbeiten

Es wurden die folgenden CICS-Demonstratoren bzw. Testszenarios aufgebaut und getestet:

CICS-Demonstrator 1:

Eine Montagestation für Modellautos mit fünf Teilstationen wurden in einem Beispielprojekt in der IIoT-Plattform WOAS mit fünf Instanzen einer CICS-SMM ausgerüstet und steht öffentlich für Testzwecke zur Verfügung. Die CICS-Steuerungen laufen in zwei Beispielszenarios wahlweise in einer Private Cloud (Server der Hochschule Düsseldorf: <http://woas.ccad.eu>) und in einer Public Cloud beim Anbieter STRATO (<https://woass.ccad.eu>). Durch die CICS-Steuerungen werden 135 Prozessdaten der Anlage gesteuert.

CICS-Demonstrator 2:

Für laufende Test- und Präsentationszwecke wurde eine mobile Bearbeitungs- und Prüfstation mit einer CICS-CM sowie wahlweise mit einer CICS-SMM ausgerüstet. Für die CICS wurden gleichfalls unterschiedliche Cloud-Strukturen getestet (Private Cloud in der Hochschule, Private Cloud bei Formware, Public Cloud bei STRATO, lokale Cloud vor Ort).

CICS-Demonstrator 3:

Für eine Demonstration beim PA-Mitglied Formware wurde ein Remote-Szenario mit einer CICS-SMM gesteuert. Dabei arbeitete die Bearbeitungs- und Prüfstation in Düsseldorf mit einem Roboter in Nußdorf zusammen. Damit konnte die Fähigkeit einer CICS zum Aufbau global verteilter Steuerungssysteme demonstriert werden. Der Aufbau dieses Demonstrators erfolgt gemeinsam mit der Forschungsstelle 2 (Roboter).

CICS-Demonstrator 4:

Für die Messe SPS/IPC/Drives 2016 erfolgte gemeinsam mit Forschungsstelle 2 Aufbau und Betrieb einer Fertigungszelle mit zwei Bearbeitungs- und Prüfstationen sowie

einem Beschickungsroboter. Zur Steuerung der Teilstationen sowie zur Synchronisierung mit dem Roboter wurden zwei CICS-SMM-Instanzen eingesetzt.

Weitere Testszenarios:

Es wurden weitere Testszenarios aufgebaut und getestet:

- Testszenario gemeinsam mit Siemens Köln, bei dem eine CICS-SMM eine Demo-Anlage mit Antriebstechnik (Paketsortieranlage) steuert.
- Testszenario für die Trainings- & Forschungsfabrik Fab21 des CCAD, bei der eine CICS-SMM in der Zone C (Fertigungsautomatisierung) verschiedene Stationen steuert.

Zur Anbindung der Gerätetechnik der Demonstratoren an das IP-Netz wurden verschiedene industrielle Gateways getestet sowie auch eigene Gateways entwickelt und evaluiert.

- **Erzielte Ergebnisse**

Die CICS-Komponenten (CICS RT und CICS-R) sind als Dienste aufgebaut und werden als JavaScript-Objekte instanziiert und über einheitliche und konsistente Methoden genutzt. Sie können deshalb sowohl in einer verfügbaren IIoT-Plattform als auch auf Stand-Alone-Webseiten genutzt werden. Für die CICS-Demonstratoren wurde die frei verfügbare IIoT-Plattform WOAS genutzt (<http://woas.ccad.eu>). Die entsprechenden Testanwendungen wurden in der IIoT-Plattform browser-basiert in einem EDIT-Modus projektiert.

Die CICS-CM wurde erfolgreich am *CICS-Demonstrator 2* getestet und 2015 auf der Automatisierungsmesse SPS/IPC/Drives in Nürnberg vorgestellt (Bild 9).

Das SPS-Programm arbeitet bei der Anwendung in Bild 25 als zyklische Zustandsmaschine, welche auf die Anwesenheit eines Werkstücks in Position 1 wartet, danach weitere Aktionen in den anderen Positionen durchführt und den Rundtisch wieder nach Position 1 zurückführt. Das IEC 61131-3-Programm besteht aus 253 Zeilen AWL-Code und nutzt 60 digitale und analoge Prozessvariablen. Die Anbindung der E/A-Module (von W&T) der Station sowie des Servoantriebs für den Drehtisch an die CICS-Steuerung erfolgt über IoT-Gateways für die proprietäre TCP-Schnittstellen von W&T sowie über Modbus TCP für den Servoantrieb.

Einige Vorteile der CM-Lösung sind:

- Der Server ist zur Laufzeit nicht mehr erforderlich. Der Server bzw. die Cloud dient nur zur Speicherung der CICS-Dienste einschl. der Steuerungsprogramme.
- Die Prozessdatenkommunikation zwischen den Geräten und der CICS-Steuerung kann auf das lokale Netz begrenzt werden, wenn sich der Client ebenfalls in diesem Netz befindet.
- Auf einem Client können, je nach Leistung des Client, mehrere CICS-Steuerungs-Instanzen laufen und damit unterschiedliche CPS-Komponenten (Geräte) steuern.
- Prinzipiell ist jeder beliebige Client (PC, Tablet, Smartphone) in der Lage als CICS-Steuerung zu arbeiten.



a)



b)

CICS-CM 2 2 1 1	Digital Input 0	Digital Input 1	Digital Output 0	Digital Output 0	Analog Input 0	Analog Output 0
LOAD	PrtAva %IX0.0	%IX1.0	TstDwn %QX0.0	St0 %QX1.0	TablePd 50 %ID0.0	TablePd 60 %QD0.0
RUN	TstIsD %IX0.1	%IX1.1	Fixer_ %QX0.1	St1 %QX1.1	%ID0.1	0 %QD0.1
STEP	TstIsP %IX0.2	%IX1.2	DrDwn_ %QX0.2	St2 %QX1.2	%ID0.2	0 %QD0.2
RESET	DrIsUp %IX0.3	%IX1.3	DrOn_Q %QX0.3	St3 %QX1.3	%ID0.3	0 %QD0.3
	DrIsPo %IX0.4	%IX1.4	DrUp_Q %QX0.4	St4 %QX1.4		
	DrIsDo %IX0.5	%IX1.5	Flippe %QX0.5	St5 %QX1.5		
	%IX0.6	%IX1.6	%QX0.6	St6 %QX1.6		
	%IX0.7	%IX1.7	%QX0.7	St7 %QX1.7		

c)

Bild 25: Anwendungsbeispiel mit dem CICS-Demonstrator 2 für eine CICS-Steuerung im Client Mode (CM); a) Bearbeitungs- und Prüfstation; b) E/A-Module, IoT-Gateway und Switch; c) HMI-Proxy im Webbrowser für die Bedienung der CICS-CM

Die CICS-SMM-Steuerung wurde in Anwendungsbeispielen mit dem CICS-Demonstratoren 1 – 4 umfangreich getestet. Eine Anwendung steuert z.B. die Montage-/Demontagestation für Modellautos der FST 1 bestehend aus fünf Teilstationen entsprechend *CICS-Demonstrator 1* (Bild 26).

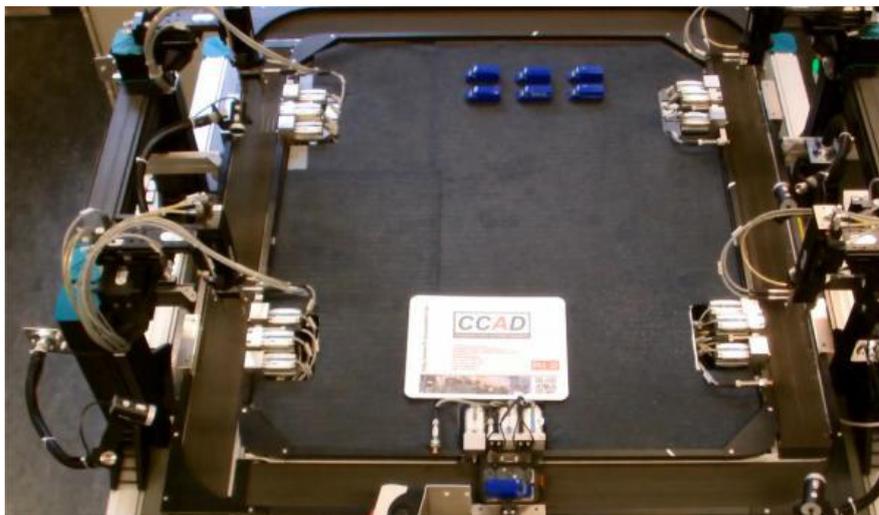


Bild 26: Montagestation für Modellautos mit fünf Teilstationen (CICS-Demonstrator 1)

Die Anlage nach Bild 26 wird im Standardbetrieb mit fünf klassischen SPS-Steuerungen (pro Teilstation jeweils eine SPS) von Siemens und Phoenix Contact betrieben. Die durchschnittliche Montage-/Demontagezeit für ein Modellauto beträgt damit 40 s.

Für den CICS-Test wurden alle fünf Teilstationen über Device Gateways/Webconnectoren für OPC DA an das Internet angeschlossen. Damit stehen alle 135 Prozessdaten der Stationen in der Web-Welt zur Verfügung. Die IEC 61131-Steuerungsprogramme für die fünf Stationen beinhalten insgesamt 1355 Zeilen Quellcode in ST.

Die Projektierung des CICS-Steuerungssystems für die Anlage erfolgte mittels der IIoT Plattform WOAS vollständig im Webbrowser. Für jede der fünf Montage-/Demontagestationen wird eine eigene CICS-Steuerungs-Instanz genutzt. Alle fünf HMI-Proxies für diese Instanzen sind auf einer Webseite positioniert.

Mit der CICS-SMM-Steuerung aus der Cloud erhöht sich die Montage-/Demontagezeit für ein Modellauto um ca. 15%. Dafür können aber fünf SPS-Steuerungen als Hardware sowie deren Wartung entfallen. Tests mit einer Anbindung der Geräte über OPC UA und einem Webconnector auf Basis von Node-RED ergab ähnliche Ergebnisse.

Zum *CICS-Demonstrator 2* liegen eine Reihe von Testergebnissen vor. Dazu gehören u.a. Ergebnisse zu Funktionstests mit unterschiedlichen am Markt verfügbaren Web-Gateways (WISE-5231, NIO100Y) sowie mit selbst entwickelten Gateways auf Basis von Node-RED (z.B. MQTT auf Modbus TCP).

Der *CICS-Demonstrator 3* wurde auf den Firmentagen von Formware in Nußdorf am 22/23.09.16 erfolgreich präsentiert.

CICS-Demonstrator 2 wurde mit einer CICS-CM auf der Messe SPS/IPC/Drives 2015 und *CICS-Demonstrator 4* mit einer CICS-SMM auf der SPS/IPC/Drives 2016 in Nürnberg auf dem Fraunhofer-Gemeinschaftsstand mit großem Besucherinteresse erfolgreich präsentiert.

Bild 27 zeigt als Beispiel die Komponentenstruktur des Messeaufbaus von 2016.

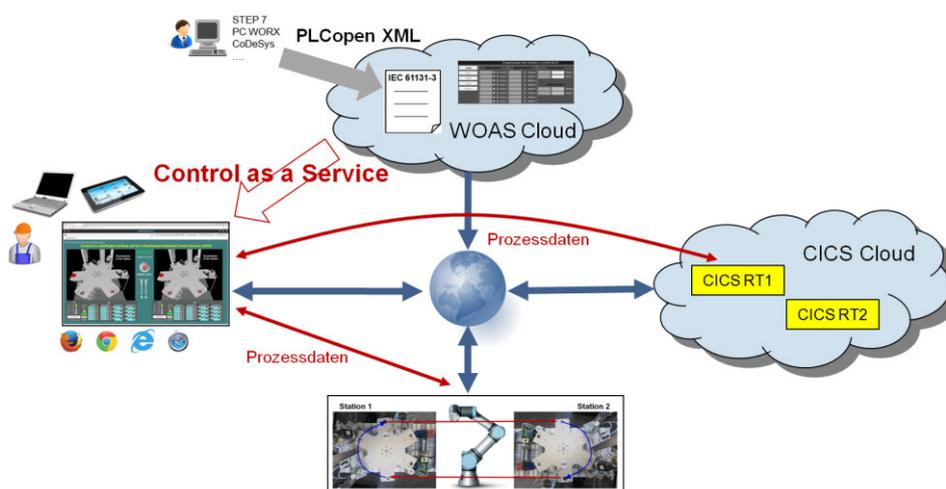


Bild 27: Komponentenstruktur des CICS-Demonstrator 4 auf der SPS/IPC/Drives 2016

Berichte:

- Bericht „Entwicklung und Test eines Industrie 4.0 Gateways für den Internetzugriff auf Modbus TCP und proprietäre TCP Protokolle“
 - Bericht „OPC-UA-Gateway auf Basis von node-RED über die Protokolle WebSocket und MQTT“
 - Bericht „NodeRED Modbus TCP Gateway Dokumentation“
 - Bericht „Test WISE-5231“
 - Bericht „Test einer cloudbasierten Steuerung zur Automatisierung einer Produktionsanlage“
 - Schaltungsunterlagen und Aufbaubeschreibungen für die Demonstratoren
-
- Gegenüberstellung Ziele / Ergebnisse

Gegenüber dem ursprüngliche Ziel des Arbeitspakets zum Aufbau nur eines CICS-Demonstrators wurden zwei ständig verfügbare CICS-Demonstratoren sowie zwei weitere temporäre CICS-Demonstratoren aufgebaut und getestet. Damit wurden die Ziele des Arbeitspaketes vollständig erreicht. Die Evaluierung ergab z.B. für den CICS-Demonstrator 1 (Montage-/Demontagestation für Modellautos), dass der Betrieb der Anlage mit fünf CICS-Steuerungen aus einer öffentlichen Cloud (STRATO) über das Internet gegenüber dem klassischen lokalen Betrieb mit fünf SPS-Steuerungen (von Siemens und Phoenix Contact) die Montage-/Demontagezeit nur um 15% erhöht (45 s mit CICS gegenüber 39 s mit lokaler SPS). Damit konnte die Anwendbarkeit des CICS-Konzepts für Steuerungen mit „weichem“ Echtzeitverhalten erfolgreich nachgewiesen werden.

3.9 Arbeitspaket 9 (FST 1 + FST 2)

Know-How-Transfer und Vorbereitung der wirtschaftlichen Verwertung der Forschungsergebnisse.

Das Arbeitspaket gliederte sich in folgenden Teilaufgaben:

AP 9.1: Vorbereitung eines Arbeitsflyers sowie einer Projekt-Internetpräsenz zum Forschungsprojekt.

AP 9.2: Präsentation eines Fachvortrags auf dem jährlichen VDI/VDE-GMA-Kongress und/oder SPS/IPC-Drives-Kongress sowie mindestens auf einem weiteren internationalen Fachkongress.

AP 9.3: Vorbereitung und Durchführung eines Workshops zur weborientierten Automatisierung mit interessierten Industrievertretern.

AP 9.4: Präsentation der CICS-Architektur sowie einer Funktionsmuster-Lösung auf der SPS/IPC/Drives (oder Hannover-Messe) auf einem eigenen Messestand.

AP 9.5: Anfertigung des Schlussberichts für das Projekt

Für dieses AP sind beide Forschungsstellen gemeinschaftlich verantwortlich.

- Durchgeführte Arbeiten

Im Folgenden wird nur eine Auswahl der Know-How-Transfer-Maßnahmen aufgeführt. Eine vollständige Übersicht findet sich unter Punkt 6.

- Entwicklung von vier Projektflyer sowie ein universell einsetzbares RollUp
- Zum internen Zugang zu den CICS-Dokumenten wurde über die FST 2 ein internes Dokumentenportal angelegt, über die auf Projektdokumente zugegriffen werden kann. CICS-Veröffentlichungen können z.T. aber auch über die IloT-Plattform WOAS heruntergeladen werden.
- Es wurden verschiedene Veröffentlichungen im Rahmen nationaler und internationaler Konferenzen vorbereitet und durchgeführt.
- Ein neuer IGF-Projektvorschlags „Potential, Struktur und Schnittstellen von Big Data-Diensten für die CPS-basierte Automation (Big Data Service for CPS-based Automation - BDAUT)“, der Teilergebnisse aus dem CICS-Projekt für die Datenspeicherung und -analyse von Echtzeitdaten aufgreift, wurde eingereicht und durch die Gutachter prinzipiell auch positiv bewertet. Da die erzielte Bewertungspunktzahl aber für eine zeitnahe Förderung nicht ausreicht, wurde der Antrag zurückgezogen. Die Einreichung eines überarbeiteten Antrags soll in 2018 erfolgen.
- Einreichung eines Forschungsgroßgeräte-Antrags durch die FST 1 (gemeinsam mit dem Fachbereich Medien der HS Düsseldorf) für eine Industrie 4.0-Modellfabrik Fab40. Der Antrag wurde durch die DFG nicht als förderwürdig begutachtet.
- Es erfolgte die Mitwirkung an der Einreichung eines Antrags im NRW-Programm FH-Kompetenz „RIMIT - Rheinisches Institut für Mensch-Interaktion-Technik“. Die FST 1 sollte dabei für die Säule Industrie 4.0 im Arbeits- und Lernraum verantwortlich sein. Dieser Vorschlag wurde aber gleichfalls nicht bestätigt.
- Durchführung von vier Bachelorarbeiten an der HS Düsseldorf zu CICS.
- Durchführung von zwei Masterarbeiten an der HS Düsseldorf zu CICS.
- Durchführung von 15 Projektarbeiten im Masterkurs „Modellierung&Simulation“ zu ergänzenden Automatisierungsdiensten für CICS.
- Im Rahmen verschiedener Besuche von chinesischen Delegationen aus Universitäten und Unternehmen in der FST 1 wurde das CICS-Projekt vorgestellt.
- Durchführung einer Vielzahl von Gesprächen mit interessierten Unternehmen zum Einsatz einer CICS-Steuerung.
- Vorbereitung und Veröffentlichung von insgesamt 24 Beiträgen in Fachzeitschriften sowie nationalen und internationalen Konferenzen

- **Erzielte Ergebnisse**

Neben den Informationen zu CICS auf der Homepage von FST 1 und FST 2 sind im Folgenden eine Auswahl von Transferergebnissen aufgeführt. Eine vollständige Übersicht findet sich unter Punkt 6 und 7 dieses Berichtes.

Info durch Dritte zum Projekt in:

- Computer&Automation, 12/2014, S.6
- atp Newsletter 11/2014
- etz elektrotechnik & automation 11/2014
- VDI nachrichten, 47/2017

Patentanmeldungen:

- Deutsche Patentanmeldung „Steuerungseinrichtung mit einem Steuerungsprogramm und einer Runtime-Maschine zum Betreiben eines Automatisierungsgerätes“; DE 10 2015 221 652.7, Anmeldedatum: 4.11.2015.
- Deutsche Patentanmeldung „Steuerungseinrichtung mit einem Steuerungsprogramm und einer Gerätekonfiguration zum Betreiben eines Automatisierungsgerätes“; DE10 2015 221 650.0, Anmeldedatum: 4.11.2015
- PCT-Anmeldung „Steuerungseinrichtung mit einem Steuerungsprogramm und einer Gerätekonfiguration zum Betreiben eines Automatisierungsgerätes“ PCT/EP2016/076625, Anmeldedatum: 4.11.2016

Veröffentlichung in Fachzeitschriften (Auswahl):

- Industrial Cloud – Status und Ausblick, HMD (2015) 52:647–664, SpringerVerlag (Leitartikel)
- Cloud-basiertes IEC 61131-Steuerungssystem. - SPS-Magazin, Nov. 2015
- Neuer Standard Industrie 4.0 - Die künftige Rolle der IT für die Produktion im deutschen Mittelstand, Unternehmermagazin, 7/2015
- Flexibel steuern mit der Datenwolke. – SPS-Magazin, November 2016
- Steuerungsdienste aus der Cloud auf Basis IEC 61131 - SPS der nächsten Generation? – atp edition 4/2017 (Hauptbeitrag)
- Cloud-basierte Steuerungen – aber Wie? – Computer & Automation, 5/2017

Beitrag auf Konferenzen/Messen (Auswahl):

- Web of Things in an Industrial Environment. - 11th International Conference on Web Information Systems and Technologies WEBIST-2015, Lissabon, Juni 2015
- Cloud-based Industrial Control Services. – 14th International Conference on Remote Engineering & Virtual Instrumentation, New York, März 2017
- Eine SPS-Steuerung als Industrie 4.0-Komponente. - AALE 2016, Lübeck, März 2016

- Konzept und Implementierung einer I40-Komponente für Steuerungen aus der Cloud. - 4. Markt&Technik Industrie 4.0 & Industrial Internet Summit, München, Oktober 2016
- Cloud basierte IEC 61131 Steuerungsdienste auf Basis von Web-technologien. – ASW 2016, Wildau, März 2016
- IEC 61131 & Industrial Internet of Things – Bewährte Praxis mit zukunftsfähigen Technologien verbinden. – VDI-Fachtagung Industrie 4.0, Düsseldorf, Januar 2016
- Messepräsentation des CICS-Projekts auf dem Fraunhofer-Gemeinschaftsstand zur SPS/IPC/Drives 2015 in Nürnberg
- Messepräsentation des CICS-Projekts auf dem Fraunhofer-Gemeinschaftsstand zur SPS/IPC/Drives 2016 in Nürnberg

Studium und Lehre:

- Aufnahme der Informationen zur CICS-Methodik in die Vorlesung „Echtzeitsysteme“ des Bachelorstudiengangs Elektrotechnik/Vertiefung Automatisierungstechnik der Hochschule Düsseldorf ab dem WS 2016/17

4 Verwendung der Zuwendung

- wissenschaftlich-technisches Personal (Einzelansatz A.1 des Finanzierungsplans)

Arbeitspaket 1: Erarbeitung eine Studie zum Anwendungs- und Nutzungspotential von verteilten, Cloud-basierten Steuerungsdiensten

Ergebnisse	Anwendungs- und Nutzungspotential von CICS-Steuerungen; Klassifikation einer CICS-Steuerung als Industrie 4.0 (I40)-Steuerung; Vor- und Nachteile der für CICS geeigneten Kommunikationsstrukturen; Patentrecherche zu CICS	
Personaleinsatz	FST 1: HPS A: 3 PM	FST 2: HPS A: 2 PM
Erfüllungsgrad	100%	

Arbeitspaket 2: Bewertung der Anforderungen und Eigenschaften klassischer Steuerungssysteme in Gegenüberstellung zu potentiellen Lösungen auf Basis von Webtechnologien

Ergebnisse	Erarbeitung eine Studie zum Anwendungs- und Nutzungspotential von verteilten, Cloud-basierten Steuerungsdiensten; SWOT-Analyse zur Strategie für ein CICS-Steuerungssystem	
Personaleinsatz	FST 1: HPA A: 2 PM	FST 2: 0
Erfüllungsgrad	100%	

Arbeitspaket 3: Referenzarchitektur und Schnittstellen

Ergebnisse	Referenzarchitektur und Schnittstellen für ein CICS-System unter Berücksichtigung der zu automatisierenden Prozessklassen; Aufbau und Test einer Simulationsumgebung zur Evaluierung heterogener Kommunikations- Infrastrukturen für den Einsatz in CICS	
Personaleinsatz	FST 1: HPA A: 3,3 PM HPA B: 2 PM	FST 2: HPA A: 9 PM
Erfüllungsgrad	100%	

Arbeitspaket 4: Analyse und Bewertung industrieeüblicher Schnittstellen in der prozessnahen Ebene

Ergebnisse	Untersuchungen und Beispielimplementierungen für Schnittstellen aus dem IoT-Umfeld und Evaluierung hinsichtlich einer Anwendung in CICS	
Personaleinsatz	FST 1: 0	FST 2: HPA A: 6 PM
Erfüllungsgrad	100%	

Arbeitspaket 5: Analyse von IT-Sicherheitsproblemen

Ergebnisse	Analyse hinsichtlich Datensicherheit und Maschinensicherheit bei einem CICS-Steuerungssystem; Vorschläge für erforderliche IT-Schutzmaßnahmen; Aufbau und Test von Beispielszenarios für eine sichere CICS-Plattformumgebung	
Personaleinsatz	FST 1: HPA A: 0,3 PM HPA B: 2,5 PM	FST 2: 0
Erfüllungsgrad	100%	

Arbeitspaket 6: Funktionsmuster einer Web-geeigneten IEC 61131-Runtime-Umgebung

Ergebnisse	Implementierung des Prototyps einer CICS im Client-based Mode (CM) sowie im Server-based Mixed Mode (SMM); Integration der CICS-CM und CICS-SMM als Dienste in die IIoT-Plattform WOAS; Analyse industrietypischer IEC 61131-Programmsysteme für die Erstellung von CICS-Steuerungsprogrammen	
Personaleinsatz	FST 1: HPA A: 10,5 PM HPA B: 4,2 PM	FST 2: 0
Erfüllungsgrad	95% (Es erfolgt keine größere Berücksichtigung der IEC 61499, s. dazu Erläuterungen in Pkt. 1.4, erzielte Ergebnisse)	

Arbeitspaket 7: Funktionsmuster Web-Connector und Evaluierung

Ergebnisse	Aufstellung der Anforderungen sowie Beispielimplementierung und Test eines Funktionsmusters für einen „schlanken“ Web-Connector	
Personaleinsatz	FST 1: 0	FST 2: HPA A: 6,5 PM
Erfüllungsgrad	100%	

Arbeitspaket 8: CICS-Demonstrator

Ergebnisse	Entwicklung, Aufbau und Betrieb von vier CICS-Demonstratoren sowie weiterer Testszenarios für CICS-Steuerungen; Funktionstests mit unterschiedlichen am Markt verfügbaren Web-Gateways; Entwicklung spezifischer IoT-Gateways für Modbus TCP und OPC UA	
Personaleinsatz	FST 1: HPA A: 3,5 PM HPA B: 1,9 PM	FST 2: 0
Erfüllungsgrad	100%	

Arbeitspaket 9: Know-How-Transfer und Vorbereitung der wirtschaftlichen Verwertung der Forschungsergebnisse

Ergebnisse	Veröffentlichung von 24 Beiträgen in Fachzeitschriften und Konferenzen; Realisierung von 4 Bachelorarbeiten und 2 Masterarbeiten zu CICS; Durchführung einer Vielzahl von Gesprächen mit interessierten Unternehmen; weitere Ergebnisse unter Punkt 6/7	
Personaleinsatz	FST 1: HPA A: 1,1 PM HPA B: 0,5 PM	FST 2: HPA A: 1,5 PM
Erfüllungsgrad	100%	

5 Notwendigkeit und Angemessenheit der geleisteten Arbeit**FST 1:**

Alle durchgeführten Arbeiten entsprachen im Wesentlichen den vorgesehenen Arbeitsschritten aus dem Arbeitsplan (Lösungsweg zur Erreichung des Forschungsziels) und waren deshalb, wie geplant, erforderlich und angemessen. Die folgende Übersicht zeigt den tatsächlichen Personaleinsatz für die einzelnen Jahre.

Jahr	Wiss.-techn. Personal (Einzelansatz A.1) [PM bei 100%]	HPA Gruppe A [PM bei 100%]	HPA Gruppe B [PM bei 100%]
2014	3,0	3,0	-
2015	13,6	11,1	2,5
2016	18,2	9,6	8,6
2017	2,4	2,4	-
Summe	37,2	26,1	11,1

Gegenüber dem Einzelfinanzierungsplan wurden für 2015 ca. 19% mehr Personenmonate mit HPA A und ca. 17% weniger HPA B eingesetzt. Grund dafür war die inhaltlich komplizierte Thematik der Forschungsarbeiten, für die nicht genügend qualifizierte HPA B und Hilfskräfte (z.B. mit Programmierkenntnissen in Webtechnologien) in 2015 gefunden werden konnten.

2016 wurden gegenüber dem Einzelfinanzierungsplan ca. 6% mehr Personenmonate mit HPA A und ca. 4% weniger mit HPA B gleichfalls wegen der komplizierten Thematik der Forschungsarbeiten eingesetzt.

Gegenüber dem Einzelfinanzierungsplan wurden für 2017 ca. 37% mehr Personenmonate mit HPA A eingesetzt. Dies resultiert aus der Verlängerung des Projekts vom März 2017 bis September 2017 wegen Personalproblemen in der FST 2. Damit mussten auch von der FST 1 noch Zuarbeiten in 2017 realisiert werden.

Zusammengefasst ergeben sich gegenüber dem ursprünglich geplanten Einzelfinanzierungsplan für die FST 1 mit 22,5 PM HPA A und 12 PM HPA B ein tatsächlicher Personaleinsatz von 26,1 HPA A und 11,1 HPA B. Damit ergibt sich gegenüber der Planung eine Erhöhung bei den HPA A von ca. 14% und für die HPA B eine Reduzierung von ca. 7%. Wie bereits oben erwähnt ergab sich die Erhöhung der HPA A-Personalressourcen aus der komplexen Softwarethematik, die für das eingesetzte Personal für die Implementierungsaufgaben sehr gute Softwarekenntnisse und –erfahrungen voraussetzte.

Erforderliche Mehrausgaben für die Lösung entsprechender Arbeitsaufgaben in der FST 1 wurden durch Kürzungen an anderen Stellen gegenfinanziert.

FST 2:

Die durchgeführten Arbeiten entsprachen im Wesentlichen den vorgesehenen Arbeitsschritten aus dem Arbeitsplan und waren deshalb, wie geplant, erforderlich und angemessen. Die folgende Übersicht zeigt den tatsächlichen Personaleinsatz für die einzelnen Jahre.

Jahr	Wiss.-techn. Personal (Einzelansatz A.1) [PM bei 100%]	HPA Gruppe A [PM bei 100%]
2014	0	0
2015	7,5	7,5
2016	12	12

2017	6	6
Summe	25,5	25,5

Aufgrund der Problematik zu Projektanfang einen geeigneten Bearbeiter für das Projekt CICS zu finden konnte erst im Januar 2015 mit den Arbeiten begonnen werden. Wegen des Ausscheidens des Bearbeiters im Juni 2015 mussten die Arbeiten am Projekt zwischen dem 1. Juli und dem 14. November 2015 ruhen. Aufgrund der Einarbeitung in die abgebrochenen Arbeiten durch den neuen Bearbeiter bzw. durch eine Erweiterung der AP's 1 und 3 ist der Aufwand für diese AP's gegenüber dem Antrag um 1 PM gestiegen. Mangels qualifizierter Hilfskräfte musste die Erstellung des Web-Connector Funktionsmusters und deren Evaluierung in AP 7 von dem wissenschaftlichen Mitarbeiter übernommen werden, was noch einmal zu einem Mehraufwand von 1,5 PM geführt hat.

Insgesamt ergab sich im Projekt für FST 2 für HPA Gruppe A ein Anstieg des Aufwands von ca. 10 % um 2,5 PM.

Erforderliche Mehrausgaben für die Lösung entsprechender Arbeitsaufgaben in der FST 2 wurden durch Kürzungen an anderen Stellen gegenfinanziert.

6 Darstellung des wissenschaftlich-technischen und wirtschaftlichen Nutzens der erzielten Ergebnisse insbesondere für KMU

Wissenschaftlich-technischer Nutzen:

Der wissenschaftlich-technische Nutzen des Projekts besteht in der Entwicklung einer web-orientierten Architektur und Schnittstellen für ein flexibel verteiltes Steuerungssystem in einer Cloud. Damit wurde eine grundsätzlich neue Lösung für eine verteilte Steuerungsdienstestruktur erforscht und als Prototyp umgesetzt. Basierend auf CICS können beliebige, anwendungsbezogenen Steuerungsdienste erzeugt, manipuliert und im IP-Netz verteilt werden. Die Ausführung einer konkreten Steuerungsinstanz kann sowohl in der Cloud (im Server) als auch in beliebigen Webclients erfolgen.

Es wird davon ausgegangen, dass CICS-Steuerungssysteme einen wesentlichen Beitrag zu intelligenten Produktionssysteme sowie für die Realisierung vernetzter Produktionen mit übergreifenden Logistik- und Wertschöpfungsketten leisten können. Damit unmittelbar verbunden sind Ressourceneffizienz, qualitatives Wachstum, verbesserte Klima/Energieverhältnisse sowie Mobilität.

Im Einzelnen besteht der wesentliche wissenschaftlich-technische Nutzen

- in der Schaffung einer offenen und herstellerunabhängig nutzbaren informationstechnische Modellierung und Architektur für ein weborientiertes Steuerungssystems auf Basis von Steuerungsdiensten,
- in der Definition von Syntax und Semantik zugehöriger Schnittstellen sowie
- im Aufbau von Funktionsmustern zur Evaluierung der Architektur als Basis für die Entwicklung beispielhafter und praxisnaher CICS-Steuerungslösungen.

Die wissen.-techn. Ergebnisse des Forschungsvorhabens lassen sich unabhängig vom konkreten Wirtschaftszweig branchenübergreifend zur Entwicklung weborientierter Steuerungs- und Automatisierungslösungen universell und flexibel einsetzen.

Mit der Umsetzung der Ergebnisse des Projekts ergeben sich weitere technische und wirtschaftliche Nutzungsmöglichkeiten. Dazu gehören z.B.:

- Die CICS-Architektur schafft eine übergreifende Strategie für Analyse und Erstellung von weborientierten industriellen Steuerungsanwendungen. Mit Hilfe kombinierbarer Steuerungsdienste lassen sich neue Anwendungen schneller entwickeln und flexibel einsetzen; Unternehmen können damit besser auf neue Kundenwünsche reagieren
- Konfiguration und Erzeugung von Steuerungsdiensten erfolgt systematisch und strukturiert, herstellerunabhängig und transparent auf Basis verfügbarer Industriestandards (PLCopen). Es entstehen offene und flexible weborientierte Steuerungssysteme, die unabhängig von den verwendeten Plattformen, Programmiersprachen und Protokollen sind.
- Offene und wohldefinierte Schnittstellen zwischen den CICS-Komponenten ermöglichen durch die Integration verschiedener Steuerungsfunktionen (herstellerspezifische, herstellerunabhängige, plattformübergreifende) den Aufbau von adaptiven und wandlungsfähigen Steuerungssystemen, wie sie für Industrie 4.0 gefordert sind. Steuerungsdienste sind in hohem Grad wiederverwendbar und sichern damit eine hohe Entwicklungsproduktivität.
- Auf der Basis der vorliegenden Modell- und Schnittstellennotationen können Web-Connectoren und Anschlusskomponenten geschaffen werden, die eine benutzerfreundliche und effiziente Integration von Feldgeräten in ein IP-Netz ermöglichen (Plug-and-Work).

Wirtschaftlicher Nutzen insbesondere für KMUs:

Hauptnutzung der Projektergebnisse erfolgt im Fachgebiet Mess-, Steuerungs- und Automatisierungstechnik. Die Forschungsergebnisse können aber prinzipiell in allen Wirtschaftszweigen und Branchen für die Steuerung bzw. Automatisierung technischer Prozesse genutzt werden. Wegen der Möglichkeit zur Schaffung hochflexibler und „schlanker“ Steuerungslösungen für „weiches“ Echtzeitverhalten werden als Einsatzbereich aber eher spezialisierte KMU-Lösungen im Sektor des kundenspezifischen Geräte-/Maschinen- und Anlagenbaus sowie für den Betrieb verteilter Anlagen (Gebäudeautomatisierung, Windparks, verteilte Energiesysteme) gesehen.

Viele KMUs beschäftigen sich mit Entwicklung und Vertrieb automatisierungstechnischer Gerätetechnik auf der MSR³-Ebene (Sensoren, Aktoren, Feldgeräte und geeignete TCP-Anschlussmodule) und müssen für kundenspezifische Gesamtlösungen Systemkomponenten für Steuerungsaufgaben (z.B. SPS-Systeme) dazukaufen, da für entsprechende eigene FuE-Arbeiten die Personalkapazität in den KMUs meist nicht ausreicht.

Die Forschungsergebnisse ermöglichen es diesen KMUs durch industrienaher Folgeprojekte eigene angepasste und „schlanke“ Steuerungsprodukte und –leistungen für die prozessnahe Ebene mit tragbarem Aufwand zu entwickeln und damit ihr Portfolio zu erweitern, so

³ MSR – Messen, Steuern, Regeln

dass sie moderne weborientierte Automatisierungslösungen aus einer Hand anbieten können. So lassen sich z.B. die bisher bei den KMUs verfügbare SCADA/HMI- bzw. Prozessleit- und Steuerungssysteme mit CICS-Steuerungsdiensten ergänzen und damit ihren Anwendungs- und Einsatzbereich erweitern. Damit können verfügbare Systeme erheblich aufgewertet und ihre Nutzungsdauer erhöht werden. Dies steigert den Innovationsgrad und die Produktverantwortung der KMUs und damit ihre Leistungsfähigkeit.

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 Cloud-basierten Steuerungsdiensten und deren kundenspezifische Anpassung gibt den KMUs einen wesentlichen Wettbewerbsvorteil gegenüber diesen Firmen und bietet neue Marktchancen bei der Lieferung schlüsselfertiger Lösungen insbesondere für die zeitunkritische Steuerung einfacher Anlagen (Klimaanlagen, Heizungsanlagen, verteilte Förderpumpen, Windräder, Sonnenkollektoren usw.).

Darüber hinaus ermöglichen die Projektergebnisse prinzipiell die Schaffung neuer Geschäftsmodelle durch das Vermieten von Steuerungsfunktionen (zeitabhängige Nutzung der Steuerungsprogramme/-algorithmen als SaaS-Modell), die sich durch die erforderliche Flexibilität und Agilität insbesondere für KMUs sowie auch für Existenzgründungen eignen und hier den KMUs neue Märkte mit hohem Wachstumspotential erschließen können.

Da die FuE-Kapazität bei KMU i.d.R. sehr begrenzt ist und eine aktive Beteiligung an aktuellen Forschungsrichtungen bzw. -trends damit kaum möglich ist, bot das Projekt den beteiligten KMUs des projektbegleitenden Ausschusses (PA) die Beteiligung an neuen Forschungsergebnissen im gegenwärtig stark nachgefragten Forschungsumfeld der vernetzten Industrie (Industrie 4.0) und digitalen Transformation der Produktion. Durch die Zusammenarbeit der KMUs mit führenden Global-Playern der Automatisierungstechnik (Nicht-KMU) im projektbegleitenden Ausschuss erhielten die KMUs weiterhin Zugang zu entsprechenden Netzwerken und es erfolgte eine Bewusstseinssteigerung für den Einsatz neuer Informations- und Kommunikationstechnologien bei den beteiligten KMUs.

7 Plan zum Ergebnistransfer in die Wirtschaft

- konkrete Beschreibung der bereits durchgeführten Transfermaßnahmen (mit Zeitangaben):

Maßnahme	Ziel	Rahmen	Datum/Zeitraum
Projekt-meetings	Beteiligung des PA am Projekt	unterschiedliche Orte (FST1 +2)	10.11.14; 16.03.15; 05.11.15; 12.05.16; 16.12.16; 19.07.17
Projekt-präsentation	Information der ZVEI-Forschungsgemeinschaft Automation	Sitzungen der FG Automation in Frankfurt/Main und Hannover	23.02.16; 14.09.17
	Information eines breiten Industriefachpublikums; Aquisition interessierter Firmen für eine Nutzung	SPS/IPC/Drives in Nürnberg (FST1 +2)	24.11. – 26.11.15; 22.11. – 24.11.16

	<p>VDI-Fachtagung Industrie 4.0 (FST1+2)</p> <p>Fachforum Industrie 4.0 (FST2)</p> <p>Hausmesse Formware (FST1+2)</p> <p>Fraunhofer Workshop „Digitale Produktion“ (FST2)</p> <p>KommA 2016 , Lemgo (FST2)</p> <p>Fachbeitrag in atp edition (FST 1 + 2)</p> <p>Fachbeitrag in Computer&Automation (FST 1 + 2)</p>	<p>Januar 2016</p> <p>Mai 2016</p> <p>September 2016</p> <p>September 2016</p> <p>November 2016</p> <p>April 2017</p> <p>Mai 2017</p>
Information int. wissen-techn. Web-Spezialisten über Teilaspekte von CICS	<p>11th Int. Conf. on Web Inf. Systems and Technologies (WEBIST-2015), Lissabon (FST1)</p> <p>AALE 2016, Lübeck (FST1)</p> <p>14th International Conference on Remote Engineering and Virtual Instrumentation (REV2017), New York (FST1 + 2)</p>	<p>Juni 2015</p> <p>März 2016</p> <p>14. – 16.03.17</p>
Information der Fachwelt der Wirtschaftsinformatiker über Cloud in Automation	HMD (2015) 52, Springer Verlag, (Leitartikel) (FST1+2)	Oktober 2015
Information für Unternehmen aus dem Maschinenbau/IT-Security	Arbeitskreis IT Security für Industrie (FST2)	Mai 2015

	Information von Unternehmen aus dem Maschinenbau	Veranstaltung des Clusters Mechatronik bei Fujitsu (FSW2) ASW 2016, Wildau (FST 2) HANSER Konstruktion (FST2)	Juli 2015 März 2016 Oktober 2016
	Diskussion mit Fachleuten im Automatisierungsumfeld	Delegation von Universitäten aus Australien (FST1)	April 2017
	Allg. Information für Unternehmen	Fraunhofer Innovisions (FST 2)	Mai 2015
Training für chinesische Spezialisten	Ausbau der Trainingskompetenz in Industrie 4.0 der FST 1 für den chinesischen Markt	Integration der CICS-Thematik in das Industrie 4.0-Training für chinesische Spezialisten sowie Delegationen aus Universitäten und Unternehmen (FST1)	1 x 8 Wochen Training im II.HJ 2015 3 x 4h-Training in 2015 3 x 4h-Training in 2016 3 x 4h-Training in 2017
Vertiefte Gespräche mit interessierten Unternehmen zum Einsatz von CICS	Dr.-Ing. Hans Egermeier	Zusammenarbeit zur Nutzung von CICS bei Kunden aus dem Maschinenbau	Gespräche laufen
	EUROS	Test bzw. Nutzung des Echtzeit-Browsers von EUROS als CICS-Client (FST 1)	
	TE Connectivity	Vorbereitung eines Drittmittelprojekts unter Nutzung von CICS-Ergebnissen (FST2)	
	qmd4, MicroSys	Zusammenarbeit zur Integration des Web-basierten IEC 61131-3-Editors von qmd4 in CICS (FST1+2)	
	SpaceNet, HS Rosenheim, Fraunhofer EMFT, SEMIKRON, Erhardt & Leimer, infoteam, XYDE, Monkeyworks, assystem, óce print	Diskussion eines möglichen Einsatzes einer CICS-Steuerung (FST1+2)	

	systems, linutronix, SOTEC, HARTING, B&R, Berner&Mattner, KAPSCH, INSYS		
Vorbereitung neuer Projekte	Gründung eines Start-Up zur Vermarktung einer IIoT Suite WOAS unter Nutzung von CICS als Steuerungsdienst	Projekt wurde im EXIST-Programm eingereicht, aber abgelehnt (FST1+2)	Januar 2017
	Huawei	Nutzung von CICS in einem X-Ethernet von Huawei (FST 2)	Projektvertrag ist abgeschlossen Projektlaufzeit 01.01.18 – 31.12.18
	Abschluss eines ZIM-Projekts	Einsatz einer modifizierten CICS-Steuerung im Leitsystem der Fa. WEBfactory (FST1+2)	Projektgespräche haben sich verzögert, da WebFactory einen neuen Eigentümer hat
	Verfahren/Plattformen für die Automatisierung einer Mikropumpenfertigung	Einsatz von CICS in der MEMS Fertigung (FST1+2)	Projektgespräche laufen

- konkrete Beschreibung der noch geplanten Transfermaßnahmen:

Maßnahme	Ziel	Rahmen	Datum
Projektpräsentation	Information eines Industriefachpublikums; Aquisition interessierter Firmen	1-tägiges Workshop in München „SMART AUTOMATION - Cloudbasierte Steuerungsdienste & Smarte Anwendungen“ (FST1+2)	11.01.18
	Information int. wissentechn. Spezialisten über Teilaspekte von CICS (I40-Komponente für CICS)	15th International Conference on Remote Engineering and Virtual Instrumentation – REV2018, Düsseldorf (FST1)	21. – 23.02.16
Training für chinesische Spezialisten	Weiterer Ausbau der Trainingskompetenz der FST 1 für den chinesischen Markt	Vorstellung von CICS in weiteren Trainingsmaßnahmen für chinesische Spezialisten und Delegationen (FST 1)	5 x Training geplant in 2018

8 Zusammenstellung aller Arbeiten, die im Zusammenhang mit dem Vorhaben veröffentlicht wurden

Im Zeitraum 2015 – 2017 wurde folgende Veröffentlichungen durch die Forschungsstellen 1 und 2 vorgenommen:

Jahr	Quelle	FST	Art der Veröffentlichung	Titel
2015	11th Int. Conf. on Web Inf. Systems and Technologies (WEBIST-2015), Lissabon	1	Konferenzbeitrag	Web of Things in an Industrial Environment
2015	HMD (2015) 52, Springer Verlag,	1 + 2	Leitartikel	Industrial Cloud - Status und Ausblick
2015	Arbeitskreis IT Security für Industrie	2	Vortrag	Einführung in Steuerungsdienste in der Cloud mit IEC 61131
2015	Veranstaltung des Clusters Mechatronik bei Fujitsu	2	Vortrag	Cloudbasierte Steuerungsdienste für die Produktion von morgen
2015	SPS/IPC/Drives in Nürnberg	1 + 2	Demonstration	Vorführung einer CICS-CM am CICS-Demonstrator 2
2015	Fraunhofer Innovisions	2	Vortrag	Produktion im Browser, Fertigungssteuerung via Internet
2015	Patentregister	1 + 2	deutsche Patentanmeldung	Steuerungseinrichtung mit einem Steuerungsprogramm und einer Runtime-Maschine zum Betreiben eines Automatisierungsgerätes
2015	Patentregister	1 + 2	deutsche Patentanmeldung	Steuerungseinrichtung mit einem Steuerungsprogramm und einer Gerätekonfiguration zum Betreiben eines Automatisierungsgerätes
2015	Unternehmermagazin	1	Zeitschriftenartikel	Neuer Standard Industrie 4.0 - Die künftige Rolle der IT für die Produktion im deutschen Mittelstand
2016	Patentregister	1 + 2	PCT-Anmeldung	Steuerungseinrichtung mit einem Steuerungsprogramm und einer Gerätekonfiguration zum Betreiben eines Automatisierungsgerätes
2016	VDI-Fachtagung Industrie 4.0	1 + 2	Konferenzbeitrag	IEC 61131 & Industrial Internet of Things – Bewährte Praxis mit zukunftsfähigen Technologien verbinden.
2016	Fachforum Industrie	2	Vortrag	KLASSISCHE

	4.0,			RUNGSTECHNIK IM ALTER DER CLOUD
2016	Hausmesse Formware	1 + 2	Demonstration	Vorführung einer Fertigungszelle mit CICS-SMM-Steuerung
2016	Fraunhofer Workshop „Digitale Produktion“	2	Vortrag	Kommunikationstechnologien für die Industrie von morgen
2016	Komma 2016 , Lemgo	2	Konferenzbeitrag	A Comparison of current Web Protocols for usage in Cloud based Automation Systems
2016	AALE 2016, Lübeck	1	Konferenzbeitrag	Eine SPS-Steuerung als Industrie 4.0-Komponente
2016	SPS/IPC/Drives in Nürnberg	1 + 2	Demonstration	Vorführung einer Fertigungszelle mit einer CICS-SMM-Steuerung
2016	ASW 2016, Wildau	1 + 2	Konferenzbeitrag	Cloud basierte IEC 61131 Steuerungsdienste auf Basis von Web-technologien
2016	HANSER Konstruktion	2	Zeitschriftenartikel	Wie die SPS in die Wolke wandert
2016	4. Markt&Technik Industrie 4.0 & Industrial Internet Summit	1 + 2	Konferenzbeitrag	Konzept und Implementierung einer I40-Komponente für Steuerungen aus der Cloud
2016	SPS-Magazin	1 + 2	Zeitschriftenartikel	Flexibel steuern mit der Datenwolke
2017	14th International Conference on Remote Engineering and Virtual Instrumentation (REV2017), New York	1 + 2	Konferenzbeitrag	Cloud-based Industrial Control Services
2017	atp edition	1 + 2	Hauptbeitrag	Steuerungsdienste aus der Cloud auf Basis IEC 61131
2017	Computer&Automation	1 + 2	Zeitschriftenartikel	Cloud-basierte Steuerungen – aber Wie?
2017	NEXT	2	Zeitschriftenartikel	Der Befehl aus der Cloud - SPS-Steuerungen aus der Cloud
2018	15th International Conference on Remote Engineering and Virtual Instrumentation (REV2018), Düsseldorf	1	Demonstration	Vorführung von CICS am CICS-Demonstrator 2

9 Einschätzung zur Realisierbarkeit des vorgeschlagenen und aktualisierten Transferkonzepts

Die wirtschaftlichen Erfolgsaussichten für die Umsetzung der Projektergebnisse in den nächsten zwei Jahren werden als hoch eingeschätzt. Die Gründe dafür sind:

- Zum Projektende (2017) wird generell ein verstärkter Einsatz von Webtechnologien in der Steuerungs- und Automatisierungstechnik in Zusammenhang mit modernisierten Produktionsstrukturen nach den Industrie 4.0-Paradigma erwartet und akzeptiert.
- Die Forschungsergebnisse sind prinzipiell für alle zeitunkritischen Steuerungen branchenübergreifend für einen ressourcenschonenden und effizienten Betrieb einsetzbar.
- Ab 2017/18 stehen zumindest für lokale Netze schnelle und deterministische IP-Netze wie z.B. TSN oder X-Ethernet (Huawei) in der erforderlichen Qualität (Quality of Services) und mit der erforderlichen Sicherheit für die Automatisierungstechnik zur Verfügung, so dass auch über diese Netze deterministische Echtzeit-Steuerungen möglich sind.
- Die Forschungsergebnisse sind auch partiell und als Bestandteil übergeordneter Systeme effizient nutzbar. Dies erweitert gleichfalls den Einsatzumfang.
- Die wirtschaftliche Nutzbarkeit kann abhängig vom Einsatzgebiet und den jeweiligen Anforderungen etappenweise erfolgen, so dass das wirtschaftliche Risiko überschaubar bleibt.

Dabei sind folgende Etappen bzw. Nutzungsarten vorstellbar:

- Nutzung der CICS-Modellarchitektur und deren Schnittstellen als theoretische Basis für neu zu entwickelnde Produkte (ab 2018).
- Einsatz des Web-Connectors nach einer Überarbeitung des Prototyps zur schnellen Anbindung von Sensoren/Aktoren/Geräten an ein IP-Netz (2018).
- Nutzung der Web-basierten IEC 61131-Runtime-Umgebung (nach Erweiterung und Re-Implementierung) als verteiltes Steuerungssystem für E/A-Geräte am IP-Netz (2018).
- Entwicklung eines Miet-Business-Modell zur Nutzung von Steuerungsdiensten aus der Cloud unter Nutzung eines geeigneten Webportals auf Basis von CICS als Dienstleistung (2018/19).

Abhängig von den aufgeführten Etappen ergeben sich unterschiedliche Finanzierungsaufwände für eine industrielle Umsetzung der Forschungsergebnisse. Die folgende Tabelle veranschaulicht eine Schätzung der erforderlichen Entwicklungskosten für eine industrielle Umsetzung der CICS-Projektergebnisse aus heutiger Sicht (ohne Verwaltungs-, Werbungs- und Kundenaquisitionskosten):

Nutzungsart	Zeitraum	Kosten [€]	Bemerkungen
CICS-Modellarchitektur und deren Schnittstellen	2018	-	Die theoretische Basis von CICS kann ohne Kosten unmittelbar als Basis für

			Neuentwicklungen genutzt werden.
Web-Connector	2018	30.000 (ca. 3 PM)	Re-Design/ Re-Implementierung erforderlich
Web-basierte IEC 61131-Runtime- Umgebung	2018/2019	120.000 (ca. 12 PM)	Erweiterung / Re-Implementierung erforderlich
CICS-Webportal mit Miet-Business-Modell	2018/2019	180.000 (ca. 18 PM)	Neuentwicklung auf Basis von CICS unter Nutzung von CICS-Teilkomponenten

Es wird weiterhin angestrebt unter Beteiligung interessierter Mitglieder des PA bzw. anderer Unternehmen ein StartUp zu gründen, welches neu entwickelte Produkte und Dienstleistungen auf Basis von CICS anbieten soll.