Interaktion von AR-Devices und Cyber Physical Systems

Hauptprojekt Ausarbeitung

Sascha Waltz

Sommersemester 2017

HAW Hamburg

Zusammenfassung. Das Ziel dieser Ausarbeitung ist es, eine Kommunikation zwischen einem mobilen Endgerät und einem Cyber-Physical System umzusetzen. Hierzu sollen die Möglichkeiten der Absicherung und die Benutzerinteraktion dargestellt und implementiert werden. Zur Kommunikation werden selbstauskunftsfähige Objekte genutzt, die über ein Agentensystem kommunizieren können.

1 Einleitung

Die Interaktion mit physischen Objekten die in der virtuellen Welt abgebildet werden ist heutzutage relativ leicht umsetzbar. Viele Objekte können sich selbst in der virtuellen Welt abbilden und sind so für User oder andere Maschinen erreichbar.

1.1 Vorangegangene Arbeit

Die bisherige Arbeit befasste sich mit dem Thema "Sicherheit in Cyber-Physical Systems", also mit Systemen, die physische Komponenten in der virtuellen Welt abbilden und diese auch zugreif- und steuerbar machen. Hierzu wurden Kommunikationsmöglichkeiten zwischen diesen Systemen untersucht, die zur Verfügung stehenden Möglichkeiten solcher Systeme analysiert und verschiedene Lösungsansätze erstellt.
Die Interaktion mit einem Benutzer oder mit einem weiteren System stand hierbei noch nicht im Vordergrund sondern wurde nur sporadisch nebenbei betrachtet. Das Hauptaugenmerk lag auf der Kommunikation zwischen den Systemen bzw. Agenten bei der Abfrage oder Übermittlung von Daten. Wie oder von wem diese Daten abgefragt wurden, war dabei nebensächlich.

1.2 Zielsetzung

Aufbauend auf der im Grundprojekt umgesetzt Authentifizierung eines Users, welcher mit einem Cyber-Physical System (CPS) interagieren möchte, soll in diesem

Hauptprojekt die Interaktion mit CPS umgesetzt werden. Der authentifizierte User soll hierzu Daten abfragen oder Befehle übermitteln können. Die Interaktion soll hierbei über ein Augmented Reality (AR) Interface erfolgen, welches das CPS-Gerät anhand eines QR-Code-Tags erkennt und die Verbindung herstellt. Hierbei soll der User automatisch authentifiziert werden. Wenn der User nicht berechtigt ist, wird die Verbindung abgelehnt.

2 Authentifizierung und Visualisierung in einer Cyber-Physical System-Umgebung

Um die in [4] beschriebene Authentifizierung zu testen und zu nutzen, wird ein Interface benötigt, über welches ein User mit dem Gerät des Cyber-Physical Systems interagieren kann. Dieses sollte sich automatisch verbinden, authentifizieren und Daten abrufen oder senden können. Als Grundlage wird die in der Grundprojekt-Ausarbeitung [4] entwickelte Authentifizierungsumgebung genutzt. An dieser werden die User authentifiziert und der Zugriff gewährt oder verweigert. Die Entwicklung des Interfaces erfolgt zunächst für Android-Devices mit Unity3D[1] und Vuforia Engine[2]. Diese Kombination bietet eine sehr gute und umfangreiche Oberfläche zur Entwicklung von Augmented-Reality Apps und bringt sämtliche benötigten Funktionen mit.

2.1 Projektaufbau

Der grundlegende Aufbau der Projektumgebung ist in Abbildung 1 zu sehen. In diesem Projekt wird überwiegend die Eingabe-Komponente, also das Smartphone oder ein anderes AR-Gerät, behandelt. Die zu bedienenden CPS-Geräte werden mit einem QR-Code gekennzeichnet, welcher die Bezeichnung des Gerätes enthält, den die App dann nutzt um eine Verbindung aufzubauen. Um die Verbindung aufzubauen, wird zunächst die Freigabe vom Request-Agent benötigt, dieser wiederum erfragt die erforderlichen Berechtigungen beim Authentification-Agent. Wird der Zugriff gestattet, darf die App mit dem CPS interagieren.

2.2 Authentifizierung am Agenten

Die Authentifizierung am Agenten hat diverse Hintergründe. Vordergründig geht es darum, sicherzustellen, dass nur befugte Personen oder Systeme auf bestimmte Daten oder Vorgänge zugreifen können und unbefugten der Zugriff verwehrt wird. Des Weiteren lässt sich so sicherstellen, dass sowohl das Gerät als auch die Daten, auf die zugegriffen wird, authentisch und integer sind und Herkunft und Erhalt nicht abstreitbar sind.

[1] Unity3D - https://unity3d.com/de/; letzter Zugriff: 02.12.2017
[2] Vuforia-Developer Library - https://library.vuforia.com/; letzter Zugriff: 02.12.2017

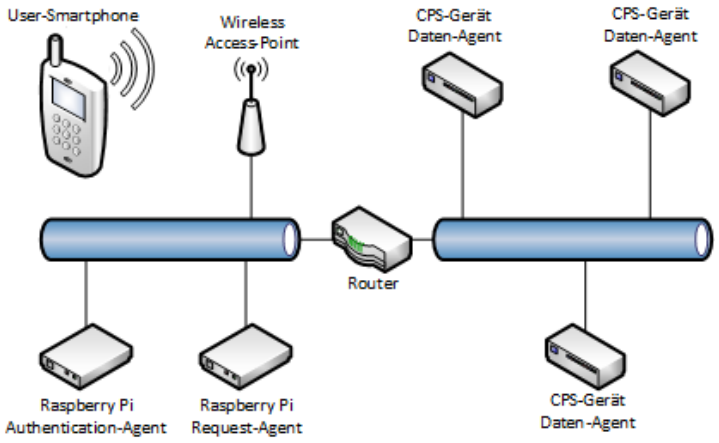


Abb. 1: Projektaufbau mit Raspberry Pi und Ethernet (vgl.) [4]

Authentizität

Die Authentizität bezeichnet die Eigenschaft, das jemand wirklich derjenige ist, der er vorgibt zu sein oder das Informationen von der Quelle stammen, von der sie stammen sollen. Hierzu werden Identitäten überprüft und es gibt verschiedene Wege und Möglichkeiten um dies zu tun(vgl.) [4]:

Wissen Der Benutzer oder das System haben Kenntnis von etwas, das nur sie wissen können, ein Passwort, die Antwort auf eine bestimmte Frage oder Ähnliches

Besitz Benutzer oder System sind im Besitz von etwas, das nur sie besitzen können, einen Schlüssel, eine Karte, einen Dongle etc.

Biometrie Hier werden die biometrischen Merkmale einer Person überprüft, beispielsweise Fingerabdruck oder Iris, welche einen Menschen ebenfalls eindeutig ausweisen können

Die einfachste Variante für dieses Projekt ist Wissen oder Besitz. Zur Authentifizierung kann ein Passwort oder ein Schlüssel abgefragt werden. Um die Authentifizierung automatisch ablaufen zu lassen, ist es am einfachsten, einen SSH-Key eines Users zu übermitteln, wenn dieser versucht sich zu verbinden. Im optimalen Fall ist nur der Benutzer im Besitz dieses Schlüssels und hat ihn im Zweifelsfall auch noch mit einem Passwort gesichert.
Es ließe sich hier auch eine Zwei-Faktor-Authentifizierung realisieren, diese ist aber nur mit der Interaktion des Users möglich, da zusätzlich zu dem SSH-Key noch die Hürde einer serverseitigen Passwortabfrage genommen werden muss. Serverseitig deshalb, damit die Kontrolle des Passwortes dem System obliegt, auf welches zugegriffen werden soll. Das SSH-Key-Passwort zählt hier nicht als Zwei-Faktor-Authentifizierung, da die Kontrolle des Passwortes bei dem Benutzer liegt und das System, auf welches zugegriffen wird, nicht beide Faktoren kontrollieren kann.

Nichtabstreitbarkeit

Der Versand und der Empfang von Daten soll nicht in Abrede gestellt werden können. Diese Aspekte der Nichtabstreitbarkeit müssen grade bei der Übertragung von wichtigen und sensiblen Daten sichergestellt werden. Die Übertragung selbst lässt sich in der Regel anhand von Log-Dateien festhalten, den vollständigen Empfang der Daten muss man sich aber vom Client bestätigen lassen. Die Bestätigung erfolgt anhand einer kurzen Nachricht, die ebenfalls in das Log des Servers eingetragen wird.

Integrität

Um die Integrität der Daten und des Systems selbst zu gewährleisten dürfen Daten nicht oder nicht unbemerkt verändert werden. Lassen Daten sich nicht unbefugt manipulieren, liegt eine starke Integrität vor. Lassen Daten sich zwar verändern, aber es ist nicht möglich dies unbemerkt zu tun, spricht man von einer schwachen Integrität. Da hier nur Daten abgerufen werden, kann der Client nicht prüfen, ob die Daten vor dem Versenden verändert wurden. Aus diesem Grund werden Daten nur über eine verschlüsselte Verbindung versendet, so dass sichergestellt ist, dass auch unveränderte Daten beim Empfänger angekommen sind.

Public-Key-Verfahren

Ein Public-Key-Verfahren bringt die nötigen Eigenschaften mit, um die zuvor aufgeführten Punkte abzudecken. Voraussetzung ist natürlich, dass der Benutzer oder das System alleinigen Zugriff auf den jeweiligen privaten Schlüssel haben. Das Signieren von Daten mit einem privaten Schlüssel sorgt dafür, dass der

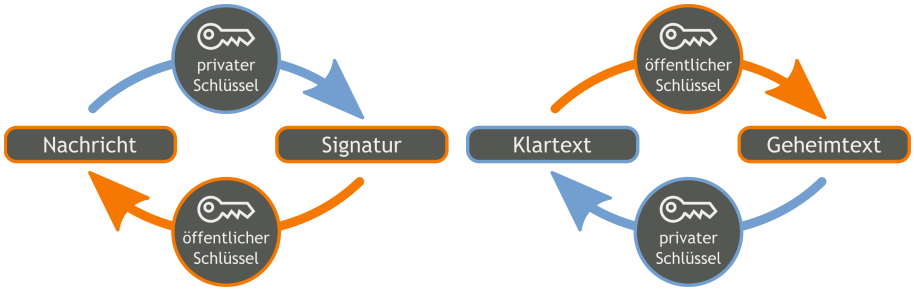


Abb. 2: Signieren [5] (links) und Verschlüsseln [6] (rechts) mit einem Public-Key-Verfahren

Empfänger durch das entschlüsseln der Nachricht mit dem öffentlichen Schlüssel sicher sein kann, dass die Daten den richtigen Ursprung und Inhalt haben. Sollten die Daten auf dem Weg zum Empfänger verändert worden sein, wäre es nicht

möglich, sie erneut mit dem korrekten privaten Schlüssel zu signieren, da dieser nur im Besitz des Absenders sein darf. Somit sind die Punkte Authentizität und Integrität abgedeckt, wenn ein solches Verfahren genutzt wird. Diese Punkte sind in diesem Projekt von vorrangiger Bedeutung und wurden mittels SSH umgesetzt. Die Verschlüsselung von Daten sorgt dafür, dass Daten nicht in falsche Hände gelangen können. Hierzu werden die Daten vor der Übertragung mit dem öffentlichen Schlüssel des Empfängers verschlüsselt. Dieser wiederum ist als Einziger im Besitz seines dazugehörigen privaten Schlüssels, mit dem die Daten wieder entschlüsselt werden können. Hierzu muss allerdings der öffentliche Schlüssel des Empfängers auf dem System vorhanden sein. In diesem Projekt wird davon ausgegangen, dass die Daten nicht sensibel sind und somit unverschlüsselt übertragen werden können.

2.3 Anzeige eines AR-Interfaces für ein Cyber-Physical System

Um mit einem CPS interagieren zu können, muss zunächst erkannt werden, um welches CPS es sich handelt. Als Augmented Reality Gerät können heutzutage Smartphones oder Wearables wie Brillen (Microsoft HoloLens[3]) Ein gutes Beispiel

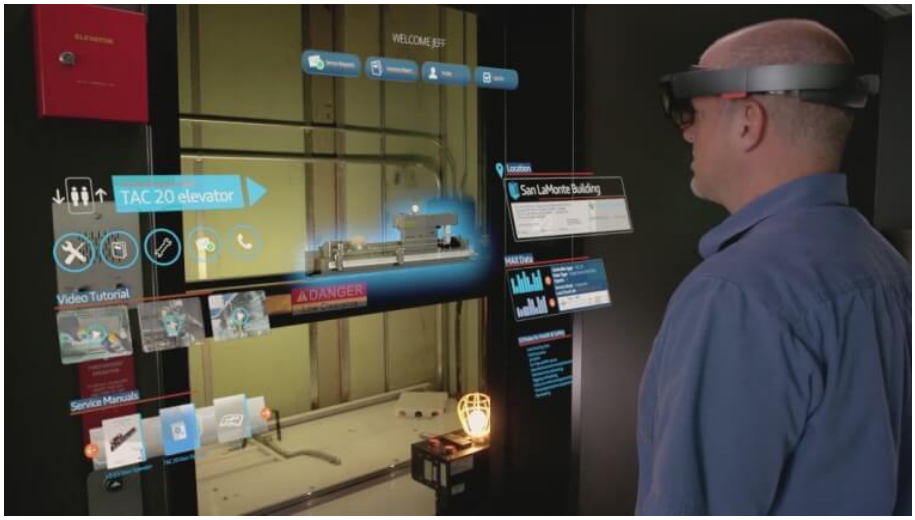


Abb. 3: Konzept eines AR-Interfaces zur Fahrstuhl-Wartung [2]

für ein AR-Interface hat die Firma ThyssenKrupp geplant, wie in Abbildung 3 zu sehen. Hier soll ein Techniker eine Fahrstuhlwartung durchführen und diese mit Hilfe einer HoloLens und einem AR-Interface vornehmen. Dieses Interface

[3] Microsoft HoloLens - https://www.microsoft.com/de-de/hololens; letzter Zugriff: 30.12.2017

sollte natürlich nicht für jedermann nutzbar sein, da es sich um ein sehr sensibles und kontrollbedürftiges System handelt, zu dem kein Unbefugter Zugriff haben sollte.
Das Interface soll praktischer Weise in der Nähe oder über dem zu benutzenden Objekt angezeigt werden. Hierzu muss es entsprechend im Raum positioniert werden. Die Positionierung lässt sich über verschiedene Faktoren erreichen:

Erkennen des Objektes Das Augmented Reality Gerät erkennt das Objekt an sich als ein Objekt mit dem eine Interaktion stattfinden kann. Bei dieser Methode müssen markante Merkmale des Objektes bekannt sein um eine eindeutige Bestimmung zu ermöglichen.

Erkennen eines Tags Mit Hilfe eines QR-Codes oder einer ähnlichen Markierung des Objektes wird dieses erkannt und eine Interaktion eingeleitet. Es muss sich dabei nicht um einen optischen Tag handeln, auch ein Bluetooth- oder NFC-Beacon wären hier denkbare Möglichkeiten.

Position des AR-Gerätes Anhand der Position des Smartphones oder der AR-Brille wird das entsprechende Interface eingeblendet. Hierzu muss die Position des Objektes als auch die Position des AR-Gerätes zum Objekt bekannt sein.

Das Erkennen von Objekten erfordert eine sehr umfangreiche Datenbank um die Objekte anhand ihrer Merkmale eindeutig identifizieren zu können. Ein QR-Code mit den benötigten Informationen oder ein Bluetooth- oder NFC-Beacon, die zur Verbindung und Anzeige des Interfaces erforderliche Informationen überträgt, sind hier wesentlich einfacher umzusetzen.

2.4 Unity3D

Die Umsetzung eines Prototypen für dieses Projektes erfolgte mit dem Programm Unity3D([4]). Unity3D ist ein Editor zum erstellen von 2D und 3D Anwendungen und Oberflächen, als auch Augmented- und Virtual Reality Umgebungen. (vgl. [3])
Um mit Objekten in der realen Welt in AR-Anwendungen zu interagieren, müssen diese zunächst erkannt werden. Hierfür eignet sich die Vuforia-Engine, welche seit Version 2017.2 in Unity integriert ist. Mit dieser Engine lassen sich Bilder oder Objekte durch ein AR-fähiges Gerät (Smartphone, AR-Brille, etc.) erkennen, so dass darauf in der Anwendung reagiert werden kann. (vgl. [1])
Die Erkennung von Bilder, bzw. Tags, oder Objekten ist ein wichtiger Part bei der Interaktion. Nur wenn ein Tag oder ein Objekt erkannt werden, lassen sich die Abläufe zur Authentifizierung starten. So wurde in diesem Projekt zunächst daran gearbeitet, entsprechende Tags in Form von Bildern zu erkennen.
Unity und Vuforia bieten hierzu eine AR Camera, welche die Main Camera bei der Entwicklung in Unity ersetzt. Um mit dieser Bilder zu erkennen, müssen die Bilder zunächst einer Datenbank hinzugefügt werden. Dies geschieht über die Vuforia-Homepage. Vuforia bringt auch einige Bilder mit, die der Engine gute Identifizierungsmerkmale bieten. Auf den ersten Blick zeigen die Bilder

[4] vgl. https://unity3d.com/de/ - letzter Zugriff: 18.02.2018

Abb. 4: Vuforias mitgelieferte Target-Images (vgl. [1])

keine besonders auffälligen Merkmale, die sie übermäßig gut erkennbar machen. Dennoch gehören sie laut Vuforia zu den gut identifizierbaren Bildern. Aus diesem Grund wurden diese Bilder für das Projekt gewählt.
Mit Hilfe von Unity und Vuforia lassen sich Augmented-Reality Objekte auf diesen Tags platzieren und bewegen sich mit, wenn die Tags bewegt werden und weiterhin im Blickfeld des AR-Gerätes bleiben.

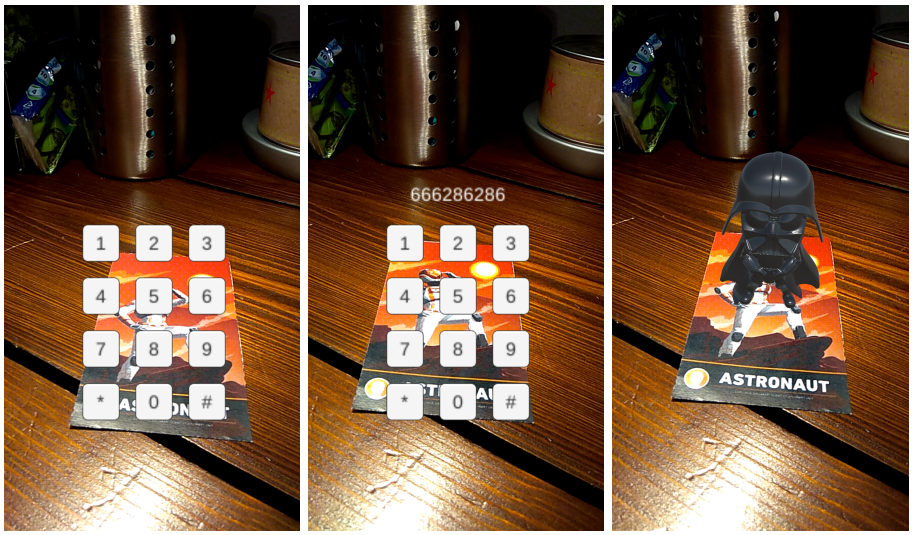
Abb. 5: Eingeblendetes Objekt nach erkennen des Tags

Um die Bilder wiedererkennen zu können, werden sie als Objekte in der Unity-Umgebung platziert, weitere Objekte, Interfaces oder Aktionen werden

dann hier als Kinder des Bild-Objekts eingefügt.
Es lassen sich nach der Erkennung auch Abläufe programmieren, die an dem Tag angeheftet werden können, so dass sich Objekte, Töne oder auch bewegte Objekte einblenden lassen. Auch die Interaktion mit den Objekten ist möglich, solange der Tag im Sichtfeld bleibt und erkannt wird.
Diese Abläufe und die Interaktion lassen sich einfach zur Authentifizierung nutzen, was im nächsten Abschnitt behandelt wird.

2.5 Kombination Authentifizierung und AR

Die Vuforia-Engine macht das Erkennen eines Objekts oder Tags recht einfach und zuverlässig, also muss als nächster Schritt eine Interaktion bzw. eine Authentifizierung stattfinden. Wie schon im Grundprojekt beschrieben, lassen sich 3 Methoden zur Authentifizierung nutzen: Wissen, Besitz und Biometrie (vgl. [4], Abschnitt 3.1).



(a) Erkennen des Tags (b) Eingabe des PINs (c) Anzeige des Objekts

Abb. 6: Ablauf Interaktion, Authentifizierung, Anzeige des Objekts

Diese Interaktion und Authentifizierung ist für alle Arten von Geräten (Smartphones, AR-Brillen und Ähnlichen) praktikabel, nur die Auswahl der Ziffern erfolgt per tippen oder Gestensteuerung. Da nur der User des AR-Gerätes dieses Interface sehen und bedienen kann, ist es fast nicht möglich, dass jemand den Pin bei der Eingabe sehen kann. Eingabe per Sprache oder Geste wäre dahingehend nicht so vorteilhaft, da jeder mithören oder sehen kann, was eingegeben wird.

In diesem Projekt wurde zunächst nur Wissen in Form eines Pins abgefragt:

Abbildung 6a zeigt das eingeblendete Interface nach erkennen des Tags
Abbildung 6b veranschaulicht den über das Interface eingegebenen Pin
Abbildung 6c stellt das Objekt nach erfolgreich eingegebener Pin auf dem Tag dar

Die Umsetzung in Unity3D gestaltet sich für dieses Projekt recht simpel, Vuforia bringt die nötigen Funktionen und Elemente mit, das Interface lässt sich mit Unity-eigenen Mitteln gestalten und die Abläufe wurden in einer Controller-Klasse implementiert, die an die Buttons gehängt wurde.

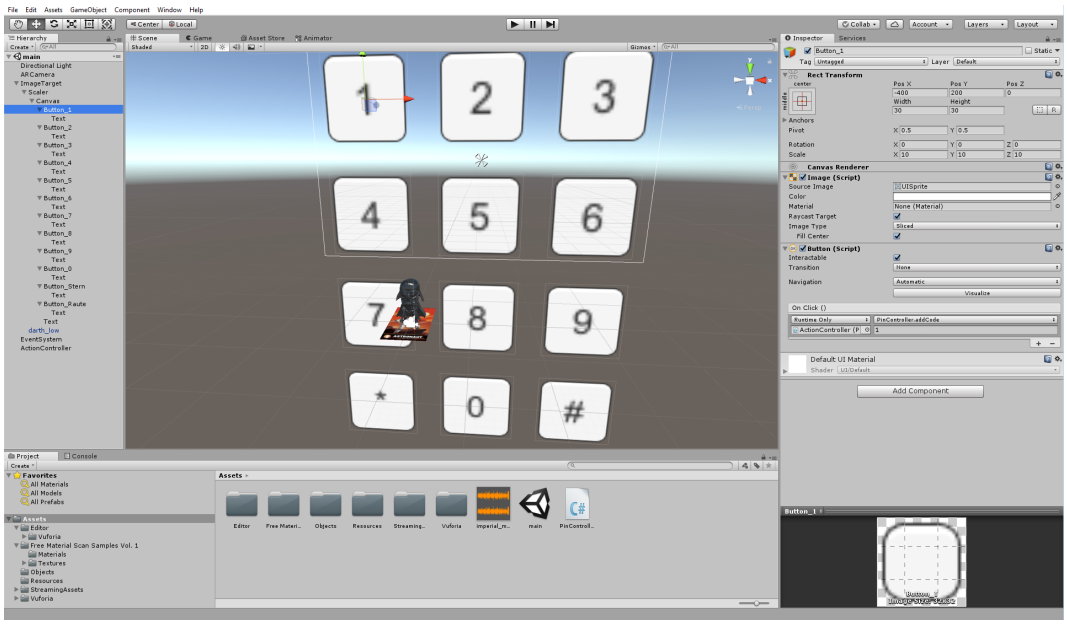


Abb. 7: Übersicht des Projektes in Unity

Dieser Controller überprüft den eingegebenen Pin bei betätigen der #-Taste und wenn der Pin korrekt ist, wird das Interface aus- und das Objekt eingeblendet.

3 Fazit

In diesem letzten Abschnitt wird das Projekt noch einmal zusammen gefasst, der aktuelle Stand beleuchtet und ein Ausblick auf die weiteren Möglichkeiten gegeben.

3.1 Zusammenfassung und derzeitiger Stand

Das Projekt beinhaltet einen Prototypen zur Interaktion per Augmented Reality Device, Smartphone oder Wearable o.ä., die Erkennung von Tags, ein Interface zum Pin-Eingabe und ein Objekt, das danach angezeigt wird. Die Passwort- bzw. Pin-Abfrage arbeitet zur Zeit noch nicht zentralisiert über einen entsprechenden Agenten oder Server, sondern direkt über ein im Controller hinterlegtes Passwort.
Die Abbildungen im vorangegangenen Kapitel beschreiben die umgesetzte Simulation und zeigt, dass die Interaktion mit virtuellen Interfaces anhand von Tags, welche sich auf einem CPS platzieren lassen, möglich und praktikabel ist.
Es wurde gezeigt, dass für den einfachen Prototypen die Unity3D-eigenen Mittel ausreichen, um eine Interaktion und Authentifizierung zu ermöglichen.
Die Implementierung eines Public-Key-Verfahrens, wie in Abschnitt 2.2 beschrieben, war im Rahmen dieses Projektes nicht mehr möglich und hätte sowohl Zeit als auch Umfang des Projektes gesprengt.

3.2 Ausblick

Zukünftige Arbeiten können sich mit einigen Punkten befassen, die in diesem Projekt nicht umgesetzt wurden. Zum einen wäre hier die Anbindung an die Middleware des CSTI um eine zentrale Verwaltung von Passwörtern und Pins zu ermöglichen. Diese können Personen- oder aber auch Objektbezogen sein, also ein Passwort für ein bestimmtes CPS, welches jeder User kennen muss um damit zu interagieren.
Eine weitere Möglichkeit ist die Authentifizierung per Sprach- oder Stimmerkennung, bei der sich der User anhand seiner Stimme authentifiziert.
Ein wichtiger Punkt, der in zukünftigen Arbeiten behandelt werden könnte, ist die Kommunikation der Middleware mit dem CPS, nachdem der User sich erfolgreich angemeldet hat. Diese Kommunikation könnte beispielsweise über entsprechende Agenten ablaufen. Es ist auch denkbar, dass die Kommunikation direkt zwischen AR-Gerät und dem CPS stattfindet und der User bei erfolgreicher Authentifizierung ein Token erhält, welches ihm erlaubt, mit dem CPS direkt zu kommunizieren.

Literatur

1. Inc., P.: Getting started with vuforia in unity (2018), https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity-2017-2-beta.html
2. Ridder, M.: Thyssenkrupp treibt digitalisierung des weltweiten aufzugsservice weiter voran: Microsoft hololens verringert wartungszeit (2016), https://www.thyssenkrupp.com/de/newsroom/pressemeldungen/press-release-114208.html
3. Unity3D: Unity user manual (2017.3) (2018), https://docs.unity3d.com/2017.3/Documentation/Manual/UnityManual.html
4. Waltz, S.: Sicherheit in cyber physical systems. Tech. rep., Hochschule für Angewandte Wissenschaften Hamburg, http://users.informatik.haw-hamburg.de/ ubicomp/projekte/master2017-proj/waltz.pdf (2017)

5. Wikipedia: Asymmetrisches kryptosystem (2017), https://de.wikipedia.org/wiki/Datei:Orange_blue_digital_signature_de.svg
6. Wikipedia: Asymmetrisches kryptosystem (2017), https://de.wikipedia.org/wiki/Datei:Orange_blue_public_key_cryptography_de.svg