

# Azure Digital Twins: Eine Plattformanalyse und prototypische Umsetzung

1<sup>st</sup> Lucas Weiss

Fakultät für Informatik

Technische Hochschule Rosenheim

Rosenheim, Deutschland

lucas.weiss@stud.th-rosenheim.de

2<sup>nd</sup> Raphael Wudy

Fakultät für Informatik

Technische Hochschule Rosenheim

Rosenheim, Deutschland

raphael.wudy@stud.th-rosenheim.de

**Zusammenfassung**—Diese Arbeit untersucht die Anwendung der Microsoft Azure Digital Twins (ADT)-Plattform zur Erstellung eines digitalen Zwillings einer Windturbine. Sie umfasst eine Analyse der Plattform sowie eine prototypische Umsetzung. Der Fokus liegt auf der Evaluierung der ADT-Funktionalitäten, der Analyse zweier wissenschaftlicher Arbeiten und der Implementierung eines digitalen Schattens der Windturbine vom Typ AN Bonus 1000/54. Wichtige Ziele sind die Integration von IoT-Datenströmen, die Berechnung der Rotorblattstellung und die Visualisierung mittels des ADT Explorers. Ein Ansatz eines mathematischen Modells der Turbinenleistung wird umrissen.

Die Methodik umfasst theoretische Analysen, Dateninterpolation und die Modellierung der Systemarchitektur. Einschränkungen, wie der begrenzte Zugang zu Echtzeitdaten und andere Einschränkungen, werden berücksichtigt. Trotz dieser Herausforderungen ermöglicht der digitale Zwilling die Visualisierung von Systemverhalten.

**Index Terms**—Digital Twin (DT), Microsoft Azure Digital Twin (ADT), Windturbine, DT Architektur

## I. EINLEITUNG

Im Bereich der digitalen Zwillinge (eng. Digital Twins (DT)) erschienen in den letzten Jahren zahlreiche Plattformen zur Umsetzung wie z. B. Bosch IoT Things [1], Digital Twin in Google Cloud Plattform [2], AWS IoT TwinMaker [3], Ansys Twin Builder [4], Microsoft Azure Digital Twins [5]. Ein DT ist ein hilfreiches Werkzeug, um beispielsweise Veränderungen an einem physikalischen System (PS) zu testen, bevor diese Änderungen tatsächlich am PS durchgeführt werden. Ein Grund für das Erscheinen der zahlreichen Plattformen zum jetzigen Zeitpunkt und nicht bei der Entstehung des konzeptionellen Gedankens (ca. 1960 [6]) eines DTs, ist der Anstieg in der verfügbaren Rechenleistung und die breite Verfügbarkeit von IoT-Geräten [7]. Ohne diese sogenannten 'Enabler' ist die Umsetzung eines digitalen Zwillings erschwert. Neben dem Anstieg der Plattformen ist ein Anstieg in den veröffentlichten wissenschaftlichen Arbeiten zu verzeichnen [8].

## II. DIGITAL TWINS

Ein DT ist das virtuelle Abbild eines physikalischen Systems. Dabei stehen die Systeme in kontinuierlichen Datenaustausch, erhalten regelmäßige Updates über den Produkt-Lebenszyklus hinweg. Wichtig dabei ist die volle Integration der beiden Systeme zueinander. Eine Änderung in einem System führt zur Änderung im anderen. Ist dies nicht der

Fall, treffen die Definitionen von einem digitalen Schatten oder digitalen Modell zu (siehe Abbildung 1). Bei einem manuellen Update des digitalen Systems kann auch von einem digitalen Unterstützungssystem die Rede sein. Dabei ist zu beachten, dass ein DT nicht zwingend ein exaktes 3D-Modell des PS sein muss [7].

## III. METHODE

Diese Arbeit untersucht zwei relevante Paper aus dem Bereich Digital Twins unter der Verwendung der Microsoft Azure Digital Twins Plattform (ADT). In Bezug auf die Evaluation der Paper stehen der Anwendungsfall, die Architektur des DT, verwendete Technologien, Ergebnisse des Einsatzes des DT und Herausforderungen im Vordergrund. Neben der Dokumentenrecherche wird ein digitaler Zwilling in der ADT-Umgebung prototypisch umgesetzt.

## IV. DIE PLATTFORM MICROSOFT AZURE DIGITAL TWINS

„Azure Digital Twins ist ein PaaS-Angebot (Platform as a Service), das die Erstellung von Zwillingsgraphen auf der Grundlage von digitalen Modellen ganzer Umgebungen ermöglicht. Bei einer solchen Umgebung kann es sich um Gebäude, Fabriken, Höfe, Energieversorgungsnetze, Eisenbahnstrecken, Stadien und mehr handeln. Sogar ganze Städte lassen sich modellieren. Diese digitalen Modelle liefern wichtige Einblicke, um Produkte zu verbessern, Vorgänge zu optimieren, Kosten zu senken und die Benutzerfreundlichkeit zu revolutionieren.“ [10]

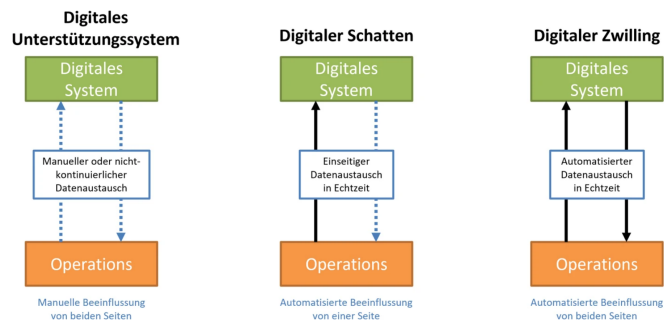


Abbildung 1. Arten von digitalen Systemen und der Kopplung zu einem physikalischen System [9]

### A. Überblick und Schwerpunkt der Plattform

Mit ADT ist der Anwender in der Lage ein eigenes Vokabular zu definieren und den digitalen Zwilling in den selbstdefinierten Begriffen des Unternehmens zu erstellen. Anwender definieren Modelle, die einer Klasse der objektorientierten Programmiersprache ähneln. In diesem Konstrukt wird die Datenform der realen Arbeitsumgebung festgelegt. Der digitale Zwilling entsteht später aus den Modell-Definitionen. ADT-Modelle werden in Digital Twin Definition Language dargestellt. Diese Sprache basiert auf JSON-LD [11].

Über Beziehungen zwischen den Zwillingen entstehen Zwillinggraphs. Diese sind visuell darstellbar und zeigen Beziehungen/Abhängigkeiten der Zwillinge in der gesamten Umgebung an [12]. Zwillinggraphs sind über eine ADT-Abfragesprache (ähnlich einer SQL-Abfragesprache) abrufbar. Das Abfragen von Eigenschaften, Modellen Beziehungen und deren Eigenschaften ist möglich [13]. Des Weiteren bietet das ADT-Environment Entwicklerschnittstellen. CLI-Befehle [14], APIs, SDKs [15] und einen ADT-Explorer [16] zur Visualisierung und Interaktion mit den Daten.

Der Dateneingang kann über das IoT-Hub von Azure oder allgemein über Cloud-native Integrationsdienste wie Function Apps realisiert werden. Dies ermöglicht eine Live-Datenintegration. ADT-Daten zur Speicherung oder zusätzlichen Verarbeitung sind über Endpunkte an andere Azure Dienste verfügbar [17].

Im Zusammenspiel mit der Live-Datenintegration bietet die Plattform Dienste zur ereignisgesteuerten Architektur an. Dadurch sind Automatismen und Benachrichtigungen bei Zustandsänderungen oder Ereignissen im digitalen Zwilling möglich [18].

Neben diesen Aspekten bietet die ADT-Plattform Funktionen zur Verwaltung und Sicherheit an. Damit sind sensible Daten geschützt und durch rollenbasierten Zugriff ist eine sichere Nutzung gewährleistet [19].

Die Plattform wurde entwickelt, um sich nahtlos in andere Azure-Dienste und Drittanbieter zu integrieren. Bestehende IoT-Systeme und Datenquellen können dadurch angebunden werden. Des Weiteren ermöglichen Echtzeitdatenverarbeitung und -analyse Abbildungen komplexer physikalischer Umgebungen [10].

### B. Evaluation Paper: Digital Twin of Wind Turbine Based on Microsoft® Azure IoT Platform [20]

1) *Anwendungsfall*: 'Digital Twin of Wind Turbine Based on Microsoft® Azure IoT Platform' beschreibt ein Proof of Concept zur Anwendbarkeit von DTs in komplexen mathematischen Modellierungen von Windturbinen (WT) unter der Verwendung von online verfügbaren Daten-Streaming. Umgesetzt mit Azure als PaaS.

2) *Architektur*: Das physikalische System ist ein Zusammenspiel aus einem mathematischen Modell einer Windturbine des Typen 4A IEC 6400-27-1-2020 und online verfügbaren Daten-Streaming-Diensten (Umgesetzt mit Matlab). Die Kopplung des physischen Systems mit der Azure Umgebung ist mit

einem MQTT Broker auf IoT Geräten in einem Cloud Gateway umgesetzt. Das IoT-Hub verteilt die eingehenden Daten an das Event-Hub, an den Digital Twin und an die mathematische Abbildung in Azure. Simulationen und Vorhersagen in Azure werden durch die Komponente Azure Machine-Learning bereitgestellt. Die generierten Daten aktualisieren den DT im Azure Data Explorer. Außerdem speichert die Anwendung die Daten in einem persistenten Datenspeicher (Azure Storage). Der 'Endanwender' kann sich die Daten über ein Custom App anzeigen lassen. Die Custom App bezieht die Daten aus dem Azure Storage.

Somit bildet Matlab mit dem mathematischen Modell und Online-Data-Streaming das physikalische System (PS), das Cloud Gateway und das IoT-Hub sind die verbindenden Elemente (CI), der DT mit mathematischem Modell bildet das virtuelle System (VS), Azure Machine-Learning kann dabei als Service (SVC) angesehen werden und der Azure Storage ist die Systemdatenbank (SD) der Umsetzung [20].

#### 3) *Eingesetzte Technologien*:

- Datenspeicherung: SQL Datenbank
- Datenübertragung: IoT mit MQTT Broker
- Datenerzeugung: mathematische Modelle, Maschinelles Lernen
- Datenquelle: Open Source Daten-Streaming

4) *Erfolg des DT*: Das Proof of Concept zeigt, dass der Einsatz eines DT mit ADT prinzipiell möglich ist und Zeit einsparen kann. Performance von operationalen Assets ist verständlicher dargestellt. Des Weiteren zeigt der DT bei der mechanischen Messung dieselbe Kurve wie das PS mit minimalen Abweichungen. Dadurch verifiziert der DT neben dem Proof of Concept das verwendete mathematische Modell der WT.

5) *Herausforderungen*: Einige Berechnungen müssen in der Cloud erfolgen, was Datenübertragung, Latenz und Bandbreitenlimitation zur Folge hat. Berechnungen sind ggf. langsamer und kostenintensiver durch Nutzung von Cloud Computing. Außerdem bleiben Aussagen zu Beginn des Papers ungeklärt. Wie z.B. der DT bei der angesprochenen, notwendigen Kostenreduktion behilflich sein kann. Auch auf die angesprochene Fehlerdiagnose wird nicht mehr näher eingegangen. Generell ist Vorgehensweise der Umsetzung nicht eindeutig nachvollziehbar, da die Autoren keine Implementierungsansätze und Datenquellen direkt nennen.

### C. Evaluation: Digital twin with Machine learning for predictive monitoring of CO2 equivalent from existing buildings [21]

#### 1) *Anwendungsfall*:

'Digital twin with Machine learning for predictive monitoring of CO2 equivalent from existing buildings' demonstriert die technische Machbarkeit eines digitalen Zwilling, der Echtzeitdaten aus IoT-Sensoren sammelt, analysiert und in ein 3D-BIM-Modell integriert. Es zeigt, wie ein solcher digitaler Zwilling genutzt werden kann, um Emissionen zu überwachen, Vorhersagen zu treffen und datengetriebene Entscheidungen zu ermöglichen. [21]

2) *Architektur*: Das physikalische System besteht aus einem Wohnraum-Szenario in einem kleinen Apartment in Belfast, Nordirland. Dieses ist mit IoT-Sensoren (SGP30 und AHT20) ausgestattet, die die Temperatur, relative Luftfeuchtigkeit, Total Volatile Organic Compounds (TVOC) und CO<sub>2</sub>-Äquivalent (eCO<sub>2</sub>) messen. Diese Daten werden von einem ESP32-Mikrocontroller erfasst und mittels MQTT-Protokoll über eine Wi-Fi-Verbindung an die Microsoft Azure Cloud übertragen.

Das Azure IoT Hub verteilt die eingehenden Sensordaten mittels Event Grid und Azure Functions an die Azure Digital Twin-Umgebung und speichert sie parallel in Azure Blob Storage Containern. In dem Azure Digital Twin Explorer werden die Daten in einer digitalisierten Abbildung des Gebäudemodells (3D-BIM-Modell) visualisiert.

Die Vorhersage künftiger CO<sub>2</sub>-Emissionen erfolgt durch mathematische Berechnung in Azure Machine Learning, das die Sensor Daten, in JSON-Format, aus den Blob Storage verarbeitet. Die Ergebnisse werden, nach erneuter Abspeicherung in Blob Storage, über eine Function App in die Twin Umgebung zurück eingespeist.

Die Visualisierung erfolgt über ein interaktives Dashboard in Microsoft Power BI, das sowohl aktuelle als auch vorhergesagte Emissionen darstellt. Nutzer können damit Trends und Anomalien erkennen, basierend auf Echtzeit- und historischen Daten.

Somit bildet das Wohnraumszenario mit IoT-Sensoren das physikalische System (PS), das Cloud-Gateway und das IoT-Hub sind die verbindenden Elemente (CI), der Azure Digital Twin mit integrierten Sensordaten bildet das virtuelle System (VS), Azure Machine Learning fungiert als Service (SVC), und Azure Blob Storage dient als Systemdatenbank (SD). [21]

#### 3) *Eingesetzte Technologien*:

- Datenspeicherung: Azure Blob Storage container, JSON-Format für strukturierte Speicherung
- Datenübertragung: IoT mit MQTT Broker, Wi-Fi zur Verbindung der IoT-Geräte mit der Cloud
- Datenerzeugung: ESP32 mit Sensoren (AHT20 für TVOC und eCO<sub>2</sub>, SGP30 für Temperatur und relative Luftfeuchtigkeit)
- Datenquelle: IoT-Sensoren in den Räumen: Wohnzimmer, Küche, Schlafzimmer, zusätzlich 3D-BIM-Modell

4) *Erfolg des DT*: Das Proof of Concept zeigt, dass der Einsatz eines Digital Twins mit Azure Digital Twin und maschinellem Lernen prinzipiell möglich ist und eine effiziente Überwachung von Emissionen in Echtzeit erlaubt. Die Integration von IoT-Daten und durch maschinelles Lernen gesammelten Daten in ein 3D-BIM-Modell ermöglicht eine benutzerfreundliche Visualisierung der Luftqualität. Darüber hinaus zeigen die vorhergesagten CO<sub>2</sub>-Emissionsdaten ein ähnliches Verhalten wie die realen Messdaten, mit einem Mean Squared Error (MSE) von maximal 0,6. Dies verifiziert neben dem Proof of Concept auch die Robustheit des eingesetzten maschinellen Lernmodells und die technische Machbarkeit der Architektur.

5) *Herausforderungen*: Einige Herausforderungen ergeben sich durch die notwendige Integration von IoT-Geräten, Cloud-Computing und Digital Twin Technologien. Eine Lösung für die langfristige Kalibrierung der eingesetzten Sensoren, wie z. B. des SGP30, ist erforderlich. Die Interoperabilität zwischen den IoT-Geräten, dem 3D-BIM-Modell und der Azure Digital Twin Umgebung wird durch die Nutzung standardisierter Modelle erleichtert, jedoch ist die Umsetzung auf größere und komplexere Szenarien nicht vollständig evaluiert. Der Digital Twin wird als 'digitaler Schatten' beschrieben, da er derzeit nur unidirektionale Datenflüsse (von Sensoren zum Digital Twin) unterstützt, und keine Maßnahmen zur Automatisierung oder bidirektionalen Steuerung implementiert sind. Der Fokus liegt primär auf der Überwachung und Vorhersage von CO<sub>2</sub>-Emissionen, ohne dass weiterführende Funktionen wie Fehlerdiagnosen oder Steuerungen in diesem Proof of Concept realisiert wurden.

## V. PROTOTYPISCHE UMSETZUNG

### A. Zielsetzung

Der Prototyp wird auf der Plattform Azure Digital Twin umgesetzt. Das betrachtete System ist eine Windturbine des Typen AN Bonus 1000/54. In der realen Welt gibt es viele verschiedene Typen von Windturbinen. Allerdings fiel die Entscheidung wegen drei Hauptgründen auf diese Windturbine.

- 1) Technische Daten zu dieser WT sind Open Source verfügbar (Windturbinen Model AN Bonus 1000/54 [22])
  - 2) Neben den technischen Daten sind historische Daten zur Leistungskurve verfügbar.
  - 3) Diese Windturbine ist vom Typen Onshore.
1. und 2. sind aus Gründen der Nachvollziehbarkeit und der Vorgaben zur Verwendung von Open Source Quellen notwendig. 3. lässt ein realitätsnahes Szenario abbilden, da Umgebungsdaten (Windgeschwindigkeit, Temperatur etc.) über Städte abgefragt werden und nicht über Küsten und Meere. Weitere Limitationen werden in V-D beschrieben.

### B. Anwendungsfall

Windturbinen sind dezentrale Einheiten eines Gesamtsystems (Stromerzeugung durch erneuerbare Energien). Diese dezentralen Einheiten sind an Räumlichkeiten gebunden und fernab von sofortigen Zugriff. Anders als bei einer Produktion ist das Maintenance-Team meistens nicht direkt vor Ort. Somit ist ohne Überwachung und Kopplung der Windturbine mit IoT der Status nur bekannt, wenn der Windturbinen-Betreiber zyklisch Wartungsarbeiten oder Überprüfungen der Windturbine vornimmt. Aus diesem Grund ist die Verbindung der Windturbine mit IoT und die Erstellung eines Digital Twins ein wünschenswerter Anwendungsfall. Neben der Überwachung des aktuellen Status, können Optimierungen und Simulationen durchgeführt werden, ohne die Windturbine vor Ort aufzusuchen oder in speziellen Testing-Facilities (sofern existierend) aufzubauen. Das bedeutet, ein Windturbinen-Betreiber kann proaktiv Wartungspläne anhand historischer Daten erstellen oder Algorithmen für die Anpassung der Rotorblattstellung mit Echtzeitdaten im digitalen Zwilling

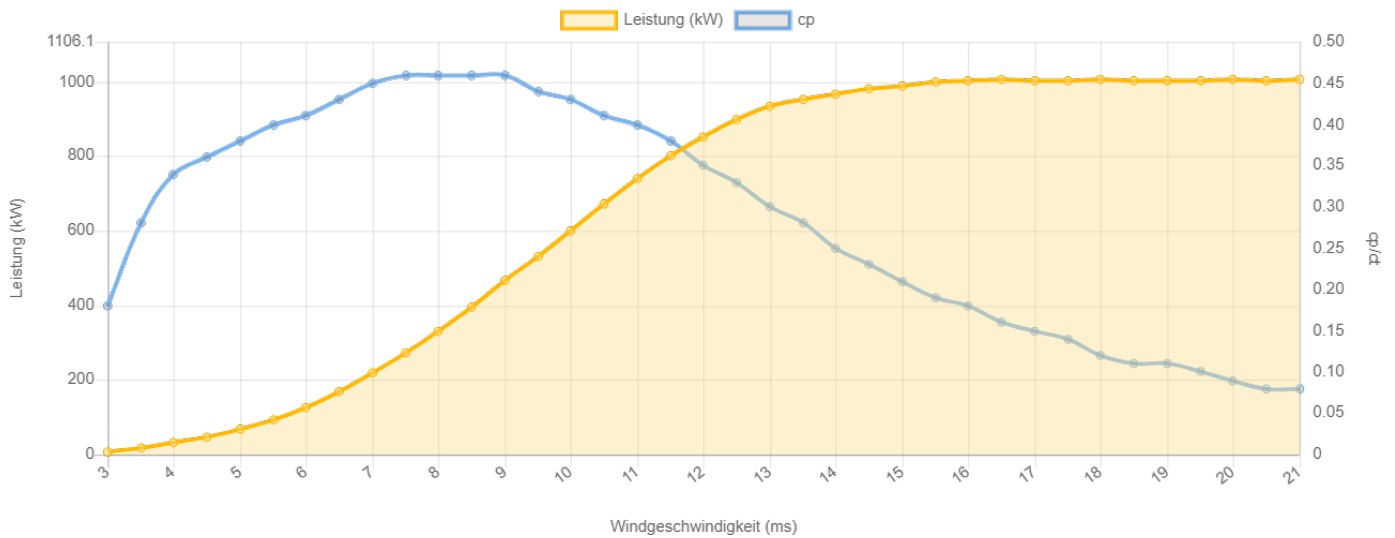


Abbildung 2. Leistungsdaten der WT AN Bonus 1000/54 [22]

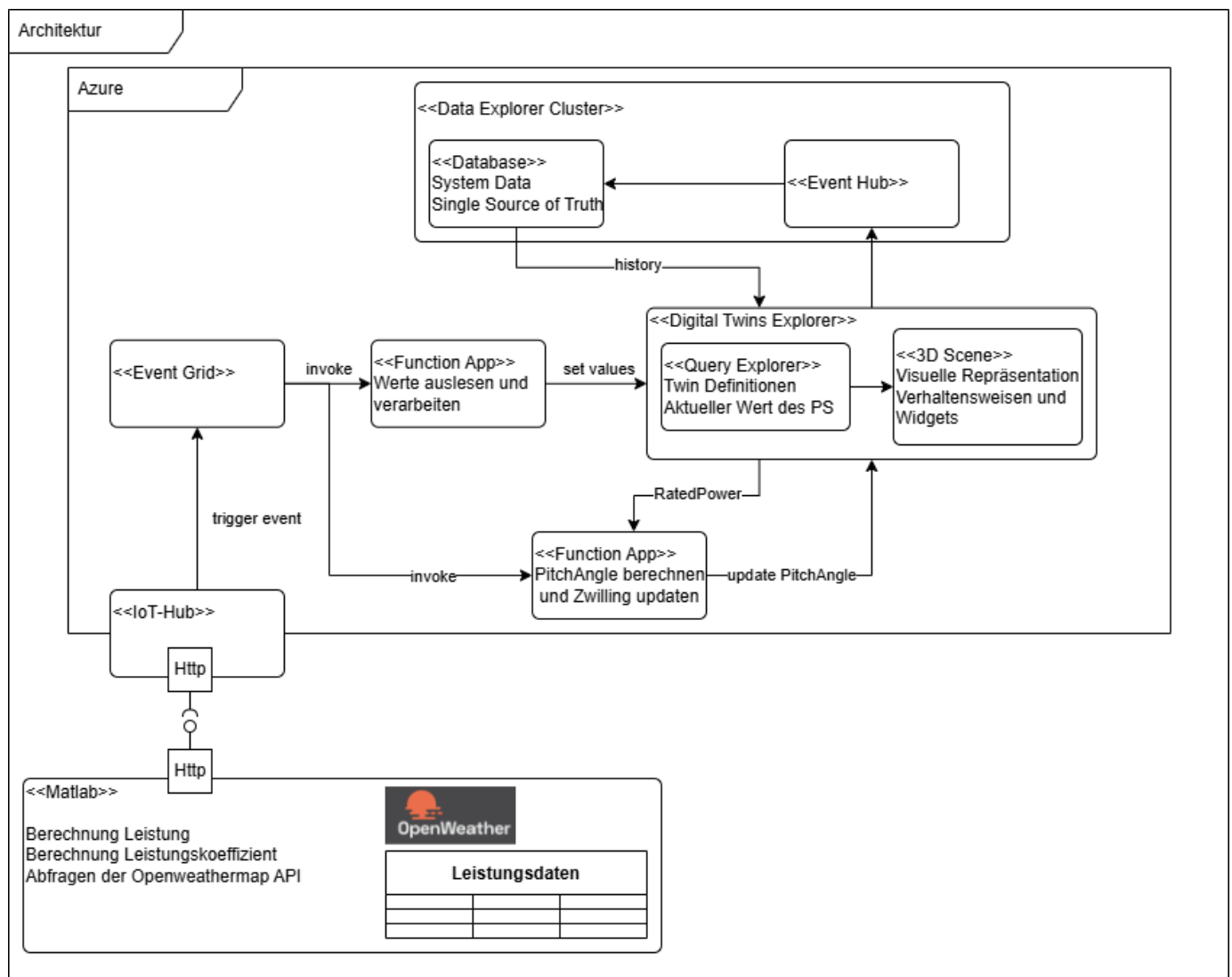


Abbildung 3. Architektur Diagramm des Gesamtsystems

optimieren und deren Auswirkungen einschätzen bevor das physikalische System angepasst wird.

Außerdem bekommt ein Maintenance-Team andere Einblicke in den Service-Fall. Bereits vor Eintreffen an der Windturbine, sind Informationen über den Status der WT bekannt. Ist der Rotor defekt? Erzeugt der Generator Strom? usw.

### C. Anforderungen

#### Physisches System:

- Leistungsdaten der Windturbine müssen Open Source verfügbar sein.
- Die Leistungsdaten der Windturbine sind mittels IoT, an die Azure Plattform zu übermitteln.
- Sensordaten zur Umwelt müssen über IoT an die Azure Plattform gesendet werden.

#### Kommunikations-Interface:

- Das Datenübertragungsformat müssen die Entwickler festlegen.
- Die Entwickler müssen sich an das festgelegte Datenübertragungsformat halten.
- Anpassungen am Datenübertragungsformat muss unter gemeinsamer Absprache der Entwickler stattfinden.

#### Services:

- Die eingehenden Daten sind Grundlage für Berechnungen in der digitalen Welt
- Wenn historische Daten vorhanden sind, müssen die Berechnungen die historischen Daten repräsentieren.
- Eine zu starke Abweichung spiegelt kein realitätsnahe Abbildung des physikalischen Systems wider.
- Festlegung eines Gates von einem maximalen Fehler  $< 0.05$  in Berechnungen.

#### Virtuelles System:

- Die eingehenden Daten aktualisieren das virtuelle System.
- Die Daten sollen visuell repräsentiert werden.
- Die visuellen Repräsentation sollen Verhaltensweisen aufweisen.

#### System Daten:

- Der digitale Zwilling muss die eingehenden Daten speichern.
- Der digitale Zwilling muss die berechneten Werte speichern.

### D. Limitationen

Da in der Recherchezeit dieser Arbeit keine Open Source Quellen zur Echtzeitdatenverarbeitung von Sensordaten, einer im Betrieb befindlichen Windturbine, gefunden wurden, wird das PS mit Limitationen umgesetzt. Dies betrifft Sensordaten und Leistungsdaten der Windturbine. Die Sensordaten zur Umwelt werden über die Openweathermap API [23] angefragt. Diese API erlaubt im kostenlosen Plan bis zu 1000 Anfragen am Tag. Aus diesem Grund ist keine Echtzeitübertragung mit sekundlichen Updates möglich. Hier würden 86.400 Anfragen täglich das Limit bei weitem überschreiten. Zur Annäherung sendet das PS alle drei Minuten die Daten.

Des Weiteren sind keine Leistungsdaten über eine API zum ausgewählten Windturbinen-Typ Open Source verfügbar. In den meisten Fällen unterliegen diese Daten der Geheimhaltung. Allerdings können, wie unter V-A erwähnt, technische Daten und historische Leistungsdaten über ein Datenblatt und Diagramm extrahiert werden. Dies ist nur händisch möglich. Aus diesen Gründen ist es nicht möglich einen digitalen Zwilling für eine Windturbine zu erstellen. Durch die fehlende Rückkopplung an das PS durch die genannten Limitationen wird ein digitaler Schatten implementiert. Lediglich der letzte Schritt, das Zurücksenden der berechneten Daten über IoT an einen Sensor o.ä. der Windturbine, ist nicht möglich.

Ein Vergleich der berechneten optimalen Rotorblattstellung mit der realen ist ebenfalls nicht möglich, da hier keine Sensordaten oder historischen Daten Open Source verfügbar sind. Deswegen kann das virtuelle System nur mathematische Annäherungen ohne Vergleich mit realen Daten anzeigen. Neben diesen Fakten ist die aktuelle Umsetzung und deren Datenhaltung ebenfalls limitiert. Über den Premium-Plan in Azure für ein Data Cluster können die Daten sieben Tage rückwirkend gespeichert werden. Der Premium-Plan ist mit Kosten verbunden. Diese sind mit den Vorgaben für das Studienprojekt zu vereinbaren.

### E. Architektur und Aufbau

Die eingangs erwähnte Windturbine des Typen AN Bonus 1000/54 weist Leistungs- und technische Daten als Open Source Quelle auf und dient daher als Grundlage für das PS. Die Leistungsdaten können der Abbildung 2 entnommen werden. Die Daten sind in 0.5er Abständen vorhanden (vgl. Tabelle I). Damit ein glatter Verlauf bei Windgeschwindigkeiten wie 8.7 oder 3.9 möglich ist, werden die über die Openweathermap API [23] angefragten Wetterdaten in einem Skript verarbeitet und über die historischen Daten interpoliert. Dieses Konstrukt dient als PS und sendet über einen Http-Client

- Technische Daten der Windturbine (Name, CutIn-Speed, CutOut-Speed etc.)
- interpolierte Leistungsdaten (Kilowatt und  $C_p$ )
- Rotordaten (Rotorradius)
- Umweltdaten (Standort, Windgeschwindigkeit, Temperatur, Luftfeuchtigkeit etc.)

an das Azure IoT-Hub (vgl. Abbildung 3). Dort werden die Daten empfangen und mittels Event Grid und einer Function App an die digitale Zwillings Definition gesendet. Die Function App aktualisiert dabei den neuesten Wert im digitalen Zwilling.

Im Azure Digital Twin Explorer ist neben der Definition eine 3D-Szene angelegt. Diese repräsentiert die Windturbine als digitales Abbild. Das Abbild ist mit Verhaltensweisen an die Variablen des digitalen Zwillings gekoppelt und erhielt bestimmte Regeln wie z. B. eine grüne Einfärbung des Generator-Blocks, wenn die ausgehende Leistung der Windturbine über 900 KW liegt oder rote Rotorblätter, wenn die Rotorblattstellung nicht berechnet, also 0 ist.

Durch das aufgesetzte Azure Data Cluster speichert Azure die aktuellen Werte des digitalen Zwillings in einer Azure

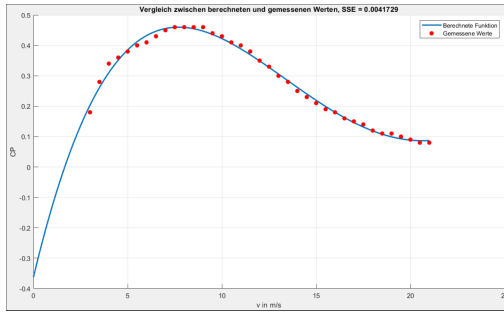


Abbildung 4. Kurve des mathematischen Modells der Windturbine und historisch gemessene Datenpunkte im Vergleich

Tabelle I  
WINDGESCHWINDIGKEIT  $v$ , LEISTUNGSKOEFFIZIENT UND LEISTUNG DER  
AN BONUS 1000/54 IN TABELLARISCHER UND REDUZIERTER FORM ZU  
ABBILDUNG 2

$v$ (m/s)	$C_p$	Leistung in KW
3.0	0.180	7.43
3.5	0.280	17.11
4.0	0.340	30.89
4.5	0.360	46.37
5.0	0.380	67.45
...	...	...
21.0	0.080	1005.58

Storage Cluster Datenbank, welche für Anwendungen dieser Art ausgelegt ist. Dadurch sind z.B. Datenverläufe als Widgets in der 3D-Szene konfigurierbar. Ein Event Hub koordiniert den Datenfluss zwischen den Azure Storage-Clustern und dem Digitalen Zwilling und sorgt dabei für eine effiziente und zuverlässige Datenübertragung.

Die Rotorblattstellung berechnet sich bei Dateneingang des PS über das IoT-Hub neu. Die dafür zuständige Function App aktualisiert mit den berechneten Werten den digitalen Zwilling. Ab diesem Zeitpunkt übernimmt das Event Hub und das Data Cluster die Datenhaltung.

## VI. THEORETISCHE GRUNDLAGEN FÜR BERECHNUNGEN IM DIGITALEN ZWILLING

### A. Berechnung der Rotorblattstellung

Aus wissenschaftlichen Artikeln zur Windkraft und Windturbinen geht hervor, dass die Rotorblattstellung (Pitch-Angle) individuell über einen Algorithmus direkt an der Windturbine gesteuert wird. Für eine realistische Berechnung der Pitch-Angle sind Daten der Windturbinen notwendig. Die Steuerung der Pitch-Angle in diesem Prototypen richtet sich nach der aktuellen und Nennleistung der Windturbine.

Unter Berücksichtigung der Einschalt-, Abschalt- und Nennwindgeschwindigkeiten der Windturbine gelten (mit leichten Abweichungen je nach Typ) drei Winkelbereiche. 0 - 5°, ab Einschaltgeschwindigkeit, damit die Rotorblätter sich zu drehen beginnen. 15 - 20° bis zur Nennwindgeschwindigkeit und 90° ab Abschaltgeschwindigkeit zum Schutz vor Überlast. Zwischen diesen Bereichen liegt in diesem Fall ein linearer Verlauf (siehe Gleichung 1).

$$\theta(v_{\text{wind}}, C_p) = \begin{cases} \theta_{\min} & \text{für } P_{\text{turbine}} < P_{\text{rated}} \\ \theta_{\max} & \text{für } P_{\text{turbine}} \geq P_{\text{rated}} \end{cases} \quad (1)$$

Linearer Verlauf

Unter Einhaltung der gesetzten Bedingungen, berechnet sich die Pitch-Angle in Abhängigkeit zur Leistung über:

- 1)  $(P_{\text{turbine}} < P_{\text{rated}}) : \theta = 5$
- 2)  $(P_{\text{turbine}} > P_{\text{rated}}) : \theta = 15 + \left( \frac{P_{\text{turbine}} - P_{\text{rated}}}{P_{\text{rated}}} \right) \cdot 75$ ; Zusatz:  
Wenn  $\theta > 90$  dann  $\theta = 90$
- 3) Ansonsten :  $\theta = 5 + \left( \frac{P_{\text{turbine}}}{P_{\text{rated}}} \right) \cdot 10$

### B. Mögliches mathematisches Modell der Windturbine

Nicht in der Umsetzung enthalten, aber erwähnenswert, ist das anhand der Leistungsdaten errechnete mathematische Modell der Windturbine. Eine Implementierung fand aus zeitlichen Gründen nicht mehr statt. Durch dieses Modell könnten Simulationen im digitalen Zwilling ermöglicht werden.

Mit der linearen Ausgleichsrechnung und den vorhandenen Leistungsdaten war es möglich, anhand der Abbildung 2 eine Modellfunktion (siehe Gleichung 2) abhängig von der Windgeschwindigkeit  $v$  zu formulieren.

$$C_p(v; k_0, k_1, k_2, k_3, k_4) = k_0 v^4 + k_1 v^3 + k_2 v^2 + k_3 v + k_4$$

mit  $k_0 = \phi_0, k_1 = \phi_1, k_2 = \phi_2, k_3 = \phi_3$  und  $k_4 = \phi_4$  (2)

Mit allen Datenpunkten aus dem Graphen 2 zu  $C_p$  ergibt sich eine 37x5 Matrix nach dem Schema

$$\begin{bmatrix} \phi_0(v^{(1)}) & \phi_1(v^{(1)}) & \phi_2(v^{(1)}) & \phi_3(v^{(1)}) & \phi_4(v^{(1)}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_0(v^{(37)}) & \dots & \dots & \dots & \phi_4(v^{(37)}) \end{bmatrix}$$

Zur Errechnung der Parameter

$$k_0, k_1, k_2, k_3 \text{ und } k_4$$

fand die Matlab Funktion 'polyfit' Verwendung. Die daraus resultierende optimale Anpassung (Gleichung 3) ist:

$$C_p(v) = -1.0588 \cdot 10^{-5} v^4 + 9.4546 \cdot 10^{-4} v^3 - 0.026546 v^2 + 0.25998 v - 0.36238 \quad (3)$$

Mit einer aufsummierten Fehlersumme (SSE) von 0.0041729. Übereinander gelegt beschreibt die Gleichung unter 3 die Kurve aus Abbildung 4. Neben dieser Gleichung ist die allgemeine Formel zur Berechnung der Windkraft notwendig. Diese ist definiert durch Gleichung 4 [24]. Wird das Ergebnis der Gleichung 3 mit  $P_{\text{wind}}$  aus Gleichung 4 multipliziert, ist das Ergebnis die Leistung der Windturbine abhängig vom Leistungskoeffizienten  $C_p$  in Watt.

$$P_{\text{wind}} = \left( \frac{1}{2} \right) \cdot \rho \cdot A \cdot v^3 \quad (4)$$



Abbildung 5. DT Definition des Power-Models

## VII. UMSETZUNG IN AZURE

### A. IoT-Hub

Das IoT-Hub erlaubt es den Entwicklern Geräte zur Kommunikation zu registrieren. Die verwendeten Geräte müssen keinen physikalischen Gegen-Part besitzen. In dieser Umsetzung wurde ein Gerät angelegt ohne Gegenspieler. Allerdings erstellt das IoT-Hub für dieses Gerät Verbindungs-Strings und Keys. Hier per Shared Access Token (SAS) und Verbindungs-String via Http. Diese Kommunikation ist, im Gegensatz zu einer Socket-Verbindung per MQTT, unidirektional nach Azure.

### B. Function Apps

In diesem Projekt spielt eine Function App eine zentrale Rolle bei der Arbeit mit dem ADT, da sie die Daten vom IoT-Hub an den ADT weiterleitet. Eine Function App ist ein serverloser Compute-Dienst, der auf definierte Trigger reagiert. In diesem Fall wird ein EventGrid-Trigger verwendet, der in den Events-Einstellungen des IoT-Hubs konfiguriert ist. Alle JSON-Payloads, die an den IoT-Hub gesendet werden, werden an die Function App weitergeleitet. Dort werden sie deserialisiert und anschließend in ein JsonPatchDocument aus der Bibliothek Azure.DigitalTwins.Core serialisiert. Diese Bibliothek verwaltet die Kommunikation mit dem ADT. Die Authentifizierung erfolgt über die Rolle Azure Digital Twins Data Owner.

Für die Berechnung des PitchAngle wird eine weitere Function App verwendet, die ein ähnliches Setup wie die vorherige besitzt. Im Gegensatz zur direkten Weiterleitung der Werte an den Digitalen Zwilling werden in dieser Function App Berechnungen durchgeführt. Dazu werden zusätzlich Daten aus dem Digitalen Zwilling benötigt, die ebenfalls über die Azure.DigitalTwins.Core-Bibliothek abgerufen werden. Die berechneten Ergebnisse werden anschließend wieder in den Digitalen Zwilling zurückgespielt.

### C. Digital Twin Explorer

Für die Verwendung des Azure Digital Twins muss zunächst eine Definition eines Datenmodells festgelegt sein. Das verwendete Format ist Digital Twin Definition Language (DTDL) [25], [26]. Diese Definitionen (vgl. Abbildung 5) sind im Digital Twin Explorer angelegt und beinhalten die Variablen, Datentypen und den Kontext. Diese definierten Models können in der Azure Umgebung angesprochen werden, für bspw. ein Update der Eigenschaft 'Kilowatts'.

Neben der Definition des DTs im Digital Twin Explorer kann eine visuelle Repräsentation mit Verhaltensweisen und Widgets erstellt werden. Die sogenannte 3D-Szene des DTs ist eine 3D-Abbildung im Format .glb oder .gltf (vgl. Abbildung 6). Ist die Modellierung segmentiert und die einzelnen Elemente sind auswählbar, so lassen sich die einzelnen Variablen des DTs mit dem 3D-Modell verbinden. Erst jetzt sind Verhaltensweisen und Widgets konfigurierbar. In dieser Umsetzung sind verschiedene Verhaltensweisen und Widgets implementiert (vgl. Abbildung 7).

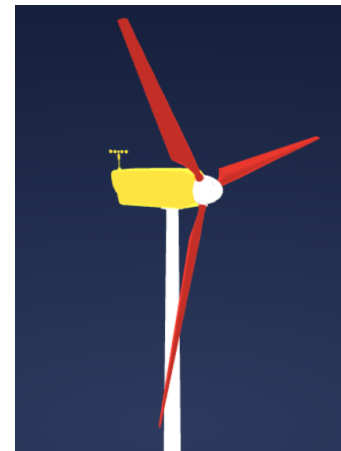


Abbildung 6. 3D-Modell der Windturbine in ADT mit angewendeten visuellen Regeln

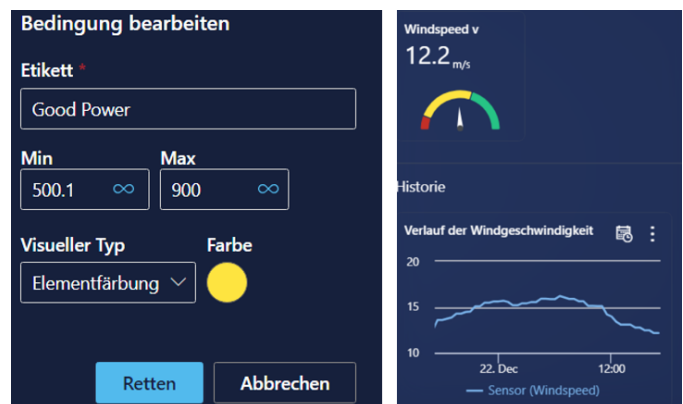


Abbildung 7. Links: Visuelle Regeln für die Leistung der Windturbine; rechts: Konfiguriertes Widget der gemessenen Windgeschwindigkeit (aktuell und im historischen Verlauf)



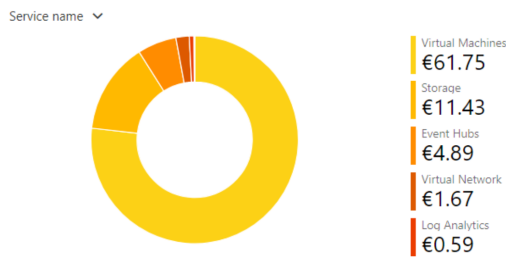


Abbildung 10. Kostenverteilung nach Services

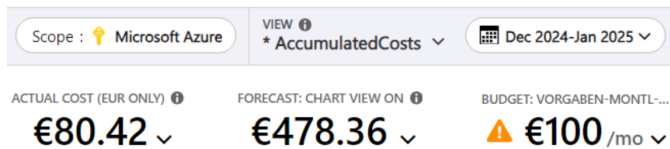


Abbildung 8. Aktuelle Kosten, Kostenprognose und monatliches Budget. Zeitraum Dezember 2024 - Januar 2025

#### D. Data Cluster und Storage

Damit der digitale Zwilling in ADT sein volles Potential zeigen kann, ist ein kostenpflichtiges Abonnement notwendig. Über bspw. einen Premium-Plan kann ein Data Cluster mit Data History und einer Datenbank eingerichtet werden. Nach erfolgreicher Einrichtung und Verbindung des Data Clusters samt Datenbank mit dem Event Hub, müssen keine weiteren Schritte zur Datenhaltung vorgenommen werden. Die Funktionalität zur Data History ist im Digital Twin Explorer freigeschalten und kann konfiguriert werden.

### VIII. KOSTENBERICHT

Wie in Abschnitt V-D erwähnt, ist ein Datenverlauf nur möglich, indem ein Azure Data Cluster im Premium-Plan verwendet wird. Diese damit verbundenen Kosten schienen zunächst unerheblich und den Vorgaben entsprechend. Allerdings zeigt die Prognose von Azure und die ersten 11 Tage des Zahlplans ein anderes Bild.

Der benötigte Premium-Plan ist mit 22€/Monat für sieben Tage rückwirkenden Storage und einem Data Cluster im Budget (100€/Monat) der einzelnen Projekte. Was nicht im Premium-Plan enthalten ist und auf den ersten Blick nicht ersichtlich war, ist die Miete der virtuellen Maschinen, die im Hintergrund laufen und weitere Kosten verursachen (vgl. Abbildung 10)). Diese Kosten sind nicht unerheblich und limitieren die Verwendung der Plattform weiter. Alleine in den ersten 11 Tagen nach Einrichten des Azure Data Clusters samt Storage und dazugehörigen Event Hubs stiegen die Kosten von 0.02€ auf 80.88€ (vgl. Abbildung 9).

Da die Kosten in diesem kurzen Zeitraum stark angestiegen sind und die Prognosen die Vorgaben weit übersteigen (vgl. Abbildung 8), wurden zwischen der Zwischenpräsentation und einer Woche vor Abgabe der Datentransfer zwischen dem physikalischen System und Azure reduziert bzw. eingestellt.

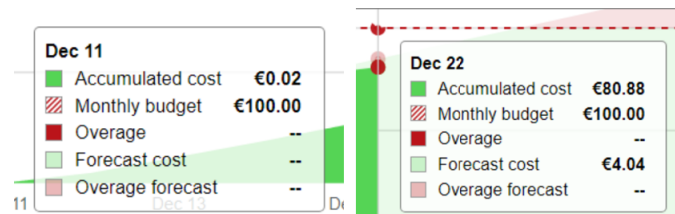


Abbildung 9. Aufsummierte Kosten 11.12.-22.12.2024, Stand 18:00 Uhr

### IX. DISKUSSION

Die Plattform Azure Digital Twins ist ein guter Werkzeugkasten, um einen digitalen Zwilling umzusetzen. Es gibt unterschiedliche Möglichkeiten dies zu erreichen. Neben Software und Systems Engineering Themen sind 3D-Szenen für andere Zielgruppen direkt in die Plattform integriert. Auch mit der Programmiersprache C# ist ein Framework vorhanden, welches neben der Plattform selbst ein großes Forum an Dokumentation und Know-How zur Verfügung stellt. Dadurch, dass die Dokumentation der Plattform öffentlich einsehbar ist und die Plattform selbst sich an Standard-Protokolle und Definitionen hält, ist es ebenfalls möglich, sich einfache Konfigurationsschritte interaktiv mit ChatGPT beizubringen. Dies ermöglicht eine leichtere Lernkurve.

Einige wenige Funktionen, wie z. B. die Verhaltensweisen Einstellungen in der 3D-Szene sind intuitiv gestaltet, andere sind dagegen eher ein technischer Verwaltungsakt. Deswegen sind Vorkenntnisse für eine reibungslose Umsetzung notwendig. Ohne diese Vorkenntnisse und Dokumentation ist die Plattform nicht selbst erklärend oder gibt einem nur wenig Hilfestellung. Jede Applikation in der Azure-Plattform muss eingerichtet werden. Es müssen für bspw. Function Apps spezielle Berechtigungen für Aktionen gesetzt werden usw. Berechtigungen für z. B. den Datenverlauf benötigen die 'Endanwender' der 3D-Szene. Es ist zwar beschrieben, wie Berechtigungen dafür einzurichten sind, aber erst nach tiefer Recherche und einigen Chatanfragen über ChatGPT war bekannt, um welche Berechtigungsrolle es sich genau handelt.

Auch die Übersetzung der Plattform in die deutsche Sprache lässt zu wünschen übrig. In der heutigen Zeit und IT-Branche ist es zwar Usus, mit der englischen Sprache umzugehen, aber wenn die Voreinstellungen der Anwendung auf der Einstellung des Browsers beruhen, dann sollte die Übersetzung auch sitzen. Beispiele hierfür wären: 'Retten' statt 'Save' (vgl. Abbildung 7) oder 'Entlassen' für 'Dismiss' und 'Immobilie' für 'Property'. Im Grunde ist die Anwendung schwer nutzbar mit der deutschen Übersetzung.

### X. FAZIT

Sind Kenntnisse im Umgang mit Azure und C# vorhanden, so ist es möglich, einen Digitalen Schatten/Zwilling in angemessener Zeit zu implementieren. Diese Umsetzung wurde innerhalb eines Semesters entwickelt und fertiggestellt. Die Ergebnisse zeigen einen soliden Grundstock, der für die Weiterentwicklung genutzt werden kann. Es ist ein vollständig eingerichtetes IoT-Hub mit Verbindungsbeispielen und Daten



vorhanden, die Definition des digitalen Zwillings ist eingerichtet und mit Verhaltensweisen im Ansichtsmodus des Azure Digital Twin Explorers vorhanden. Die Daten sind sieben Tage rückwirkend vorhanden (dies kann mit einem Update des Abos erhöht werden). Ebenso können Berechnungen mit realen Daten als Grundlage über das IoT-Hub erfolgen. Alles in allem ist die Zielsetzung erfolgreich umgesetzt, und der Anwendungsfall ist abgedeckt. Der reale Einsatz und damit einhergehende Verifizierung des gesamten Konzepts bleibt in diesem Fall aus, da diese Arbeit im Rahmen einer Studienarbeit erstellt wurde und der produktive Einsatz nicht weiterverfolgt wird.

## LITERATUR

- [1] B. GmbH, "Bosch iot suite - things dokumentation," 2024, zugriff am: 01. November 2024. [Online]. Available: <https://docs.bosch-iot-suite.com/things>
- [2] C. Global, "Digital twin in gcp," 2024, zugriff am: 01. November 2024. [Online]. Available: <https://www.c2global.com/articles/digital-twin-in-gcp-2531>
- [3] I. Amazon Web Services, "Was ist aws iot twinmaker?" 2024, zugriff am: 01. November 2024. [Online]. Available: [https://docs.aws.amazon.com/de\\_de/iot-twinmaker/latest/guide/what-is-twinmaker.html](https://docs.aws.amazon.com/de_de/iot-twinmaker/latest/guide/what-is-twinmaker.html)
- [4] I. Ansys, "Ansys digital twin," 2024, zugriff am: 01. November 2024. [Online]. Available: <https://www.ansys.com/de-de/products/digital-twin>
- [5] M. Corporation, "Azure digital twins," 2024, zugriff am: 01. November 2024. [Online]. Available: <https://azure.microsoft.com/de-de/products/digital-twins>
- [6] S. D. I. Software, "Apollo 13: The first digital twin," <https://blogs.sw.siemens.com/simcenter/apollo-13-the-first-digital-twin/>, 2024, zugriff am: 01. November 2024.
- [7] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020.
- [8] Z. Lv and E. Fersman, *Digital Twins: Basics and Applications*. Springer, 2022.
- [9] J. Neugebauer, L. Heilig, and S. Voß, "Anforderungsanalyse zur umsetzung eines digitalen zwillings im containerterminal," *HMD*, vol. 60, pp. 110–131, 2023, received: 05 November 2022; Accepted: 13 December 2022; Published: 23 January 2023. [Online]. Available: <https://doi.org/10.1365/s40702-022-00941-1>
- [10] M. Corporation, "Azure digital twins documentation," <https://learn.microsoft.com/de-de/azure/digital-twins/>, 2024, zugriff am: 01. November 2024.
- [11] —, "Azure digital twins: Konzepte und modelle," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-models>, 2024, zugriff am: 02. November 2024.
- [12] —, "Azure digital twins: Konzepte des twin-graphen," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-twins-graph>, 2024, zugriff am: 02. November 2024.
- [13] —, "Azure digital twins: Konzepte der abfragesprache," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-query-language>, 2024, zugriff am: 02. November 2024.
- [14] —, "Azure digital twins: Konzepte der befehlszeilenschnittstelle (cli)," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-cli>, 2024, zugriff am: 02. November 2024.
- [15] —, "Azure digital twins: Konzepte von apis und sdks," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-apis-sdks>, 2024, zugriff am: 02. November 2024.
- [16] —, "Azure digital twins: Konzepte des azure digital twins explorers," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-azure-digital-twins-explorer>, 2024, zugriff am: 02. November 2024.
- [17] —, "Azure digital twins: Konzepte des datenflusses (ingress und egress)," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-data-ingress-egress>, 2024, zugriff am: 02. November 2024.
- [18] —, "Azure digital twins: Konzepte der ereignisbenachrichtigungen," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-event-notifications?tabs=eventgridevents>, 2024, zugriff am: 02. November 2024.
- [19] —, "Azure digital twins: Sicherheitskonzepte," <https://learn.microsoft.com/de-de/azure/digital-twins/concepts-security>, 2024, zugriff am: 02. November 2024.
- [20] R. Issa, M. S. Hamad, and M. Abdel-Gelil, "Digital twin of wind turbine based on microsoft® azure iot platform," in *2023 IEEE Conference on Power Electronics and Renewable Energy (CPERE)*, 2023, pp. 1–8.
- [21] A. Arsiwala, F. Elghaish, and M. Zoher, "Digital twin with machine learning for predictive monitoring of co<sub>2</sub> equivalent from existing buildings," *Energy and Buildings*, vol. 284, p. 112851, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378778823000816>
- [22] An bonus-1000/54 wind turbine. Wind-Turbine-Models.com. Zugriff am 30. November 2024. [Online]. Available: <https://www.wind-turbine-models.com/turbines/396-an-bonus-1000-54>
- [23] Openweathermap api. OpenWeatherMap. Zugriff am 30. November 2024. [Online]. Available: <https://openweathermap.org/api>
- [24] OpenEI, "Gateway: Wind," 2024, zugriff am: 23.12.2024. [Online]. Available: <https://openei.org/wiki/Gateway:Wind>
- [25] A. D. T. Team, "Digital twins definition language (dtdl) v3," 2024, zugriff am 23.12.2024. [Online]. Available: <https://github.com/Azure/opensdtwin/blob/master/DTDL/v3/DTDL.v3.md>
- [26] Microsoft, "Digital twins definition language: Models," 2024, zugriff am 23.12.2024. [Online]. Available: <https://learn.microsoft.com/en-us/azure/digital-twins/concepts-models>