

ERARBEITUNG UND REFLEXIONSPHASE

Niraj Bista/10781594

Inhaltsverzeichnis

Erarbeitung und Reflexionsphase	2
Abteilung der fachlichen Anforderungen	2
UML- Use-Case-Diagramm.....	2
Technische Umsetzung der Systemkommunikation	3
Sequenzdiagramm der Bewässerungsvorgang.....	4
Komponenten Diagramm	6
Things	7
Zustandsdiagramm Sensor Thing	8
Zustandsdiagramm Pump Thing	10
Arduino Cloud	12
GitHub-Repository	13
Abbildungsverzeichnis	14

Erarbeitung und Reflexionsphase

In der Erarbeitungsphase wurde das Smart Bewässerungssystem konzipiert, modelliert und technisch vorbereitet.

Abteilung der fachlichen Anforderungen

Das System soll die Bodenfeuchtigkeit erfassen, diese im Cloud Dashboard anzeigen und bei Unterschreiten eines konfigurierbaren Schwellwerts automatisch eine Bewässerung auslösen. Zusätzlich soll der Benutzer die Bewässerung manuell über das Web-Dashboard starten können.

Damit ergeben sich folgende zentrale Funktionen des Systems:

- Erfassung von Sensordaten
- Visualisierung der Messwerte der Bewässerung
- Manuelle Steuerung durch den Benutzer

UML- Use-Case-Diagramm

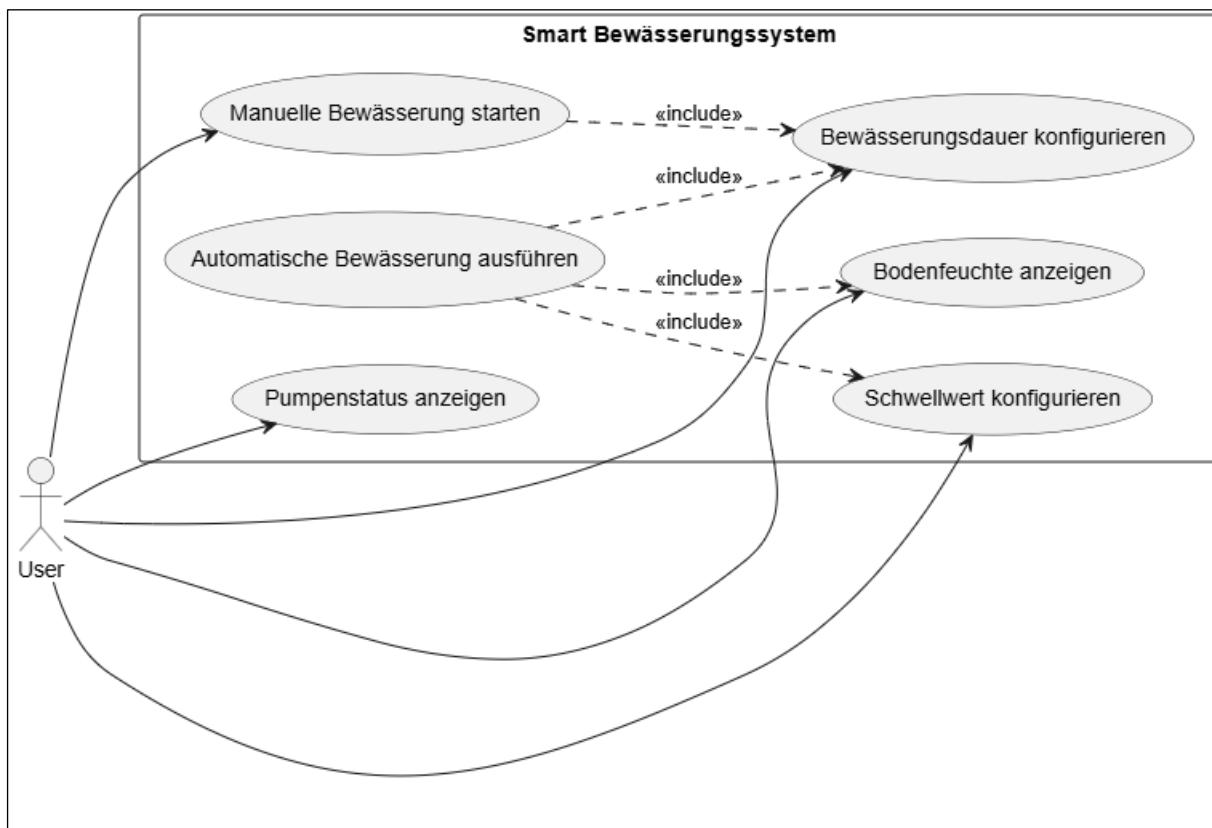


Abbildung 1: Use-Case-Diagramm der Anwendung.
(Quelle: eigene Darstellung)

Das Use-Case-Diagramm in Abbildung 1 beschreibt die funktionalen Interaktionen zwischen dem Akteur User und dem System „Smart Bewässerungssystem“.

Der User kann die aktuelle Bodenfeuchtigkeit anzeigen, den Schwellwert für die automatische Bewässerung konfigurieren, die gewünschte Bewässerungsdauer festlegen, eine manuelle Bewässerung starten sowie den aktuellen Pumpenstatus einsehen. Die automatische Bewässerung wird als eigener Anwendungsfall modelliert. Dieser Anwendungsfall beinhaltet die Anzeige der aktuellen Bodenfeuchtigkeit sowie die Berücksichtigung des konfigurierten Schwellwertes und der eingestellten Bewässerungsdauer. Diese funktionalen Abhängigkeiten werden im Diagramm durch <>include<>-Beziehungen dargestellt.

Auch der Anwendungsfall „Manuelle Bewässerung starten“ verwendet die zuvor konfigurierte Bewässerungsdauer und ist daher ebenfalls über eine <<include>> Beziehung mit dem Anwendungsfall „Bewässerungsdauer konfigurieren“ verbunden.

Technische Umsetzung der Systemkommunikation

Zur technischen Umsetzung der Kommunikation zwischen Sensorik, Backend und Web-Dashboard wurde das leichtgewichtige Nachrichtenprotokoll MQTT vorgesehen.

MQTT eignet sich besonders für IoT-Anwendungen, da es nur geringe Netzwerkbandbreite benötigt, eine asynchrone Kommunikation erlaubt und eine lose Kopplung der beteiligten Systemkomponenten unterstützt.

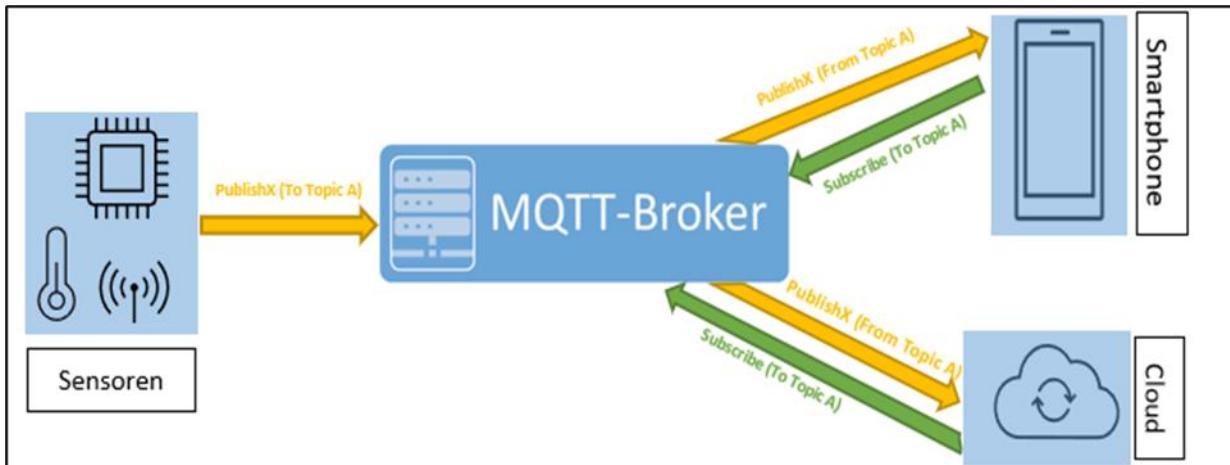


Abbildung 2: Schematische Darstellung der Kommunikation über MQTT-Broker.
(Quelle: eigene Darstellung)

Das MQTT-Protokoll verwendet das Publish/Subscribe Modell wie in Abbildung 2 zu sehen. Alle Daten werden zunächst von verschiedenen Geräten, d.h. Client oder Sensoren an einen sogenannten Broker in einem Topic veröffentlicht. Für jeden Client oder Sensor kann ein anderes Topic ausgewählt werden. Da sich alle veröffentlichten Daten bei einem Broker befinden, ist es nun möglich, diese Daten bei einem anderen Gerät (Client) zu veröffentlichen, wofür das Gerät das Topic, in dem die Daten veröffentlicht werden, abonnieren muss. So-lange das Gerät ein Thema abonniert hat, wird der Broker die neuen Daten bei jeder Aktualisierung an das Gerät senden. Ein Client kann sowohl Publisher als auch Subscriber werden.

Sequenzdiagramm der Bewässerungsvorgang

Das Sequenzdiagramm beschreibt den zeitlichen Ablauf des Bewässerungsvorgangs im System „Smart Bewässerungssystem“ und zeigt die Interaktion zwischen den beteiligten Akteuren und Systemkomponenten. Beteiligt sind der Akteur User, das Web-Dashboard als Benutzerschnittstelle, die Arduino IoT Cloud als zentrale Koordinationsinstanz sowie ein SensorNode zur Erfassung der Bodenfeuchte und ein PumpNode zur Steuerung der Bewässerung.

Zu Beginn übermittelt der SensorNode zyklisch den gemessenen Bodenfeuchtewert an die Cloud. Die Cloud verarbeitet diese Messung und aktualisiert anschließend die Anzeige im Web Dashboard.

Im Fall der automatischen Bewässerung prüft die Cloud, ob die gemessene Bodenfeuchte unter dem konfigurierten Schwellwert liegt. Ist dies der Fall, wird der PumpNode mit der zuvor festgelegten Bewässerungsdauer angesteuert. Der PumpNode schaltet die Pumpe ein, wartet die definierte Dauer ab und schaltet sie anschließend wieder aus. Der aktuelle Status wird danach an die Cloud zurückgemeldet und im Web Dashboard angezeigt.

Ist die Bodenfeuchte ausreichend, wird keine Bewässerung ausgelöst.

Zusätzlich wird der Ablauf einer manuellen Bewässerung dargestellt. Der User startet den Vorgang über das Web Dashboard. Die Anforderung wird an die Cloud weitergeleitet, welche wiederum den PumpNode mit der konfigurierten Bewässerungsdauer aktiviert. Nach Abschluss des Bewässerungsvorgangs meldet der PumpNode den Status an die Cloud zurück, die diesen im Dashboard darstellt.

Das Diagramm verdeutlicht die zeitliche Reihenfolge der Nachrichten und macht sichtbar, dass die Cloud als zentrale Steuerungs- und Entscheidungsinstanz fungiert.

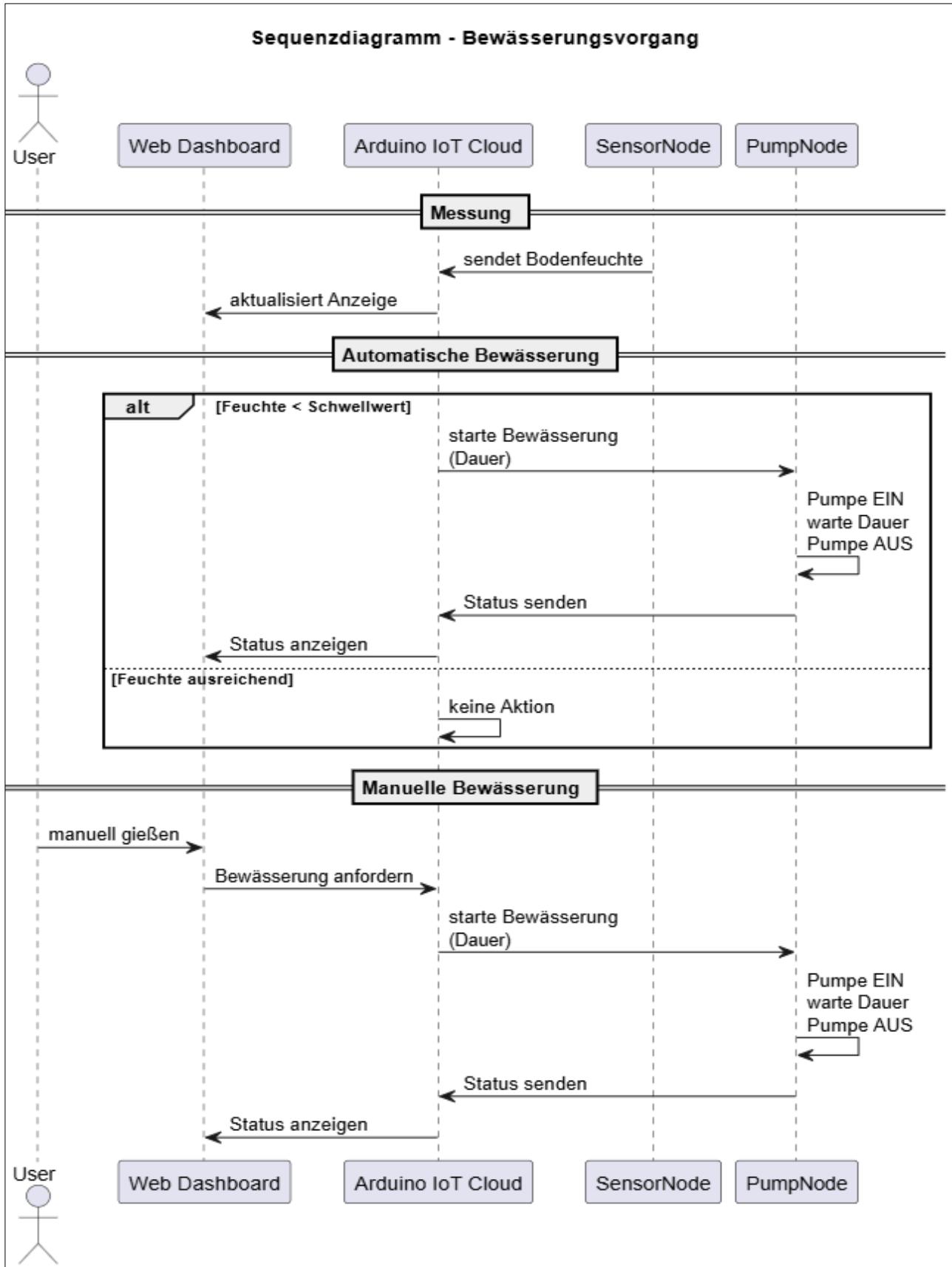


Abbildung 3: Sequenzdiagramm der Bewässerungsvorgang im Smart Bewässerungssystem.
(Quelle: eigene Darstellung)

Auf dieser Basis wurde die Systemarchitektur mit 2 „Things“ entworfen.

Komponenten Diagramm

Das Komponentendiagramm stellt die logische Architektur des Smart Bewässerungssystems dar. Das System besteht aus den Hardwarekomponenten SensorNode und PumpNode, die jeweils auf einem Arduino Nano 33 IoT betrieben werden, sowie aus der Software- und Integrationskomponente Arduino IoT Cloud und dem Web Dashboard als Benutzerschnittstelle.

Der SensorNode erfasst die aktuelle Bodenfeuchte und überträgt die Messwerte an die Arduino IoT Cloud. Die Cloud übernimmt die zentrale Verarbeitung der Daten und stellt diese dem Web Dashboard zur Anzeige zur Verfügung. Über das Web Dashboard kann der Benutzer Konfigurationsdaten eingeben sowie eine manuelle Bewässerung auslösen. Diese Informationen werden an die Arduino IoT Cloud übermittelt, die daraufhin entsprechende Bewässerungsbefehle an den PumpNode sendet. Der PumpNode setzt die Steuerbefehle um und steuert die Pumpe.

Das Diagramm verdeutlicht, dass die Arduino IoT Cloud als zentrale Kommunikations- und Integrationskomponente fungiert, über die sowohl die Sensordaten als auch die Steuerbefehle zwischen den Hardwarekomponenten und der Benutzerschnittstelle ausgetauscht werden.

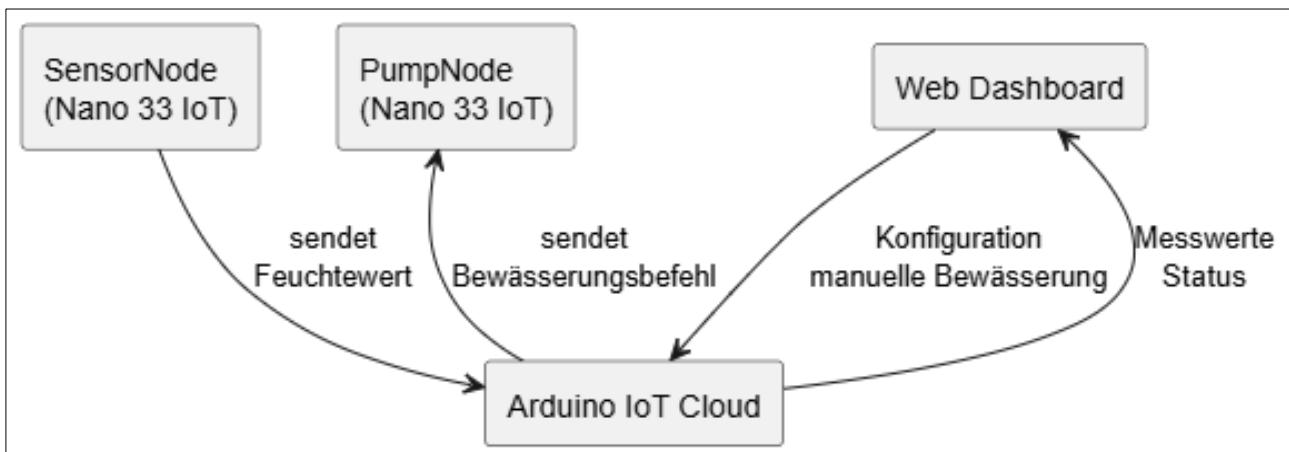


Abbildung 4: Komponenten Diagramm.
(Quelle: eigene Darstellung)

Things

Things sind physische Geräte oder Objekte (z. B. Sensoren, Aktoren, Maschinen), die mit Software, Netzwerkverbindung und oft Cloud-Anbindung ausgestattet sind, sodass sie Daten erfassen, senden und empfangen können.

Die Erstellung der Things bezieht sich auf den Prozess, bei dem verschiedene Geräte (auch als „Things“ bezeichnet) miteinander verbunden werden, um Daten auszutauschen und über eine Cloud-Plattform zu kommunizieren. Dies ermöglicht die Fernüberwachung, Steuerung und Verwaltung der Things über das Internet.

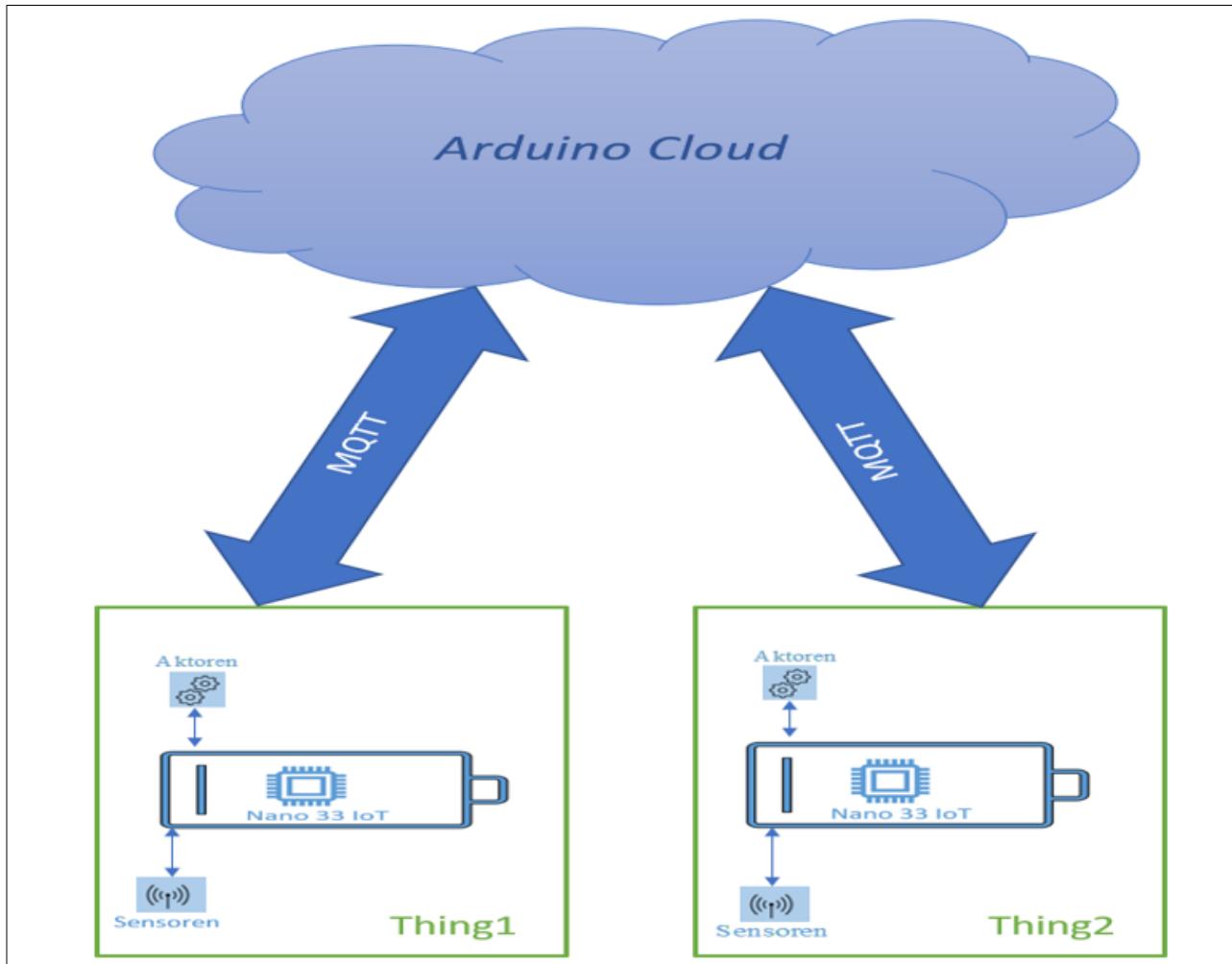


Abbildung 5: System-Architektur der Cloud-integrierten Smartbewässerungseinrichtung mit 2 Things.
(Quelle: eigene Darstellung)

Das erste Thing repräsentiert die Sensoreinheit zur Erfassung der Bodenfeuchtigkeit. Dieses Thing besteht aus einem Mikrocontroller mit angeschlossenem Bodenfeuchtesensor und ist dafür verantwortlich, Messwerte zyklisch zu erfassen und an den MQTT-Broker zu übertragen. Das zweite Thing repräsentiert die Bewässerungseinheit. Diese Einheit steuert die Wasserpumpe und empfängt über MQTT entsprechende Steuerbefehle zur Aktivierung oder Deaktivierung der Bewässerung. Die beiden Things kommunizieren nicht direkt miteinander, sondern ausschließlich über den MQTT-Broker.

Das Backend übernimmt in der Architektur die Auswertung der Messdaten und prüft kontinuierlich, ob der konfigurierte Schwellwert für die Bodenfeuchtigkeit unterschritten wurde. Wird dieser Zustand erkannt, veröffentlicht das Backend automatisch einen Steuerbefehl an das entsprechende Topic der Bewässerungseinheit.

Zusätzlich kann der Benutzer über das Web-Dashboard manuell einen Bewässerungsvorgang auslösen. In diesem Fall wird ebenfalls eine MQTT-Nachricht an das Steuer-Topic der Bewässerungseinheit gesendet.

Durch diese Aufteilung entstehen klar abgegrenzte Verantwortlichkeiten zwischen Sensorik, Geschäftslogik, Benutzeroberfläche und Aktorik.

Zustandsdiagramm Sensor Thing

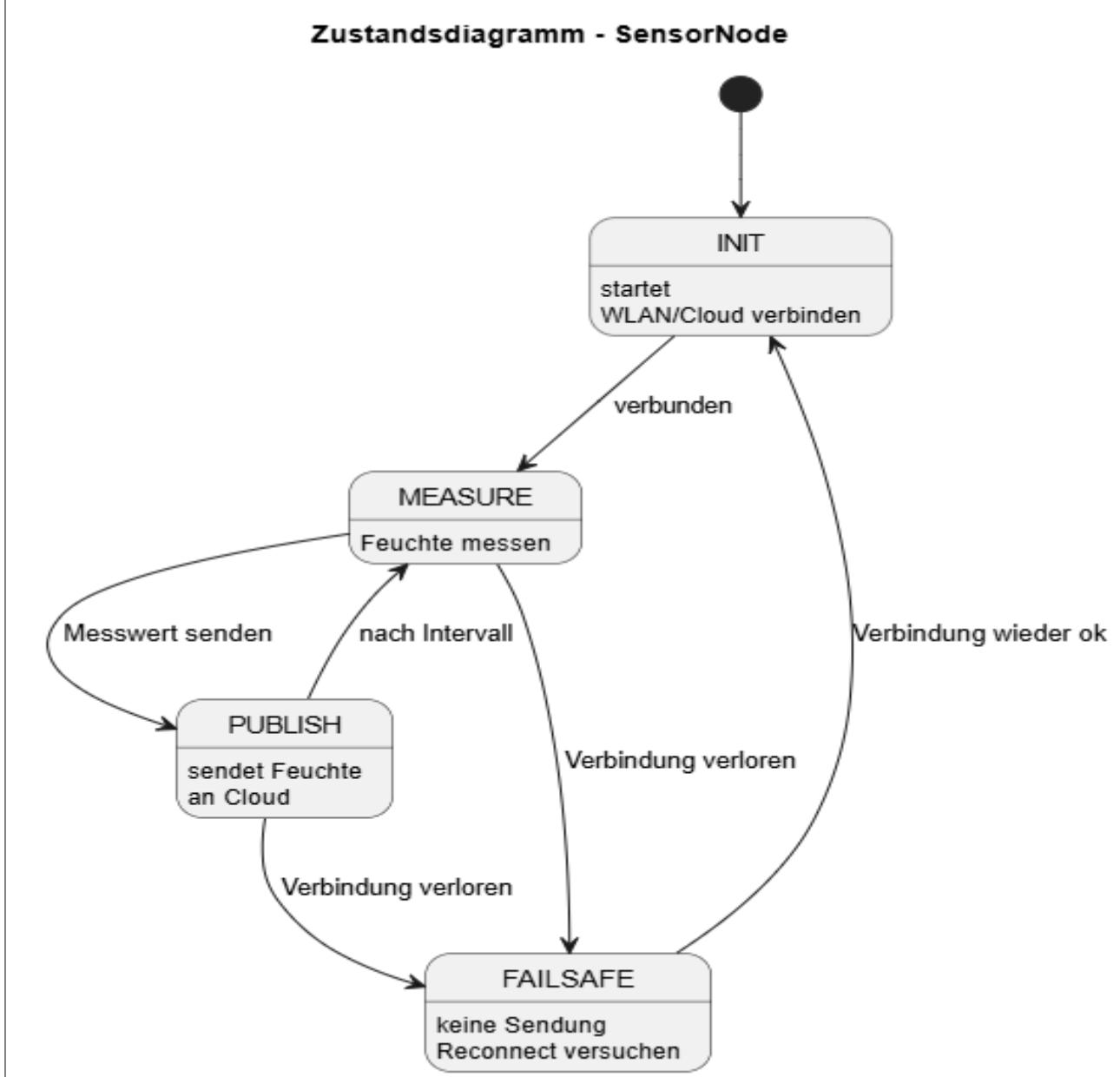


Abbildung 6: Zustandsdiagramm Sensor Thing.
(Quelle: eigene Darstellung)

Das Zustandsdiagramm beschreibt das interne Laufzeitverhalten des SensorNode innerhalb des Smart Bewässerungssystems.

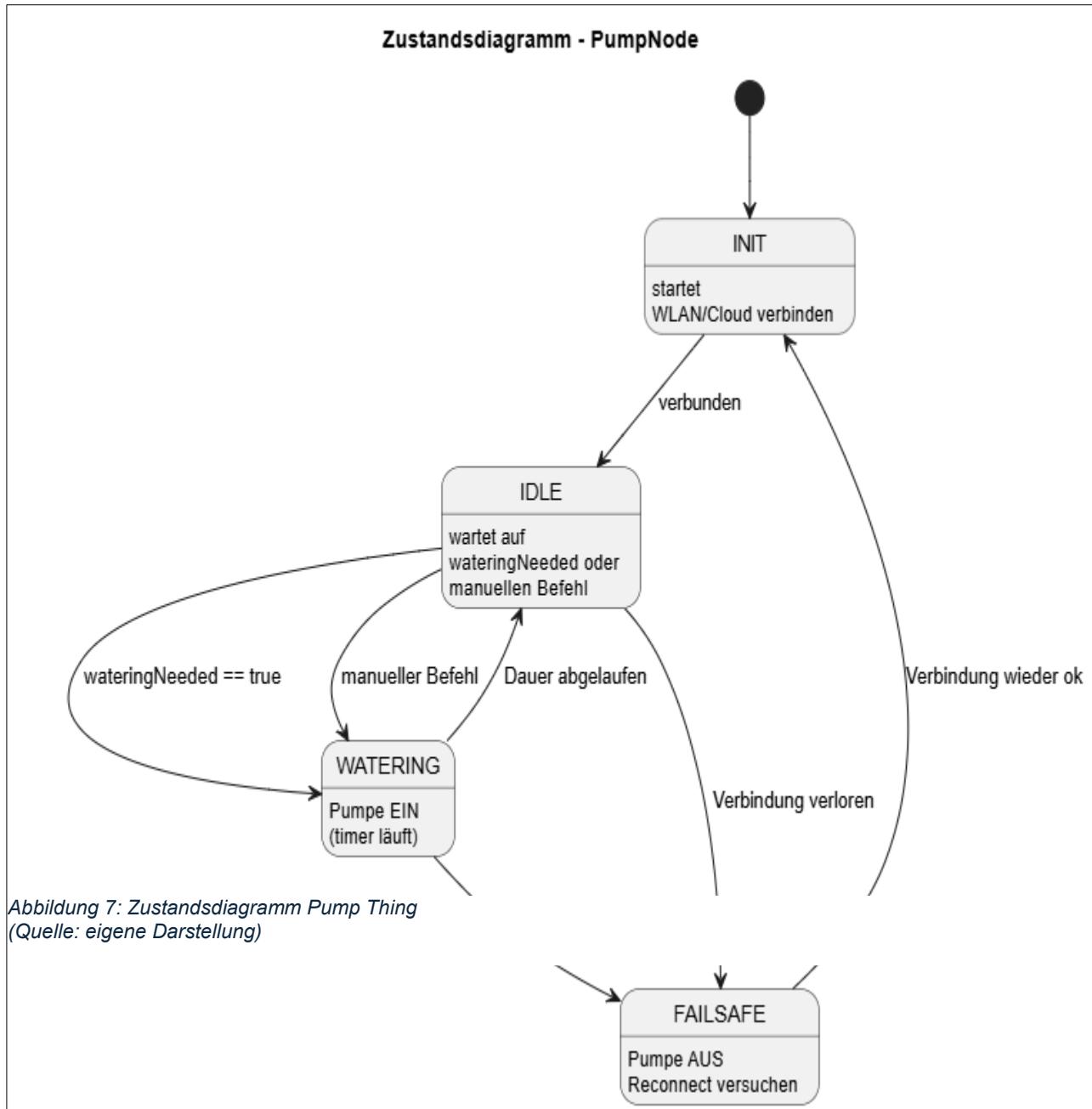
Nach dem Start befindet sich der SensorNode zunächst im Zustand **INIT**, in dem die Verbindung zum WLAN und zur Cloud aufgebaut wird. Sobald die Verbindung erfolgreich hergestellt wurde, wechselt das System in den Zustand **MEASURE** und führt eine Messung der Bodenfeuchte durch. Anschließend wird in den Zustand **PUBLISH** gewechselt, in dem der gemessene Feuchtewert an die Cloud übertragen wird. Nach erfolgreichem Senden kehrt der SensorNode zyklisch in den

Messzustand zurück, um die nächste Messung nach dem definierten Intervall durchzuführen. Tritt während der Messung oder der Übertragung ein Verbindungsverlust auf, wechselt der SensorNode in den Zustand FAILSAFE. In diesem Zustand werden keine Messwerte gesendet und es wird versucht, die Verbindung zur Cloud erneut herzustellen. Nach erfolgreichem Wiederaufbau der Verbindung erfolgt der Rücksprung in den Initialisierungszustand. Das Diagramm verdeutlicht, dass der SensorNode kontinuierlich zwischen Messung und Veröffentlichung der Daten arbeitet und über einen eigenen Fehlerzustand verfügt, um Verbindungsprobleme kontrolliert zu behandeln.

Zustandsdiagramm Pump Thing

Nach dem Systemstart befindet sich der PumpNode im Zustand INIT, in dem die Verbindung zum WLAN und zur Cloud aufgebaut wird. Nach erfolgreicher Verbindung wechselt das System in den Zustand IDLE, in dem der PumpNode auf einen Bewässerungsbefehl wartet.

Ein Übergang in den Zustand WATERING erfolgt entweder durch eine automatisch ausgelöste Bewässerung oder durch einen manuellen Befehl aus der Benutzeroberfläche. Im Zustand



WATERING wird die Pumpe eingeschaltet und ein interner Timer überwacht die konfigurierte Bewässerungsdauer. Nach Ablauf der Dauer wechselt der PumpNode wieder in den Wartezustand IDLE. Kommt es im Wartezustand oder während der laufenden Bewässerung zu einem Verbindungsverlust, wird in den Zustand FAILSAFE gewechselt. In diesem Zustand wird die Pumpe abgeschaltet und es wird versucht, die Verbindung zur Cloud erneut herzustellen. Nach erfolgreicher Wiederverbindung erfolgt der Neustart über den Initialisierungszustand.

Das Diagramm zeigt, dass der PumpNode sowohl automatische als auch manuelle Bewässerungsanforderungen verarbeitet und sicherstellt, dass bei Kommunikationsfehlern keine unkontrollierte Bewässerung stattfindet.

Arduino Cloud

Die Arduino Cloud bietet eine umfassende Lösung für IoT-Anwendungen, indem sie Entwicklern eine leistungsstarke Plattform zur Verfügung stellt. Ihr Zweck besteht darin, die Verwaltung von IoT-Geräten zu erleichtern. Die Arduino Cloud ermöglicht das einfache Hinzufügen, Überwachen und Verwalten von Geräten sowie das Aktualisieren der Firmware. Sie bietet auch nahtlose Datenübertragung, unterstützt MQTT-Kommunikationsprotokoll und ermöglicht die Analyse und Visualisierung gesammelter Daten. Die Plattform kann problemlos in bestehende Systeme integriert werden und bietet Skalierbarkeit und Flexibilität für IoT-Projekte. Insgesamt bietet die Arduino Cloud eine umfassende Lösung für die Entwicklung von IoT-Anwendungen, die den Entwicklungsprozess vereinfacht und die Leistungsfähigkeit maximiert.

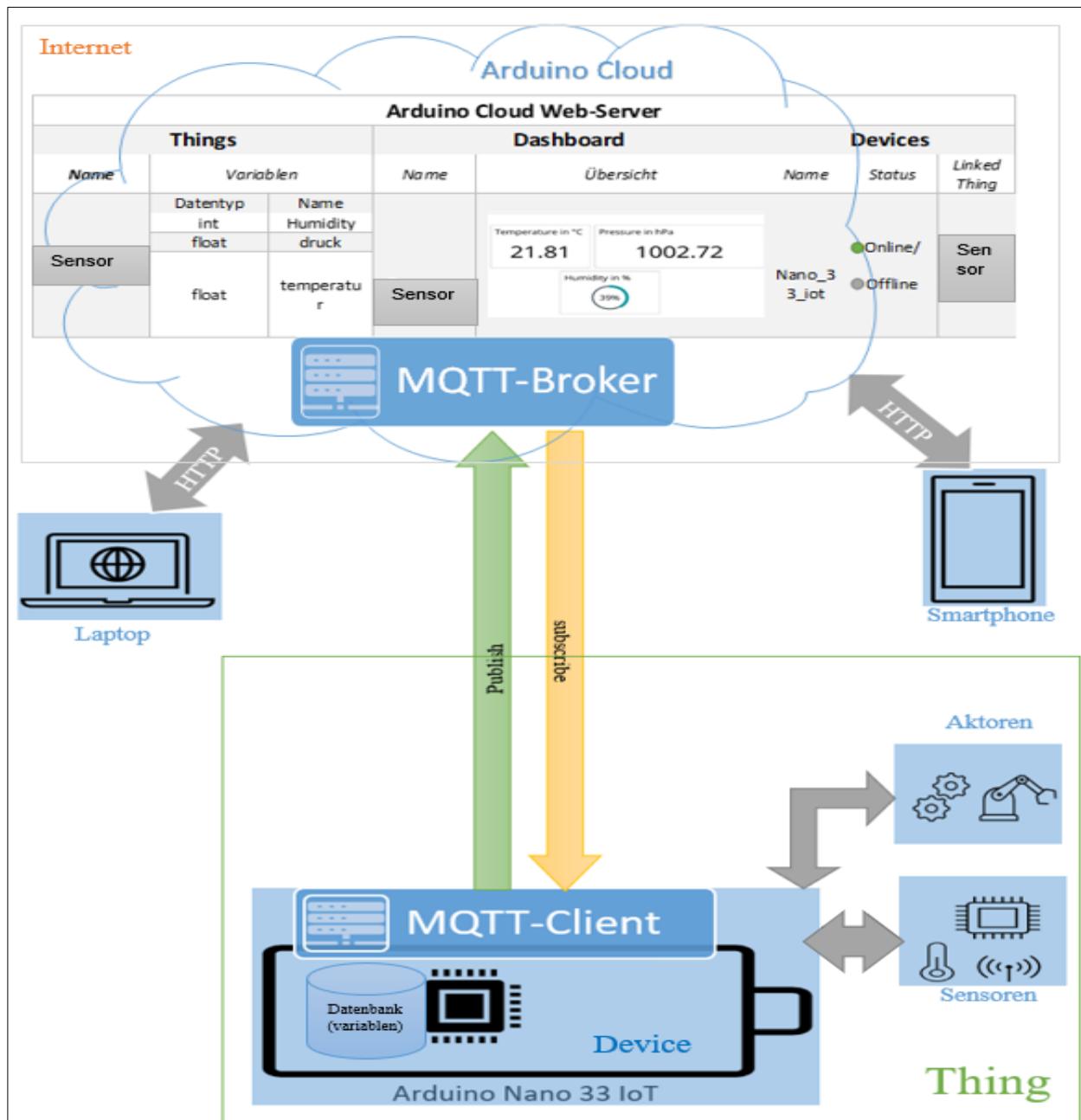


Abbildung 8: System-Architektur der Arduino IoT Cloud.
(Quelle: eigene Darstellung)

GitHub-Repository

<https://github.com/Niraj-Bista/smartBewaesserung>

Abbildungsverzeichnis

Abbildung 1: Use-Case-Diagramm der Anwendung. (Quelle: eigene Darstellung)	2
Abbildung 2: Schematische Darstellung der Kommunikation über MQTT-Broker. (Quelle: eigene Darstellung)	3
Abbildung 3: Sequenzdiagramm der Bewässerungsvorgang im Smart Bewässerungssystem. (Quelle: eigene Darstellung)	5
Abbildung 4: Komponenten Diagramm. (Quelle: eigene Darstellung).....	6
Abbildung 5: System-Architektur der Cloud-integrierten Smartbewässerungseinrichtung mit 2 Things. (Quelle: eigene Darstellung).....	7
Abbildung 6: Zustandsdiagramm Sensor Thing. (Quelle: eigene Darstellung).....	8
Abbildung 7: Zustandsdiagramm Pump Thing (Quelle: eigene Darstellung).....	10
Abbildung 8: System-Architektur der Arduino IoT Cloud. (Quelle: eigene Darstellung).....	12