Internet of Things: Begriffe, Gefahren, Messwerte von Sensoren holen und Aktoren schalten

# Was ist das IOT

Das Internet der Dinge bezeichnet die Verknüpfung eindeutig identifizierbarer physischer Objekte (*things*) mit einer virtuellen Repräsentation in einer Internet-ähnlichen Struktur

„Intelligente“ Gegenstände sollen dabei den Menschen unterstützen. Sie stellen Informationen über ihren Zustand oder den Zustand ihrer Umgebung zur Verfügung oder reagieren darauf in einer Form, die den Menschen unterstützt. Beispiel Kühlschrank, der eine Benachrichtigung sendet, wenn zB die Milch aus ist oder sogar direkt nachbestellt.

Begriff hat viele Überschneidungen zB mit embedded Systems, Web2.0, ubicomp, Sensornetzwerken.

Problematisch ist dabei evtl der Datenschutz, da die „Dinge“ dabei fast immer an einem realen Netz hängen und damit mit dem Internet der Menschen verbunden sind.

Klassische Beispiele

* Alles was mit smart anfängt smart-TV, smart-Home, smart-meter usw
* IP-Überwachungskamera, mit der man von überall Haus und Hof überwachen kann
* Heizungssteuerung, die per App abgerufen/bedient werden kann
* Vernetzte Alarmanlagen
* homePod und co, alexa und co
* Ampelsystem einer Stadt, das zentral verwaltet wird
* Das überregionale Stromnetz, das so geschaltet wird, dass Überkapazitäten in andere Gebiete umgeleitet werden, die gerade mehr Leistungen benötigen (zB Unfall in Kaprun, wo eine Trafostation abbrannte weil das IP Netz flasch konfiguriert war)
* Ungewollte „Things“ zB Selbstfahrende Autos, die zB über das Autoradio im Internet hängen

# Gefahren

Standardnetzwerk ist dabei das ethernet (kabelgebunden oder mobilfunk oder WLAN)  
im Endeffekt meist TCP/IP und darüber liegende Schichten mit eigenen Protokollen  
„things“ im Haushalt senden ihre Informationen üblicherweise über den Router ins Internet, dh dieser muss richtig konfiguriert sein und die Firewall muss entsprechen durchlässig für die Ports sein, Ein normaler Käufer ist aber kein IT Experte, der seine Firewall konfiguriert, damit der Kühlschrank funktioniert. Deshalb wird oft einfach http als Protokoll (dh meist Port 80 oder 443, die sind sowieso fürs Browsen offen) verwendet und darin das spezifische Protokoll eingepackt.   
Zusätzlich haben viele Router dynamische Adressen, so dass eine dynamische Identifizierung notwendig wird (zB DynDns oder noIP). Die Kommunikation erfolgt daher fast immer über die cloud des Verkäufers, der die Vermittlung zwischen „Thing“, dem Internet und dem Benutzer zur Verfügung stellt. (also zB „thing“-„Hersteller-cloud“-„Nutzer App“). Schaut für den Benutzer so aus, also würde die App direkt mit dem Gerät kommunizieren, es läuft aber über den Hersteller.  
Bei WLAN geht’s direkt, problematisch ist dabei meist die erste Einrichtung des Gerätes, da in vielen Fällen das WLAN Kennwort unverschlüsselt zum Gerät übertragen wird (ein Staubsauger hat keine Tastatur in die man das WLAN Passwort am Anfang zum Konfigurieren eingeben kann)  
Oft behält sich der Verkäufer einen Verbindungskanal zB für Firmware-Updates offen.  
Standardkennwort wird vom Kunden nicht geändert (war bei der Hofer IP Cam so, dass jeder die Bilder jeder Kamera anschauen konnte)  
Geräte mit Linux als OS, ssh und Standardbenutzer und Kennwort

# Bekannte Geräte mit groben Schutzmängeln:

* Bekanntestes Datenschutzproblem war die Aldi-IP Kamera bei der über http folgendes von außen möglich war:  
  Videos mitschauen, Mikrofon einschalten und mithören, Konfiguration auslesen mit: WLAN Passwort des Heimnetzes sowie email und ftp Kennwörter des Benutzers, wenn Benachrichtigung aktiviert war.
* Und nochmal Aldi: WLAN Steckdose mit ausgereiften, nicht dokumentierten Webinterface, dass aus dem Internet erreichbar war, mit Standardpasswort, **Mikrofon zum Einschalten**, Auslesemöglichkeit von Kennwörtern.
* Staubsauger der das WLAN Kennwort im Klartext anfordert und über WLAN fernsteuerbar ist und mittels Lasertaster den Raum abscannt, einer hatte sogar ein webcam drauf, da konnte man dann das Zimmer über den Staubsauger beobachten
* Gewerbe-Tiefkühlschränke, die im Internet hängen, auf shodan gelistet sind und per Fernwartung abgeschaltet werden können (davon hat es allein in Deutschland ein paar Hundert gegeben, Herstellerfirma hat aber nicht reagiert)
* Smart TV Boxen mit Linux, ssh Zugang, fixen admin-admin account und Packetmanager, wo man mit apt-get Pakete nachinstallieren konnte. zB einen Voice Recorder.

Problematisch ist meist die Erstkonfiguration über WLAN, da ist mithören mit Wireshark ganz leicht möglich

Später mittels man in the middle Attacken (SSLsplit, mitmproxy, certificate spoofing) auch bei https Verbindungen möglich, weil die Geräte keine Schutzmechanismen haben (Browser meldet ja, dass zB ein selbst signiertes Zertifikat verwendet wird, Staubsauger wohl nicht)

Geräte haben oft mehr Sensoren drauf, als angegeben sind, weil Massenproduktion mit komplexer Ausstattung billiger ist als eigene Platinen für jede Version eines Gerätes zu machen. HW-Funktionen werden dann per Firmware aus oder zugeschaltet.

# Schutz:

Segmentierung des Netzes nach Geräteklassen (dafür gibt es von Cisco eine Einteilung von 0=ungeschützt bis 100=Hochsicherheit):  
zB folgende getrennte Segmente:

* HochsicherheitsLAN (PCs mit vertraulichen Daten)
* Gäste LAN bzw Gäste-WLAN (Geräte die nur eine Internetverbindung brauchen)
* Medien-Zone (Smart-TV, Chromecast und co, HomePod, AmazonEcho, Alexa..)
* IOT-Zone (things brauchen meist nur Internet und keinen Zugriff zum lokalen Netz)
* DMZ (demilitarisierte Zone: web-Server, mail-Server)
* Geräte, die keiner Zone zuordenbar sind, kommen in die höchste und werden entsprechend durchgeschleust
* Eigene Zonen für Geräte, die zB per Sprachsteuerung (Internet) Videos aus der heimischen Medienzone abspielen sollen

Insgesamt ein riesiger Aufwand, evtl mit Neukonfiguration, wenn ein Gerät seine Rolle tauscht oder durch ein Firmwareupdate plötzlich mehr kann.  
Oft reichen aber zumindest 2 Zonen oder VLANs

Einrichtung:

Normale Router können meisten nur 2 Zonen (Normales und Gäste WLAN)

### Möglichkeit 1 – Router kaskadieren:

Man kann 2 oder mehrere Router kaskadieren (und dann die Firewallregeln und Portweiterleitungen einrichten. Dann hat man evtl schon 4 Zonen: 2\*normales Netz und 2\*Gästenetz. zB Fritzboxen um 80€. Die sicherste Zone liegt am untersten/letzten Router   
Nachteil doppelter Stromverbrauch, Konfiguration aufwändiger, man muss etwas von portforwarding oder outing verstehen

1. Router1:  
   GästeWLAN am Gäste-LAN   
   IOT am normalen LAN
2. Router2: (hängt mit seinem WAN Port an einem LAN Port von Router1):  
   Medien am Gäste-LAN  
   vertrauliche Daten am normalen LAN
3. Usw den nächsten Router wieder an einem LAN Port anhängen

### Möglichkeit 2 – MultiLAN Router:

Bezeichnung Mutli-LAN Router oder Layer-3 Switches. Diese haben von Haus aus mehrere Ports, kosten ab 150€ und können VLANs betreiben. Gute, zentrale Konfigurationsmöglichkeiten dadurch relativ einfach einzurichten, man muss aber evtl Routen mit Regeln von einem VLAN ins andere legen  
gibt’s zB von TP-Link mit 5 Ports, load balancing, ipv6,… um 200€

### Möglichkeit 3 – HW Firewall:

Eigene HW Firewall bauen, mit mehreren Netzwerkkarten und Netzwerksegmenten und zB pfSense (freie Firewall/Routing Software), Einarbeitung in die Firewall nötig, Stromverbrauch recht hoch.

Möglichkeit 4 – Raspberry Projekt:

Der Raspberry 3B hat LAN und WLAN, da könnte man auch Routen bauen und mit IPTables schützen

Insgesamt: ohne Netzwerk-Know How geht gar nichts.

# IoT - Interaktion mit Sensoren und Aktoren

Ein sehr gebräuchlicher Chip ist von espressif, heißt ESP8266 oder ESP32 (kostet wenige Euro)

Sehr kleiner Mikrocontroller mit WLAN, digitalen Ein- und Ausgängen, Permanentspeicher (NVS – non volatile storage). Den gibt’s mit draufgesteckten Relais und Spannungsadapter, so dass man ihn ans Stromnetz anschließen kann, Controller bekommt seine 5V davon und kann übers Relais 240 V AC Geräte schalten. Kommt aus China, ist in vielen fernbedienbaren „smart“-Geräten (zB fernbedienbare Lampen, Jalousien-Steuerungen, Garagentore, usw) drinnen.

Ist über die Arduino IDE oder mit Visual Studio Code und platform io einfach programmierbar.

Es gibt für sehr viele Sensoren Bibliotheken und Beispielprogramme (in C, C++, python, Java)

Bibliothek mit einem Webinterface für die Erstinstallation zB um das lokale WLAN Passwort einzugeben oder zurückzusetzen, dieses wird dann im NVS gespeichert und der Controller hängt im Netz

## Komponenten:

1. Sensordaten einlesen: meist mit polling, fertige Bibliothek verwenden, Messwerte kommen über die GPIO Ports per Pulsweitenmodulation oder über den I2C Bus. Sourcecode besteht meist aus wenigen Zeilen: SensorBibliothek einbinden, Sensor-Objekt anlegen und getdata Methode verwenden
2. Verbinden zu einem Broker und Messwerte senden (publishen):  
   entweder liegt der Broker in der Cloud (vom Hersteller) oder der Controller arbeitet selber als Broker und hat den entsprechenden Dienst laufen, verwendet werden das mqtt-Protokoll oder websockets.  
   - client-Instanz anlegen (meist eine Bibliothek von paho, da gibt man im Konstruktor den Namen des Brokers an)  
   - connect Methode: client.connect(Broker Adresse, Port); mqtt port ist meist 1883  
   - publish Methode des clients verwenden und Messwerte unter einem topic publishen:   
   client.publish(topic, message, QoS)  
   QoS: drei Stufen, wie die Werte für den Empfänger bereitgestellt werden:  
    0 (fire and forget), 1 at least once, 2 one time
3. Abholen der Messwert und Darstellung in der GUI einer APP  
   Kann per websockets oder einfach über das MQTT Protokoll erfolgen:  
   Verbindung zum Broker (wie in Punkt 2)  
   client.subscribe(topic, asynch callback, die aufgerufen wird, falls was zu dem Topic kommt)

Natürlich könnten auch eigene REST Services verwendet werden oder alles über tcp oder datagramm sockets laufen. Ist aber aufwändiger, Brokerkonzept ist sehr verbreitet, bietet auch security Möglichkeiten per accounts und per SSL

## Daten von GUI zu einem Aktor

zB Einschalten einer Lampe: Ablauf genau wie oben, aber mit umgekehrten Rollen, jetzt ist der ESP der subscriber und die GUI der Publisher.

GUI hat Button, im Eventhandler des Buttons wird ein true/false über ein passendes topic zum Broker gepublished,

ESP ist subcriber und wartet auf das topic, wenn was daherkommt schaltet er übers Relais den Aktor (zB die Lampe) ein/aus.