

**Auf der  
Heft-DVD**Über 7 GByte  
für EntwicklerSponsored Software:  
Intel Parallel Studio  
XE 2016Entwicklungswerkzeuge:  
Eclipse, Arduino, AtomTools, Frameworks, Sprachen  
und mehr: openHAB, AllJoyn,  
Kura, Elixir, Lua, Apache-Projekte  
aus den Bereichen Big und  
Smart Data sowie JavaScript-  
Tools für das Internet  
der Dinge

# Entwickeln für das Internet der Dinge

**Grundlagen:****Elektronik, Sensorik und Software****Open Source im Internet der Dinge****Know-how:****Nutzerschnittstellen  
konzeptionieren****IoT-Architektur  
richtig umsetzen****Sensordaten sammeln  
und verarbeiten****Kommunikation:****Protokolle harmonisieren****Die wichtigsten Messaging-  
und Sensorprotokolle im Überblick**

## IoT-Projekte sicher planen

**!HOSTSERVER**

# Managed Hosting

**zertifiziert nach ISO 27001 und ISO 9001**

- ✓ Applikationsserver
- ✓ Private Cluster
- ✓ Private Cloud

**Garantierter Betriebsstandort  
Frankfurt am Main (DE)**



## Managed Hosting für IT-Projekte und Webanwendungen

Individuelles Managed Hosting am Standort Deutschland vom Server bis zum IT-Projekt mit Fokus auf Verfügbarkeit, Sicherheit und Datenschutz.

- Managed Firewall
- Managed Loadbalancer
- DDoS Protection Service
- Varnish Edge-Caches

Wir bieten den zertifizierten Betrieb von Servern, Clustern, Private Cloud, Projekten sowie Anwendungen, mit individuellem SLA und ITIL-Service Management.

Nutzen Sie 15 Jahre Erfahrung in IT-Betrieb, Compliance Management und Application-Hosting.

**Sprechen Sie uns an: 030 – 47 37 55 50**  
[www.hostserver.de/hosting](http://www.hostserver.de/hosting)

# Niemand lebt für sich allein

In der Welt der IT folgt Hype auf Hype. Nach Big Data und der Cloud hat die Industrie nun das sogenannte Internet der Dinge für sich entdeckt. Heimautomatisierungsszenarien, Maker-Projekte und eine Reihe zweifelhafter Produkte, scheinbar ohne Sinn und Verstand zusammengeflanscht und mit Internetanschluss versehen, sollen eine neue Zeitrechnung einläuten. Und auch die Berichterstattung lädt nicht unbedingt dazu ein, sich als Nicht-Insider auf professioneller Ebene näher mit der neuen Domäne zu befassen: Zu häufig liegt der Fokus auf Sicherheitsproblemen und Skurrilitäten.

Häufig werden die Schwachstellen im Zuständigkeitsbereich der IoT-Entwickler gesucht – etwa im Fall des Jeeps, dessen Fahrer nach einem System-Hack die Kontrolle über das Gefährt verlor. Folglich scheint es so, als erhielten Softwareentwickler und -architekten nicht nur ein neues Einsatzgebiet, sondern auch noch einen Haufen Verantwortung. Ab sofort liegt es nicht nur in ihrer Hand, dass die Software den formulierten Ansprüchen gerecht wird, sie müssen unter anderem auch dafür sorgen, dass die Kommunikation mit der Hardware und anderen Infrastrukturelementen sicher und problemlos über Systemgrenzen hinweg funktioniert. Zudem sollen sich Daten im Zweifel in Echtzeit auswerten lassen. Und in allen Projekten sind Mechanismen vorzusehen, die gewährleisten, dass einmal ausgelieferte Geräte auch über die Dauer ihres Betriebs vor Angriffen geschützt sind, der Anbieter also jederzeit Aktualisierungen oder Wartungen durchführen kann – ganz ohne die beliebten Hintertüren.

Neulinge können all diese Anforderungen kaum erfüllen. Zwar gibt es das Internet der Dinge in der einen oder anderen Form und unter anderen Namen schon eine ganze Weile, trotzdem ist die Disziplin als Ganzes noch zu jung, um bereits mit allgemeingültigen Handlungsempfehlungen aufwarten zu können. Selbst aus der Telekommunikationstechnik oder der Embedded-Programmierung bekannte Eigenheiten scheinen

ganz neue Dimensionen einzunehmen. Spätestens aber, wenn der Vorgesetzte von den wirtschaftlichen Prognosen erfährt, die an mehreren Stellen im Heft erwähnt werden, geht kein Weg mehr am IoT vorbei.

Die zahlreichen Open-Source-Projekte des IoT sind Zeugnis des Stellenwerts, den Kommunikation abseits der technischen Ebene in ihm spielt. Gemeinsam lassen sich viele Probleme lösen und häufig besser, als das allein möglich gewesen wäre. So gut wie jeder kann etwas zum IoT beitragen, was wohl auch einen Teil seiner Faszination ausmacht.

Dieses Heft kann zwar niemandem die Einarbeitungsmühen abnehmen, die für ein erfolgreiches Mitmischen in der schönen neuen Welt nötig sind. Einen Wegweiser für die Navigation in ihren Tiefen kann es aber auf alle Fälle bieten und im Zweifel zeigen, dass man mit vielen Überlegungen, die sich in der ersten Zeit aufdrängen, nicht allein steht.

Hypes sind (nicht nur, aber auch) ein Zeichen dafür, was Menschen bewegt. Ein offenes Ohr und Kommunikationsfähigkeit können also enorm helfen, wenn man die Zukunft mitgestalten möchte. Das Web lässt grüßen.

JULIA SCHMIDT



# Grundlagen

Für viele Entwickler bedeutet der Einstieg in das Internet der Dinge, sich mit Themen auseinanderzusetzen, mit denen sie vorher keinen bis wenig Kontakt hatten. Allerdings gibt es Analogien zum gewohnten Arbeitsumfeld, sodass niemand gezwungen sein sollte, noch einmal bei Null zu beginnen.

ab Seite 7



# Know-how

IoT-Projekte sind ihrer Natur nach recht komplex und bedürfen einer sorgfältigen Planung, was ihre Architektur, die Energieversorgung und Aspekte der Vernetzung angeht. Stärker denn je sollten Entwickler zudem die Bedürfnisse der Nutzer von Anfang an bedenken, da nur so erfolgreiche Produkte entstehen.

ab Seite 35

## Grundlagen

<b>Stand der Dinge</b>	
Protokollgewirr, Privatsphäre und Open Source im IoT	8
<b>Vorwissen</b>	
Womit sich Backend-Entwickler nun beschäftigen müssen	16
<b>Technik</b>	
Als Softwareentwickler Elektronik lernen	20
<b>Sensoren</b>	
Datenerfassung in Industrieanwendungen	24
<b>Kreativität</b>	
Ideen für das Internet der Dinge entwickeln	28
<b>Ethik</b>	
Entwicklermanifeste für das Internet der Dinge	32

## Know-how

<b>Systementwurf</b>	
System- und Softwarearchitekturen für das Internet der Dinge	36
<b>Modellierung</b>	
Realisieren komplexer Projekte	46

<b>Usability</b>	
Anwenderschnittstellen von Beginn an berücksichtigen	50
<b>Datenverarbeitung</b>	
Prototypen mit Big-Data-Anbindung entwickeln	54
<b>Versorgung</b>	
Energiegewinnung für autarke Systeme	58
<b>Security</b>	
IT-Sicherheit und Datenschutz in IoT-Projekten umsetzen	62

## Protokolle

<b>Internetprotokolle</b>	
Internet- und Anwendungsprotokolle als IoT-Grundlage	68
<b>MQTT</b>	
Message Queue Telemetry Transport fürs Internet der Dinge und Dienste	74
<b>Kommunikationssicherung</b>	
MQTT sicher verwenden	78
<b>Standards</b>	
OPC UA 1.03 und MTConnect 1.3: Neue Kommunikationsstandards für das IoT	82
<b>Konvergenz</b>	
Sensorprotokolle gemeinsam einsetzen	90



# Protokolle

Ob es nun um den Sensor geht, der Messwerte an eine zentrale Analyseeinheit verschickt, oder ein Stellglied, das auf Anweisungen wartet: Der Austausch von Daten ist ein zentraler Punkt des IoT. Wer also für dieses Geschäftsfeld entwickeln möchte, kommt nicht daran vorbei, sich mit einer Vielzahl von Protokollen zu beschäftigen.

ab Seite 67



# Sprachen

Open Source ist eine treibende Kraft im IoT und bietet eine ganze Reihe von Frameworks und Bibliotheken, die Entwicklern den Einstieg in die neue Welt erleichtern. Nicht nur C- und C++-Programmierer werden versorgt, auch Java- und JavaScript-Nutzer haben ihren Platz. Und wer einmal etwas Neues ausprobieren will, kann Lua und Elixir einen Blick schenken.

ab Seite 99

## Sprachen

### C und C++

C-Sprachen in Open-Source-Projekten für das IoT **100**

### Java

Open Source Java-APIs für die Entwicklung von IoT-Applikationen **108**

### JavaScript

Internet of Things für Einsteiger **114**

### Elixir

Mit Erlangs Erben in die Zukunft **122**

### Lua

Alternative Programmiersprache für das IoT **126**

## Praxis

### Infrastruktur

Digitale Transformation der Gebäudetechnik **132**

### Interoperabilität

Erfahrungen aus dem Aufbau moderner IoT-Architekturen **134**

### Objektkommunikation

Einführung in das Physical Web **138**

### Messaging

Tipps und Tricks zur Umsetzung eines Messaging-Systems in C# **144**

### Heimautomatisierung

Anbindung einer mit Codesys programmierten SPS an OpenHAB **148**

## Sonstiges

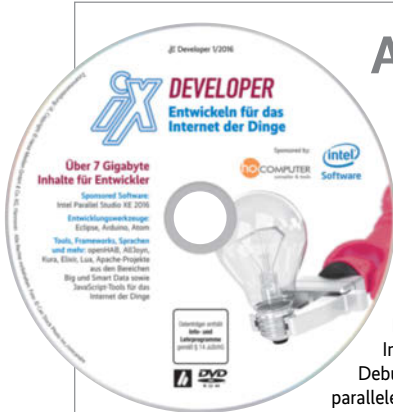
Editorial **3**

Impressum **6**

Inserentenverzeichnis **6**

 Alle Links: [www.ix.de/ix1517004](http://www.ix.de/ix1517004)

Artikel mit Verweisen ins Web enthalten am Ende einen Hinweis darauf, dass diese Webadressen auf dem Server der iX abrufbar sind. Dazu gibt man den iX-Link in der URL-Zeile des Browsers ein. Dann kann man auch die längsten Links bequem mit einem Klick ansteuern. Alternativ steht oben rechts auf der iX-Homepage ein Eingabefeld zur Verfügung.



## Auf der DVD im Heft

Eine Reihe wichtiger Projekte, zum Großteil Open Source, erleichtert Software-Entwicklern den Einstieg ins Internet der Dinge.

### Sponsored Software

#### Intel Parallel Studio XE 2016:

Intels Tool Suite zum Entwickeln, Debuggen und Verbessern von auf die parallele Abarbeitung ausgelegtem C++ und Fortrancode kann Programmieren dabei

helfen, schnellere Software zu erstellen und die Prozessorleistung bestmöglich zu nutzen.

### IDEs und Editoren

**Eclipse 4.5.1:** Die erste Mars-Service-Release der Entwicklungsumgebung für Java-Programmierer.

**Atom:** Atom ist ein quelloffener Editor aus dem JavaScript-Umfeld.

**Arduino:** Das Arduino-Projekt stellt seinen Nutzern eine eigene IDE zur Verfügung.

### Big und Smart Data

Daten sammeln und verarbeiten ist eine zentrale Aufgabe des IoT. Hilfestellungen können unter anderem die auf der DVD zu findenden Apache-Projekte aus dem Big-Data-Umfeld geben: Flume, Hadoop, Hive, Kafka, Spark, Zookeeper

### Sprachen

**Lua und Elixir** sind vielleicht bereits aus anderen Zusammenhängen bekannt, das IoT bietet jedoch ein interessantes neues Umfeld für beide Sprachen.

**JavaScript:** Die Lingua Franca des Web erlangt derzeit verstärkt als verbindendes Element des IoT Aufmerksamkeit. Webentwicklern steht mittlerweile eine Vielzahl an Projekten zur Verfügung, die bei der Umsetzung von Teilaufgaben helfen können. Auf der DVD finden sich daher Cylon.js, Johnny-Five, macchina.io, Mosca, MQTT.js, node-coap, Node.js, Node-RED, Socket.IO und Zetta.

### openHAB

Die quelloffene Plattform soll dabei helfen, Komponenten zur Gebäude-automatisierung hersteller- und protokollneutral zu verbinden.

### Physical Web

Zwar steht das auf Interaktion ausgelegte Smart-Object-Projekt noch ziemlich am Anfang seiner Entwicklung. Wen der Artikel allerdings neu gierig gemacht hat, kann auf der DVD einen Blick auf den Quellcode der Physical-Web-App werfen.

### Frameworks

**AllJoyn:** Das Linux Foundation Collaborative Project soll die Kommunikation zwischen Anwendungen und Geräten sowie das gegenseitige Entdecken erleichtern.

**Eclipse SmartHome:** Über System- und Protokollgrenzen hinweg soll das Abstraktions- und Übersetzungs-Framework Interaktion ermöglichen.

**Kura** ist Teil des Eclipse-IoT-Projekts, nimmt Java/OSGi zur Grundlage und dient als Framework für IoT-Gateways.

### Weitere Helfer

Betriebssystem für das IoT: Riot; Middleware-Plattform: Kaa; Messaging: Mosquitto und Paho

### Hinweis für Käufer der digitalen Versionen

- PDF- und iPad-Version: In der iX-App finden Sie einen Button zum Download des DVD-Images.
- PDF-E-Book: Folgen Sie im Browser der unter „Alle Links“ angegebenen URL.

Alle Links: [www.ix.de/ix1517006](http://www.ix.de/ix1517006)

MAGAZIN FÜR PROFESSIONELLE INFORMATIONSTECHNIK

### iX Developer 1/2016 – Internet der Dinge

Postfach 61 04 07, 30604 Hannover; Karl-Wiechert-Allee 10, 30625 Hannover

Redaktion: Telefon: 05 11/53 52-387, Fax: 05 11/53 52-361, E-Mail: [post@ix.de](mailto:post@ix.de)

Chefredakteur: Jürgen Seeger (js) -386

Konzeption und redaktionelle Leitung: Julia Schmidt (jul) -378, E-Mail: [jul@ix.de](mailto:jul@ix.de)

**Autoren dieser Ausgabe:** Philip Ackermann, Dr.-Ing. Oliver Alt, Louay Bassbous, Stephanie Bayer, Stefan Brandys, Frank Cloheim, David Fuhr, Sebastian Golasch, Lars Gregori, Kai Hackbarth, Tam Hanna, Oliver Heuser, Matthias Kassner, Konstantin Kersten, Wolfgang Klimt, Daniel Koch, Henryk Konsek, Dr. Axel Krauth, Moritz Minzlaff, Carsten Müllers, Susanne Nolte, Dominik Obermaier, Günter Obiltschnig, Lisa Öztürkoglu, André Riesberg, Golo Roden, Susanna Roden, Michael Stal, Markus Türber, Matthias Wagler, Ramon Wartala, Dr. Olaf Wilken, Maik Wojcieszak, Marko Wolf, Sascha Wolter

**Abbildungen © Can Stock Photo Inc.:** Arogant (S. 35, 46), Baloncici (S. 36), CarolinaSmith (S. 67, 74), cocooo (S. 99, 108), cookelma (S. 4, 16), CWMGary (S. 67, 82), dja65 (S. 5, 78), Erdosain (S. 122), GeorgeJmclittle (S. 131, 148), GNB\_designs (S. 131, 134), Jagree (S. 131, 144), jas0420 (S. 67, 68), lokes (S. 7, 20), maggee (S. 7, 24), magraphics (S. 8), maxxyustas (S. 35, 50), mtoome (S. 4, 58), nmccandre (Titel, DVD, DVD-Hülle, DVD-Inhalt), paulista (S. 2), PicsFive (S. 99, 100), PixelRobot (S. 132), svanhorn (S. 99, 114), uatp1 (S. 35, 62), vilax (S. 90), vydrin (S. 138), yellowcrest (S. 54)

**Abbildung © Fotolia.com:** gradt (S. 28)

**Abbildungen © iStockphoto LP:** 3alexnd (S. 5, 126), artistico (S. 139)

**Redaktionsassistent:** Michael Mentzel (mm) -153, Carmen Lehmann (cle) -387

**Korrektorat:** Kathleen Tiede, Hinstorff Verlag, Rostock

**Layout und Satz:** Enrico Eisert, Matthias Timm, Hinstorff Verlag, Rostock

**Titelidee:** iX

**Verlag:** Heise Medien GmbH & Co. KG, Postfach 61 04 07, 30604 Hannover; Karl-Wiechert-Allee 10, 30625 Hannover; Telefon: 05 11/53 52-0, Telefax: 05 11/53 52-129

**Geschäftsführer:** Ansgar Heise, Dr. Alfons Schröder

**Mitglieder der Geschäftsleitung:** Beate Gerold, Jörg Mühle

**Verlagsleiter:** Dr. Alfons Schröder

**Anzeigenleitung:** Michael Hanke (-167), E-Mail: [michael.hanke@heise.de](mailto:michael.hanke@heise.de)

**Druck:** Dierichs Druck + Media GmbH & Co. KG, Frankfurter Straße 168, 34121 Kassel

**DVD-Herstellung:** Klaus Ditze

**Verantwortlich:** Textteil: Jürgen Seeger; Anzeigenteil: Michael Hanke

**iX Developer 1/2016 – Internet der Dinge:** Einzelpreis € 12,90, Österreich € 14,20, Schweiz sfr 25,80, BeNeLux: € 14,80, Italien: € 16,80

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlages verbreitet werden; das schließt ausdrücklich auch die Veröffentlichung auf Websites ein.

Printed in Germany

© Copyright by Heise Medien GmbH & Co. KG

### Die Inserenten

Aicas	<a href="http://www.aicas.com/cms/en/jobs">www.aicas.com/cms/en/jobs</a>	79
dpunkt Verlag	<a href="http://www.dpunkt.de">www.dpunkt.de</a>	17, 31
Intel GmbH; ho-Computer Software GmbH	<a href="http://www.hocomputer.de">www.hocomputer.de</a>	156
Hostserver	<a href="http://www.hostserver.de">www.hostserver.de</a>	2
Lieber Lieber Software	<a href="http://www.lieberlieber.com">www.lieberlieber.com</a>	11
Logiball	<a href="http://www.logiball.de/karriere.html">www.logiball.de/karriere.html</a>	39
Rheinwerk Verlag	<a href="http://www.rheinwerk-verlag.de">www.rheinwerk-verlag.de</a>	53
wibu Systems	<a href="http://www.wibu.de">www.wibu.de</a>	23
ZIB	<a href="http://www.zib.de">www.zib.de</a>	79

Die hier abgedruckten Seitenzahlen sind nicht verbindlich. Redaktionelle Gründe können Änderungen erforderlich machen.



## Grundlagen

Während sich eine Reihe von Software-Entwicklern durch die Mitarbeit in Embedded- und Telekommunikationsprojekten schon einige IoT-Grundlagen aneignen konnte, stellt das Internet der Dinge andere vor ganz neue Herausforderungen. Ganz von vorn sollte aber niemand anfangen müssen, gibt es doch einige Analogien zum gewohnten Terrain.

Protokollgewirr, Privatsphäre und Open Source im IoT	8
Womit sich Backend-Entwickler nun beschäftigen müssen	16
Als Softwareentwickler Elektronik lernen	20
Datenerfassung in Industrieanwendungen	24
Ideen für das Internet der Dinge entwickeln	28
Entwicklermanifeste für das IoT	32

Protokollgewirr, Privatsphäre und Open Source im IoT

# (Un-)Schöne neue Welt



**Sebastian Golasch**

Vernetzung kann nicht nur den Komfort erhöhen, sie erzeugt auch Daten, deren Verbleib und Einsatz nicht immer nachvollziehbar ist. Wer die Kontrolle behalten möchte, sollte die Open-Source-Bewegung unterstützen.

**T**ee, Earl Grey, heiß!“ – Worte, die seit mehr als 25 Jahren Sehnsucht in Star-Trek-Freunden wecken. Seit Captain Picard zum ersten Mal an den Replikator trat, kennt die Welt eine neue Vision des Komforts: Der Mensch spricht etwas aus, der Computer setzt es um.

Doch nicht nur mit der Replikator-Technik zeigte die Serie, wie sich der Umgang mit den Maschinen in Zukunft gestalten könnte. Sieht man heute etwa, wie Counselor Troi per Sprachbefehl das Licht im Quartier auf 20 Prozent dimmt, scheinen derartige Annehmlichkeiten nicht mehr all zu weit entfernt zu sein. Schließlich gibt es Apples Siri, Amazons Alexa und Googles Spracherkennung auf der einen und via Smartphone steuerbare Lampen unterschiedlicher Anbieter auf der anderen Seite. Während viele Menschen Ersteres bereits ihr eigen nennen, kommt Letzteres langsam in allgemein erschwingliche Preisregionen. Es könnte sich daher die Frage stellen, warum Counselor Trois Quartier nicht längst eine Alltäglichkeit ist.

## IoT und was, bitte?

Eine simple Antwort darauf gibt es leider nicht, zu viele Faktoren bestimmen momentan den Markt der „Smart Home“-Angebote. Um die Zusammenhänge im Verlauf besser verstehen zu können, lohnt sich ein Blick zurück zu den Anfängen des vernetzten Zuhauses und des Internets der Dinge.

Der Begriff „Internet der Dinge“ ist zum Überbegriff für elektronische Einheiten jeglicher Art geworden, die über ein Netzwerk miteinander kommunizieren. Dabei ist die Grundidee so alt wie das Internet an sich, denn das Internet der Dinge umfasst nichts anderes als Computer, die miteinander in Verbindung treten. Was jedoch einen ordinären Rechner in

dem Kontext adelt, ist das, was seine Besitzer an ihn anschließen. Verbindet man etwa einen Raspberry Pi oder Arduino mit einer Kaffeemaschine, ergibt sich zwar noch kein Replikator, aber ein Gerät, das sich ohne Bedenken in die Schublade „IoT“ stecken lässt. Ein Telefon mit Computer im Hosentaschenformat: Internet of Things; ein USB-Stick am Fernseher, der Video- und Audiosignale überträgt: Internet of Things; ein Wasserkocher mit via Bluetooth steuerbarem An/Aus-Schalter: Internet of Things und so ließe sich die Liste eine ganze Weile fortführen.

Obwohl die Technik, die derartige Projekte ermöglicht, schon lange existiert, findet der Begriff „Internet of Things“ erst seit circa fünf Jahren größere Beachtung [a] (siehe Abbildung 1).

Der technische Fortschritt und die Verbreitung entsprechender Bausteine sowie anderer Produkte auf dem breiten Markt sind wichtige Faktoren, die den Trend erklären. Zum einen wurden Computer viel günstiger, schneller, kleiner und wesentlich energiesparender – ein Punkt, der im Laufe des vorliegenden Artikels noch ein paar mal Thema sein wird. Zum anderen ist auch der Zugang zum Internet wesentlich günstiger, schneller, verfügbarer und mehr Menschen zugänglich geworden. Der Mensch ist als dritte Komponente einer derartigen Rechnung generell nicht zu vernachlässigen: Ein Großteil der Weltbevölkerung akzeptiert das Smartphone als alltäglichen Begleiter, die Bindung zwischen Mensch und Maschine wird enger, die Angst vor der Technik schwindet.

## Viele Köche verderben den Brei

Das dadurch neue Geschäftsfelder entstehen, ist Unternehmen im Elektronik- und Kommunikationsbereich nicht verborgen ge-



blieben. Und doch hielten sich Branchengrößen à la Microsoft, Apple und Google lange zurück. Derweil besetzten viele kleinere Hersteller Nischenbereiche oder schlossen sich zu Allianzen zusammen, um gemeinsam den Markt bespielen zu können. Im Zuge der Markteroberung der „vielen Kleinen“ entstand jedoch eine Marktfragmentierung, die im IT-Umfeld ihres Gleichen sucht.

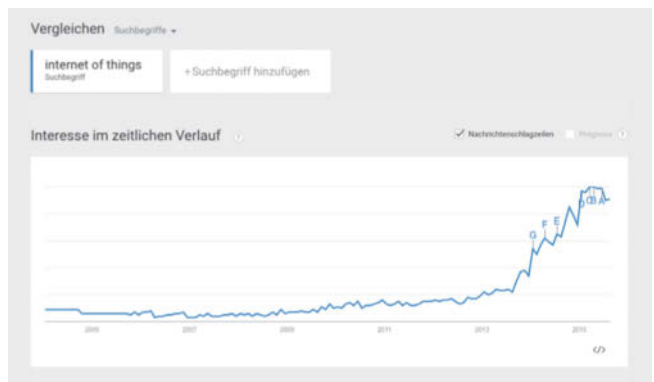
Ein Blick auf die am Markt verbreiteten Funkprotokolle im Bereich der Hausautomatisierungen bestätigt diese These. Das im deutschsprachigen Markt erfolgreiche Protokoll BidCos der Firma Homematic zeigt eine erstaunliche Verbreitung. Das Produkt der ZigBee-Allianz, das ZigBee-Protokoll [b], das sich durch seine fast ein Dutzend meist zueinander inkompatiblen Subprotokolle auszeichnet, hat in den letzten Jahren zunehmend an Popularität gewonnen, vor allem durch die starke Verbreitung des Hue-Beleuchtungs-Systems der Firma Philips. Verlässt man den deutschen Markt und wendet sich England und den USA zu, trifft man oft auf Produkte, die auf dem Z-Wave-Protokoll [c] basieren und sich langsam auch hier zunehmender Beliebtheit erfreuen.

Im Bereich des Energy Harvesting schickt sich derweil das EnOcean-Protokoll [d] an, die entsprechende Nische zu besetzen. Teil der Strategie ist es, die nötige Energie eines Schaltbefehls, der zum Beispiel bei der Betätigung eines Wandschalters abgesendet wird, durch die kinetische Energie zu generieren, die durch seine Benutzung entsteht. Auch alte Bekannte wie das drahtlose Telefonie bekannte DECT oder das vielseitig verwendbare Bluetooth-Protokoll sind in diesem Segment mit ihren Ultra-Low-Energy-Varianten vertreten. Seltener zu finden waren lange Zeit WLAN-Netze, doch auch hier ließ sich im vergangenen Jahr ein Zuwachs der Produkte verzeichnen, die auf die bestehende drahtlose Infrastruktur aufsetzen.

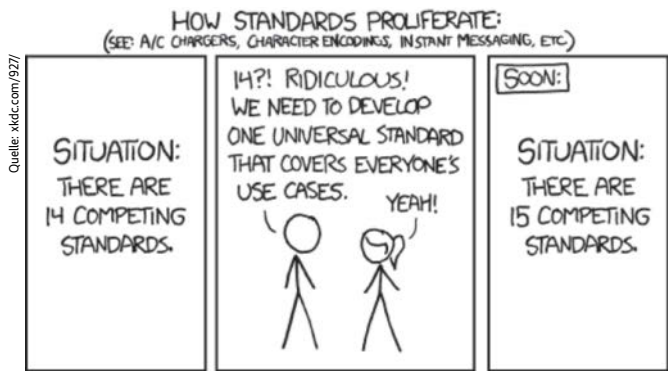
## Funkprotokolle in a Nutshell

Die Mehrzahl der Protokolle ist, obwohl sie zumeist auf den gleichen Frequenzen senden, zueinander inkompatibel und unterliegt zudem diversen firmenpolitischen und markenrechtlichen Beschränkungen. Mal sind die Spezifikationen öffentlich zugänglich, mal unterliegen sie der Geheimhaltung.

Ein Ende der Protokollmanie ist nicht absehbar – mit dem Thread-Protokoll von Nest/Google oder dem neuen Ableger aus deutschen Landen, Homematic IP, dringen weitere Protokolle in den Markt ein, eine Konsolidierung ist, zumindest kurzfristig, nicht absehbar.



**In den letzten Jahren ist das Interesse am „Internet of Things“ sprunghaft angestiegen – zumindest, wenn Suchergebnisse als Indikator dienen dürfen (Abb. 1).**



**Leicht überzeichnet dargestellt, scheint die Standardlandschaft sich im IoT-Bereich tatsächlich so oder ähnlich zu entwickeln (Abb. 2).**

Was viele dieser Funkprotokolle jedoch gemein haben, ist die Nutzung der sogenannten ISM-Frequenzbänder [e] (Abkürzung für Industrial, Scientific, Medical), die von der Vollzugsordnung für den Funkdienst weltweit vorgegeben sind. In Deutschland wird deren Einhaltung von der Bundesnetzagentur überwacht, in anderen Ländern von vergleichbaren Behörden (siehe Tabelle 1 auf S. 10).

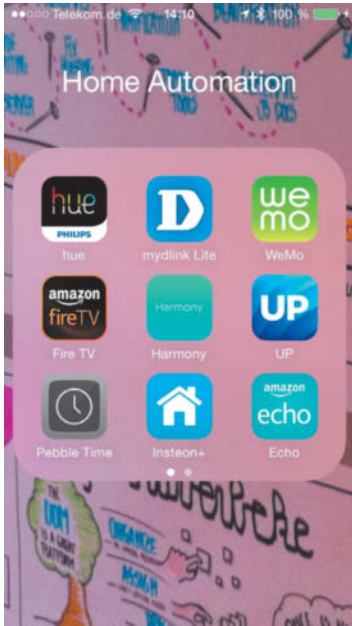
Nimmt man Abstand von den sogenannten „Baumarkt-Steckdosen-Sets“, die zumeist auf einer Frequenz von 433 MHz operieren, aber nur unidirektional agieren können, kann man sein Augenmerk im europäischen Raum auf die Frequenzen 868 MHz und 2,4 GHz richten.

Als Randnotiz sei dabei bemerkt, dass das 868-MHz-Band nicht zu den ISM-Bändern zählt und in Europa exklusiv für den Einsatz von Funkkommunikation über Kurzstrecken reserviert ist. Aus dem Grund kommt es oft zu Inkompatibilitäten zwischen EU- und Nicht-EU-Geräten, obwohl sie auf das gleiche Protokoll setzen. So senden Z-Wave-Geräte in den USA auf dem ISM-Band der Region 2 mit 915 MHz, während das typgleiche Modell auf dem europäischen Markt auf der SDR-Frequenz 868 MHz funkt (ein Link zu einer entsprechenden Quelle ist, neben einigen anderen, unter der nach „Alle Links“ am Ende des Artikels aufgeführten URL zu finden). Das 868-MHz-Band kommt unter anderem bei BidCos, Homematic IP, Z-Wave, EnOcean und teilweise ZigBee zum Einsatz.

Weltweit einheitlich sind hingegen die Regularien für das 2,4-GHz-Band. Seine berühmtesten Vertreter sind sicherlich die nach IEEE 802.11 (gebräuchliches Haus-WLAN) agierenden Geräte und Bluetooth. Aber auch einige ZigBee-Protokolle senden auf der Frequenz.

## Individuelles Vorgehen

Die Kommunikation der Geräte untereinander ist trotz des gleichen Frequenzbands unterschiedlich geregelt, während Z-Wave beispielsweise ein Mesh-Netzwerk [f] der Komponenten untereinander aufbaut, nutzt das BidCos-Protokoll eine Mischung aus Stern- und Mesh-Topologie (siehe S. 36). Bei Z-Wave kommt die Mesh-Technik vor allem zum Einsatz, um die Reichweite zu erhöhen, was im Fehlerfall einer Komponente zum Zusammenbruch des gesamten Netzwerks führen kann, inklusive dessen bis zu 10 minütiger Neuaufbauphase und Unerreikbaarheit. BidCos hingegen nutzt eine derartige Komponentenverbindung zur effektiveren Steuerung von Einzelkomponenten. So liegt BidCos generell eine Sterntopologie



**Die Vielzahl von Apps, die durch fehlende Interoperabilität allein im Bereich Home Automation verfügbar sind, lässt sich mittlerweile schwer überblicken (Abb. 3).**

funkfähigen Tablets, gesehen hat, sollte verstehen, was gemeint ist. Dabei sind solche Geräte in der Regel noch mit hochwertigen Stromspeichern ausgestattet, während viele der am Markt befindlichen Smart-Home-Geräte handelsübliche AA-, AAA-Batterien oder gar Knopfzellen für den Betrieb brauchen. Eine stete Verbindung und damit Erreichbarkeit aus einem WLAN ist damit heutzutage unmöglich.

Eine bizarre Szene, die sich aus dem Protokollwirrwarr ableitet, ist in Abbildung 3 zu sehen: Da kein Standard existiert, benötigen Nutzer fast für jedes der Geräte eine eigene Steuer-Software, die heutzutage meist in Form einer Smartphone-Applikation zur Verfügung steht. Nicht nur, das dadurch aus dem Internet der Dinge proprietäre „iPhone-Apps der Dinge“ werden, viel schlimmer ist, dass so die Interaktion der Geräte untereinander unterbunden ist.

zugrunde, in der eine zentrale Einheit die Komponenten steuert. Zusätzlich lässt sich aber auch beispielsweise ein Thermostat direkt mit der Heizungssteuereinheit verknüpfen, um unter anderem eine schnellere Befehlsverarbeitung zu realisieren.

### Bund und EU spielen mit

### Das Intranet der Dinge

Der beschriebene Umstand ist einer der Gründe, warum sich Dienste wie „If this than that“ (siehe Abbildung 4), kurz IFTTT, großer Beliebtheit erfreuen, können sie doch den x Euro teuren Lichtschalter von Marke A mit der Glühbirne von Firma B interagieren lassen. Die Kehrseite der Medaille ist aber, das Nutzer einem Drittanbieter Zugriff auf noch mehr persönliche Daten geben müssen, nur um die Grundfunktionen ihrer Installation sicherzustellen. So ergibt sich ein Vendor Lock-in, der viele Hersteller freuen dürfte, da er den Kunden dazu zwingt, alles aus einer Hand zu kaufen. Allerdings sorgt er auch dafür, dass Abnehmer nur eingeschränkt die Option haben, sich die für ihren häuslichen Anwendungszweck geeigneten Geräte zuzulegen und sie kombiniert zu nutzen.

Ein genereller Irrglauben scheint allerdings auch in der Art und Weise zu herrschen, wie Geräte und Subnetzwerke an das Internet angebunden sind. Kaum eines der am Markt erhältlichen Geräte verbindet sich direkt mit dem Internet. Zum Großteil basieren die Hausnetzwerke auf dem Subnetz des jeweiligen Funkprotokolls, das sich über eine Art Router, bei vielen Herstellern auch Coordinator oder Bridge genannt, über einen zentralen Punkt Zugang zum Internet verschafft. In diesem Schema wissen die einzelnen Geräte nichts über das, was hinter dem Router vor sich geht. Dass dahinter ein noch viel größeres Netzwerk steht, ist ihnen schlicht unbekannt.

Als wären Protokollwirrwarr und Herstellerinteressen nicht schon genug, auch die Behörden spielen im intelligenten Zuhause mit. So ist es in einigen Bundesländern mittlerweile Pflicht, unter bestimmten Bedingungen Rauchmelder mit mindestens 10 Jahren Batterielaufzeit [g] zu installieren. Das bedeutet, dass bei einem vernetzten Rauchmelder neben der reinen Sensorik auch die datenverarbeitende Logik, sprich der eingebaute Minicomputer, mit Strom zu versorgen ist. Das gleiche gilt für die Send- und Empfangseinheit, die für die Kommunikation mit dem Router zuständig ist.

Einer der treibenden Gründe für die „Intranetifizierung“ der Hausautomationsprotokolle ist der Energiehunger, den eine „Always on“-Internetverbindung mit sich bringt. Wer einmal den Verbrauch eines WLAN-Geräts, zum Beispiel eines nicht mobil-

Da die heutige Technik nicht für die Konzeption eines immer sende- und empfangsbereiten Geräts ausgelegt ist, zwingen die Regularien Hersteller dazu, in ihrer Entwicklung Kreativität zu zeigen. Beispielsweise melden sich Rauchmelder nur in großen Intervallen bei ihrer Basisstation und fahren ihre Aktivitäten außerhalb der Meldeintervalle auf ein Minimum herunter, sollte kein Sonderfall wie erhöhte Rauchentwicklung vorliegen.

Von der VO Funk ausgewiesene ISM-Bänder			
Von	Bis	Typ	Anmerkungen
6,765 MHz	6,795 MHz	A	Short Range Devices (SRD)
13,553 MHz	13,567 MHz	B	SRD
26,957 MHz	27,283 MHz	B	SRD
40,66 MHz	40,70 MHz	B	SRD
433,05 MHz	434,79 MHz	A	SRD, nur Region 1 (Europa, Afrika, Nachfolgestaaten der UdSSR und Mongolei)
902 MHz	928 MHz	B	nur Region 2 (Nord- und Südamerika)
2,4 GHz	2,5 GHz	B	-
5,725 GHz	5,875 GHz	B	-
24 GHz	24,25 GHz	B	-
61 GHz	61,5 GHz	A	-
122 GHz	123 GHz	A	-
244 GHz	246 GHz	A	-

Quelle: <https://de.wikipedia.org/wiki/ISM-Band> (Stand: August 2015)

Allerdings ist das nicht der einzige Fall, in dem es Einschränkungen zu beachten gilt: Einen viel größeren Einfluss auf die heute gebräuchlichen Funkprotokolle im 868-MHz-Band hat der „Duty Cycle“ beziehungsweise die „1%-Regel“ [h]. Da mehrere Protokolle die Frequenz nutzen, ist auszuschließen, dass sie sich gegenseitig stören, weshalb die Bundesnetzagentur in dem Fall maximale relative Frequenzbelegungsdauern festschreibt. Ein Gerät, das auf der Frequenz abseits des genutzten Protokolls agiert, darf pro Stunde maximal ein Prozent der Zeit auf Sendung gehen, umgerechnet also 36 Sekunden. Die Menge der Daten, die es dabei übertragen darf, ist nicht festgelegt. Daher kann es bei Protokollen mit langsamerer Datenübertragung schneller zu Engpässen kommen als bei Protokollen mit höherer Übertragungsrate.



# ENTERPRISE ARCHITECT

## Enterprise Architect 12.1 mit UML 2.5 Unterstützung

Softwareentwicklung kann so einfach werden:

- ⇒ Unterstützung für UML, SysML, SOMF, BPMN, BPEL uvm.
- ⇒ Besseres Design & Verständnis der Architektur im gesamten Team
- ⇒ Validieren der Anforderungen
- ⇒ Nachvollziehbarkeit der Testfälle und der Testabdeckung
- ⇒ Unterstützung verschiedener Vorgehensmodelle
- ⇒ Für mehr als 10 Programmiersprachen



**Jetzt Enterprise Architect 12.1 testen!**  
**Besuchen Sie [www.sparxsystems.de](http://www.sparxsystems.de)**

# LieberLieber

UML Debugging

## Embedded Code Generation

## Requirements Tracing

## Model Versioning

Addins & Customizations

Make Enterprise  
Architect your tool.



Für den Nutzer bedeutet das, dass beispielsweise eine Funksteckdose nach x Minuten, in denen regelmäßig gesendet wurde, nicht mehr auf Befehle reagiert, da die maximale Nutzungsdauer überschritten wurde. Tritt der Fall ein, sind weitere n Minuten zu warten, bis das Gerät wieder für den Einsatz freigegeben ist. Im Regelfall wird solch eine Grenze selten erreicht, allerdings gibt es schon im normalen Hausgebrauch Situationen, in denen dieses Regularium negativ auffallen kann.

## Sicherheit wird kleingeschrieben

Eines der entscheidenden Themen im Bereich der Hausautomatisierung ist die Sicherheit der Protokolle – nicht umsonst werben viele der Hersteller damit, das sich durch den Einsatz ihrer Systeme die eigene Sicherheit erhöhen lässt. Im besonderen der Einbruchschutz ist einer der am häufigsten zitierten Anwendungsfälle.

Es ist sicherlich nicht von der Hand zu weisen, dass Systeme, die bei Abwesenheit das Licht und die Rollläden steuern und so Hausbewohner simulieren, abschreckend auf Einbrecher wirken können. Auch Meldungen über das unerwartete Öffnen von Fenstern und Türen oder der klassische Einsatz eines Bewegungsmelders während der Abwesenheit kann dazu führen, das sich Verbrechen schneller melden und aufklären lassen.

Bei diesen gern vom Hersteller ausgemalten Anwendungsfällen ist allerdings auch die Gegenfrage zu stellen: Machen sich Nutzer durch den Einsatz der Sensoren anfälliger für Einbrüche, da der Dieb eventuell den Status von Türen und Fenstern auslesen kann, um zu erkennen, ob ein einfaches Eindringen möglich ist? Oder: Können Eindringlinge ein Smartlock, ein funkgesteuertes Türschloss, hacken und ohne jegliche Spuren in die Wohnung eindringen (Link dazu unter der „Alle Links“-URL)?

Eine klare Antwort auf diese Fragen gibt es nicht, jedoch muss man sich vor Augen führen, dass viele der Protokolle es mit der Sicherheit nicht so genau nehmen. Eine Verschlüsselung der übertragenen Daten ist in der Hausautomatisierung eher die Ausnahme als die Regel. Selbst wenn die am häufigsten genutzten Protokolle eine Verschlüsselung der Daten in ihren Spezifikationen beschreiben, so ist die Umsetzung für die Hersteller von Endgeräten oft optional.

Um sichere Verschlüsselungsverfahren in Echtzeit zu betreiben, benötigt man mehr Rechenleistung, was wiederum einen höheren Energieverbrauch nach sich zieht. Überträgt man das wieder auf das Beispiel des Rauchmelders, ist es unmöglich, die vorgeschriebene Batteriebensdauer einzuhalten und die Daten zudem noch zu verschlüsseln. In einem Netzwerk mit Stern-topologie ist das, abhängig von der Art des Geräts, noch relativ unbedenklich, aber in einem Mesh-Netzwerk können sich durchaus weitere Angriffsvektoren finden, sollte ein ungesicherter Knoten Kommunikation von sicheren Gegenstellen weiterleiten.

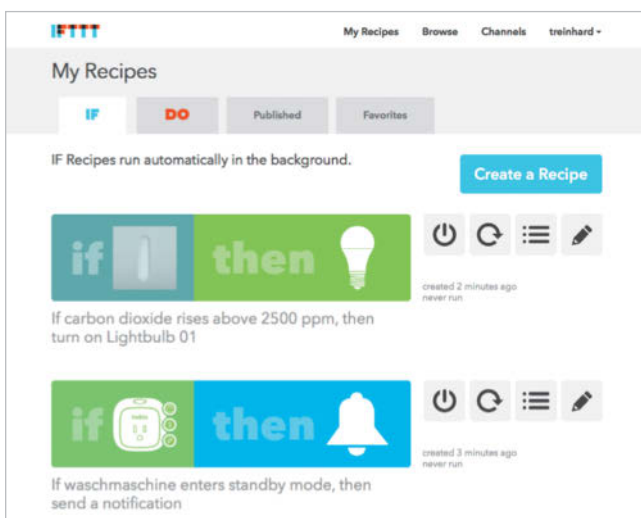
Um der Sicherheit dennoch Genüge zu tun, zumindest im Bereich der Steuerung, bieten viele Protokolle die Option, beim Anlernen des Geräts an der Basiseinheit, dem Router, einmalig einen Sicherheitsschlüssel auszutauschen, mit dem sich dann die Steuerbefehle signieren lassen. Das Verfahren gilt als sicher genug, um zumindest die Steuerung der Geräte vor Fremdeinfluss zu schützen. Allerdings ist zu beachten, dass so mehr Kommunikation zwischen Gerät und Router entsteht, wodurch die Batteriebensdauer sinkt und man schneller den Duty Cycle erreicht. Zudem ist das Verfahren in den Protokollspezifikationen oft als optional angegeben, weshalb es auch in dem Fall Geräte gibt, die es nicht unterstützen.

2013 zeigte der Chaos Computer Club am Beispiel von Home-matic, dass auch dieser Mechanismus seine Schwächen hat (ein Link zu einem entsprechenden Vortrag findet sich unter der „Alle Links“-URL). An der Stelle sollte man allerdings erwähnen, dass die düsteren Zukunftsszenarien von Einbrechern, die sich den Eintritt in die fremden vier Wände erhacken, noch eine Weile Zukunftsmusik bleiben werden. In Deutschland, gerade im städtischen Raum, gehen fast 20 Prozent der Einbrüche auf das Konto der Beschaffungskriminellen, eine Klientel, das wohl weiterhin den Ziegelstein dem Computer als Einbruchwerkzeug vorziehen wird. In den Statistiken zu Wohnungseinbrüchen sind bisher keine Fälle von „erhacktem“ Zugang aufgeführt (siehe „Alle Links“).

## Privatsphäre als Gratisdreingabe

Nebst der ungebrochenen Aktualität des Themas „Datenschutz“ durch NSA-Skandal, Snowden-Leaks und Vorratsdatenspeicherung kann der Kauf eines intelligenten Helferleins für den Hausgebrauch das Problem ganz akut ins eigene Heim bringen. Als Anstoß für ein simples Gedankenexperiment kann schon der Kauf einer funkfernsteuerbaren Glühbirne auf Grundlage des ZigBee-Protokolls dienen. Was sie über ihre Nutzer verrät, hängt von der Art und dem Umfang der Nutzung der angebotenen Funktionen ab. Die Otto Normalverbraucher überlegen sich im Normalfall allerdings nicht, ob sie etwa Geofencing ein oder ausschalten, für sie handelt es sich dabei lediglich um eine Komfortfunktion, keine Einladung zum Spähen.

Nutzen sie die vom Anbieter der Lampe bereitgestellte Geo Location Services, damit das Licht ausgeht, wenn sie die Wohnung verlassen und sich automatisch wieder anschaltet, kurz bevor sie nach Hause kommen, geben sie Einiges preis. Der Hersteller weiß theoretisch, wann der Glühbirnenkäufer zuhause ist, bekommt die Möglichkeit, seinen Aufenthaltsort zu bestimmen und Muster daraus abzuleiten. Nicht zuletzt erfährt er so den Wohnort des Käufers. Darüber hinaus lassen sich aus den Nutzungsinformationen Rückschlüsse auf den Schlaf-/Wachrhythmus ziehen und beim Einsatz mehrerer Lampen können Anbieter durch die Signalstärke und die davon ableitbare Entfernung vom Router annähernd bestimmen, wie groß die Wohnung oder das Haus ist, in dem die Birnen Verwendung finden.



IFTTT hilft zwar beim Kommunikationsproblem, verlangt dafür allerdings auch Nutzerdaten (Abb. 4).



# FÜR ROOTINIERS.

iX. WIR VERSTEHEN UNS.

**Jetzt auch für Android!  
Das Mini-Abo testen:**

3 Hefte + 16GB USB-Stick nur 13,50 Euro  
[www.iX.de/digital](http://www.iX.de/digital)



Sie wollen Zugriff auf alle Fakten? Nehmen Sie ihn sich – iX ab sofort auch als Android-App. Testen Sie 3 aktuelle Ausgaben jetzt komplett papierlos auf Ihrem Android/iOS-Tablet & -Smartphone per HTML5 oder PDF zum Vorzugspreis. **Jetzt zugreifen: [www.iX.de/digital](http://www.iX.de/digital)**



Das Vorhandensein allein bedeutet sicherlich noch nicht, dass die Hersteller all diese Möglichkeiten aktiv nutzen, existent sind sie allerdings und kein Anbieter lässt sich gerne bei der Datenauswertung in die Karten blicken. Wenn man sich vor Augen führt, dass sich derartige Daten nur durch eine simple Glühbirne sammeln lassen, will man sich gar nicht erst vorstellen, was man durch den Einsatz anderer Geräte alles preisgibt. Mit „Always On“-Audio-Capture-Geräten, Smart TVs, Fitnessarmbändern, Smart Watches, Netzwerkkameras und Umweltkontrollgeräten lässt sich eine Menge an interessanten Kundendaten sammeln. Letztere verarbeitet der Hersteller zumeist nicht lokal, sondern in den eigenen Rechenzentren – das sich auf dem Weg dorthin noch weitere Spionagemöglichkeiten bieten, ist ein Fakt, keine Fiktion.

Angesichts dieser Möglichkeiten fällt es nicht schwer, sich auszumalen, was staatliche Organe mit solchen Informationen anstellen könnten. In erster Linie gehen die Daten jedoch an Unternehmen, denen es nicht um die Kontrolle der Bürger, sondern um monetäre Vorteile geht – Daten sind sprichwörtlich Gold wert. Definiert sich ein Unternehmen über die Masse an gesammelten Daten, hängen Arbeitsplätze und Wohlstand an ihnen, was sich als gefährliche Kombination erweisen kann.

## Zurück zur Selbstbestimmung

Möchte man dem etwas entgegensetzen, sollte man nicht damit warten, sich zu engagieren. Spätestens, wenn derartige Produkte zur Standardausstattung von Mietwohnungen gehören, hilft nur noch der Ausweg, als Eremit in den Bergen zu leben.

Eine Alternative zu Unternehmensangeboten bietet derzeit die DIY- und Maker-Szene, die Open-Source-Hardware wie das Arduino Board hervorgebracht hat und mit Erweiterungen und Sensoren Optionen bietet, um Heimnetze selbst aufzubauen. Open-Source-Software mit entsprechender Verschlüsselungstechnik kann das Übrige erledigen, und sollte Anbindung an externe Netze nötig sein, lässt sie sich zu eigenen Konditionen einrichten. Das es sich dabei um kein massentaugliches Szenario handeln dürfte, sollte jedem bewusst sein.

Die vielversprechendste Option ist daher, den Gedanken von Open-Source-Software und -Hardware in die Unternehmen zu tragen, und wenn möglich, nicht nur den Gedanken, sondern gleich die entsprechenden Projekte. Eines der interessantesten ist in der Hinsicht das von der Eclipse Foundation getragene Eclipse Smart Home [i]. Das Projekt kommt bisher nicht nur in der DIY-Szene zum Einsatz, sondern findet auch in Unternehmen Beachtung, die letztlich dafür sorgen, dass es in Endkundenprodukten auf dem Markt landet.

Der Codeanteil derartiger Produkte ist offen einsehbar, der geneigte Entwickler hat folglich die Option, aktiv einzugreifen und so ohne Unternehmenszugehörigkeit Einfluss zu nehmen. Übernimmt wie im Fall von Smart Home eine unabhängige Stelle Überwachung und Regulierung des Projekts, sind die Möglichkeiten für Firmen, negativ Einfluss zu nehmen, begrenzt. Momentan bergen derartige Entwicklungen sicherlich die besten Chancen, Privatsphäre und Kontrolle über das eigene Heim zu behalten.

## Veränderungen im Kleinen

Wessen Java-Kenntnisse sich in Grenzen halten, oder wer nicht die Zeit opfern mag oder kann, sich in ein gewaltiges Projekt wie Eclipse Smart Home einzuarbeiten, kann trotzdem etwas

bewegen. Wichtig ist das Experimentieren und Aufbrechen der Vendor Lock-ins, die Programmiersprache ist dabei zweitrangig. Als Schweizer Taschenmesser hat sich mittlerweile etwa Node.js herausgestellt. Die JavaScript-Plattform ermöglicht das schnelle Prototyping von Ideen und hat zudem eine steile Lernkurve.

Node.js bietet Zugriff auf USB-Ports, Arduino-Entwicklung, Bibliotheken für UPNP oder Bonjour-Netzwerk-Discovery, wobei alles mit vergleichsweise wenig Aufwand zu erlernen und dank guter Tutorials recht zugänglich ist. Hat man einen Prototyp entwickelt, kann man sich auf Basis seines Codes immer noch andere Entwickler ins Boot holen, die in der Lage sind, einem beim Portieren für ein größeres Projekt wie Eclipse Smart Home zu helfen.

## Fazit

Bis „Tee, Earl Grey, heiß!“ Realität ist, müssen Konsumenten wohl noch etwas warten. In der Zwischenzeit ist es jedoch die Aufgabe mündiger Entwickler und Endanwender, die Hersteller in die richtige Richtung zu lenken. Sie alle haben die Chance, etwas zu bewirken – sei es durch Kaufkraft und die Unterstützung von Unternehmen, die Open-Source-Software in ihre Produkte integrieren oder die Mitarbeit an Projekten wie Eclipse Smart Home. Schließlich wurde auch in Star Trek nirgends erwähnt, dass die Enterprise nicht mit einem Stack aus Open-Source-Projekten betrieben wird. (jul)



### Sebastian Golasch

arbeitet als Fach-Senior-Manager-Software-Developer an den Smart Home UIs der Telekom-eigenen Hausautomatisierungsplattform Qivicon. In seiner Freizeit trägt er zu Open-Source-Software in den verschiedensten Bereichen bei.

## Onlinequellen

- [a] Google Trends  
[google.com/trends/explore#q=Internet%20of%20things%20](https://www.google.com/trends/explore#q=Internet%20of%20things%20)
- [b] ZigBee-Netzwerkspezifikationen  
[zigbee.org/zigbee-for-developers/network-specifications](https://www.zigbee.org/zigbee-for-developers/network-specifications)
- [c] Protokollübersicht Z-Wave  
[wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243\\_2\\_Z\\_Wave\\_Protocol\\_Overview.pdf](https://wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243_2_Z_Wave_Protocol_Overview.pdf)
- [d] EnOcean-Funkstandard  
[enocean-alliance.org/de/enoclean\\_standard/](https://www.enocean-alliance.org/de/enoclean_standard/)
- [e] Bundesnetzagentur: Funkanwendungen auf den ISM-Bändern  
[emf3.bundesnetzagentur.de/pdf/ISM-BNetzA.pdf](https://www.emf3.bundesnetzagentur.de/pdf/ISM-BNetzA.pdf)
- [f] Z-Wave-Topologie und -Routing  
[en.wikipedia.org/wiki/Z-Wave#Topology\\_and\\_routing](https://en.wikipedia.org/wiki/Z-Wave#Topology_and_routing)
- [g] VDS-Anforderungen für Rauchmelder  
[vds.de/fileadmin/vds\\_publicationen/vds\\_3131\\_web.pdf](https://www.vds.de/fileadmin/vds_publicationen/vds_3131_web.pdf)
- [h] EQ3 Guide „Duty Cycle“  
[eq-3.de/faq.html?id=128](https://www.eq-3.de/faq.html?id=128)
- [i] Eclipse SmartHome  
[eclipse.org/smarthome](https://www.eclipse.org/smarthome)



## // EINTAUCHEN INS INTERNET DER DINGE

**D**as Internet der Dinge verspricht viel: eine Fülle neuer Produktideen und Geschäftsmodelle sowie weitreichende Veränderungen im privaten und beruflichen Leben. Während es an kreativen Ideen nicht mangelt, ist die technische Umsetzung nicht trivial, und viele Softwareentwickler stehen vor großen Herausforderungen wie einem schwer überschaubaren Ökosystem aus Hardware, Protokollen und Softwarekomponenten. Auf der anderen Seite müssen sich Entwickler mit Hardware-Erfahrung andere, für sie neue Themen erarbeiten, unter anderem Vernetzung / Connectivity, Sicherheit und Datenschutz.

Die **building IoT 2016** setzt hier an und hilft Ihnen, sich im Dschungel der neuen Aufgaben zurechtzufinden. Hier treffen Sie auf die Gestalter des Internet of Things, das sich langsam zum Internet of Everything mausert. Vieles ist hier noch im Fluss, allerdings helfen das in Vorträgen vermittelte Grundlagenwissen und die Möglichkeit zum Austausch über aktuelle Forschungsvorhaben dabei, Ihre eigenen Projekte erfolgreich anzugehen und umzusetzen.

## Programm im Januar 2016 online!

### THEMEN:

- // Hardware-, Elektronik- und IT-Grundlagen für Entwickler
- // Werkzeuge, Programmiersprachen, Frameworks und Bibliotheken
- // IoT-Plattformen
- // Architektur, Modellierung und Prototyping von IoT-Projekten
- // Security und Safety
- // Projektorganisation
- // Nutzerschnittstellen, UI, UX
- // Protokolle und Kommunikationsstandards
- // IoT und Big Data



Womit sich Backend-Entwickler nun beschäftigen müssen



# Weniger ist mehr

**Henryk Konsek**

Wer bisher bei der Planung seiner Softwareprojekte auf stabile Verbindungen und unbegrenzten Speicher setzen konnte, muss im IoT-Umfeld in einigen Punkten umdenken. Darüber hinaus lässt sich aber auch ein Großteil des bisher angesammelten Wissens weiterverwenden.

**G**laubt man den Prognosen, wird das Internet der Dinge eine große Sache, die durch mehrere Millionen vernetzter Geräte und noch mehr Dollar die Wirtschaft bis 2020 antreiben soll. Schon heute beginnen Unternehmen, Unmengen von Daten heterogener Geräte zu sammeln. Ihr Ziel ist es, die gesammelten Informationen für neue Produkte zu nutzen, sich so weiterzuentwickeln und auf lange Sicht effizienter und ertragreicher zu sein. Allerdings kommen mit dem Wandel der Geschäftsmodelle für das IoT neue Aufgaben auf die in Unternehmen angestellten Backend-Entwickler zu, sollen sie den Prozess aktiv begleiten.

Um einen besseren Eindruck davon zu bekommen, welche Herausforderungen in den kommenden Jahren auf Softwareentwickler warten, ist es erst einmal wichtig, sich die Unterschiede zwischen Embedded- und Rechenzentrumsentwicklung vor Augen zu führen und zu überlegen, wie sie mit dem IoT zusammenhängen. Embedded-Entwickler erstellen in der Regel Software, die auf einer bestimmten Hardware laufen soll. Letztere wiederum ist meist in einem Produkt des Unternehmens, etwa einem Kühlschrank oder Auto, verbaut und soll in dem Kontext vom Endnutzer eingesetzt werden. Normalerweise kommen in dem Bereich Hardware-nahe Sprachen wie C oder C++ zum Einsatz. Die Plattform, für die Programmierer entwickeln, ist zudem hochspezifisch und nur für bestimmte Geräte ausgelegt. Embedded-Code ist daher in der Regel nicht portierbar und enthält Hardware-spezifische Referenzen.

Rechenzentrums- oder Backend-Entwickler schreiben derweil meist Software mit höherem Abstraktionsgrad in Sprachen wie Java oder .NET. Code in dem Bereich lässt sich für unterschiedliche Hardware-Plattformen portieren und Anwendungen können auf den meisten Servern laufen. Häufig befassen sich Backend-Entwickler nicht mit der verwendeten Hardware, da es

in der Regel nicht nötig ist, zu wissen, worauf die Anwendung im Produktivbetrieb läuft. Solange der Server mit wenigstens einem Gigabyte Speicher und einem Prozessor mit x64-Architektur ausgerüstet ist, gibt es keine Probleme.

## Komplexe Architektur

Beschäftigt man sich mehr mit dem Internet der Dinge, scheint es, als ob die Prinzipien der Embedded-Programmierung Einzug in die Welt der Datenzentrumssysteme halten. Den Marktprognosen nach zu urteilen, werden sich immer mehr traditionelle Softwareanbieter mit Embedded-Programmierung befassen, besonders aufgrund der damit verbundenen Möglichkeiten und der Verfügbarkeit günstiger Bauteile. Einplatinencomputer und Sensoren sind mittlerweile nicht mehr den Hardwareherstellern vorbehalten, weshalb es einfacher ist, damit Versuche zu starten und auf ihrer Grundlage Geschäftsmodelle zu entwickeln.

Die größte Umstellung muss ein Backend-Entwickler in der IoT-Welt wahrscheinlich in puncto Architektur durchmachen. Meist lassen sich Datenzentrumssysteme als Backend-Anwendung sehen, die mit einer Datenbank verknüpft sind und dem Client Zugriff auf bestimmte Daten gewähren. Normalerweise kommt ein Webclient zum Einsatz, etwa ein Browser, der das HTTP-Protokoll zugrunde legt.

Eine typische IoT-Architektur (siehe auch S. 36) lässt sich als Übermenge dieses Aufbaus beschreiben. Zwar ist immer noch ein Backend-Dienst vonnöten, um den Webclients die Inhalte zu liefern, gleichzeitig ist er aber auch mit einer Vielzahl weiterer Geräte in Verbindung, die ihm mit unterschiedlichen Protokollen Nachrichten zukommen lassen. Eine derartige Architektur ist in der Abbildung auf Seite 18 zu sehen. Zudem ist die Gateway-