Future Internet –

The Private Things

and Services of Internet

Das private, d.h. nicht öffentliche Internet der Dinge und Dienste
in der “Private Cloud” auf dem heimischen Rechner.

Version 1.14 vom 22. Januar 2013

Zum Inhaltsverzeichnis

Dieter H. Herold

64285 Darmstadt

Tel. 0 61 51 / 951 96 80

dieter.h.herold@web.de

1 Das Internet der Dinge

Ob Sie es glauben oder nicht, aber wir befinden uns bereits am Anfang der vierten industriellen Revolution, die alles bisher Bekannte, Gewusste und Bewusste auf den Kopf stellt! Kein Stein bleibt auf dem anderen, nichts bleibt so wie es ist!

>> Als Industrielle Revolution wird die tiefgreifende und dauerhafte Umgestaltung der wirtschaftlichen und sozialen Verhältnisse, der Arbeitsbedingungen und Lebensumstände bezeichnet, die in der zweiten Hälfte des 18. Jahrhunderts begann und verstärkt im 19. Jahrhundert, zunächst in England, dann in ganz Westeuropa und den USA, seit dem späten 19. Jahrhundert auch in Japan und weiteren Teilen Europas und Asiens zum Übergang von der Agrar- zur Industriegesellschaft geführt hat. Als wichtigste an dieser Umwälzung beteiligte Gesellschaftsklassen standen sich kapitalistische Unternehmer und lohnabhängige Proletarier gegenüber.

Die Industrielle Revolution führte zu einer stark beschleunigten Entwicklung von Technik, Produktivität und Wissenschaften, die, begleitet von einer starken Bevölkerungszunahme, mit einer neuartigen Zuspitzung der sozialen Missstände einherging: Es kam zu einer Teilverlagerung des Pauperismus vom Lande in die Städte, ohne dass hinreichende Wohnunterkünfte vorhanden waren; und in den entstehenden Fabriken, für die Arbeitskräfte gebraucht wurden, konzentrierte sich ein Lohnarbeiterproletariat. Daraus ergab sich als ein gesellschaftspolitisches Kernproblem die Soziale Frage, verbunden mit wiederkehrenden Arbeiterunruhen und Bemühungen von Sozialreformern, die akute Not zu lindern und deren Ursachen zu bekämpfen.

In weltgeschichtlicher Perspektive wird der Industriellen Revolution eine ähnliche Bedeutung zugemessen wie dem Übergang vom Nomadentum zur Sesshaftigkeit in der Neolithischen Revolution. Bezüglich der Industriellen Revolution bildeten sich mit der Zeit zwei Begriffsebenen heraus: Die eine meint die mit der Entstehung der Großindustrie verbundene Epochenbezeichnung, während die andere auf einen unabgeschlossenen Prozess fortlaufenden Gesellschaftswandels zielt. Die in vor- und frühindustrieller Zeit am meisten benachteiligten proletarischen Schichten profitierten im weiteren Verlauf auch ihrerseits von der industriellen Revolution, indem eine große innerstaatliche soziale Ungleichheit als Problem begriffen wurde und breitere Bevölkerungsschichten in die Lage kamen, sich einen relativen Wohlstand zu erarbeiten.

Einige Wirtschaftshistoriker und Sozialwissenschaftler kennzeichneten spätere historische Umbrüche in den Wirtschafts-, Produktions- und Arbeitsformen als zweite und dritte Industrielle Revolution. Der französische Soziologe Georges Friedmann sprach 1936 erstmals von einer zweiten industriellen Revolution. Er datierte sie auf die Jahrzehnte um 1900 und identifizierte als deren Charakteristika die intensivierte Mechanisierung, den weitverbreiteten Gebrauch von Elektrizität und die Massenproduktion von Gütern (Taylorismus und Fordismus). Die mikroelektronische Revolution seit Mitte der 1970er-Jahre wird als technologischer Kern einer neuen,

dritten Industriellen Revolution angesehen, so zum Beispiel von dem US-amerikanischen Soziologen Daniel Bell. << (Quelle: Wikipedia)

Bereits im Jahr 2006(!) ließ das ZDF/3sat eine Spezialausgabe von „neues“ zum Thema „Das Internet der Dinge“ produzieren: [Video 1/3] [Video 2/3] [Video 3/3]

Weitere interessante **Videos** zum Thema „Industrie 4.0“ bzw. „Future Internet“ finden Sie [hier].

Sollte sich die Webseite wider Erwarten nicht korrekt öffnen lassen, dann kopieren Sie den Link in die Zwischenablage und von dort direkt in die Adresszeile des Browsers!

Im Abschnitt „2.3 Anschauliche Beispiele zu den Dingen und Diensten“ geht es u.a. um Maschinen, die wissen, dass sie Maschinen sind.

1.1 Begrifflichkeit und Definition

>> Das Internet der Dinge (auch engl.: Internet of Things) bezeichnet die Verknüpfung eindeutig identifizierbarer physischer Objekte (Things) mit einer virtuellen Repräsentation in einer Internet-ähnlichen Struktur. Das Internet der Dinge besteht nicht mehr nur aus menschlichen Teilnehmern, sondern auch aus Dingen. Der Begriff geht zurück auf Kevin Ashton, der erstmals 1999 „Internet of Things“ verwendet hat[1]. Bekannt wurde das Internet der Dinge durch die Aktivitäten der Auto-ID Labs[2].

Die automatische Identifikation mittels RFID wird oft als Grundlage für das Internet der Dinge angesehen. Allerdings kann eine eindeutige Identifikation von Objekten auch mittels Strichcode oder 2D-Code erfolgen. Weitere Technologien wie Sensoren und Aktuatoren erweitern die Funktionalität um die Erfassung von Zuständen bzw. die Ausführung von Aktionen. Erweiterte Definitionen zum Internet der Dinge betonen die Zugehörigkeit zum zukünftigen Internet (auch engl.: Future Internet)[3] sowie die Abgrenzung von verwandten Forschungsthemen[4]. << (Quelle: Wikipedia)

1.2 Analyse und Bewertung

Da das Internet der Dinge die **Verknüpfung** eindeutig identifizierbarer physischer Objekte (Things) mit einer virtuellen Repräsentation in einer Internet-ähnlichen Struktur bezeichnet, könnte man in diesem Zusammenhang im Hinblick auf das **Internet und das Intranet** auch von **Hyperlink-Objekten** sprechen.

Praktisch bedeutet dies, dass sich ein **Hyperlink-Objekt** innerhalb eines Text-, Webdokuments oder einer Webseite nicht nur per Mausklick anklicken, sondern auch aufrufen, d.h. aktivieren und ausführen lässt.

1.2.1 Realität und Virtualität der Dinge

Beim Internet der Dinge muss man also zwischen den **Dingen der Realität** und denen der **Virtualität** unterscheiden.

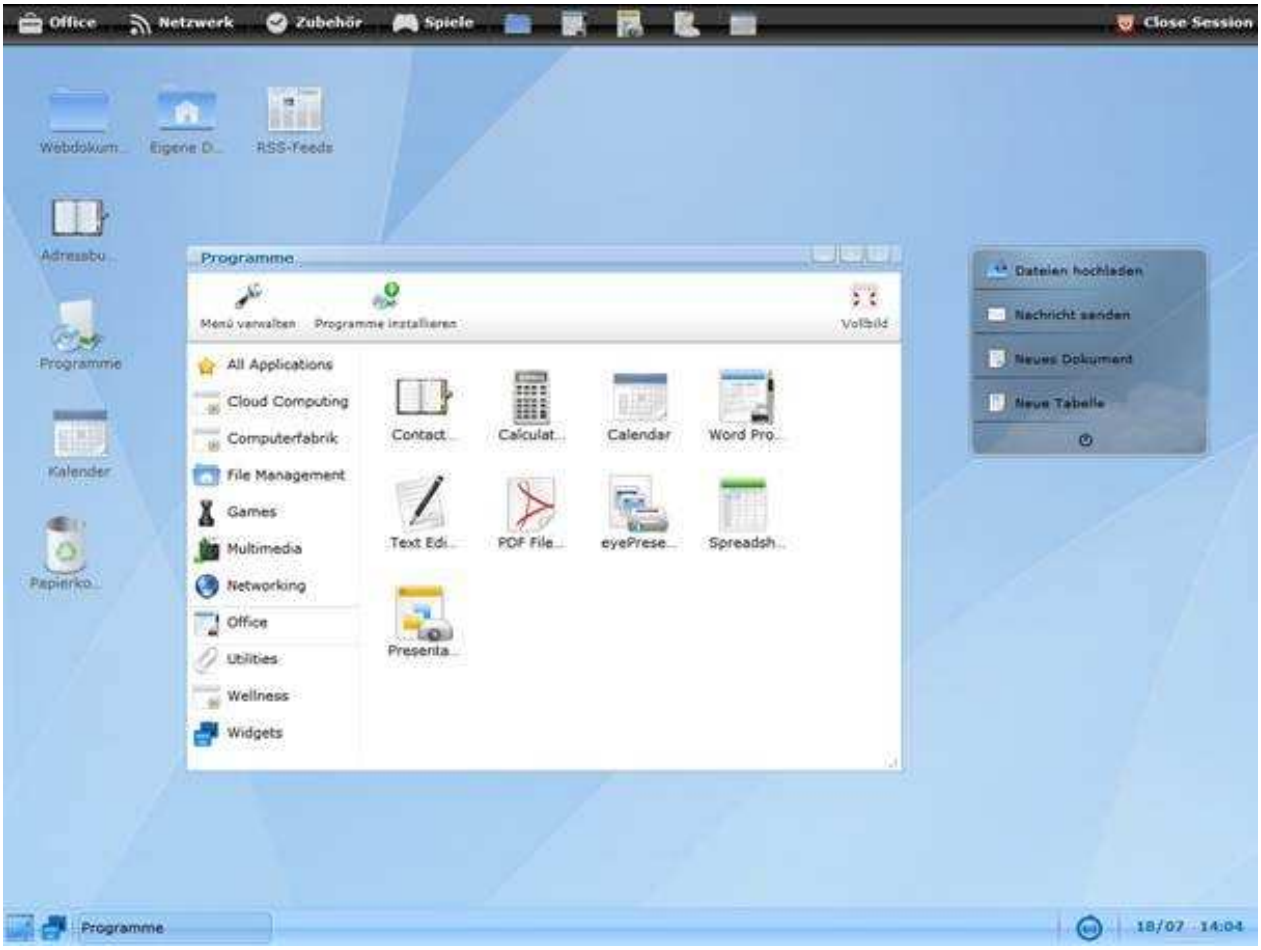
Demzufolge gibt es im Alltag, d.h. in der Gegenwart, in der Praxis, Dinge, die man nicht nur sehen und anfassen kann, sondern mit denen sich etwas anstellen und bewerkstelligen lässt, die für den Anwender einen Nutzwert haben und deshalb nützlich sind. Bei den real existierenden Dingen handelt es sich also um mehr oder weniger praktische Dinge in Form von Gegenständen oder Werkzeugen, die nicht nur nützlich sind, sondern deren Gebrauch und Anwendung sich selbst erschließt oder ausprobieren lässt. Es sind also Dinge, die sich intuitiv bedienen, anwenden oder ausprobieren lassen, sodass man auch ohne Bedienungsanleitung auskommt. Real existierende Dinge sind also meistens nützlich und selbsterklärend, sodass man sie sieht und anfassen kann.

Bei den **Dingen im Internet** handelt es sich hingegen um **virtuelle Dinge**, d.h. **Objekte** nebst **Objekteigenschaften**, die man nicht anfassen kann und nur sieht, wenn es von ihnen eine Abbildung in Form einer Skizze, einer Zeichnung, eines Bildes oder eines Ausdrucks auf Papier gibt. Inzwischen lassen sich aber auch virtuelle Dinge in Form bringen, d.h. von einem 3D-Drucker gegenständlich ausdrucken.

Obwohl sich **virtuelle Dinge** im Internet nicht anfassen lassen, so lassen sie sich trotzdem erfahren, indem man sie ausprobiert und anwendet. Demzufolge handelt es sich bei den virtuellen Dingen überwiegend um kleine Programme in Form von Apps, die auf einem (Web-) Server abgespeichert sind, von dort direkt aufgerufen oder aber auf den heimischen Personal Computer, das Smartphone oder den Tablet-PC heruntergeladen und mittels Setup installiert werden.

1.2.2 Werkzeuge der virtuellen Dinge

Da sich virtuelle Dinge stets auf dem Computer, dem Smartphone oder dem Tablet-PC abspielen, diese inzwischen stets mit dem Internet oder Intranet verbunden sind, braucht es auch einen **(Web-) Server**. Aber nicht nur! Damit sich das Internet der Dinge flexibler und individueller handhaben lässt, sollte es sich auch personalisieren lassen. Dazu gehört dann das „Private Cloud Computing“, d.h. die eigene **Datenwolke** auf dem USB-Memory-Stick oder der (USB-) Festplatte sowie einen „Web-Desktop“ nebst integrierter Standard-Anwendungen (Text, Tabelle, Präsentation) für den Browser. Dabei zählt „eyeOS“ zu den wenigen „Web-Desktop OS“-Betriebssystemen, das sich auf dem eigenen Apache-Webserver betreiben lässt. Da sich die zahlreichen und noch dazu kostenlosen Apps nur bei der älteren „eyeOS“-Version 1.9 aufspielen lassen, wird dies unter dem neuen Namen „oneye“ unter Federführung von Lars Knickrehm weiter entwickelt. Derzeit ist es in der Version 0.9.0 (1.11.0.1) verfügbar:

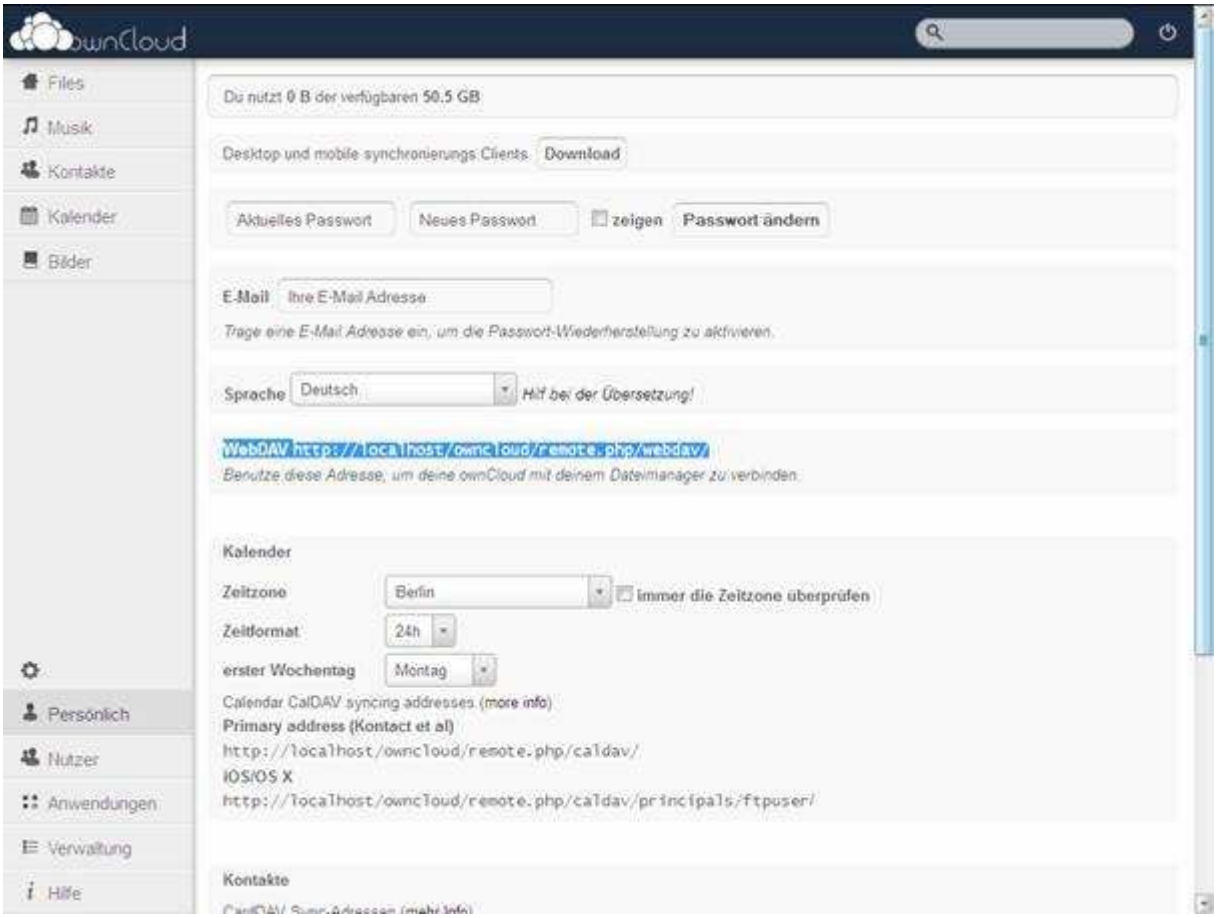


Da das „oneye“-Web-Desktop OS komplett in der Programmiersprache „PHP“ entwickelt wurde, lassen sich selbst entsprechende Apps in PHP programmieren und in den Desktop integrieren.

Weitere Einzelheiten zum „eyeOS“ bzw. „oneye“-Web-Desktop OS finden sich im Webverzeichnis \Programme2\usbwebserver\htdocs\private_cloud\index.html der **ISO-Datei „private_cloud.iso“**. Und zwar in den Abschnitten

10. „eyeOS“-Web-Desktop aufspielen und in Betrieb nehmen,
11. Ändern der „eyeOS“-Systemsprache von „English“ auf „Deutsch“,
12. Update auf eine neue „eyeOS“-Programmversion,
13. Das praktische Arbeiten in der „eyeOS“-Cloud,
14. Verschlüsselte Internetverbindung mittels „https“ und Port 443,
15. Umstieg vom USB-Webserver 8.2 auf XAMPP 1.7.4 USB,
16. Mit dem eigenen FTP-Server die Datenwolke anzapfen,
17. Ein Webdokument live in der Datenwolke publizieren.

Neben dem „oneye“-Web-Desktop Betriebssystem nebst integrierter Standard-Anwendungen (Text, Tabelle, Präsentation) bietet sich das „ownCloud“-Programm an, das als Open Source kostenlos zur Verfügung steht und ebenfalls mit PHP entwickelt wurde:



>> Die ownCloud ist eine Software-Suite, die einen ortsunabhängigen Speicherbereich für Daten zur Verfügung stellt. Das Projekt wurde im Januar 2010 vom KDE-Entwickler Frank Karlitschek ins Leben gerufen, um eine freie Alternative zu kommerziellen Cloud-Anbietern zu schaffen. Im Gegensatz zu kommerziellen Speicherdiensten kann ownCloud auf einem privaten Server ohne Zusatzkosten installiert werden. Somit können gerade bei sensiblen Daten die Bedenken gegenüber einer Datenweitergabe und der damit einhergehenden Abgabe der Kontrolle über die Daten zerstreut werden.

Als Grundlage setzt das Projekt auf PHP und einer angebundenen SQLite, MySQL oder PostgreSQL-Datenbank. Daher lässt sich ownCloud auf allen Plattformen betreiben, die diese Anforderungen erfüllen. Die ownCloud kann über eine Weboberfläche bedient werden und ist dadurch nicht an ein bestimmtes Betriebssystem gebunden. Aber auch native Programme, wie beispielsweise Dateimanager oder Groupwares, können die ownCloud über eine Schnittstelle ansprechen und Dateien und Daten lokal bereitstellen. << (Quelle: Wikipedia)

Als besonderes Sahnehäubchen verfügt „ownCloud“ auch noch über den integrierten „Sabre“-WebDAV-Server für das Dokumentenmanagement nebst Authoring und Versioning.

Weitere Einzelheiten zur „ownCloud“ finden sich im Webverzeichnis \Programme2\usbwebserver\htdocs\private_cloud\index.html der **ISO-Datei „private_cloud.iso“**. Und zwar in den Abschnitten

18. Die neue „ownCloud“-Datenwolke mit PHP,
19. Der WebDAV-Server in der „ownCloud“-Datenwolke,
20. Mit „NetDrive“ auf den „Sabre“-WebDAV-Server,
21. Mit der dynamischen IP-Weiterleitung in die „ownCloud“-Wolke,
22. „ownCloud“ aus dem Netz herunterladen,
23. „XAMPP“-Kombiserver aufspielen und starten,
24. „ownCloud“ aufspielen und starten,
25. Die ersten Schritte mit der „ownCloud“,
26. „ownCloud“-Benutzerverwaltung mit „SQLite“-Datenbank,
27. Die Datenwolke innerhalb der Datenwolke,
28. Mittels „NetDrive“ auf die „ownCloud“-Datenwolke zugreifen,
29. Über das Netz von außen auf alle „shared“-Verzeichnisse zugreifen,
30. Checkliste für die Fehlersuche bei der Online-Arbeit.

Abschließend soll noch die Frage beantwortet werden, wozu man denn überhaupt Werkzeuge wie z.B. eine „Private Cloud“, einen „Web-Desktop“ oder einen „WebDAV“-Server in der „ownCloud“ braucht.

Ganz einfach, damit das Arbeiten mit dem Internet der Dinge richtig Spaß macht. Schließlich geht es darum, dass wir das Internet der Dinge z.B. mit JavaScript, PHP und/oder Java selbst programmieren.

2 Das Internet der Dinge und Dienste

2.1 Begrifflichkeit und Definition

>> Das Ziel des Internet der Dinge ist es, die Informationslücke zwischen der realen und virtuellen Welt zu minimieren. Ein wichtiger Schritt zu diesem Ziel ist vor allem die **Standardisierung** der **Komponenten und Dienste** im **Internet der Dinge**. << (Quelle: Wikipedia)

Während es also beim Internet der Dinge um diverse **Objekte** der realen und virtuellen Welt geht, die mittels eines **Programms** veranschaulicht, nachgebildet, simuliert, berechnet oder manipuliert werden, geht es bei einem **Dienst** um >> eine technische, autarke Einheit, die zusammenhängende Funktionalitäten zu einem **Themenkomplex** bündelt und über eine klar definierte **Schnittstelle** zur Verfügung stellt.

Typische Beispiele sind hier z. B. Webservices, die Funktionalitäten für Dritte über das Inter- bzw. Intranet verfügbar machen, Netzwerkdienste, Systemdienste oder auch Telekommunikationsdienste.

Idealerweise sollte ein **Dienst** technische Funktion soweit abstrahieren, dass es nicht notwendig ist, die dahinter stehende Technik zu verstehen. Er sollte zudem auch genau definieren, welche fachlichen **Funktionen** er anbietet (z. B. in einem "Dienstvertrag").

Im Unterschied zu einem Application Programming Interface (API) kapselt ein **Dienst** üblicherweise die technische Repräsentation in fachlicher Funktionalität, ist in sich abgeschlossen und einem klar definierten **Aufgabenfeld** zugeordnet. Ein API stellt also eher technologisch und ein Dienst eher fachlich orientierte Funktionen zur Verfügung. << (Quelle: Wikipedia)

Da das Internet der Dinge und Dienste durchaus ziemlich komplex und auch kompliziert sein kann, ist es wichtig, dass ein **Dienst** und seine technischen, anwenderpraktischen **Funktionen und Methoden** verständlich, anschaulich und für den einfachen Anwender, d.h. Computerlaien nachvollziehbar sind. Schließlich geht es darum, dem Internet der Dinge und Dienste zu einer hohen Akzeptanz zu verhelfen und nicht darum, den Anwender hinsichtlich seiner teils sehr persönlichen Daten und Lebensgewohnheiten zu verunsichern. Demzufolge sollte der Anwender stets Eigentümer und Besitzer seiner Daten sein und diese in seiner „Private Cloud“ verwalten. Der Anwender sollte also stets in der Lage sein, sein persönliches Internet der Dinge und Dienste zu verstehen und zu beherrschen! –

2.2 Transparenz der Dinge und Dienste

Damit sich das persönliche Internet der Dinge und Dienste jederzeit beherrschen lässt und eine hohe Akzeptanz findet, müssen sich die Dinge und Dienste dem Anwender anschaulich und jederzeit nachvollziehbar mitteilen.

Aber weshalb braucht es überhaupt „The private Things of Internet“, das persönliche Internet der Dinge? „Private“ bedeutet in diesem Zusammenhang, dass es sich nicht um ein privates Internet der Dinge handelt, sondern dass dieses nicht öffentlich ist, die entsprechenden Programme und Daten, auf die sich „The private Things of Internet“ bezieht, nicht öffentlich, engl. „public“, sind, d.h. nicht für die Öffentlichkeit bestimmt sind. Und das ist gut so! Denn wer möchte schon, dass sein persönliches Internet der Dinge und Dienste auf einem Server von YouTube (Google) oder der Google-Suchmaschine landen?

„The private Things of Internet“ sollte sich also stets in einer „Private Cloud“, d.h. einer nicht öffentlichen „Private Cloud“ abspielen, sozusagen unter Ausschluss der Öffentlichkeit. Das ist technisch überhaupt kein Problem, schließlich lässt sich ein „Apache“-Webserver heutzutage bereits von jedem interessierten Anwender mit wenigen Mausklicks u.a. auch auf einem USB-Stick betreiben:

Dabei ist die Konfiguration des „Apache“-Servers so einfach wie nie zuvor, muss der Anwender lediglich das Webverzeichnis /root oder /htdocs sowie den Port 8080 oder 80 angeben (siehe Bild unten).

Wenn auf dem PC zuvor bereits der „**XAMPP**“**-Kombiserver** installiert wurde, dann ist das **Webverzeichnis /htdocs** bereits standardmäßig vergeben, so dass man die „**Default“-Einstellung {path}/root/** (oder ein anderes Webverzeichnis) des USB-Webservers verwenden muss:

Diesbezüglich muss noch erwähnt werden, dass der USB-Webserver des Holländers de Vries auch über einen integrierten MySQL-Datenbankserver verfügt. Dieser wird aber im vorliegenden Fall, d.h. bei den entsprechenden PHP-Programmen, nicht benötigt.

Darüber hinaus verhält es sich so, dass der „Apache“-Webserver in Verbindung mit der Skript-Programmsprache PHP selbst über einen integrierten „SQLite“-Datenbankserver verfügt, der sich z.B. beim Mozilla Firefox-Browser mittels der Erweiterung „SQLite Manager“ problemlos administrieren lässt. –

Wie man sieht, entwickelt sich die Computertechnik und damit das Internet weiterhin mit rasanter Geschwindigkeit. Demzufolge sollte sich der Anwender, der sich mit seinem PC, Notebook, Netbook oder Smartphone im Internet tummelt auch weiterbilden. Der sogenannte „Internet-Führerschein“, d.h. das Beherrschen des Browsers sowie eines E-Mail-Clients, reichen also bei Weitem nicht mehr aus, um auf dem neuesten Stand zu sein oder zu bleiben. Schließlich lassen sich einfache Webseiten schon seit geraumer Zeit praktisch mit jeder Textverarbeitung erstellen, und wie oben erwähnt, auf dem eigenen Webserver lokal oder global mittels dynamischer IP-Weiterleitung veröffentlichen. Am besten lassen sich eigene Webseiten und Webdokumente mit der kostenlosen Office-Software „OpenOffice“ oder „LibreOffice“ nebst deren Textverarbeitung gestalten, da diese nur HTML-Tags genieren und im Gegensatz zu Microsoft Office auf XML-Code verzichten, so dass die Webseiten weitestgehend übersichtlich und halbwegs gut lesbar bleiben. **[Video]**

Wenn es nun bei den „Private Things of Internet and Services“ um die Transparenz der Dinge und Dienste geht, dann bezieht sich das Ganze wider Erwarten nicht auf öffentliche Cloud-Server von Microsoft Skydrive oder Google Drive, sondern vielmehr auf unsere eigene „Private Cloud“, die auf unserem eigenen „Apache“-Webserver wie z.B. dem USB-Webserver läuft. Nur so lässt sich nämlich sicherstellen, dass wir als Eigentümer, Benutzer und Verwalter unserer Daten und Programme auch der Herrscher und Beherrscher des „Private Things of Internet and Services“ sind.

Das Problem bei Webseiten und Webdokumenten ist, dass sich diese im Internet praktisch von überall her aufrufen, anzeigen oder auch downloaden lassen. Das gilt

dann insbesondere auch für unsere „Private Cloud“ und unser „Private Things of Internet and Services“, sofern unser Webserver mittels einer dynamischen IP-Weiterleitung wie z.B. http:// oder besser https://meine-webadresse.office.net auch von außen über das Internet erreichbar ist. Um den unerwünschten oder unerlaubten Zugriff auf den Server nebst Benutzerdaten zu verhindern, bedient man sich einer entsprechenden

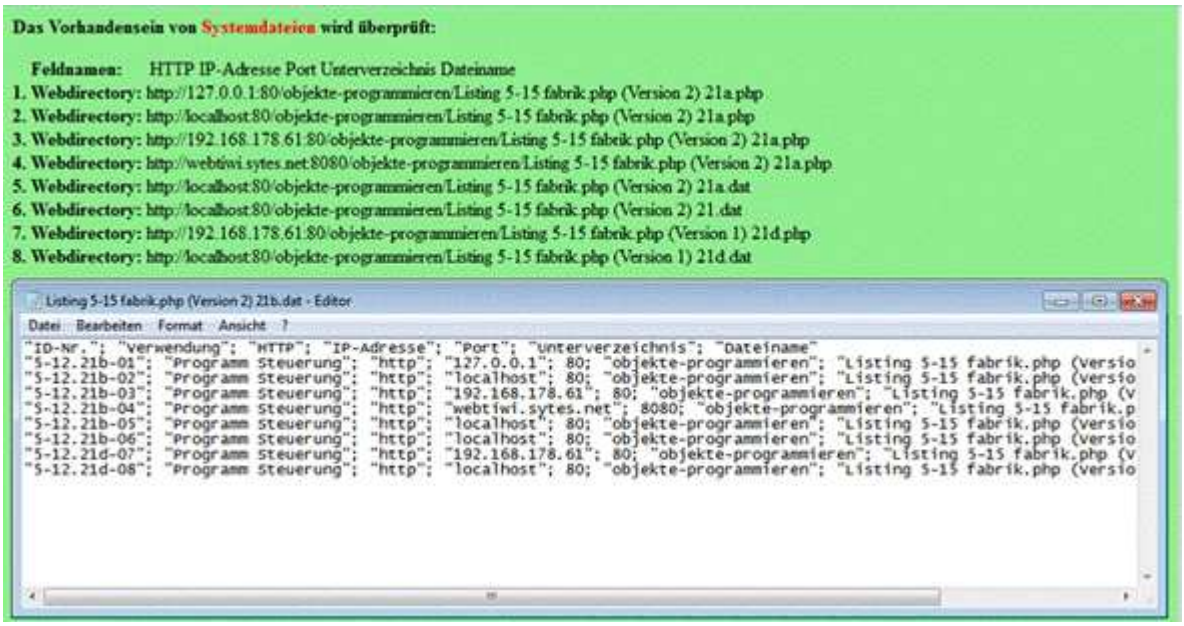
- **Benutzerverwaltung**

mittels der sich der Anwender authentifizieren muss, um Zugang zu seinen Daten zu erhalten. Darüber hinaus sollte man nur eine verschlüsselte

- **„https“-Verbindung**

verwenden bzw. den Zugriff von außen mittels dieser erlauben!

Da später beim „Private Things of Internet and Services“ auch mehrere Programme zum Einsatz kommen, die sich auch auf anderen, privaten Cloud-Servern befinden können, muss jedes Programm wissen, welche Dateien für das eigene, ordnungsgemäße Funktionieren erforderlich sind und wo sich diese befinden:



Dabei legt die

- **Programmdatendatei**

fest, welche Dateien und Programme sich wie und wo befinden, d.h. sich über bestimmte IP-Adressen, Ports und Verzeichnispfaden ansprechen lassen (siehe Bild oben).

Diesbezüglich verhält es sich normalerweise so, dass man beim Programmieren von Programmen weitere Programm- oder Dateiaufrufe statisch, d.h. mit feststehendem Dateinamen nebst Verzeichnispfad programmiert, so dass sich der Anwender um nichts zu kümmern braucht und auch nicht wirklich wissen muss, ob ein Programm weitere Programme oder Dateien aufruft oder nicht. Wenn es aber beim „Internet der Dinge und Dienste“ früher oder später um vernetzte Strukturen im Sinne eines „Smart Grid“ geht, sich die Dinge mittels der Dienste selbständig miteinander vernetzen und dabei auch noch miteinander autonom kommunizieren, dann braucht es eine entsprechende Transparenz, um das Zusammenwirken von Dingen und Diensten besser verstehen zu können. Schließlich soll ja der Anwender das „Private Internet of Things and Services“ verstehen, sich damit identifizieren, dieses selbständig beherrschen und dabei die Programme und Daten in seiner „Private Cloud“ selbst verwalten. Wenn er denn möchte und sich dazu berufen fühlt.

Damit später die autonome, strukturierte Kommunikation zwischen den Dingen und Diensten sowie den beteiligten Programmen in geordneten Bahnen verläuft, bedarf es dazu noch eines **Steuerprogramms**, das den **Datenaustausch** und die **Kommunikation** zwischen den **Programmen**, d.h. den Dingen und Diensten überwacht, koordiniert und steuert, so dass der Anwender stets den Überblick behält und sich von den Dingen und Diensten nicht überrollt oder drangsaliert fühlt. **[Video]**

2.3 Anschauliche Beispiele zu den Dingen und Diensten

2.3.1 Von Maschinen, die wissen, dass sie Maschinen sind

Kennen Sie den Unterschied zwischen dem ICE 1/ICE 2 und dem ICE 3? Sicherlich ist Ihnen der Unterschied schon aufgefallen, Sie haben ihn nur noch nicht bewusst wahrgenommen:

Bildmaterial von „Die schnellsten Züge von der Welt“.
Zum Vergrößern der Bilder bitte auf das jeweilige Bild klicken!

Beim Vergleich der beiden Bilder (siehe oben) fällt auf, dass der ICE 1 (linkes Bild) mit der Lokomotive als Antrieb recht massiv und wuchtig wirkt, während der ICE 3 (rechtes Bild) ein Fliegengewicht zu sein scheint: leicht, stromlinienförmig und ausgesprochen elegant.

Wo aber befindet sich beim ICE 3 die Lokomotive? Im Vertrauen: Sie können jetzt lange suchen und z.B. bei Wikipedia recherchieren, Sie werden keine Lokomotive beim ICE 3 finden, weil es nämlich keine gibt!

„Aber der ICE 3 muss doch irgendwie angetrieben werden!“ werden Sie sagen. Richtig! Der ICE 3 verfügt nicht nur über einen Antrieb, sondern über mehrere! Und zwar praktisch in jedem Waggon in Form von Einzelantrieben, die sich jeweils in einem Fahrgestell verbergen.

Da es also mehrere Einzelantriebe gibt, kann der ICE 3 wie eine U- oder S-Bahn äußerst schnell beschleunigen und mit hoher Geschwindigkeit Steigungen erklimmen. Demzufolge sind der ICE 3 und seine neuen Nachfolger als einzige in der Lage, die Gleise der größten Achterbahn der Welt auf der Strecke von Frankfurt am Main nach Köln zu befahren. Schneller als ca. 1,5 m/s sollte aber der ICE 3 nicht beschleunigen, da den Fahrgästen sonst kotzübel wird.

Aber weshalb erzähle ich Ihnen das Ganze? Stellen Sie sich vor, Sie sitzen mit Ihrem Smartphone im ICE 3 und fahren von Frankfurt a.M. nach Köln. Während einer länger andauernden Steigung von Frankfurt nach Limburg hätten Sie gern gewusst, wie viele Passagiere sich im Zug befinden und mit wie vielen Einzelantrieben der Zug im Moment unterwegs ist.

„Mit der richtigen App alles kein Problem!“ werden Sie sagen. Das Problem dabei ist aber, dass es eine solche App noch gar nicht gibt, weil das Future Internet, das Internet der Dinge und Dienste, noch nicht bei den Entwicklern angekommen ist und diesen der entsprechende Algorithmus auch noch nicht bekannt ist! Der Grund dafür ist der, dass das dazu erforderliche Wissen weder in den Schulbüchern noch in irgendwelchen wissenschaftlichen Abhandlungen vermittelt wird!

Trotzdem gibt es den Algorithmus schon, der es einer Maschine zu wissen erlaubt, dass sie eine Maschine ist! Wenn also der Elektromotor im Drehgestell des ICE 3 weiß, dass er ein Elektromotor ist, dann weiß er auch, dass es im ICE 3 noch weitere Einzelantriebe mit jeweils einem Elektromotor gibt. Vorausgesetzt natürlich, dass diese miteinander drahtlos z.B. per WLAN, UMTS, HSDPA, LTE oder Glasfaserkabel kommunizieren können und eine gemeinsame Sprache sprechen!

Aber woher weiß der Elektromotor, dass er sich im ICE 3 als Einzelantrieb in einem der Drehgestelle befindet? Ganz einfach, weil alle Elektromotoren über praktisch die gleichen GPS-Positionsdaten verfügen und sich mit der gleichen Geschwindigkeit auf der gleichen Strecke von Frankfurt a.M. nach Köln bewegen!

Kennen Sie noch den Werbespruch von Jägermeister „Alle für einen, einer für alle!“? Das ist sozusagen die Schwarmintelligenz der Elektromotoren im ICE 3. Aber nicht nur! Denn wer beschleunigen kann, sollte auch bremsen können! Was es also noch braucht, sind die Bremsen bzw. die Bremseinheit im zweiten Drehgestell des ICE-Waggons. Nur zur Info: auf jeder Achse eines Drehgestells befinden sich drei doppelte, innenbelüftete Scheibenbremsen. Insgesamt also sechs Scheibenbremsen, die jede für sich genommen deutlich größer ist als in einem Porsche. Diesbezüglich sollte

der ICE 3 immer nur so schnell fahren, wie er bremsen kann. Demzufolge müssen sich die Einzelantriebe untereinander verständigen und den einzelnen Brems-Drehgestellen mitteilen, welche elektrische Antriebsenergie z.B. bei einer Bergauffahrt in den Vortrieb investiert wurde. Für die Bremsleistung ist aber letztlich nur die im ICE 3 gespeicherte kinetische Energie (= Bewegungsenergie) maßgeblich, da bekanntlich ein entsprechender Teil der Antriebsenergie auch in die Überwindung der Steigung investiert werden muss. LKW-Fahrer wissen deshalb, dass man einen großen Berg auf der Autobahn nur genau so schnell hinabfahren darf, wie man bei der Steigung heraufgefahren ist. Anderenfalls wird bei der Talfahrt die im LKW gespeicherte kinetische Energie aufgrund des Transportgewichts bzw. der Masse des LKWs so groß, dass die Geschwindigkeit dramatisch zunimmt, der LKW mit mehr als 120 km/h zu Tal fährt, bei einer Notbremsung die Bremsen versagen würden!

Jetzt wissen Sie, dass die max. Antriebsleistung im Wesentlichen nicht die Bremsleistung und den damit verbundenen Bremsweg übersteigen darf! Demzufolge ist die maximal zur Verfügung stehende Bremsleistung das Maß aller Dinge bzw. das Maß für die maximale Antriebsleistung, wenn man einmal von entsprechenden Steigungen absieht für die ja auch Antriebsleistung bereitgestellt werden muss.

Bezüglich der ICE 3-Schwarmintelligenz bedeutet dies, dass es sich bei der Bremsleistung um die unabhängige Variable x und bei der Antriebsleistung um die abhängige Variable y handelt. Demzufolge ist die Antriebsleistung y stets eine Funktion der Bremsleistung x mit y = f(x). Zu der Antriebsleistung y müsste dann aber noch der entsprechende Leistungsanteil zur Überwindung der größten Steigung hinzugerechnet werden, damit der ICE 3 bei der größten Steigung nicht zum Stillstand kommt!

Bei der vernetzen Kommunikation zwischen den Antriebs- und Bremseinheiten in den jeweiligen Drehgestellen des ICE 3 gibt es also eine Hierarchie, d.h. wechselseitige Abhängigkeiten, die es z.B. bei der Programmierung zu berücksichtigen gilt!

Wann haben Sie das letzte Mal Skat gespielt? Wenn Sie Skat spielen können, dann wissen Sie, dass man stets einen dritten Mann, d.h. Mitspieler benötigt. Wenn der dritte Mann aber ein Anfänger ist und das Skatspielen nicht aus dem Effeff beherrscht, dann wird das Skatspiel für die Profis zu Qual. Wer aber ist beim intelligenten Zusammenspiel der Einzelantriebe und Bremseinheiten in den Drehgestellen beim ICE 3 der „dritte Mann“ ohne den nichts geht?

Kennen Sie das Sprichwort „Von nichts kommt nichts!“? Oder „Ohne Moos nichts los!“? Im vorliegenden Fall ist mit dem Moos nicht das Geld, sondern die elektrische Energie gemeint. Ohne Strom nichts los!

Der dritte Mann im Skatspiel ist also die zur Verfügung stehende elektrische Energie in Form des in der Oberleitung des ICE 3 transportierten Stroms I bzw. der bereitgestellten Ladungsmenge Q = I * t (bei Gleichstrom) bzw. = i(t) * t (bei Wechselstrom). Mit q(t) = C / u(t) folgt dann C = u(t) * i(t) * t für die Transportkapazität von elektrischer Ladung mittels der Oberleitung, gemessen in Megawattsekunde [MWs] = [kV * kA * s] = [MVAs] = [10^6 VAs].

Bezüglich einer Fahrdraht-Wechselspannung von u(t) = 15 kV (= fünfzehntausend Volt) und einer Antriebsleistung P = 8 MW (= acht Millionen Watt) pro Halbzug berechnet sich der zu transportierende Wechselstrom i(t) im Fahrdraht wie folgt: P = u(t) * i(t) -> i(t) = P / u(t) = 8 * 10^3 kVA / 15 kV ≈ 0,5 kA = 500 A (= Ampere für die Stromstärke). Wenn man bedenkt, dass eine Waschmaschine im Kochwaschgang so um die 10 A Strom aufnimmt, dann bräuchte man für die Antriebsleistung des ICE 3 rund 50 Waschmaschinen, die parallel geschaltet gleichzeitig betrieben werden. Wegen der geringeren Wechselspannung in den Haushalten von nur u(t) = 240 V müssten dann noch zusätzlich n = 15 10^3 V / 240 V ≈ 62 Waschmaschinen je Strang in Reihe hinzugeschaltet werden! Das wären dann zusammen 50 * 62 = 3.100 Waschmaschinen!

Spätestens jetzt wird deutlich, dass der in der Oberleitung bereit gestellten Energie eine maßgebliche Bedeutung zukommt und diese in die Berechnung der Schwarmintelligenz des ICE 3 mit einfließen sollte.

Vom sogenannten „Smart Grid“, d.h. dem intelligenten Stromnetz nebst Stromversorgung haben Sie sicherlich schon gehört. Auch vom intelligenten Stromzähler, der die elektrischen Verbraucher im Haushalt wie z.B. Waschmaschine, Wäschetrockner oder Spülmaschine, hauptsächlich nur dann einschaltet, wenn der Strom besonders günstig ist.

>> Ein intelligentes Stromnetz integriert sämtliche Akteure auf dem Strommarkt durch das Zusammenspiel von Erzeugung, Speicherung, Netzmanagement und Verbrauch in ein Gesamtsystem. Kraft- und Speicherwerke werden bereits heute so gesteuert dass stets nur so viel Strom produziert wird wie benötigt. Intelligente Stromnetze beziehen in diese Steuerung die Verbraucher sowie dezentrale kleine Energielieferanten und -speicherorte mit ein, sodass einerseits ein zeitlich und räumlich homogenerer Verbrauch entsteht und andererseits prinzipiell inhomogene Erzeuger (z.B. Windkraft) und Verbraucher (z.B. Beleuchtung) besser integriert werden können. << (Quelle: Wikipedia)

Wie man sieht, macht es also Sinn, die Energieversorgung der Bahn und des ICE 3 mit in das Future Internet, d.h. das Internet der Dinge und Dienste, einzubeziehen. Ausführliche und fachkompetente Berechnungen haben nämlich gezeigt, dass sich beim ICE 3 etliche der zur Verfügung stehenden Einzelantriebe je nach Fahrstrecke, Höhenprofil oder eventueller Fahrplanverspätung einsparen, d.h. während der Fahrt zu- oder abschalten lassen, ohne dass es zu etwaigen Leistungseinbußen oder -einschränkungen kommt.

Wie Sie vielleicht wissen, verhält es sich nämlich so, dass die Deutsche Bahn AG den Fahrstrom in den Oberleitungen ihres Schienennetzes überwiegend selbst produziert, weil die herkömmlichen Netzbetreiber und Netzversorger nicht den von der Bahn erforderlichen Netzwechselstrom der Frequenz f = 16 2/3 Hz = 16,7 Hz produzieren.

Die eigenen EVUs (= Energieversorgungsunternehmen) der Bahn produzieren also den Bahnstrom zum Selbstzweck, d.h. ohne daran etwas zu verdienen. Ferner handelt es sich bei den EVUs der Bahn um mehr oder weniger herkömmliche Kraftwerke, die

je nach Bauart eine entsprechende Strommenge in das jeweilige Oberleitungsnetz einspeisen und bis zum heutigen Tag über keine Schwarmintelligenz bzw. Smart Grid verfügen.

Wenn also auf der Rennstrecke von Frankfurt a.M. nach Köln in beide Fahrtrichtungen mehrere ICE 3-Züge pro Stunde in der Rush Hour mit bis zu 300 km/h zum Einsatz kommen und die mit der Einspeisung der dazu erforderlichen Energie beauftragten EVUs entlang der Bahnstrecke diese wider Erwarten nicht ausreichend zur Verfügung stellen können, dann kommt es unweigerlich zu Fahrplanverspätungen, weil die jeweiligen ICE 3-Züge aufgrund der Energieknappheit nicht so stark beschleunigen und schnell fahren können, wie es vom Fahrplan her vorgesehen ist. Damit aber nicht alle auf der Strecke befindlichen ICE 3-Züge der Energieknappheit und den damit verbundenen Fahrplanverspätungen ausgesetzt sind, müsste man den einen oder anderen ICE 3 während der Rush Hour vom Gleis nehmen und im Bahnhof pausieren lassen oder aber das jeweilige Energiemanagement intelligenter gestalten. Demzufolge müsste man den Energiebedarf nebst -verbrauch in der Rush Hour nur intelligent anpassen und verteilen, damit alle an der Oberleitung hängenden ICE 3-Züge gleichmäßig mit Energie versorgt werden. Dies könnte man dann z.B. dadurch bewerkstelligen, indem man sicherstellt, dass nicht alle ICE 3-Züge gleichzeitig zu einem bestimmten Zeitpunkt aus den Bahnhöfen ausfahren und beschleunigen. Das wiederum würde aber ein intelligentes Energiemanagementsystem wie beim Smard Grid voraussetzen.

Wie man unschwer sieht, macht es also Sinn, dass eine Maschine weiß, dass sie eine Maschine ist, was für Eigenschaften sie hat, wofür sie eingesetzt wird und mit welchen anderen Maschinen, Aggregaten, Gerätschaften usw. sie zusammenarbeiten kann. Bezüglich des Future Internets und der Apps bedeutet dies, dass sich der autorisierte Fachmann zwecks Kontrolle, Fehlersuche oder Wartung jederzeit auf eine Antriebs- oder Bremseinheit, das System der Energieeinspeisung sowie der Leistungs-, Steuer- und Regelelektronik schalten können sollte. Und zwar unabhängig von dem internen Steuerungssystem, d.h. parallel, autonom und redundant.

Last but not least besteht der Hauptvorteil vernetzter Systeme u.a. auch darin, dass diese aufgrund der autonomen und dezentralen Organisation selbständig potentielle Fehlerquellen und -ursachen erkennen, diese bereits vor der eigentlichen Störung sozusagen selbstheilend durch entsprechende Neuorganisation beheben, sodass es zukünftig zu keinen Totalausfällen mehr kommt! Wenn sich nämlich die Einzelantriebe eines ICE 3 untereinander finden und miteinander kommunizieren, dann können sie sich selbständig elektrisch organisieren und die Leistungsanpassung zu Zwecken der Energieeinsparung und des Einhaltens des Fahrplans entsprechend optimieren! Darüber hinaus wäre es außerdem möglich, zwei ICE 3-Halbzüge kontaktlos, d.h. ohne mechanische Kupplung mit konstantem Abstand berührungslos auf der Strecke z.B. von Frankfurt a.M. nach Köln fahren zu lassen. Je nach zu- oder abnehmendem Passagieraufkommen im ICE 3 könnte man dann während der Fahrt weitere Halbzüge hinzufügen oder wegnehmen. Bei voller Fahrt versteht sich und ohne Zwischenhalt. [Video]

Letztendlich verhält es sich nämlich so, dass zentralisierte Systeme früher oder später äußerst komplex, unüberschaubar und unbeherrschbar werden. Siehe dazu auch „Komplizierte Technik: Bahn rechnet beim ICE 3 mit weiteren Verzögerungen“. (Quelle: Spiegel Online)

2.3.2 A Tännchen please: The Future Internet is me!

Ich bin unterwegs. Auf der Autobahn nach Hannover. Zur CeBit 2020. Mit meinem Elektroauto, mit meinem 7 Zoll großen Tablet-PC und mit LTE (Long Term Evolution) im Internet, das ist bis zu 14x schneller als das alte UMTS/HSDPA.

Während das Auto wie von Geisterhand gesteuert ohne mein Zutun auf der Autobahn in der Kolonne fährt und leise vor sich hin surrt, befinde ich mich bereits auf dem virtuellen Messerundgang, um mich über die Aussteller und deren Angebote zu informieren. Ab und an gebe ich ein paar für mich wichtige Informationen und Fakten in meine kleine Datenbank ein, die als Interessen- und Datenaustauschprofil für die CeBit abgespeichert wird.

Um den automatisch richtigen Sicherheitsabstand zum Vordermann kümmern sich zwei redundant arbeitende smarte, Strom sparende Rechenknechte mit jeweils acht Prozessorkernen wie man sie auch massenhaft bei den Smartphones verwendet. Die Reisegeschwindigkeit liegt bei angenehmen und Ressourcen sparenden 115 Km/h. Kilometern pro Stunde, nicht Stundenkilometer, wie so viele Journalisten und TV-Moderatoren immer wieder sagen, weil sie von Physik und Technik keine Ahnung haben. Kilometer pro Stunde. Das ist wichtig. Drehmoment M. Auch das ist wichtig. Oder die Masse m meines Elektroautos. Die Gewichtskraft Fg berechnet sich aus dem Produkt von Masse m x (Erd-) Beschleunigung g. Ist auch wichtig. Vor allem, wenn es in den Kasseler Bergen rauf und runter geht. An der Geschwindigkeit ändert das aber nichts. Master/Slave heißt das Zauberwort. Meister und Knecht. Mein E-Mobil ist im Moment der Knecht. Muss gehorchen, sich in der Kolonne nach dem langsamsten und schwächsten Fahrzeug richten. Damit wir in der Kolonne mit konstanter Geschwindigkeit und gleichem Abstand unterwegs sind. Zwischen Göttingen und Hannover könnte es sein, dass mein Elektroauto zum Master wird, so dass sich alle anderen Fahrzeuge in der Kolonne nach meinem Wägelchen richten müssen, da dieses nicht schneller als 120 km/h fährt. Kilometer pro Stunde! Zwischen Göttingen und Hannover ist es flach. Da sieht man montags schon, wer sonntags zu Besuch kommt. Ab Hannover spricht man Hochdeutsch. Mal sehen, ob die CeBit-Besucher alle über einen spitzen Stein stolpern.

Mein Tablet-PC ist über Bluetooth kabellos mit dem Motormanagement verbunden. Gibt Impulse. Empfängt Impulse. Kleine Datenpakete. Zeigt mir an, ob mein Elektroauto Master oder Slave ist. Es sind 47 Fahrzeuge in der Kolonne unterwegs. Das sind (47 - 1) x 60 m Sicherheitsabstand + 47 x 2,50 m durchschnittliche Fahrzeuglänge = 2,8775 Km Gesamtlänge. Alle Fahrzeuge fahren mit gleichem Abstand. Lenken muss ich nicht mehr. Gasgeben auch nicht. Jedenfalls solange ich auf der Autobahn in der Computer gesteuerten Fahrzeugkolonne fahre. Das entspannt.

Mein persönlicher Gesundheitscoach meldet sich auf dem Tablet-PC. Ich soll trinken. 250 Milliliter. 250 ml. 250 Tausendstel Liter! Na ja. Kein Problem. In 35 Minuten bei Kilometer 237 werde ich mich von der Kolonne verabschieden. Die Toilette eines Rasthofes ansteuern. Sagt mein Gesundheitscoach. Auf Sächsisch. Das ist nicht so langweilig. Das habe ich so eingestellt. Bei wichtigen Infos sagt das Programm „A Tännchen please!“. Sie kennen ja den Witz bei dem ein Sachse mitten in London auf dem Weihnachtsmarkt einen Tannenbaum kaufen will und gefragt wird, was er denn kaufen möchte: „A Tännchen please!“.

Ich bin auf der CeBit am Stadtrand von Hannover angekommen. Synchronisiere mein Smartphone mit dem Tablet-PC und starte meinen Personal Information Manager. Weiter geht’s mit dem Smartphone. Das sammelt. Daten. Multimediale Visitenkarten. Präsentationen. Texte, Bilder, Audio und Videoclips. Von der CeBit. Von den Ausstellern der Halle 4 durch die ich gerade gehe. Plötzlich gibt es einen Alarm. Mein Personal Information Manager meldet, dass es in 10 Minuten am Messestand 1734 eine für mich wichtige Pressekonferenz gibt. Ich schaue mir die automatisch heruntergeladene Pressemappe an und ziehe es vor, an der Pressekonferenz nur virtuell teilzunehmen. Die Videoaufzeichnung kann ich mir ja dann auf dem Rückweg von der CeBit auf dem Tablet-PC anschauen.

Ich besuche jetzt schon die vierte Messehalle und meine Füße machen sich bemerkbar. Aber qualmen tun sie noch nicht. Das ist auch meinem Gesundheitscoach nicht unbemerkt geblieben. Er schlägt vor, dass ich eine Magnesium-Brausetablette zu mir nehme, damit ich tags drauf in der Nacht keine Wadenkrämpfe bekomme. „A Tännchen please!“ quakt es plötzlich aus meinem Smartphone. Der Datenbankspeicher ist voll. Sagt mein Personal Information Manager. Ich schaue mir die Trefferquote und die Relevanz der eingesammelten Ausstellerinfos an. Lösche alle Infos mit einer Relevanz unter 40 %. Das schafft sofort Speicherplatz und spart mir später eine Menge Zeit bei der Auswertung.

„A Tännchen please!“ quakt es schon wieder aus dem Smartphone. Eine wichtige Geschäftsbesprechung hat stattgefunden. Virtuell versteht sich. Messenger-Programme und Avatare kommunizieren nämlich inzwischen automatisch miteinander. Auch wenn man offline ist. Aber nur in Form eines Text-Chats. Etwas holprig ist das Ganze auch noch. Aber immerhin! Möglich machen das mein hinterlegtes Interessenprofil für private und geschäftliche Dinge sowie der von mir abgespeicherte und erweiterbare Frage-Antwort-Katalog in Form eines editierbaren Binärbaums und das entsprechende Kommunikationsprogramm mit Fuzzylogik und Semantik-Programmierung.

Zu allem Überfluss meldet sich jetzt auch noch mein PWPC, mein „Personal Working Process Coach“ und teilt mir mit, dass ich zu einem Weiterbildungslehrgang ‚Semantisches Programmieren für Einsteiger’ angemeldet sei. Das erfolgreiche Absolvieren des Lehrgangs sei Voraussetzung zur Teilnahme am nächsten Projekt ‚Augmented Reality with Real Time Communication’ in China.

Es knurrt und krummelt. Nein nicht auf dem Smartphone, sondern in meinem Magen. Der kleine Hunger zwischendurch meldet sich. Ganz ohne „A Tännchen please!“ Das freut mich. Endlich mal ein Gefühl, das der Gesundheitscoach auf meinem

Smartphone nicht mitbekommen hat. Liegt wohl daran, dass es noch keinen Biosensor für die Magensäfte und den Verdauungstrakt gibt. Egal. Ich starte die Indoor-Navi-Funktion, die in Kombination mit GPS und WLAN arbeitet, und lasse mir in der Halle 4 den nächstgelegenen Imbiss anzeigen. Ich entscheide mich für einen Hotdog und bezahle bargeldlos mit meinem Smartphone, das standardmäßig mit der Near Field Communication (NFC) ausgerüstet ist. Kurz darauf meldet sich doch noch mein Gesundheitscoach und teilt mir mit, dass ich wegen der fehlenden Ballast- und Nährstoffe im Hotdog meine bisher regelmäßige Verdauung gefährde. Als Ausgleich müsse ich deshalb mehr trinken und abends einen Verdauungsspaziergang machen. Na super! Das hat man davon, wenn Programme Daten miteinander austauschen und auf der papierlosen Kassenquittung „Hotdog“ draufsteht.

Nachdem ich meinen Hotdog aufgegessen hatte und mir noch eine Tasse Kaffee bestellte, meldete sich mein Lese- und Informations-Coach zu auf dem Smartphone-Display: „Ihr tägliches Informationsdefizit beläuft sich auf 11 Artikel ‚IT-Tagelöhner’ (Relevanz 42 %), sieben Artikel ‚Länger gesund arbeiten’ (67 % Relevanz), drei Artikel zum Thema ‚Mindfulness Coaching’ (73 % Relevanz) und ein Artikel ‚Wer früher stirbt ist länger tot!’ aus der Reihe ‚Der Computer unterstützte Coaching-Wahnsinn’ (Relevanz 97 %).“

Nach kurzer Überlegung entscheide ich mich dazu früher zu sterben, um dann länger tot zu sein. Natürlich nur virtuell! Wie das geht? Ganz einfach! Man muss sich nur von einem IT-Spezialisten den elektronischen Personalausweis (Perso) klonen lassen, ein anderes Passbild einspielen und einen anderen Namen nebst Geburtsdatum eintragen lassen. Außerdem lässt man sich in den Perso eintragen, dass man schon seit zwanzig Jahren arbeitslos ist, vor Kurzem einen Offenbarungseid geleistet hat und mit dem Unterhalt für vier unehelichen Kinder schon seit Jahren im Rückstand ist. Und schon lebt es sich viel gemütlicher nach dem Motto: „Ist der Ruf erst ruiniert, lebt’s sich gänzlich ungeniert!“.

Was es also unbedingt braucht, ist eine zweite Identität, ein Alter Ego, um alles das machen zu können, was richtig Spaß macht und nichts mit Arbeit zu tun hat. Das ist dann zwar ungesünder und verkürzt die Lebenserwartung, macht aber viel mehr Spaß als gesund zu leben und sich der Diktatur der elektronischen Helfer zu unterwerfen.

„A Tännchen please!“ tönt es plötzlich wieder unerwartet aus dem Smartphone, „Ihr Flug zur einsamen Südseeinsel wurde reserviert! Bitte laden Sie umgehend Ihr bargeldloses Konto auf!“ Na super! -

2.3.3 Realität und Virtualität der Dinge

„Vertrauen ist gut, Kontrolle besser!“ hat Lenin sinngemäß gesagt. Doch wer etwas kontrollieren will, muss erst einmal wissen, was kontrolliert werden soll und wie. Der Kontrolleur muss also entsprechend kompetent sein, d.h. über den zu kontrollierenden Sachverhalt entsprechend informiert sein und Bescheid wissen.

Wenn Sie z.B. einen Urlaub planen und eines Tages die Urlaubsreise antreten, d.h. die Koffer packen, dann gehen Sie in Gedanken noch einmal alles durch und prüfen

anhand einer gedanklichen Checkliste, ob Sie auch ja alles eingepackt und nichts vergessen haben. Dabei benutzen Sie vielleicht auch eine aus dem Internet heruntergeladene Checkliste, die Sie konsequent Punkt für Punkt abarbeiten. Dabei stellt die Checkliste ein Werkzeug dar, das sich einfach beherrschen und anwenden lässt. -

Stellen Sie sich vor, Sie sind stolzer Fabrikbesitzer, d.h. Herrscher und Kontrolleur über Kapital, Personal, Produktionsmittel, Fertigungsstraßen, Fertigungsroboter und den Leitstand für die Steuerung der Produktion diverser Computer. Der Knackpunkt dabei: Die Computerfabrik als Ganzes gibt es nur virtuell, d.h. in der eigenen Datenwolke, der „Private Cloud“. Außerdem befindet sich die Computerfabrik weltweit an insgesamt vier verschiedenen Standorten, die sich u.a. in Europa, den USA, China und Südkorea befinden. Dabei befindet sich die **Konzernzentrale** mit der Produktionssteuerung und -kontrolle in Deutschland, die Produktionsplanung in den USA, die Produktionsfertigung in China und die Auslieferung der Produktion (Vertrieb) in Südkorea.

Entsprechend dieser vier Standorte gibt es dann auch vier verschiedene **Programme** mit spezifischer Funktionalität je nach Standort, Aufgabe und Funktion vor Ort:

- **Konzernzentrale** mit der **Produktionssteuerung**, d.h. dem Programm Listing 5-15 fabrik.php (Version 2) 21d.php
- **Produktionsplanung** in den USA, d.h. dem Programm Listing 5-15 fabrik.php (Version 2) 21a.php
- **Produktionsfertigung** in China, d.h. dem Programm Listing 5-15 fabrik.php (Version 2) 21b.php
- **Auslieferung der Produktion** (Vertrieb) in Südkorea, d.h. dem Programm Listing 5-15 fabrik.php (Version 2) 21c.php

Diesbezüglich kann man sich leicht vorstellen, dass die **Konzernzentrale** mit dem Programm der **Produktionssteuerung** gern alle Fäden in der Hand hält, um jederzeit den Überblick zu haben und zu jederzeit an jedem Standort der ausgelagerten **Filialbetriebe** eingreifen zu können.

In der Tat laufen im Programm der **Produktionssteuerung** alle Fäden zusammen, müssen sich alle anderen Programme bei diesem anmelden. Auf diese Weise soll die **Kommunikation** aller Programme, d.h. die der ausgelagerten Filialen, kontrolliert, gesteuert und überwacht werden, so dass die **Konzernzentrale** stets über den **aktuellen Status** der **Computerproduktion**, bestehend aus Planung, Fertigung und Auslieferung (Vertrieb), informiert ist.

Die **Kontrolle** der **Konzernzentrale** über alle Aktivitäten ist aber nicht wirklich der Grund für den Einsatz des Programms der **Produktionssteuerung**, sondern nur ein nützlicher Nebeneffekt. Mit der **Produktionssteuerung** soll vielmehr der **Datenfluss, Datenaustausch und die Kommunikation** zwischen und mit den anderen Programmen geregelt werden, damit sich die Programme stets in der beabsichtigen

und gewünschten Weise miteinander austauschen und verständigen. Schließlich gibt es bei den Programmen

- **Produktionssteuerung**,
  Produktionsplanung,
  Produktionsfertigung und
  Auslieferung der Produktion (Vertrieb)

nicht nur unterschiedliche Aufgaben und Funktionen, sondern insbesondere auch **wechselseitige Abhängigkeiten**.

Wie man leicht nachvollziehen kann, lassen sich PCs erst dann ausliefern, wenn sie zuvor produziert wurden. Um aber PCs produzieren zu können, muss es eine entsprechende Fertigung geben bzw. muss die bereits vorhandene Fertigungsstraße entsprechend umgerüstet werden. Die Fertigung von PCs setzt wiederum voraus, dass die dazu erforderlichen Ressourcen, d.h. die Produktionsmittel zuvor geplant und organisiert wurden. Demzufolge ergibt sich also nachfolgende, funktionale Abhängigkeit zwischen den vier Konzernabteilungen:

> Produktionssteuerung → Produktionsplanung → Produktionsfertigung → Auslieferung der Produktion (Vertrieb)

Wie man unschwer sieht, liegt der funktionalen Abhängigkeit eine serielle Abfolge der PC-Produktion zugrunde. Dies muss aber nicht zwingend der Fall sein! Schließlich lassen sich bereits gefertigte PCs auch parallel zur laufenden Produktion neuer PC-Typen vermarkten, so dass sich die beiden Abteilungen Produktionsfertigung und Auslieferung der Produktion (Vertrieb) teilweise autonom voneinander betreiben lassen:

> Produktionssteuerung
>
> → Produktionsplanung
>
> → Produktionsfertigung
>
> → Auslieferung der Produktion (Vertrieb)

Wenn also die beiden Abteilungen Produktionsfertigung und Auslieferung der Produktion (Vertrieb) teilweise voneinander unabhängig sind, d.h. die bereits produzierten PCs sofort verkauft werden, während parallel dazu weitere produziert werden, dann müssen beide Abteilungen gleichzeitig bzw. synchron mit den einheitlichen Produktionsdaten versorgt werden. Dabei sollte der **synchrone Datenaustausch** möglichst automatisch erfolgen. Damit sich die **Datenbestände** vor, während oder nach dem Datenaustausch nicht manipulieren lassen, darf es seitens der Mitarbeiter keine zugänglichen **Datendateien** oder **Datenbanken** geben! Schließlich könnte eine betrügerische **Datenmanipulation** der **Fertigungsstückzahlen** gegenüber den produzierten, tatsächlich gelieferten und verkauften PCs dazu führen, dass z.B. Mitarbeiter in der **Qualitätssicherung** der Fertigung einzelne PCs quasi als defekt

ausmustern und an der Wertschöpfungskette vorbei auf dem örtlichen Flohmarkt gewinnbringend veräußern.

Wie aber lassen sich **Produktionsdaten** von einem Programm zum nächsten ohne **Dateitransfer**, also ohne das Vorhandensein einer **Datendatei** oder einer **Datenbank** übermitteln? Ganz einfach! Wo nichts ist, d.h. wo es keine Datendatei oder Datenbank gibt, muss bzw. lässt sich auch nichts übertragen!

Aber es gibt doch Produktionsdaten, die von einer Abteilung, d.h. von einem Programm zum nächsten übertragen werden müssen, werden Sie sagen. Richtig, es gibt Produktionsdaten, aber wenn sich diese zentral auf ein- und demselben (Apache-) **Webserver** befinden, dann muss man diese nicht eigens von einem Programm zum nächsten übertragen, sondern einfach nur mittels eines erlaubten Zugriffs verfügbar machen! Physikalisch bedeutet dies, dass die **Produktionsdaten** nur ein einziges Mal von der Produktionsdatendatei Listing 5-15 fabrik.php (Version 2) 21.dat aus eingelesen und dann im **Arbeitsspeicher** des **Webserver-PCs** abgespeichert werden. Wenn dann die beiden anderen Programme Produktionsfertigung und Auslieferung der Produktion (Vertrieb) auf den gleichen, d.h. gemeinsamen Datenbestand zugreifen wollen, dann wird der **Arbeitsspeicher** des **Webserver-PCs** einfach nur noch weitere Male ausgelesen. Vorausgesetzt, dass die beiden zuvor genannten Programme dazu entsprechend authorisiert wurden. Möglich wird das Ganze nur deshalb, weil die Computerfabrik im Internet ausschließlich **webbasiert** arbeitet, so dass sich alle Programme und Daten auf dem (Apache-) Webserver befinden. Dabei spielt es natürlich auch noch eine Rolle mit welcher **Programmiersprache** die Computerfabrik im Internet programmiert wurde. Im vorliegenden Fall also mit der Skript-Programmiersprache PHP bei der das **Programm** in Form einer **Skriptdatei** auf dem (Apache-) Webserver gespeichert und beim Programmaufruf mittels Webseite gestartet, interpretiert und ausgeführt wird. Auf der Empfängerseite greift dann der Anwender mittels eines **Webclients**, d.h. mit einem herkömmlichen **Webbrowser** auf das entsprechende Programm nebst Daten zu. Damit sich die vier obengenannten PHP-Programme datenmäßig untereinander austauschen können, müssen diese gleich zu Beginn, d.h. noch vor dem eigentlichen Programm selbst, eine sogenannte **Session** starten.

Wegen der wechselseitigen **Funktionsabhängigkeit** der **Programme**

Produktionssteuerung

→ Produktionsplanung

→ Produktionsfertigung

→ Auslieferung der Produktion (Vertrieb)

muss man im Browser als erstes das **Master-Programm** Produktionsplanung Listing 5-15 fabrik.php (Version 2) 21a.php starten, damit die Produktionsdatendatei Listing 5-15 fabrik.php (Version 2) 21.dat mit den zu produzierenden **PC-Typen** eingelesen und sofort angezeigt wird. Das strikte Einhalten der **Programmstart-Reihenfolge** ist

also immer dann erforderlich, wenn die beiden anderen, funktionsabhängigen **Slave-Programme** Produktionsfertigung und Auslieferung der Produktion (Vertrieb) ebenfalls auf die zu produzierenden **PC-Typen** der Produktionsdatendatei per **Session** zugreifen sollen.

Aber wie im richtigen Leben gibt es keine Regel ohne Ausnahme. So auch im vorliegenden Fall. Selbstverständlich lassen sich alle vier Programme (siehe oben) in beliebiger Reihenfolge starten. Niemand hindert einen daran. Wenn dabei aber wider Erwarten keine **Produktionsdaten** der zu produzierenden **PC-Typen** angezeigt werden, dann liegt es daran, dass man das **Master-Programm** Produktionsplanung Listing 5-15 fabrik.php (Version 2) 21a.php zuvor noch nicht gestartet hat! Aber das lässt sich natürlich jederzeit nachholen! -

In diesem Zusammenhang nimmt das **Programm** Produktionssteuerung Listing 5-15 fabrik.php (Version 2) 21d.php quasi eine **Sonderstellung** ein, da es weder vom **Typ „Master“** noch vom **Type „Slave“** und demzufolge völlig unabhängig ist. Wie bereits weiter oben beschrieben, dient es nämlich nur dazu, den betriebsmäßigen **Status der Programme**, d.h. der Computerfabrik im Internet anzuzeigen und zu überwachen:

Später soll das Programm der **Produktionssteuerung** auch noch weitere **Steuer- und Kontrollfunktionen** zwischen den anderen Programmen übernehmen und dafür sorgen, dass sich z.B. die beiden **Slave-Programme** Produktionsfertigung und Auslieferung der Produktion (Vertrieb) nur starten lassen, wenn zuvor das **Master-Programm** Produktionsplanung gestartet wurde! -

Außerdem wird man früher oder später sicherlich nicht umhinkommen, die eine oder andere **Datenbank** nebst **MySQL-** oder **SQlite-Server** (integraler Bestandteil von PHP) nebst eines WebDAV-Servers für das Dokumentenmanagement zu verwenden.

In diesem Zusammenhang sei abschließend nochmals darin erinnert, weshalb wir den ganzen Steuerungs- und Verwaltungsaufwand betreiben: nämlich um den **Anwender** in die Lage zu versetzen, alleiniger **Herrscher** und **Beherrscher** seiner **Programme und Daten** auf seinem eigenen **Webserver** in der eigenen **„Private Cloud“** zu sein! „The Private Things and Services of Internet“, d.h. das nichtöffentliche Internet der Dinge und Dienste wird sich nämlich nur dann auf breiter Front durchsetzen und behaupten, wenn der Anwender in die Lage versetzt wird, selbst alles im Griff zu haben. Vorausgesetzt natürlich, dass der Anwender entsprechend interessiert und ambitioniert ist und wissen möchte, wie seine Dinge und Dienste wirklich funktionieren. **[Video]**

Inhaltsverzeichnis