

Internet der Dinge (IoT) Orchestrierung

Freitag, 4. März 2016, 15:00 bis 17:00
Marcel mc-b Bernet, marcel.bernet@ch-open.ch

Agenda

- Internet der Dinge (IoT)
- Web of Things / Physical Mashups / IoT Orchestrierung
- Kitchen Helper Project
 - Software-Metriken (Thomas Fehlmann)
 - Implementierung

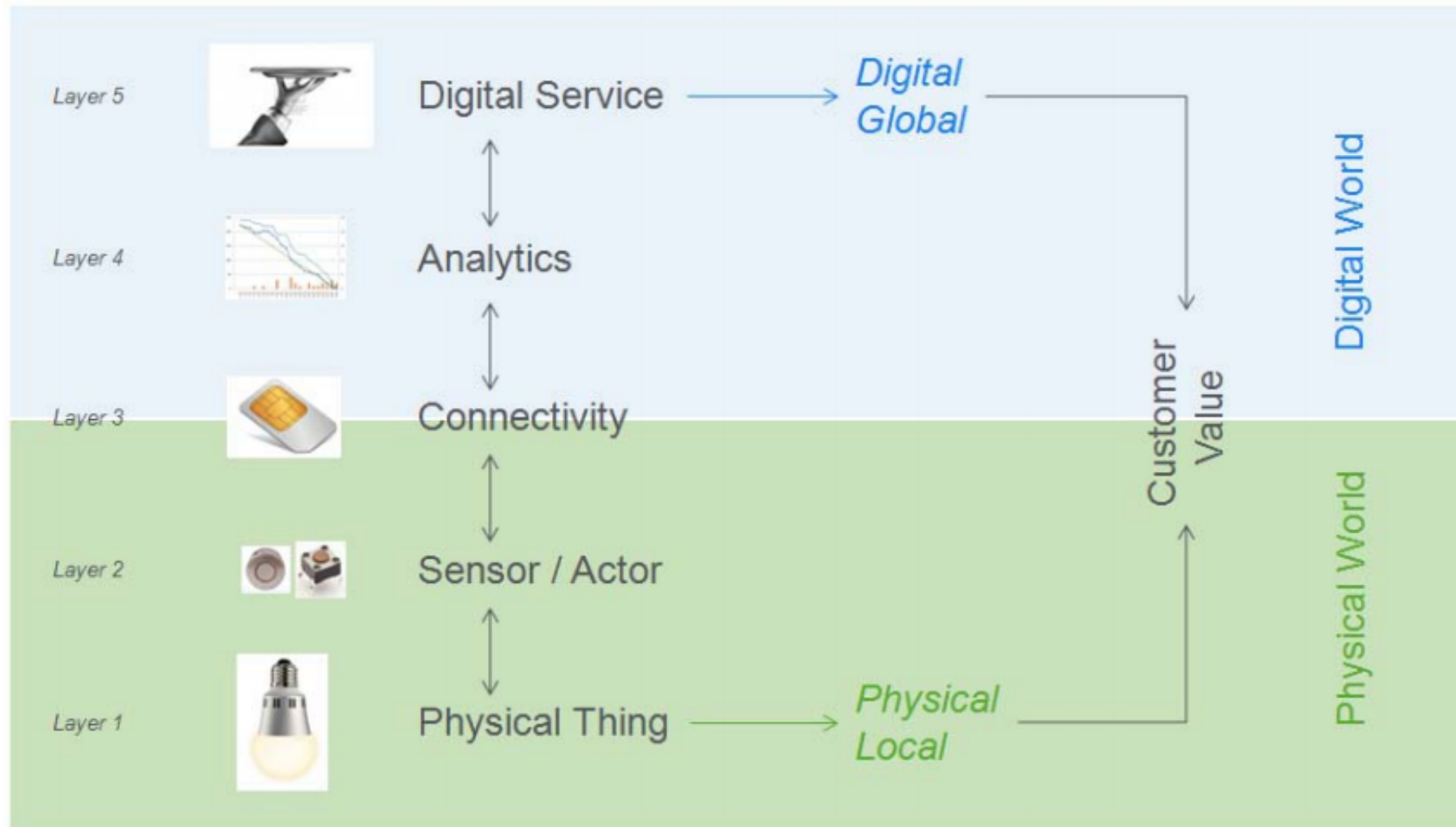
Internet der Dinge (IoT)



Das «Internet der Dinge» vereint die physische mit der digitalen Welt und bietet so die Möglichkeit für neue Geschäftsmodelle und operative Verbesserungen.

Siehe auch: <http://www.computerwoche.de/a/iot-wird-schneller-mainstream-als-erwartet,3224486>

Architektur: IoT Layer



=> **Gebrauchsgegenstände werden zu Digitalen Objekten.**

Quelle: http://www.iot-lab.ch/wp-content/uploads/2014/11/EN_Bosch-Lab-White-Paper-GM-im-IOT-1_3.pdf

IoT Ecosystem (Ökosystem)



Smart City



Wearables



Smart Home

- Industrie 4.0 ("intelligente Fabrik" oder Smart Factory)

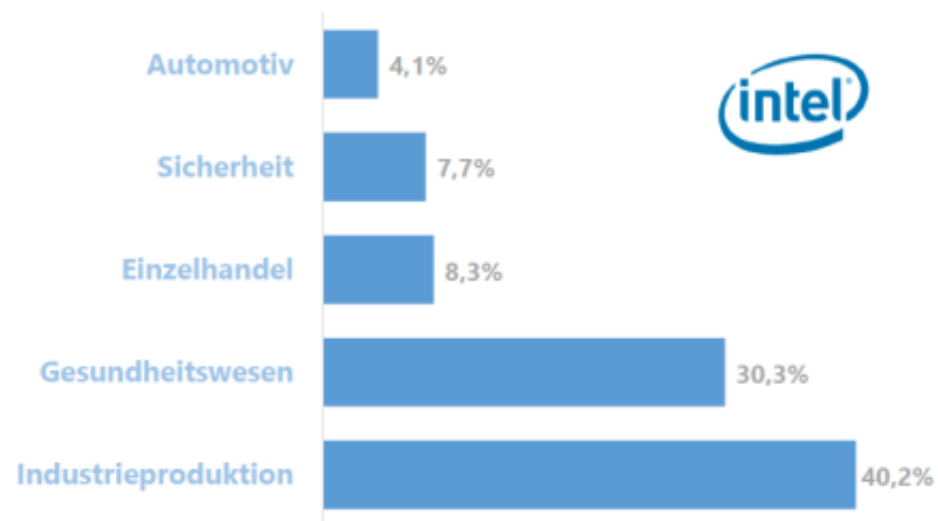
=> **IoT durchdringt alle Bereiche.**

IoT Einsatzgebiete (Zukunft)



Quelle: Intel

Eine Studie von Intel stellt fest: Mehr als 90 Prozent von IoT wird in Zukunft nicht im privaten Bereich eingesetzt.



=> IoT Anforderungen (Software Qualitätssicherung!) nehmen zu.

IoT Beispiele

- QIVICON Initiative: [QIVICON kompatible Produkte](#)
- [Samsung SmartThings](#)
- [13 Produkte für Kleinkinder](#)

Web of Things / Physical Mashups / IoT Orchestrierung

IoT Protokolle

- ZigBee - Lampen, Waschmaschinen, ...
- Bluetooth - Headsets, Fitnesstracker, ...
- X10 - Gebäudeautomation (via Strom)
- Infrarot - Unterhaltungselektronik, ...
- Funk - Steckdosen, ...
- SPI - RFID Reader, LED Strips, ...
- I2C - Sensoren wie Temperatur, Displays ...
- UART - Modems z.B. für Bluetooth
- LoRaWAN - Long Range Wide Area Network
- ...

Web

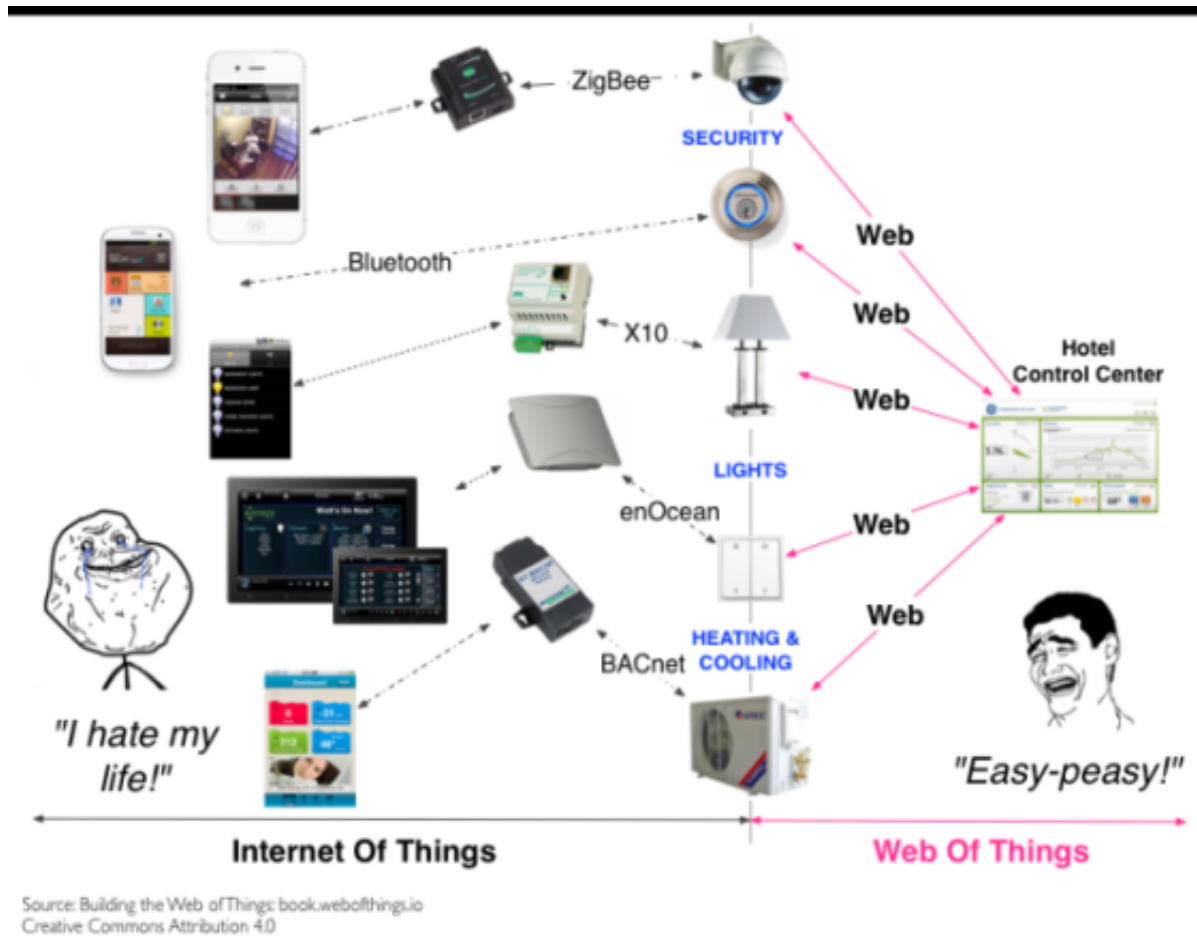
TCP/IP

- Transmission Control Protocol / Internet Protocol (TCP/IP) ist eine Familie von Netzwerkprotokollen und wird wegen ihrer grossen Bedeutung für das Internet auch als **Internetprotokollfamilie** bezeichnet.

REST

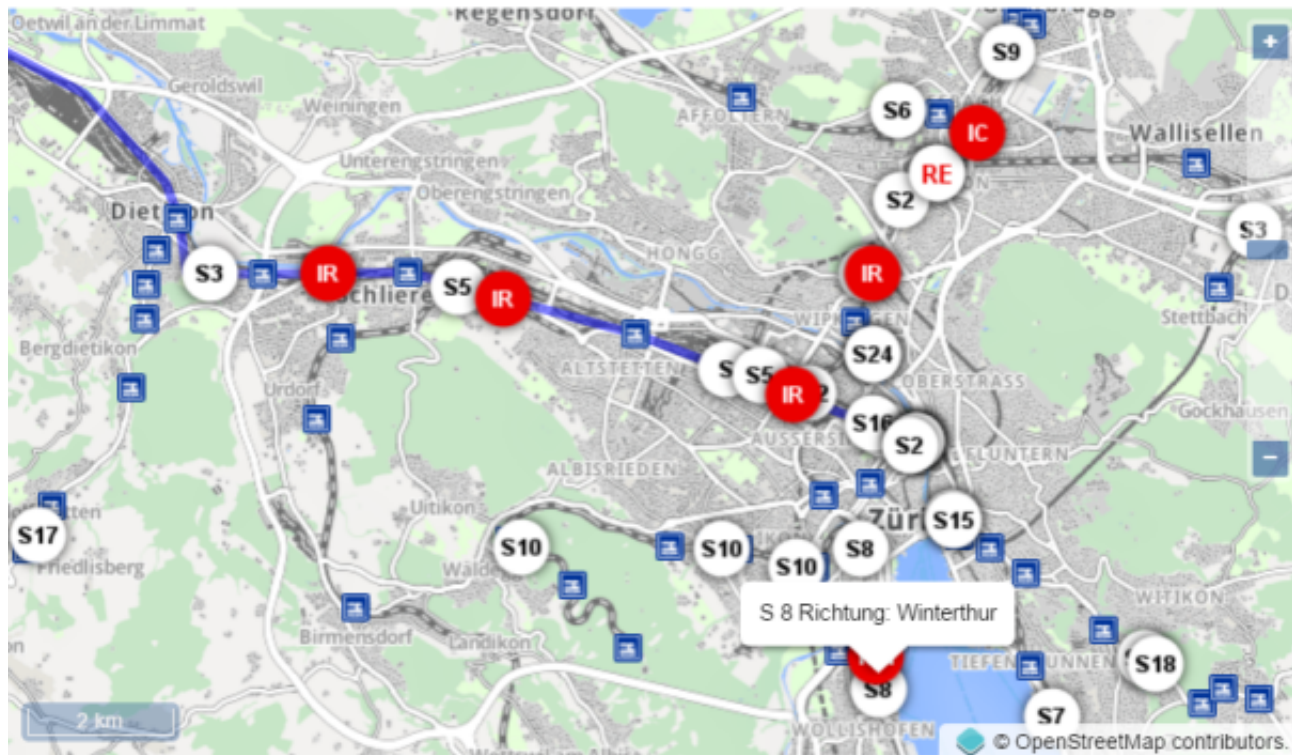
- Representational State Transfer (abgekürzt REST) bezeichnet ein Programmierparadigma für verteilte Systeme.
- REST ist eine **Abstraktion der Struktur und des Verhaltens (GET, POST ...) des World Wide Web**.
- REST fordert, dass eine Web-Adresse (URL) genau einen Seiteninhalt repräsentiert.

Web of Things



=> IoT + Web (TCP/IP, REST) = Web of Things

Mashups



Mashup (von englisch to mash für vermischen) bezeichnet die Erstellung neuer Medieninhalte durch die nahtlose (Re-)Kombination bereits bestehender Inhalte.

Physical Mashup



The screenshot shows the ProgrammableWeb website interface. At the top, there's a navigation bar with links for 'API News', 'API Directory', and 'For API Providers'. Below this is a search bar with the text 'Filter APIs'. Inside the search bar, there's a dropdown menu set to 'By Category' and a text input field containing 'HTTP'. To the right of the input field is a checkbox labeled 'Include Depreciated APIs'. A green button labeled 'Track this Search' is positioned to the right of the search bar. Below the search bar, a table lists APIs. The table has four columns: 'API Name', 'Description', 'Category', and 'Date'.

API Name	Description	Category	Date
eBay	eBay's primary web services are found in the three following API's: The [[http://www.programmableweb.com/api/ebay-trading Trading API]] is designed to provide robust support for...	Search	12.05.2005
Fitbit	Fitbit is a website that offers tools for users to track their	Fitness	03.22.2011

Using services such as IFTTT or Eclipse Smart Home, users can create **small logical rules that mix real-world devices**, such as sensors in their home **with virtual services in the cloud** for example an SMS gateway or a weather forecast service. Such applications are called physical mashups.

=> **Web of Things + Cloud Services = Physical Mashups**

Quelle: <http://webofthings.org/2016/01/23/wot-vs-iot-12/>

Cloud Services: <http://www.programmableweb.com/>

IoT Orchestrierung

- Orchestrierung (englisch orchestration, Instrumentierung, Inszenierung) ist das flexible Kombinieren mehrerer Services zu einer Komposition.
- Diese Komposition beschreibt einen **ausführbaren Geschäftsprozess**.
- Sowohl unternehmensinterne als auch unternehmensexterne Services können kombiniert werden.

=> **Physical Mashups + Geschäftsprozesse = IoT Orchestrierung**

Quelle: <https://de.wikipedia.org/wiki/Dienstekomposition#Orchestrierung>

Kitchen Helper Project: Software-Metriken

- Thomas Fehlmann

IoT sind Smarte Objecte



- Smart: geschickt, elegant, pfiffig

=> bestehende Technologien kritisch hinterfragen

IoT ist Smart

Top Business Drivers

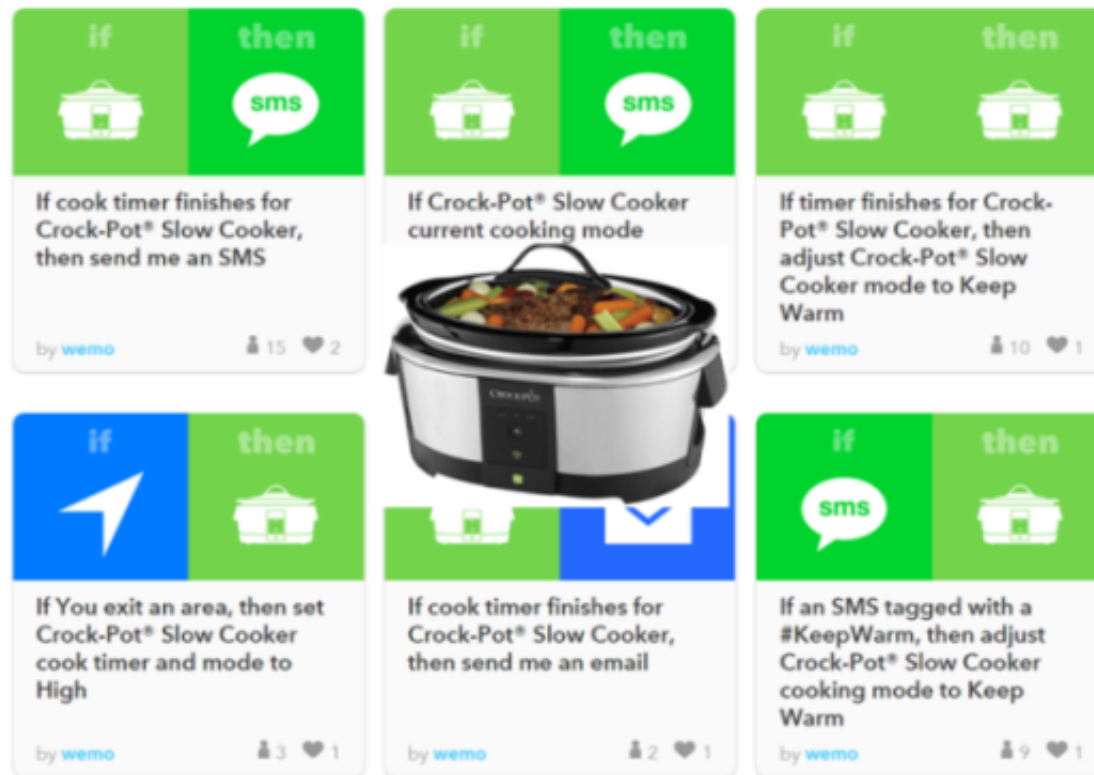
Business Drivers

<i>Top Business Drivers</i>	<i>Attributes</i>	<i>Profile</i>
D04 Get High Esteem	<i>Earn praise</i> <i>Make friends envious</i>	0.51 0.56
B01 Collect New Recipes	<i>Be special</i> <i>Make it essential</i>	0.49 0.54
C02 Willing to Recommend	<i>A special kitchen</i>	0.34 0.37
D01 Surprise Friends	<i>Feel cool</i> <i>Kitchen helper helps!</i>	0.29 0.32
A03 Customer Loyalty	<i>Customer come back</i> <i>Find special food</i>	0.28 0.31
B02 Promote Specialities	<i>Recipes promote high-yield food</i>	0.25 0.27

- Facebook, Bestehende Rezeptsites?

=> Partnerschaften, Bestehendes Verwenden?

Kitchen Helper Project = Smart K(itchen)



=> Internet fähiger Kocher kombiniert mit Cloud Services.

IFTTT: https://ifttt.com/wemo_slowcooker

Crock-Pot® Smart Slow Cooker: <http://www.belkin.com/us/F7C045/p/P-F7C045/>

Smart K(itchen): Prozesse (weniger ist mehr)

- Timer (Start in x Sekunden) setzen (optional)
- Gericht auswählen und damit Kochprozess starten
- Temperatur überwachen
- Fertigmeldung Kochprozess an Cloud Service: IFTTT
- Wenn Kochprozess abgeschlossen ist, Gericht warmhalten

Smart K(itchen): Gerichte (weniger ist mehr)

Kochvorgang	Zeitbedarf n min.	Energieverbrauch in Ah (12V DC)
Kochen von Wasser (Tee, Kaffee) 1 Liter	6,8	12
Kochen von Reis 250 g in 375 ml Wasser Kartoffeln 650 g über 500 ml Wasser Spaghetti 500 g in 1.000 ml Wasser	20 28 20	14 21 4
Gemüse garen 600 g erwärmen 600 g	12,5 / 6	14 / 10
Braten von 4 Eiern	5	7,4
4 Eier kochen (in ca. 0,7 Liter Wasser)	10	13
Kurzbraten von 2 Steaks ca. 300 g 4 Schnitzeln ca. 600 g	15	19 26
Soßen bereiten 500 ml	8	8

Quelle: <http://www.lieberherr.com/137/lebensmitteltechnik/kochen-ab-12-v-akku>

Smart K(itchen): Implementierung

- Mikrocontroller / Sensoren / Aktoren
- Connectivity / Web of Things
- Physical Mashup / Orchestrierung / Cloud

IoT mbed Plattform

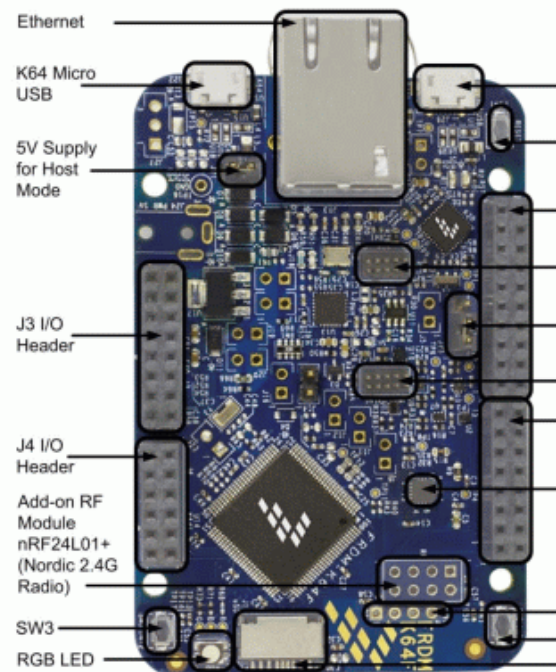


mbed ist Plattform und Betriebssystem für Internet-verbundene Geräte auf Basis von 32-Bit- ARM Cortex-M -Mikrocontroller.

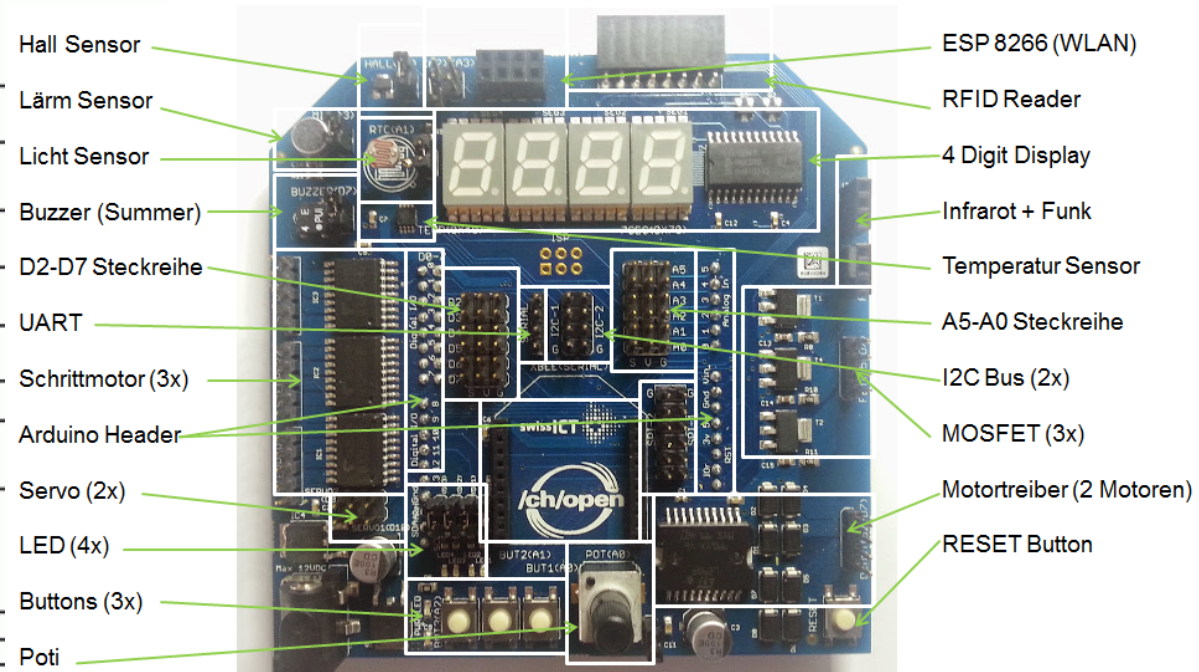
Eine Komponenten Library ergänzt die mbed Plattform.

IoT Board und Shield

FRDM K64F Mikrocontrollerboard



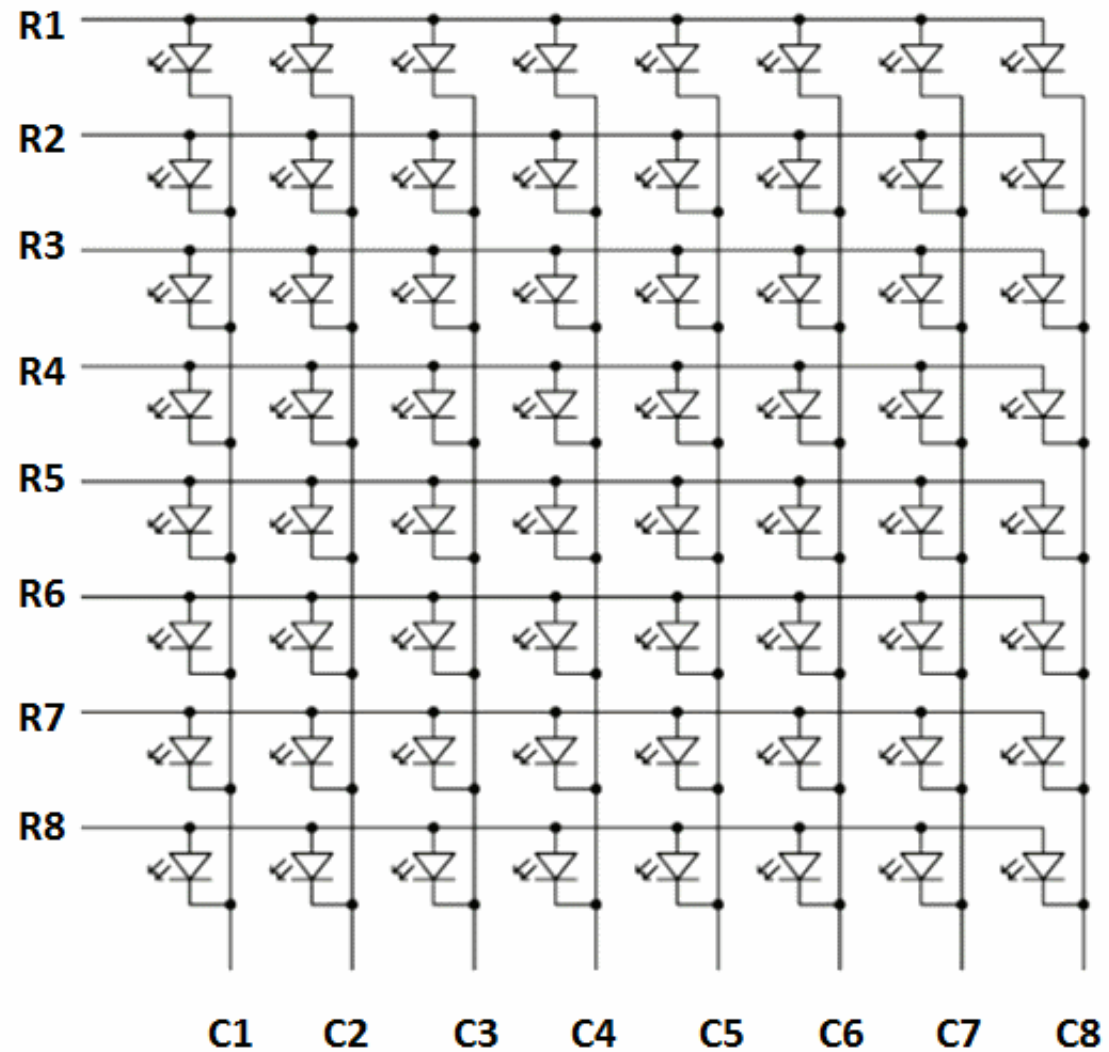
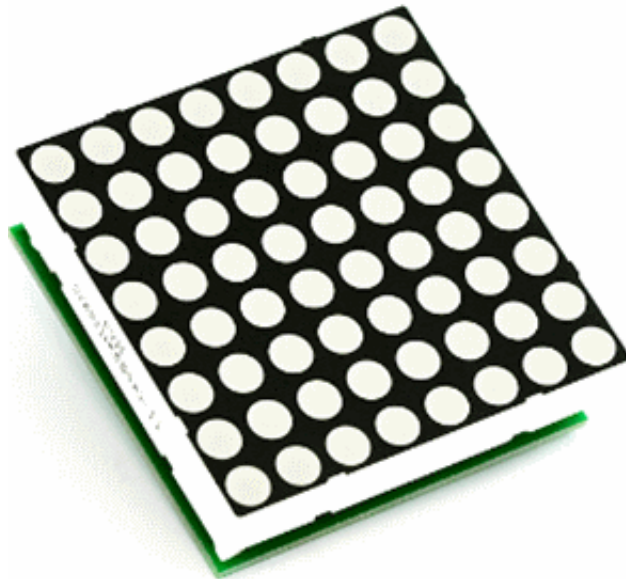
IoTKit SMD Shield



Ethernet-Anschluss, Temperatur-Sensor, MOSFET Treiber für Kocher.

=> Erweiterbar mit WLAN, RFID, Bluetooth, ZigBee, ...

Kochfeld: (Dot LED Matrix)



Kochfeld: mbed Implementierung

```
class Cooker
{
    void setTimer( long t )           // Startverzögerung setzen
    void setCookTime( long t )       // Kochdauer setzen
    void setKeepWarm( int w )        // Warmhalten on/off
    void start()                      // Kochprozess starten
    void off()                        // Kochprozess stoppen
    int run()                         // Kochprozess
    void attachFinish( void (*fptr)(void) ) // Fertigmeldung
}
```

=> **Es gibt keine Methode zum Auswählen des Gerichtes**

Source: <https://developer.mbed.org/teams/smdiotkit2ch/code/RPCHTTPServerKitchenHelper/file/26ad18018b2b/Cooker/Cooker.h>

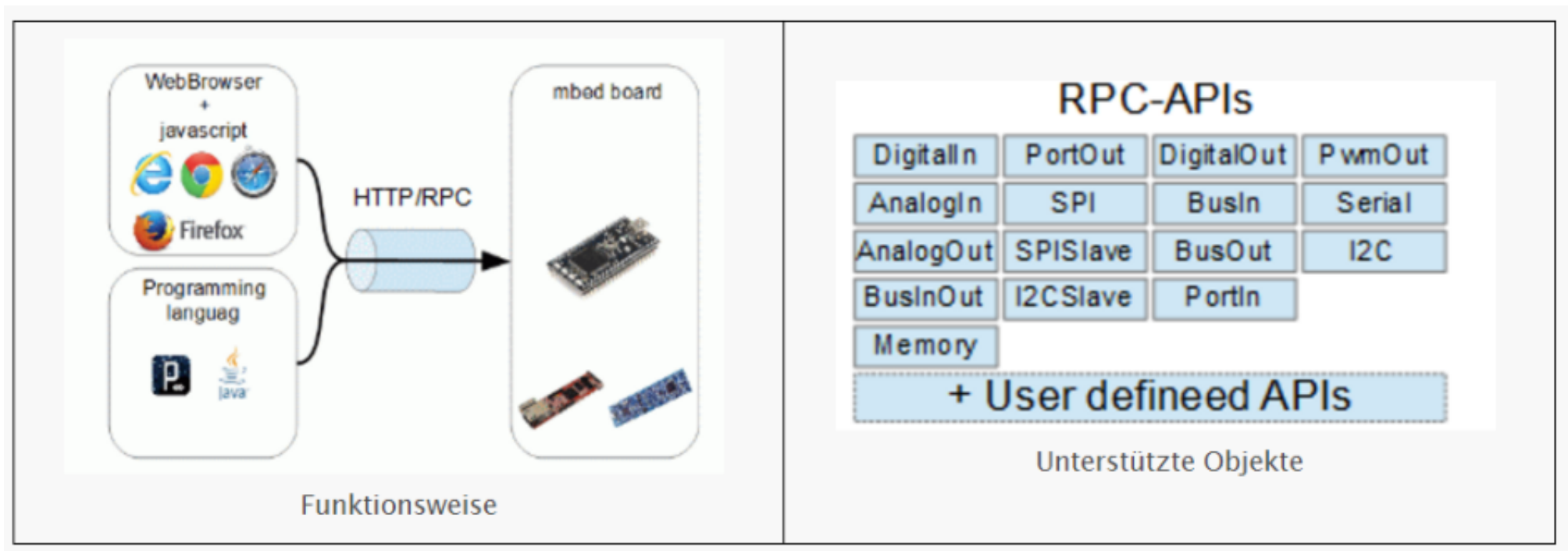
Kochfeld: Temperatur überwachen

```
class Cooker
{
    TMP175 temp;          // Temperatursensor
    AnalogIn poti;        // Schwellenwert ueberkochen

    int run()
    {
        if ( temp.fReadTemperatureTMP175() < poti.read() )
            // volle Leistung
        else
            // reduzierte Leistung
    }
}
```

Source: <https://developer.mbed.org/teams/smdiotkit2ch/code/RPCHTTPServerKitchenHelper/file/26ad18018b2b/Cooker/Cooker.h>

IoT Connectivity: Übersicht



Die mbed Plattform beinhaltet eine RPC Library (RPC - „Aufruf einer fernen Prozedur“), welche **mbed Objekte** mittels Serieller Schnittstelle oder via **HTTP Aufrufen**, zur Verfügung stellt.

=> das Projekt verwendet ein User definiertes Object RPCCooker

IoT Connectivity: mbed Implementierung

```
class RPCCooker
{
    Cooker cooker;    // Kochfeld

    /** RPC intern: verfuegbare Methoden definieren */
    virtual const struct rpc_method *get_rpc_methods()
    {
        static const rpc_method rpc_methods[] =
        {
            {"menu", rpc_method_caller<RpcCooker, int, &RpcCooker::menu>},
            {"timer", rpc_method_caller<RpcCooker, int, &RpcCooker::timer>},
            {"keepwarm", rpc_method_caller<RpcCooker, int, &RpcCooker::keepwarm>},
            RPC_METHOD_SUPER(RPC)
        };
        return rpc_methods;
    }
}
```

Source: <https://developer.mbed.org/teams/smdiotkit2ch/code/RPCHTTPServerKitchenHelper/file/16348037b99a/Cooker/RPCCooker.h>

IoT Connectivity: HTTP Aufrufe

- Timer (Start in x Sekunden) setzen (optional)

`http://<mbed Board>/rpc/cooker/timer+<Sekunden>`

- Gericht auswählen und damit Kochprozess starten

`http://<mbed Board>/rpc/cooker/menu+<Menu-Nr>`

- Wenn Kochprozess abgeschlossen ist, Gericht warmhalten

`http://<mbed Board>/rpc/cooker/keepwarm<0|1>`

=> Temperatur überwachen - wird intern abgehandelt

Source: <https://developer.mbed.org/teams/smdiotkit2ch/code/RPCHTTPServerKitchenHelper/file/16348037b99a/Cooker/RPCCooker.h>

IFTTT Cloud Service

IFTTT (die Abkürzung von "If This Then That") ist ein Dienstanbieter, der es Benutzern erlaubt, verschiedene Webanwendungen (zum Beispiel **Facebook**, Evernote, Dropbox usw.) mit einfachen Bedingten Anweisungen zu verknüpfen.

IFTTT ermöglicht Benutzern "Rezepte" nach dem Motto "If this then that" ("Wenn dies dann das") zu erstellen

- Der "this"-Teil eines Rezepts wird "Trigger"
- Der "that"-Teil eines Rezepts wird Aktion genannt.

Mittels DO Button können Aktionen, z.B. via Smartphone ausgelöst werden.

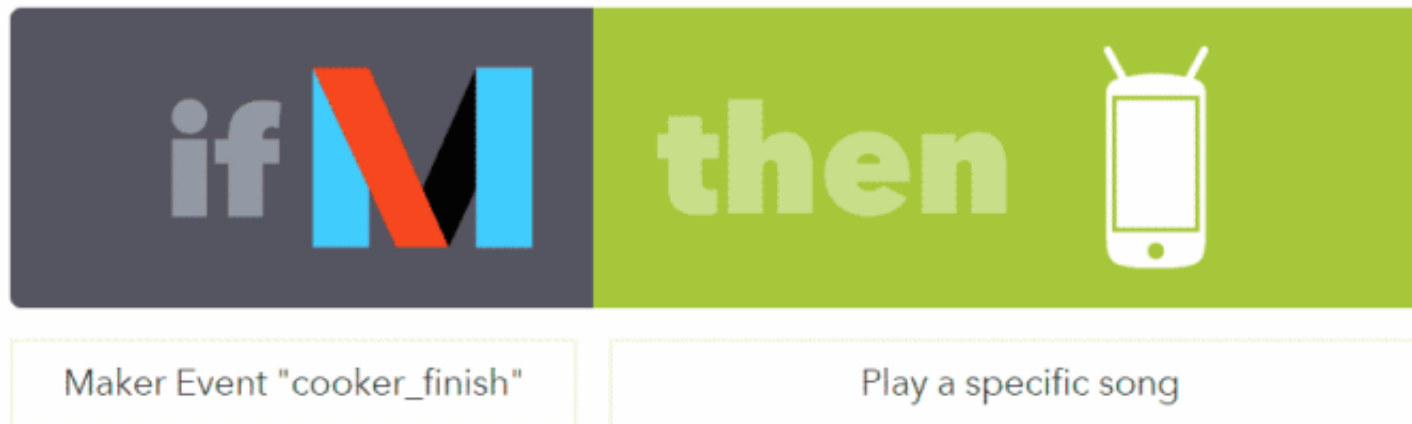
IoT Connectivity: IFTTT -> Kocher (mbed)

DO Button für folgende Actions einrichten:

- Timer, Start in 5 Sekunden
- Timer, Start in 10 Sekunden
- Gericht 1 - Kochen von Wasser
- Gericht 2 - Reis
- Gericht 3 - Spaghetti
- Gericht ...
- Gericht warmhalten

IFTTT: IFTTT -> Smartphone

Rezept für Ende Kochprozess einrichten:



Rezept Erreichbar mittels Maker Channel, bzw. HTTP Aufruf:

https://maker.ifttt.com/trigger/cooker_finish

IoT Connectivity: Kocher (mbed) -> IFTTT

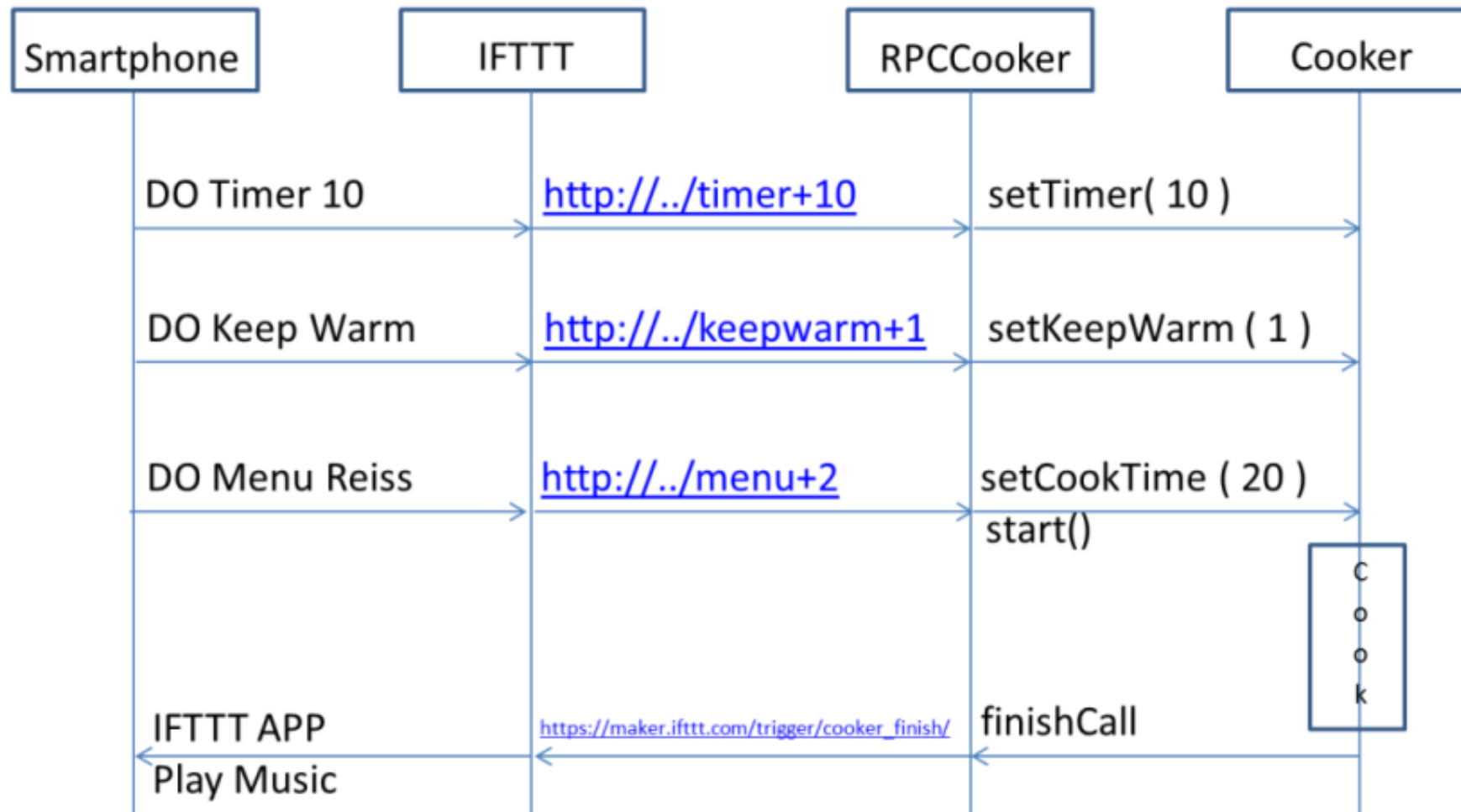
Implementiert mittels Callbacks in C/C++:

```
void finishCooker()
{
    http.get( "https://maker.ifttt.com/trigger/cooker_finish", ... );
}

main()
{
    cooker.attachFinish( &finishCooker );
}
```

=> **Trennung/Loslösung Kocher - Internet**

IoT Orchester (vereinfacht)



=> Die Klasse Cooker weiss nichts vom Internet

TODO

- Digitaler Rezepte Partner suchen (PDFs sind nicht Digital!)
 - Evtl. mehrere Varianten des Kochers Herstellen (mit und ohne Internetanschluss)
-
- Mehr IFTTT Rezepte, z.B. wenn WLAN x verlassen wird, Kocher starten
 - neuste HTTPServer Version verwenden mit `while() { srv.poll() }`
 - Kocher menu und start Kochprozess trennen
 - Kocher Fertigmeldung liefert Kochzeit, Kochende als Zeitangabe, ob KeepWarm aktiviert ist.
 - IoT Protokolle wie MQTT, CoAP statt HTTP verwenden (schneller, sicherer)

Kurse

Internet der Dinge für Business-, IT-, Software-Architekten, CIOs und Entscheidungsträger

- [IoT - Übersicht und Fallbeispiele](#)
- [IoT - für IT- und Software-Architekten](#)

Internet der Dinge für Implementierer

- [Grundlagen](#)
- [Aufbau 1](#) - Komplexe Anwendungen
- [Aufbau 2](#) - «Dinge» und die Cloud
- [Aufbau 3](#) - Raspberry Pi und Co. als Server

Unterstützt von Standortförderung – Amt für Wirtschaft und Arbeit Kanton Zürich, /ch/open und swissICT

Links

- [Business Models and the Internet of Things](#)
 - [Open Mobile Alliance](#)
 - [OMNA Lightweight M2M \(LWM2M\) Object & Resource Registry](#)
-

- [ARM mbed Platform](#)
 - [ARM mbed Komponenten](#)
 - [IFTTT](#)
 - [Crock-Pot® Smart Slow Cooker](#)
-

- [IoTKit](#) - Internet der Dinge Kit auf GitHub

Fragen / Diskussion ?

