

Nachrichtenaustausch mit der realen Welt über WebSphere MQ Telemetry



Renate Franken

WebSphere Technical Sales



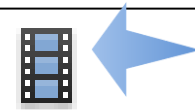
EUROPE IOT



Agenda

- **Context – Das Internet der Dinge**
- **Technology**
 - **MQTT – MQ Telemetry Transport**
 - **WebSphere MQ Telemetry**
- **Demo**
- **Future**
- **Use Cases**

The Internet of Things – Das Internet der Dinge



Video

Viele digitale Einheiten
instrumentieren unsere
Welt bereits heute



Verknüpft die virtuelle mit
der realen Welt

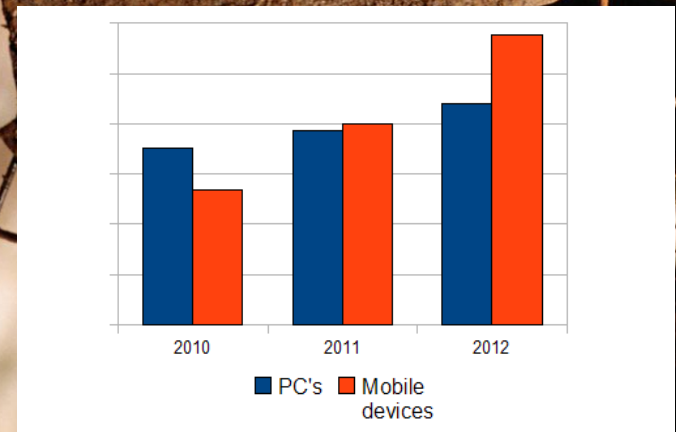
„In 2020 ist alles miteinander verbunden, geschätzte 60 Milliarden Smart Devices“

Gartner

Einige Zahlen...

2005 waren 1.3 Milliarden
RFID Tags im Umlauf

... 2011 sind es mehr als 33
Milliarden

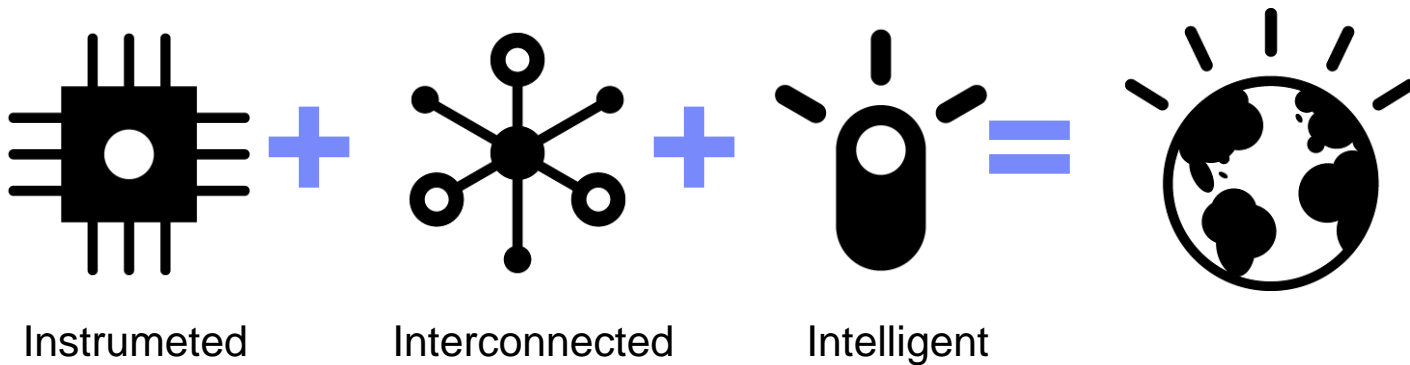


Source: Gartner

Geschätzte 2 Milliarden Menschen sind 2011 im Internet ...

... und Milliarden von verbundene Objekte – Autos, Stromzähler, techn. Geräte, Pipelines, ... – zusammengefaßt als das “Internet der Dinge”

Wir bauen einen smarten Planeten




Die Herausforderung neu zu denken und zu handeln —
ökonomisch, sozial und technisch.



Agenda

- **Context – Das Internet der Dinge**
- **Technology**
 - **MQTT – MQ Telemetry Transport**
 - **WebSphere MQ Telemetry**
- **Demo**
- **Future**
- **Use Cases**

Schlüsselfaktoren von MQ Telemetry Transport – MQTT Protokoll

- Die MQTT Spezifikationen sind **offen** 
- **Klein**, reduziert die Komplexität und Systemanforderungen
- **Einfach** / minimale pub/sub Nachrichten Semantik
 - Asynchrone (“push”) Nachrichtenauslieferung an Anwendungen
 - Einfache Aufrufe wie connect, publish, subscribe und disconnect.
- **Minimalisierte** Nachrichtenübertragungslänge
 - Protokoll mit komprimiert mit bit-wise Header und variablem Längensfeld
 - Kleinste mögliche Nachricht ist 2 Bytes.
 - Byte Array Nachrichteninhalte ohne zusätzlichen Anwendungsheader
- Eingebaute **Unterstützung bei Verbindungsabbruch**.
 - “Last will and testament” wird publiziert wenn die Client Verbindung abbricht
 - Stateful “roll-forward” Semantik und “durable” Subskriptionen.

Qualities of Service

Drei Quality of Services publizierter und subskribierter Nachrichten

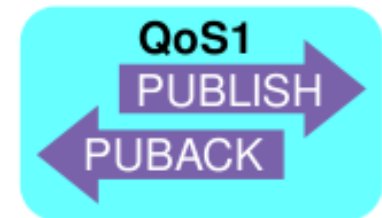
QoS 0: At most once delivery (non-persistent)

- Keine Wiederholversuche sind im Protokoll definiert.
- Die Nachricht wird höchstens einmal geschickt, die Nachricht kommt entweder einmal an oder nie



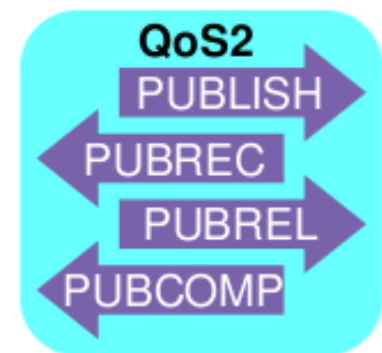
QoS 1: At least once delivery (persistent, dups possible)

- Client schickt Nachricht mit Message ID im header
- Server bestätigt mit einer PUBACK Nachricht
- Message wird wiederholt geschickt mit einem DUP bit gesetzt falls keine PUBACK erhalten wurde



QoS 2: Exactly once delivery (persistent)

- Zusätzlicher Nachrichtenaustausch um sicherzustellen dass die Nachricht nicht dupliziert geschickt wird
- Server bestätigt mit einer PUBREC Nachricht
- Client entläßt die Nachricht mit einer PUBREL Nachricht
- Server bestätigt Abschluß mit einer PUBCOMP Nachricht



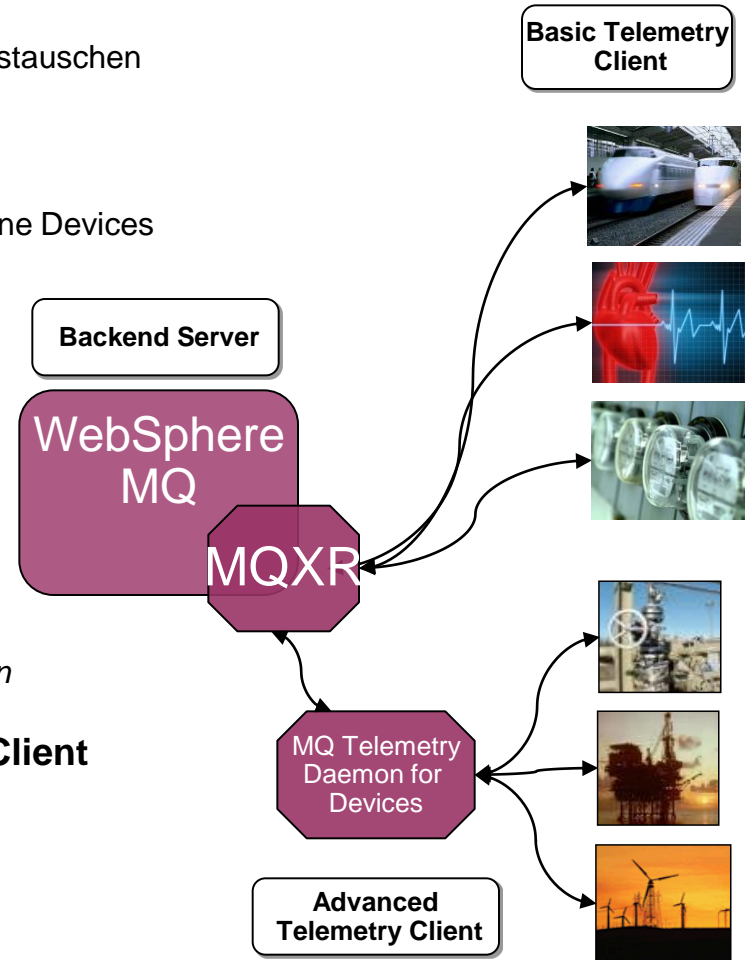
WebSphere MQ Telemetry – implementiert MQTT



- **WebSphere MQ Telemetry integriert mit WebSphere MQ**
 Publizierer und Subskriptoren können über MQI oder JMS Nachrichten austauschen
 Out of the box installierbar mit MQ7.1
- **Ist hoch skallierbar**
 An einen MQ Queue Manager können über 100,000 gleichzeitig verbundene Devices angebunden werden
- **Wird mit 2 verschiedene Clients ausgeliefert:**

 - Basic
 - *direkte Verbindung von einem Device*
 - Advanced – Deamon for Devices
 - *Agiert als "Konzentrator/Hub" for kleine Telemetry Device Netzwerke*
 - *Kann sich zu mehreren Backend Server verbinden*
 - *Unterstützt alternativen Backendserver im Fehlerfall*
 - *Kann Nachrichten zwischenspeichern oder locale Devices verbinden*
- **Zusätzlich kann jeder third party oder Open Source MQTT Client verwendet werden – unterstützt MQTT v3.1 Protokoll**
- **Umfangreiche Sicherheit**

 - Networkverschlüsselung: SSL
 - Authentifikation: JAAS
 - Authorisierung: OAM



MQ Explorer integration

The screenshot displays the IBM WebSphere MQ Explorer interface. On the left, a tree view shows the hierarchy: IBM WebSphere MQ > Queue Managers > QM_WIN_CKKS653 > Telemetry. The main window is titled 'MQTT Client Utility' and shows a 'Connection' section with Host: localhost, Port: 1883, and Client identifier: mqtt_WINCKK56S37J9H_2. The status is 'Connected'. Below this is a 'Client history' table:

Event	Topic	Message	QoS	Retained	Time
Connected					8/4/10 10:46 AM
Published	testTopic	Test Mes...	0	No	8/4/10 10:46 AM
Published	testTopic	Test Mes...	0	No	8/4/10 10:46 AM

Below the table are buttons for 'View message...', 'Clear history', and a 'Scroll lock' checkbox. The 'Subscription' section shows Topic: testTopic and Request QoS: 0 - At most once. The 'Publication' section shows Topic: testTopic, Message: Test Message, and QoS: 0 - At most once. A 'Connection Options' dialog box is open in the foreground, showing settings for new connections, including 'Clean session' (checked), 'Last Will and Testament' (empty), 'Topic' (empty), 'QoS: 0 - At most once', 'Message' (empty), and 'Retained publication' (unchecked). The dialog has 'OK' and 'Cancel' buttons. At the bottom of the main window, a status bar indicates 'Last updated: 10:47:12'.



Example: connect and send an MQTT message

```
public void sendMessage() throws MqttException {
    MqttProperties mqttProps = new MqttProperties();
    mqttProps.setCleanStart( true );
    MqttClient client = MqttClientFactory.INSTANCE.
        createMqttClient("testClient",
            "tcp://localhost:1883", mqttProps);
    client.registerCallback(this);
    client.connect();
    client.publish("abc/123",
        new MqttPayload("Hello World!").getBytes(),0),
        (byte) 2, false);
    client.disconnect();
}

public void publishArrived(String topicName,
    MqttPayload payload,
    byte qos, boolean retained,
    int msgId) {
    System.out.println("Got it!");
}
```

← Create a connection using the connection factory, this time for a clean starting client

← Register the class as a listener and connect to the broker

← Publish a message to the given topic and disconnect

← On receipt of a publication, simply print out a message on the console to say we received it

WebSphere und das 'Internet der Dinge'

Internet der Dinge

- Mobile Devices



DataPower
(with MQTT Extended Reach client for passthru to MQ)



HTTP

MQTT

WebSphere Mobile
Technology Preview -
→ Worklight

Build Control Governance

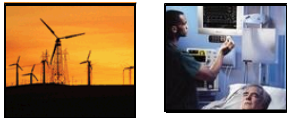
WebSphere software

- Embedded Controller



MQTT

- Sensor



Firewall

- Actuators



IBM WebSphere MQ

Queue Manager A

MQ Telemetry
Transport

WebSphere software

MQ Server is used for scale



Cast
Iron

Intelligence:
Cognos, ILOG, SPSS,
WebSphere Business
Events, BPM, ...

WebSphere
Message
Broker

WebSphere software

MQ

JMS

HTTP

WebService

Enterprise
Applications

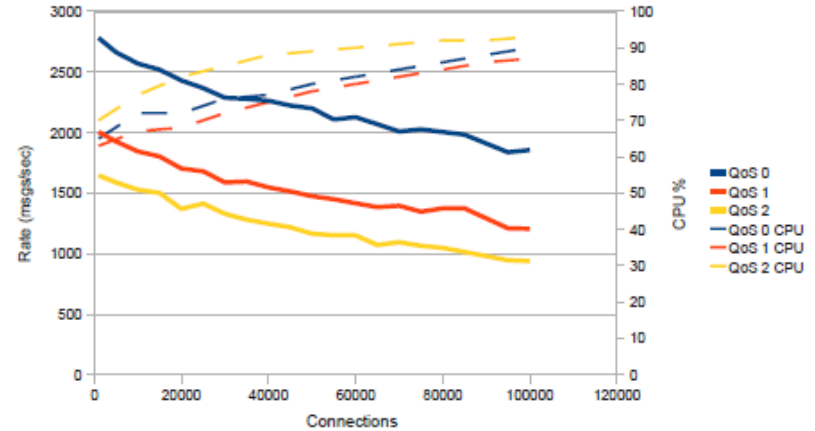
Messaging integration
hub and logic systems

MQ / JMS / HTTP



MQ Telemetry Transport versus HTTP(S)

- MQTT ist Datenzentrisch gegenüber Dokumentenzentrischen HTTP
- MQTT ist **einfach**: connect publish/subscribe disconnect
- MQTT ist **bidirektional** für 'always-on' Devices (push zum Client anstatt pull vom Server)
- MQTT hat **optimierte** Nachrichtenlänge (im Vergleich HPPT vs. MQTT QoS1)
- MQTT verteilt Nachrichten 1:1, 1:vielen über pub/sub, vs. Http über Point-to-Point
- Zuverlässiges MQTT unterstützt verschiedene **Quality of Service** (fire and forget oder fire and confirm) vs. unzuverlässigem HTTP
- Skalierbar** (>100.000 Clients /Qmgr)
- Gut integriert in **WebSphere MQ**
- Weniger **Batterieverbrauch** als HTTP



Scenario	HTTP	MQTT
10 Bytes Msg lesen vom Server	126 bytes	69 bytes
10 Bytes Msg schreiben auf den Server	141 bytes	47 bytes
100 Msgs lesen vom Server	12600 bytes	2445 bytes
100 Msgs schreiben auf den Server	14100 bytes	2126 bytes
1 Msg alle 15 Minuten 365 Tage (ca. 35000Msg)	ca. 5 MB	ca 0,7 MB
1 Msg alle 15 Minuten 365 Tage – SSL (HTTPS vs MQTT-SSL)	ca. 219 MB	ca. 23MB



Details - HTTP

Msg Flow	Data in Bytes	Data (http)
Client->Server	316	<pre>POST http://192.168.177.211:7080/httpstest HTTP/1.1 Content-Length: 7 Content-Type: text/xml; charset=utf-8 Host: 192.168.177.211:7080 SOAPAction: "" Connection: keep-alive X-Remote-Addr: 192.168.177.1 X-Remote-Host: 192.168.177.1 X-Server-Name: 192.168.177.211 X-Server-Port: 7080 X-Scheme: http Message</pre>
minimum	66	<pre>POST /post HTTP/1.1 Host: localhost Content-Length: 7 message</pre>
Server->Client	147	<pre>HTTP/1.1 200 OK Content-Type: text/xml; charset=utf-8 Content-Length: 7 Date: Mon, 10 Oct 2011 20:08:00 GMT Server: Apache-Coyote/1.1 Message</pre>
minimum	75	<pre>HTTP/1.1 200 OK Date: Mon, 10 Oct 2011 20:08:00 GMT Content-Length: 0</pre>
Total	463 (141) bytes	



Details - MQTT

Msg Flow	Data in Bytes	MQTT (send Msg QoS = 1)
Client -> Server	20	Connect -> 101200064D51497364700302001E00046D717474
Server < - Cl.	4	Connack <- 20020000 (<i>Server Acknowledge</i>)
Client -> Server	17	Publish -> 3A0F0004706F73740001 6D657373616765 (<i>send msg</i>)
Server < - Cl.	4	Pubback <- 40020001 (<i>Server Acknowledge for QoS=1</i>)
Client -> Server	2	Disconnect > E000
Total	47 bytes	

Ausblick: Erweiterte Anbindung – MQ Web Messaging

- Bietet Anbindung für
 - Business to Customer (B2C)
 - Business to Employee (B2E)
- Von **mobilen** and **festinstallierten** Einheiten mit einem modernen **web browser**
- Basiert auf Erweiterungen (MQ Telemetry) für WebSphere MQ
- MQ Web Messaging wird für die folgenden primären Absichten entwickelt:
 - 1st class web support aufbauend auf Websockets, ein Web Standard (IETF and w3C)
 - Zielgerichtet für eine neue Klasse von hochwertigen Internet Anwendungen die in einem Web Browser laufen
 - Bietet effizientes Nachrichten und Ereignis Verteilung von und zu einem Web Browser.
 - Im Gegensatz zu HTTP können Nachrichten/Events in beide Richtungen geschickt werden.
 - Ein publish/subscribe Nachrichten Paradigma zugreifbar via JavaScript API
 - Bietet traditionelles Messaging Service-Qualität für moderne Wege



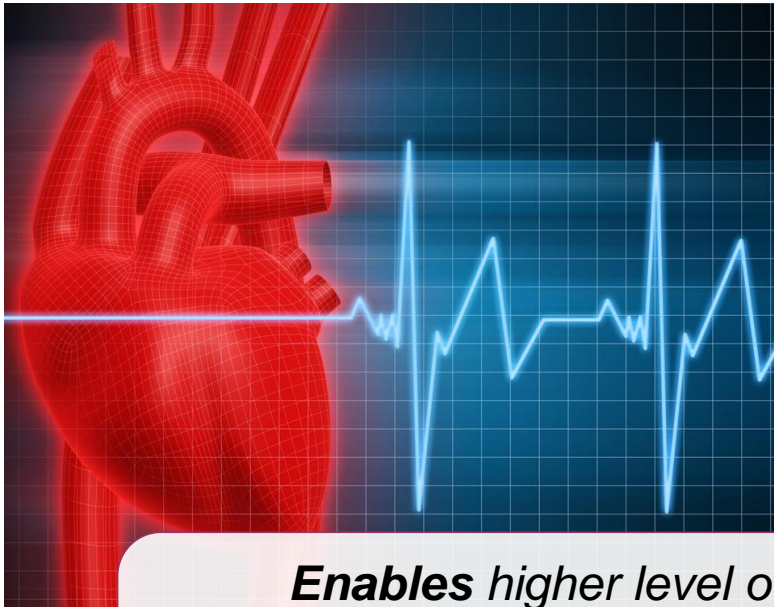


Agenda

- **Context – Das Internet der Dinge**
- **Technology**
 - **MQTT – MQ Telemetry Transport**
 - **WebSphere MQ Telemetry**
- **Demo**
- **Future**
- **Use Cases**

Smarter Healthcare

Medical organization created a remote pace-maker monitoring solution to provide better patient care



Client Pains

- Physicians needed better monitoring of cardiac patients
- Improve efficiency of checkups
- Meet healthcare data capture standards

Enables higher level of patient care and peace of mind

Improves administrative efficiency and maintenance

Helps conform to standards and ease integration of data

Improving Energy Usage

Utility company developing an Intelligent Utility Network offering for optimizing load on electricity grids



Business Partner

Needs robust middleware technology to connect to remote smart meters

Needs to be able to rapidly scale solution nationally

Able to offer daily energy savings of 15-20%

Enables utilities to reduce peaks and avoid punitive charges

Helps save electricity through better peak load management



Device Management – Car Tracking

Central European Time | English [Search](#)

Device Gateway Management

[Home](#) |
 [Device](#) |
 [Software](#) |
 [Access](#) |
 [System](#) |
 [Telematics](#)

Device Status

Services

[Live Track](#)

[Drivers Log](#)

Logged in as
Renate Franken

[Logout](#)

Drivers Log

[Search Criteria](#)

Please enter search criteria.

Device : [Device Search](#)

From:

To:

Max. Results:

[Search](#) 50 journey(s) found.



[Drivers Log](#)

Select	Start Time	End Time	Raw Distance	Raw Average Speed	Raw Max. Speed	
<input type="checkbox"/>	06-Dec-2011 20:13:31	06-Dec-2011 20:22:40	4.5	29.5	98.1	
<input type="checkbox"/>	06-Dec-2011 18:41:17	06-Dec-2011 18:52:32	4.5	24.2	79.0	
<input type="checkbox"/>	06-Dec-2011 17:06:14	06-Dec-2011 17:20:00	12.5	54.6	120.9	
<input type="checkbox"/>	06-Dec-2011 16:37:48	06-Dec-2011 16:48:57	3.7	20.3	59.6	
<input type="checkbox"/>	06-Dec-2011 16:28:32	06-Dec-2011 16:32:32	0.1	1.6	22.6	
<input type="checkbox"/>	06-Dec-2011 16:14:50	06-Dec-2011 16:25:42	5.9	32.8	97.6	
<input type="checkbox"/>	06-Dec-2011 15:42:32	06-Dec-2011 15:53:15	4.5	25.7	128.2	



Facebook Messenger

Lucy Zhang, a software engineer at Facebook, has written about their new [Facebook Messenger](#) app:

*“One of the **problems** we experienced was **long latency** when sending a message. The method we were using to send was reliable but **slow**, and there were **limitations** on how much we could improve it. With **just a few weeks** until launch, we ended up building a new mechanism that maintains a persistent connection to our servers. To do this without **killing battery life**, we used a protocol called **MQTT** that we had experimented with in Beluga. MQTT is specifically designed for applications like sending telemetry data to and from space probes, so it is **designed to use bandwidth and batteries sparingly**. By maintaining an MQTT connection and routing messages through our chat pipeline, we were able to often achieve **phone-to-phone delivery in the hundreds of milliseconds, rather than multiple seconds.**”*

MQTT Client on Android - Experiences

Push Notification



- ✓ Real-time Alerts
- ✓ Analytics Tracking
- ✓ Assured Delivery

Rich Framework

- ✓ Two-way Communication
- ✓ Rich / HTML Messages
- ✓ Status Channels
- ✓ Security
- ✓ Simple

Low Bandwidth & Cost



Very efficient bandwidth usage



Much cheaper alternative to SMS

Low Power Usage



0.3041%



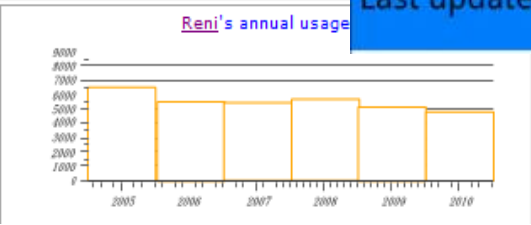
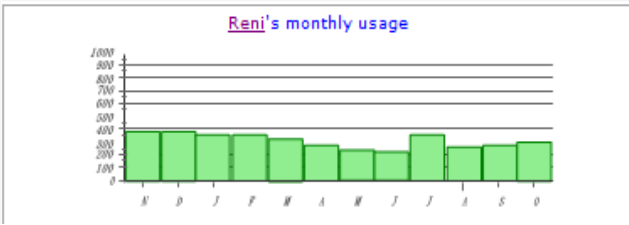
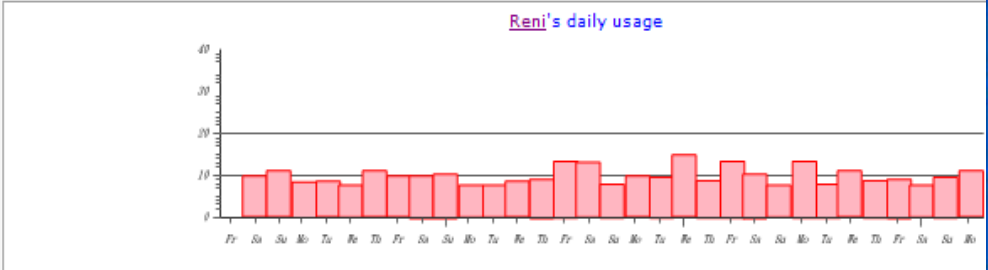
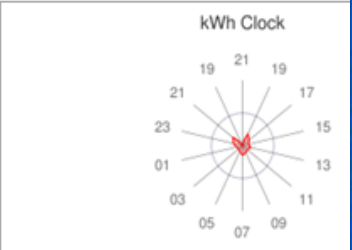
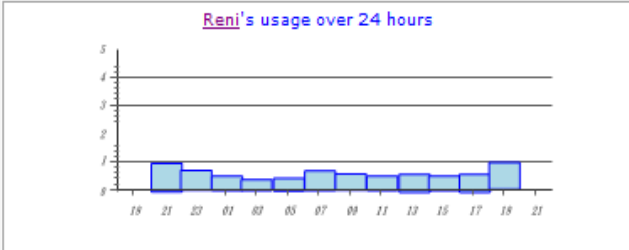
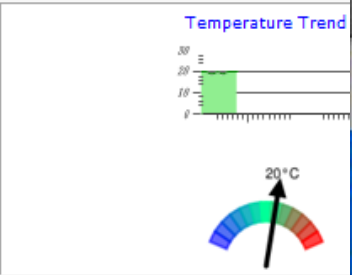
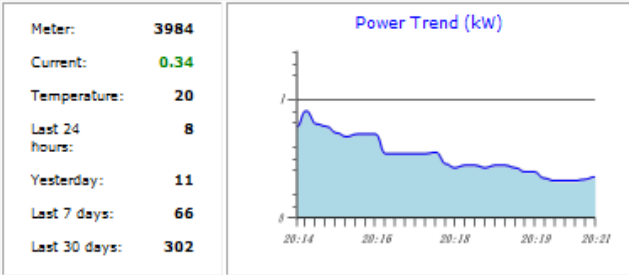
0.0047%

% Battery per Hour



Mein Stromverbrauch zu Hause mit Android Anbindung

- Current
- Dashboard
- Meters
- History
- 24 Hours
- 31 Days
- 12 Months
- 4 Years



3G 10:54 PM

UIF

Power now 470 Watt

CurrentCost - Meter

Power now: 470 watt

Channel1: 63 Watt
 Channel2: 234 Watt
 Channel3: 173 Watt

Power last 24 hours: 9.715 kW/h
 Power last 30 days: 326.939 kW/h
 Power last 12 month: 3746.25 kW/h

Power since birth: 4104.9634 kW/h
 (393 Days since birth)

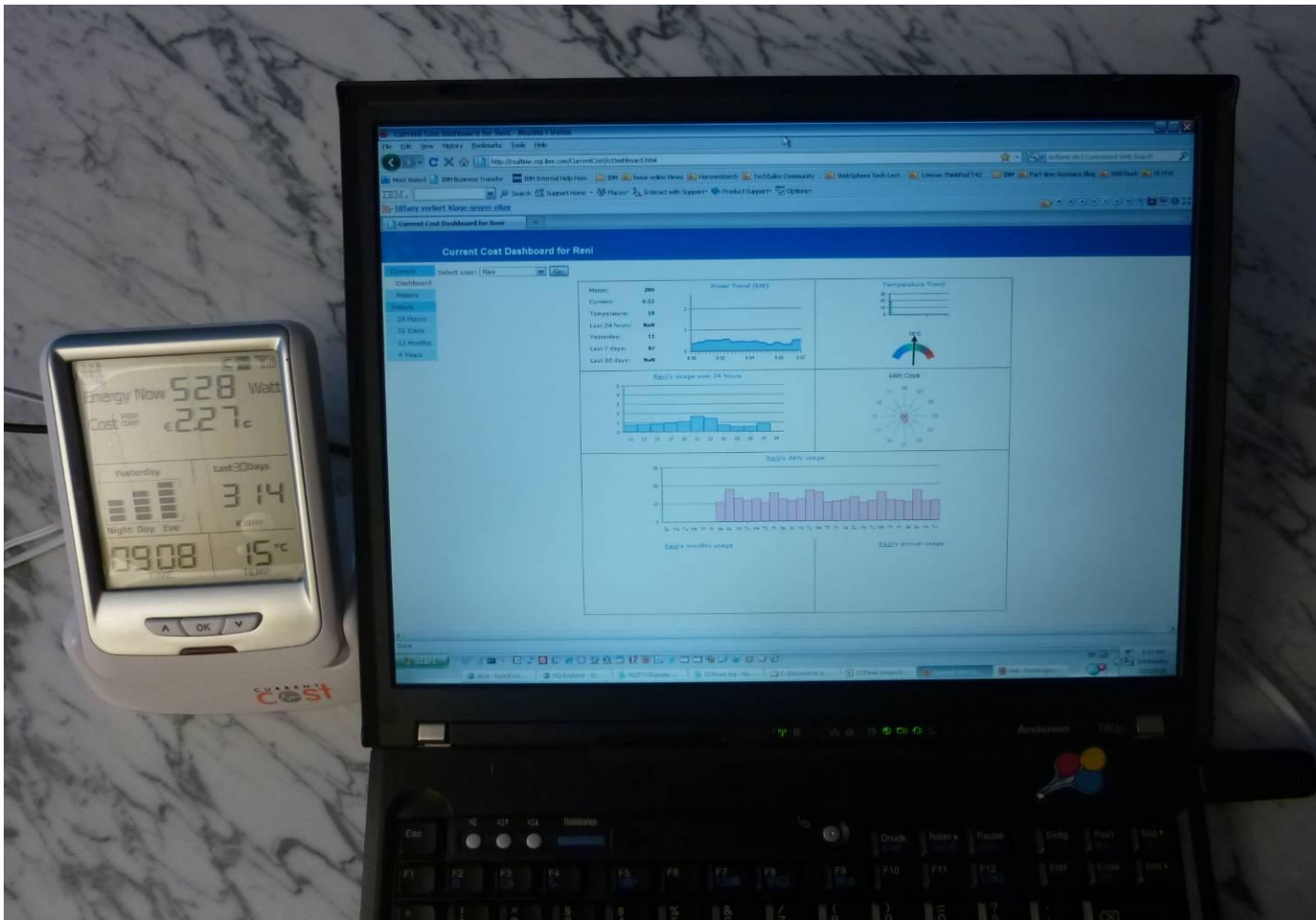
Temperature: 18.8 C

Last update: 06.12.2011 22:54:32

Mein Stromzähler



Das Display (über USB Kabel verbunden)





Easy to try out!

Really small message broker:

<http://www.alphaworks.ibm.com/tech/rsmb>

Download rsmb-1.2.0.zip

Unzip

Run `nohup ./broker >> /dev/null &`

Play with C client utils, code, and be merry!

Available for Linux IA32, IA64 kernel 2.6.8+; Linux on IBM System z; Linux for ARM XScale, kernel 2.0.0+ (Crossbow Stargate or Eurotech Viper); Windows XP; Mac OS X Leopard; Unslung (Linksys NSLU2) – Binary only, request other platforms from IBM

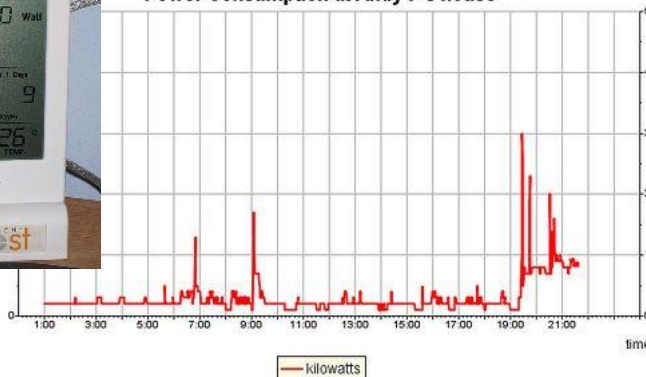


Broader integration

Simple
 Lightweight (CPU,Mem,**Net)
 Data-centric
 Distribution (pub/sub)
 Range of QoS
 => developer/community interest!



Power Consumption at Andy P's house



Clients

Note: although there are a range of options available for developers interested in MQTT, not all of the client APIs listed below are current. Some are at an early or experimental stage of development, whilst others are stable and mature. Also, some may not provide full support for all of the features of the latest MQTT specification (for example, some may only support QoS 0, or not include authentication, etc.)

Check with the provider for the current status of your preferred language implementation; and remember to respect the licenses that different implementations are published under.

Device-specific

- [Arduino Client for MQTT](#)
- [mbed client for MQTT](#)

C

- WebSphere MQ Telemetry provides a C client API
- Lotus Expeditor micro broker includes a C client API
- [IA93](#) - a WebSphere Message Broker SupportPac
- [libmosquitto](#) - a C client library provided with the mosquitto server
- [liblwmqtt](#) - a very lightweight C client

C++

- [libmosquittocpp](#) - a C++ client library provided with the mosquitto server

C# - see .NET below

Delphi

- [MQTT Client Library for Delphi](#)

Erlang

- [erlmqtt](#)
- [mqtt4erl](#)
- [my-mqtt4erl](#) - a more recent fork of mqtt4erl with numerous fixes

Java

- WebSphere MQ Telemetry provides a Java client API
- Lotus Expeditor micro broker includes two Java client APIs
- [IA92](#) - a WebSphere Message Broker SupportPac
- [MQanTT](#)

Javascript / node.js

- [Simple node.js MQTT client](#)

Lua

- [mqtt_lua](#) - Installable as a LuaRock

.NET

- Lotus Expeditor micro broker in
- [MqttDotNet](#)
- [nMQTT](#)

Perl

- [net-mqtt-perl](#)
- [anyevent-mqtt-perl](#) - Perl modules for MQTT using AnyEvent





MQTT.org: Latest News

Eclipse Paho, Open Source, and other news

November 3rd, 2011 - andyp



A big day for MQTT... "the little protocol that could"? 😊

Back in August, we [mentioned the intent to take MQTT to a standards body](#) - that process is in progress.

On November 2, IBM and Eurotech, the originators of the MQTT protocol specification, announced that they were joining Sierra Wireless and the Eclipse Foundation in [a new Machine-to-Machine \(M2M\) Industry Working Group at Eclipse](#). Sierra Wireless have already contributed M2M frameworks and tooling to the proposed [Eclipse Koneki project](#).

Today, November 3, [IBM and Eurotech also announced](#) the donation of Java and C MQTT clients to the newly-proposed [Eclipse Paho](#) M2M messaging project.

More on

<http://mqtt.org/2011/11/eclipse-paho-open-source-and-other-news#comments>



Nützliche Links

Announcement Letter

<http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&appname=iSource&supplier=877&letternum=ENUSZP10-0255>

MQTT – The Smarter Planet Protocol (blog post)

<http://andypiper.co.uk/2010/08/05/mqtt-the-smarter-planet-protocol/>

MQTT 3.1 specification – IBM developerWorks

<http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html>

WebSphere MQ Telemetry Performance Evaluation – SupportPac MP0A

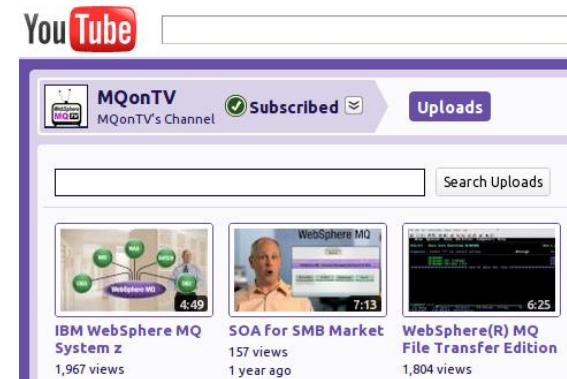
http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24027711&loc=en_US&cs=utf-8&lang=en

MQTT Community

<http://mqtt.org>

We love our users!

- developerWorks
- MQSeries.net community
 - <http://mqseries.net/>
- Global WebSphere Community GWC
 - <http://websphereusergroup.org>
- MQTT – <http://mqtt.org>
- IBMer's Blog on Messaging on developerWorks
- Twitter – @IBM_WMQ and @IBM_Broker
- YouTube – <http://youtube.com/mqontv>
 - <http://youtube.com/WebSphereClouds>
- Podcasts - <http://pipr.co/mqpodcasts>
- Store and Forward – T.Rob's WMQ Security blog
 - <http://t-rob.net/>
- Open, vibrant ecosystem
 - e.g. PyMQI, mosquitto, github projects, etc. etc.





Renate Franken

WebSphere Technical Sales



Mobile 0172-7313217

Email Renate.Franken@de.ibm.com

http://twitter.com/#!/Renate_Franken

Thank you !

