



# Smart Meter App

---

Tobias Fischer und Michael Zauner

# Team

**Tobias Fischer**



**Michael Zauner**



# Was ist ein Smart Meter?

- digitales Messgerät für den Stromverbrauch
- Schnittstelle zwischen Haushalt und Netzbetreiber
- im unseren Fall wurde ein AMIS Zähler von der Netz OÖ GmbH verwendet
  - Kundenschnittstelle wurde von Energie AG zur Verfügung gestellt
- Kundenschnittstelle AMIS Zähler
  - Infrarotschnittstelle
  - liefert jede Sekunde einen Messwert
  - Verschlüsselung der Daten mittels AES-Key



# Welche Daten liefert der Smart Meter?

*Struktur (für beide Tabellen gleich)*

Datenfeld	Bezeichnung	Einheit
ID	Id	
TIMESTAMP	Datum+Uhrzeit	
ACTIVE_ENERGY_PLUS	Zählerstand Energie A+	Wh
ACTIVE_ENERGY_MINUS	Zählerstand Energie A-	Wh
REACTIVE_ENERGY_PLUS	Zählerstand Energie R+	varh
REACTIVE_ENERGY_MINUS	Zählerstand Energie R-	varh
ACTIVE_POWER_PLUS	momentane Wirkleistung P+	W
ACTIVE_POWER_MINUS	momentane Wirkleistung P-	W
REACTIVE_POWER_PLUS	momentane Blindleistung Q+	Var
REACTIVE_POWER_MINUS	momentane Blindleistung Q-	Var
PREPAYMENT_COUNTER	Inkassozählwerk	Wh

*Tabellen*

Tabellenname	Bezeichnung
METERDATA_RT	Echtzeitdaten
METERDATA_HIST	historische Daten

*Beschreibungen*

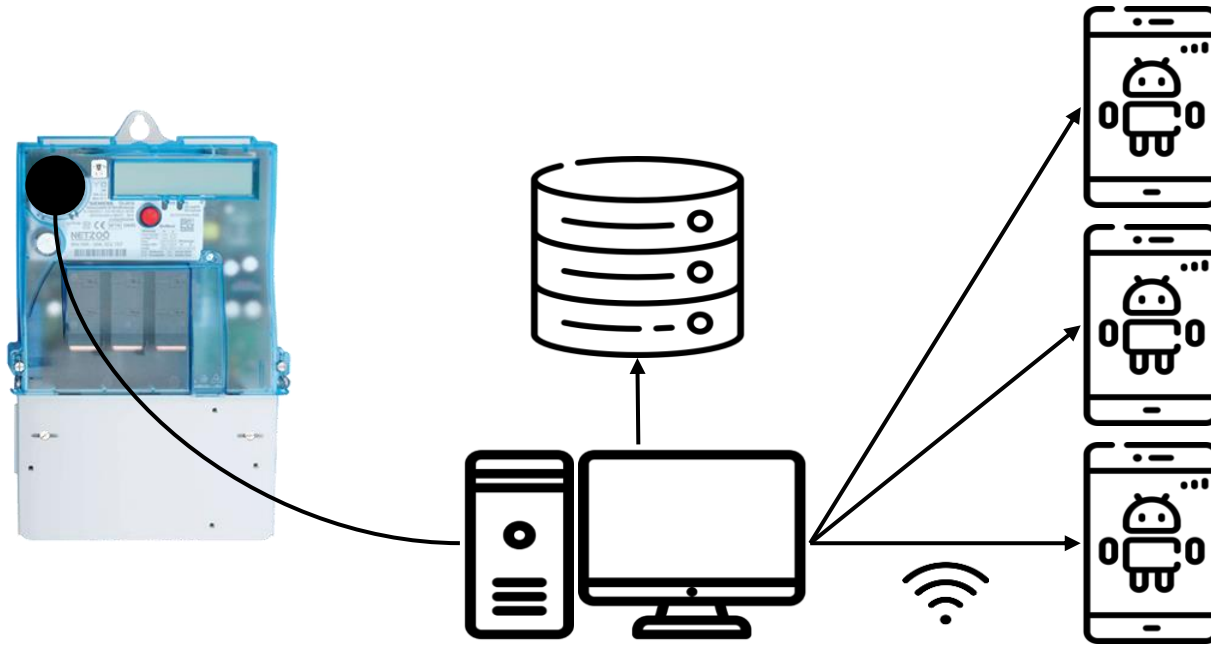
<b>Wirkleistung</b>	kann in andere Energieformen umgewandelt werden
<b>Blindleistung</b>	kann nicht in andere Energieformen umgewandelt werden durch Phasenverschiebung zwischen Quelle und Verbraucher e.g. Kondensator (kapazitiv), Spule (induktiv) Leistung negativ: Verbraucher gibt an Leistung Quelle ab
<b>Inkassozählwerk</b>	Guthaben

# Probleme

- historische Verbrauchsdaten
  - erst einen Tag später am Kundenportal sichtbar
  - Zeitauflösung kann man nicht direkt beeinflussen
- aktuelle Leistung
  - müsste direkt am Zähler abgelesen werden
  - im Kundenportal nicht sichtbar
- Energiemanagement-Lösungen erfordern Echtzeitdaten

*Kundenportal ... Webanwendung des Netzbetreibers (z.B.: Netz OÖ GmbH)*

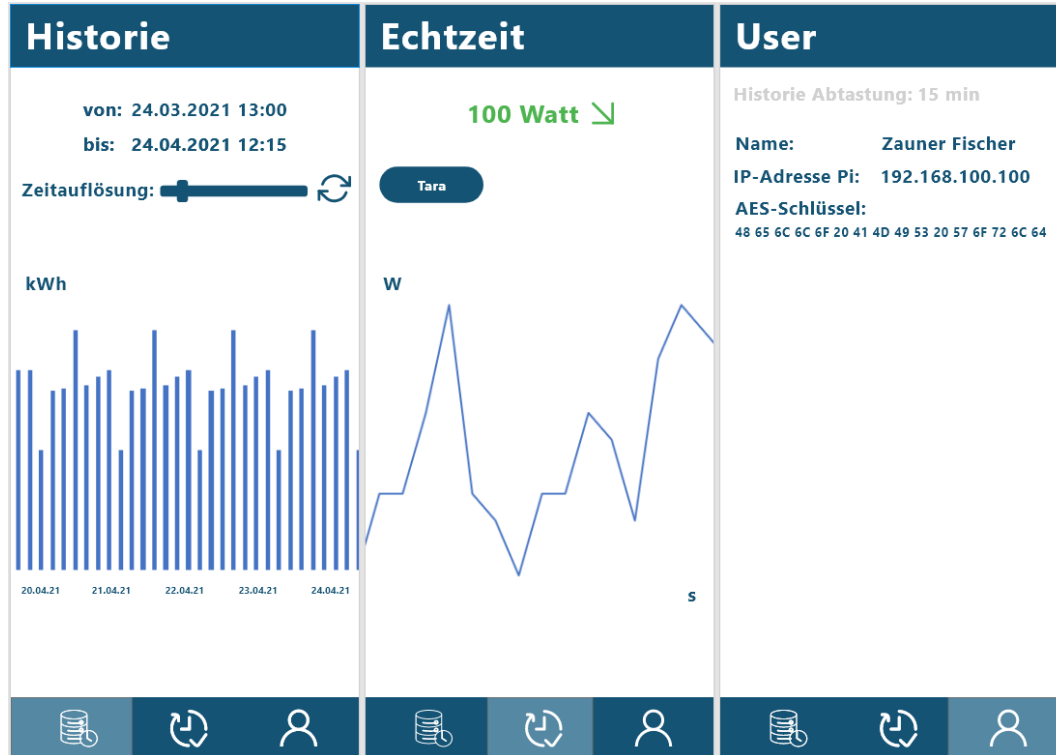
# Architektur



# Lösung

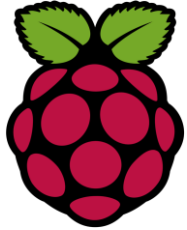
- historische Verbrauchsdaten
  - sind nun sofort verfügbar
  - Zeitauflösung kann selbst eingestellt werden
  - wenig Interesse, wie viel man vorgestern um 12:00 verbraucht hat
- aktuelle Leistung
  - kann mittels App in “Echtzeit” (near real time data) ausgelesen werden
- Erreichbarkeit
  - weltweite Analyse von dem Stromverbrauch zuhause

# Mockup





## Raspberry Pi



**ENERGIEAG**  
Oberösterreich



## Backend



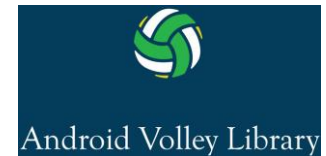
## Frontend



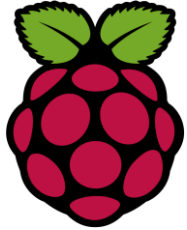
**android**



**MPAndroidChart**  
created by Philipp Jahoda



## Raspberry Pi



**ENERGIE**AG  
Oberösterreich



## Backend



## Frontend



android



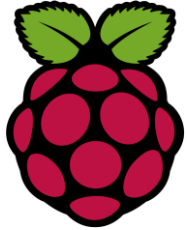
Gson



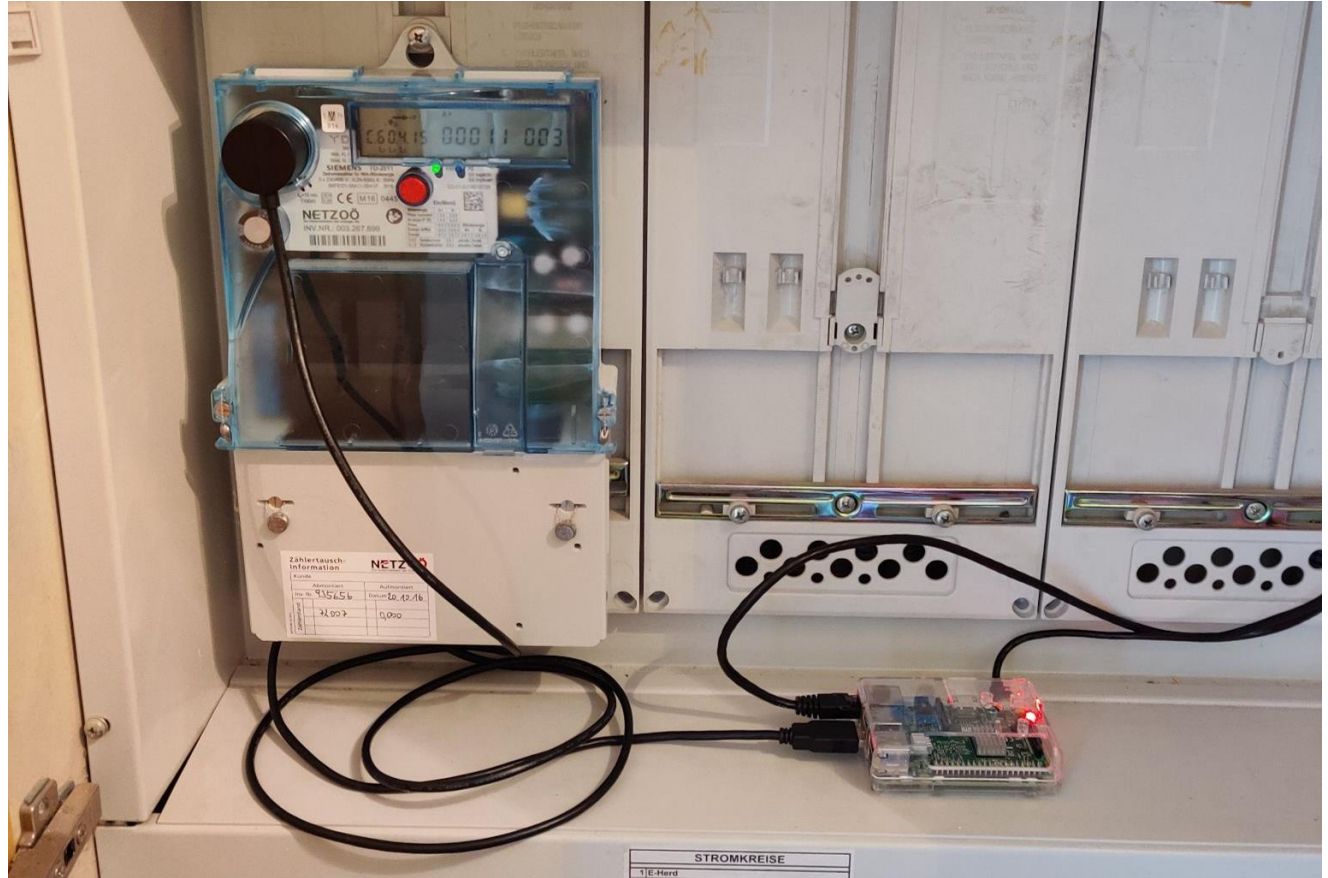
MPAndroidChart  
created by Philipp Jahoda



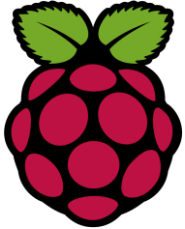
Raspberry Pi



**ENERGIE**AG  
Oberösterreich



Raspberry Pi



**ENERGIEAG**  
Oberösterreich

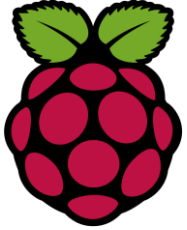


## Energie AG Library

- Verbindung mit Smart Meter
  - entschlüsseln der Daten
  - interpretieren des Datenstroms
  - speichern in lokale SQLite-Datenbank
    - jede Sekunde in Echtzeit-Datenbank
    - alle 15 min in historische Datenbank

```
pi@raspberrypi:~/Desktop/Smart_Meter/slave/mbus-slave-ima-cpp-webdemo $ ./mbus-slave
Time: 2021-06-25 11:11:13
[2021-06-25 11:11:13] Loading config-file 'config.ini' ...
[2021-06-25 11:11:13] -- loaded device1: /dev/ttyUSB0
[2021-06-25 11:11:13] -- loaded AES-Key: 42 49 D6 B1 D2 95 96 ... censored ...
[2021-06-25 11:11:13] -- loaded database_interval_rt: 1
[2021-06-25 11:11:13] -- loaded database_interval_hist: 900
[2021-06-25 11:11:13] -- loaded debug_mode: 0
[2021-06-25 11:11:13] Loading config-file 'config.ini' ... ok!
[2021-06-25 11:11:13] SQLite-Database opened/created successfully
[2021-06-25 11:11:13] Realtime Data table opened/created successfully
[2021-06-25 11:11:13] Historical Data table opened/created successfully
[2021-06-25 11:11:14] AMIS Meter: 2021-06-25 11:11:15 33744663 0 20051064 98668 308 0 53 141 0
[2021-06-25 11:11:15] AMIS Meter: 2021-06-25 11:11:16 33744663 0 20051064 98668 1163 0 53 144 0
[2021-06-25 11:11:16] AMIS Meter: 2021-06-25 11:11:17 33744664 0 20051064 98668 1606 0 53 142 0
```

## Raspberry Pi



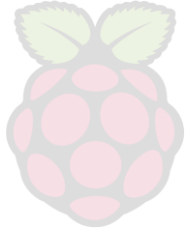
ENERGIEAG  
Oberösterreich



## Wireguard VPN

- ermöglicht weltweite Erreichbarkeit
  - Smartphone und Raspberry Pi müssen im gleichen Netzwerk sein
- sichere Lösung
  - sicherer vor Angriffen
  - Raspberry ist nicht von außen sichtbar

## Raspberry Pi



ENERGIEAG  
Oberösterreich



## Backend



## Frontend

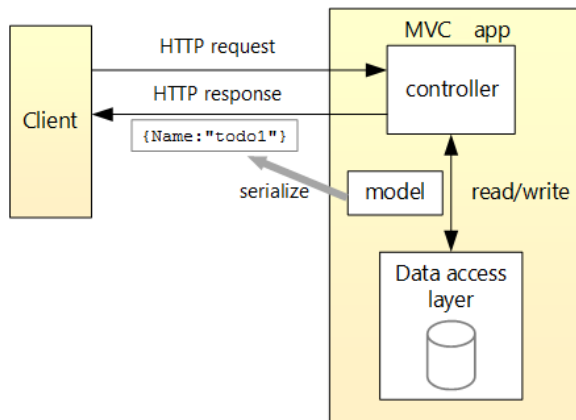


MPAndroidChart  
created by Philipp Jahoda



# Historische Daten

- Datenbankanbindung
  - *Entity Framework Core*
  - Models: Zugriff mittels .NET Objekten
  - eliminiert Abfragen durch SQL
- API
  - *ASP.NET Core MVC*
  - Controller: REST-Schnittstelle
  - Services: Datenaufbereitung
  - DTOs: übertragene Objekte
- Programmiersprache
  - C#

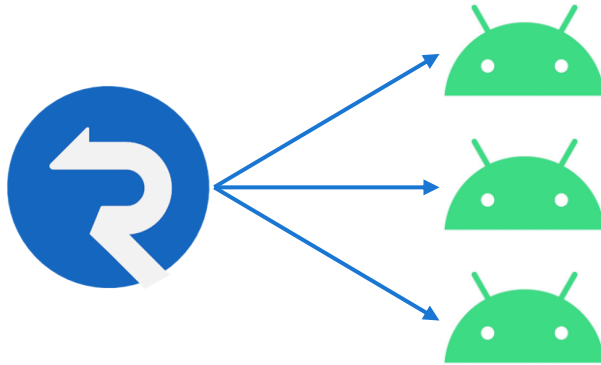


## Backend

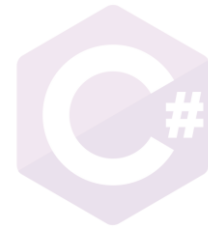


## Echtzeitdaten

- SignalR
  - Echtzeit-Funktionalität
  - Server kann dadurch Daten an Client pushen
  - verbundene Clients erhalten jede Sekunde Echtzeitdaten



## Backend





# Swagger

## Smart Meter API v1 OAS3

/swagger/v1/swagger.json

**SignalR:** /SmartMeter/RealTime -- Entry Method = "AES{AES-Key with no spaces}END" Returning MeterDataRTDto -- Get real time data every second.

### SmartMeter

**GET** /SmartMeter/CheckConnection check connection with backend

**GET** /SmartMeter/HistorySetup/{key} Information for setting up the history view

**POST** /SmartMeter/History History Data

- filtered by timestamp and time resolution
- difference between energy (Zählerstand) calculated
- min/max/avg/unit calculated

#### Parameters

Try it out

No parameters

Request body

application/json

AES Key without spaces, From/To Timestamp and Timeresolution in min

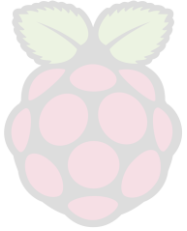
Example Value | Schema

```
{
  "aesKey": "string",
  "fromTimestamp": "2021-06-25T08:01:26.051Z",
  "toTimestamp": "2021-06-25T08:01:26.051Z",
  "timeResolution": 0
}
```

## Backend



## Raspberry Pi



ENERGIEAG  
Oberösterreich



## Backend



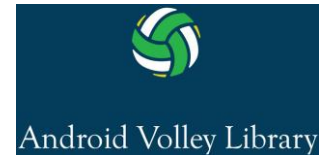
## Frontend



android



MPAndroidChart  
created by Philipp Jahoda



# Allgemein

- Konvertierung Object <> Json

- *Gson Library*

```
json = JSONObject(Gson().toJson(historyDto))
```

```
historySetup = Gson().fromJson(response.toString(), HistorySetupDto.class.java)
```

- API

- *Volley Library*
- HTTP Requests (GET, POST, ...)
- RequestQueue

- SharedPreferences - Einstellungen

- lokale XML Datei

Frontend



android



Gson



MPAndroidChart

created by Philipp Jahoda



Android Volley Library

# MPAndroid Chart

- Open-Source
  - <https://github.com/PhilJay/MPAndroidChart>
  - entwickelt in Java von Philipp Jahoda
- Mehrere Chart Typen
  - Line Chart
  - Sinus/Cosinus
  - Radar Chart
  - .....
- Bugs in BarChartRenderer Klasse
  - NullPointerException
  - IndexOutOfBoundsException
  - View.draw()

## Frontend



MPAndroidChart  
created by Philipp Jahoda



13:22



## Einstellungen



Historie Abtastung  
15

IP-Raspberry  
**192.168.100.15**

AES-Schlüssel  
**15 F1 EA 1C 80 C8 1D D5 19 00 C0 67 E9 5B 73 7A**



Historie



Echtzeit



Einstellungen



08:28

86 %

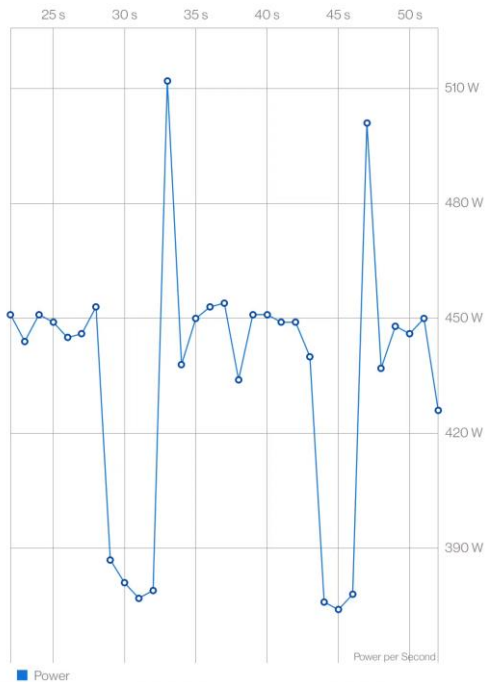
## Realtime

TARA

RESET

08:28:53 ↓

426 W



History

Realtime

Settings

18:04



## Echtzeit

TARA

RESET

00:00:00 —

0 W

Laden...



Historie



Echtzeit



Einstellungen



# Smart Meter App

---

Tobias Fischer und Michael Zauner