FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Cloud Based Distributed Data Acquisition

Exemplified on Power Quality Monitoring

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Martin Gebhard Jenny
Matrikelnummer 0728228

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Pretitle Forename Surname, Posttitle
Mitwirkung: Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle

Wien, 1. Jänner 2001

Martin Gebhard Jenny	Forename Surname

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Cloud Based Distributed Data Acquisition

Exemplified on Power Quality Monitoring

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Martin Gebhard Jenny
Registration Number 0728228

to the Faculty of Informatics

at the TU Wien

Advisor: Pretitle Forename Surname, Posttitle
Assistance: Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle
Pretitle Forename Surname, Posttitle

Vienna, 1st January, 2001

Martin Gebhard Jenny

Forename Surname

Erklärung zur Verfassung der Arbeit

Martin Gebhard Jenny
Silvrettastr. 38, 6780 Schruns

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Jänner 2001

Martin Gebhard Jenny

Danksagung

TODO: Ihr Text hier.

Acknowledgements

TODO: Enter your text here.

Kurzfassung

TODO: Ihr Text hier.

Abstract

TODO: Enter your text here.

Contents

CHAPTER 1

Introduction

TODO: Global todo's:

- Check quotation marks: "text to quote"
- Check quotation marks vs italic text
- Check citation positions: xxx [1]. or xxx[1]. or xxx. [1] ...
- List of Algorithms not populated
- Byte vs byte

CHAPTER 2

Goals

CHAPTER 3

Theoretical Foundation

3.1 Power Quality Monitoring

Electric power is often seen as self-evident. It can be consumed at any time and any point all over the country. But, this standard of comfort implies a lot of effort for the power suppliers. The wide variety of power producers makes it hard for the suppliers to deliver a power grid with stable frequency and voltage. On the one hand, nuclear power plants produce a stable high amount of baseline power, on the other hand highly dynamic techniques like solar or wind energy can make the grid unstable. As a result, the grid has to be balanced. Surplus power has to be compensated, for example by pumping water into big artificial lakes in the mountains. Especially the increasing use of electric equipment in private and industrial environments punctuates the need for a stable power grid.

To fulfill the task of keeping the grid stable, precise measurement is needed at defined points in the power grid. Therefore, the term power quality monitoring describes the the monitoring of important parameters of the power grid, such as frequency or different aspects of the voltage. Furthermore, analysis are made to react to imbalances and to predict further events like the outage of a component of the power grid.

The European Standard EN 50160 defines the parameters of the power supply network that have to be monitored to ensure the given limits by the standard. The standard describes the characteristics under normal operation conditions at a supply terminal from a customer to the public network. The following sections describes some example characteristics of the corresponding Austrian standard ÖVE/ÖNORM EN 50160[**?**]. Furthermore, the International Standard IEC 61000-4-30 defines testing and measurement techniques for power quality monitoring. [**?**]

The goal of the proposed thesis is to develop a measurement system that measures relevant characteristics of a power system for power network supplier. According to the

underlying standard (EN 50160), it is sufficient to capture the voltage on a power line to calculate the relevant properties for a power quality analysis. The scenario used in the thesis consist of synchronized measurement devices, distributed over a wider geographical area, that are connected to a cloud system. The cloud system provides functionality for managing the devices and visualization of the measured data. Furthermore, interfaces can be consumed to use the data with external systems. With this design, the power quality of a power supplier network can be captured.

3.1.1 Parameters

Effective Value

The effective value of the power supply voltage has to be 230 V ± 10% in 95% of every 10 ten minute interval and 230 V + 10%/- 15% in every ten minute interval. The following formula is used to calculate the effective value of the captured signal:

$$U_{eff} \approx \sqrt{\frac{1}{n}\sum_{n=0}^{n} x_i^2} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + x_3^2 + \cdots + x_n^2)}$$

In order to calculate the following characteristics, a Fourier transformation (see 3.1.3 for further information) has to be applied to the captured data.

Frequency

The frequency of the power supply voltage has to be 50 Hz ± 1% at 99.5% of the year and 50 Hz + 1%/- 6% at 100% of the year. After the Fourier transformation, the frequency with the highest amplitude is assumed to be the fundamental frequency f_0.

Voltage harmonics

According to the standard, all voltage harmonics up to the 25^{th} order has to be monitored. Each value has to be calculated over an interval of ten minutes and has to be tested against the values listed in the standard as described in appendix **??**. After the fundamental frequency f_0 is extracted out of the results of the Fourier transformation, the voltage harmonics can be calculated. The frequencies of the corresponding harmonics can be calculated by $f_n = n \cdot f_0$. The voltage harmonics of the desired order can then be calculated by the following formula:

$$V_n = \frac{A(f_n)}{A(f_0)} \cdot 100\%$$

where $A(f_n)$ is the amplitude of the signal at the frequency f_n.

Total harmonic distortion

In addition to the voltage harmonics, the total harmonic distortion (THD) has to be $\leq$ 8%. For this, the voltage harmonics up to the 40^{th} order are aggregated by using the following formula:

$$THD = \frac{\sqrt{\sum_{h=2}^{40} V_h^2}}{V_1} \cdot 100\% = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \cdots + V_{40}^2}}{V_1} \cdot 100\%$$

Summary

Table 3.1 summarizes all parameters that has to be observed by the proposed system.

Parameter	Description	Number of values
U_{eff}	Effective value	1
f_0	Frequency	1
$V_0 ... V_{25}$	Voltage harmonics	25
THD	Total harmonic distortion	1
Total		**28**

Table 3.1: PQM parameters

3.1.2 Measuring Electric Characteristics

TODO: better title

To calculate the selected power quality characteristics, the voltage of the power line has to be monitored. In order to accomplish this task, the analog signal has to be digitized for further computational processing. The signal has to be sampled and each sample has to be quantized to a finite number of bits representing the sample in a digital way. The accuracy of the quantization process strongly depends on the number of bits that such an analog/digital converter (ADC) offers. Assume an ADC represents a sample by B bits, the sample is transformed to a value in the range of 2^B bits, furthermore, the full-scale range R has to be taken into account. As a result of this, the full-scale range R is divided in 2^B steps and the resulting quantization with or quantization resolution can be represented using the following formula[**?**]:

$$Q = \frac{R}{2^B}$$

In the proposed scenario, a ADC with a $R = \pm$ 1200 V = 2400 V and $B = 24$ bits is used. This results in the following quantization resolution:

$$Q = \frac{2400\ V}{2^{24}\ bits} = 0.000143\ V \approx 143\ \mu V$$

Beside having a efficient quantization resolution, choosing the right sample rate is crucial. In order to reconstruct the measured signal, the time between two sample has to be chosen in a manner that on the one hand, unnecessary often taken samples of the same signal level are generated (oversampling) and on the other hand, not too less samples are taken such that the original signal cannot be reconstructed (undersampling)[**?**]. To avoid undersampling, the Nyquist sampling theorem has to be applied: Taken a real signal, that is band-limited to B Hz (only signals with a frequency below B Hz are sampled), can be reconstructed without errors with the frequency R, described by the following formula[**?**]:

$$R > 2 \cdot B$$

In the proposed scenario, we examine total harmonic distortion up to the 40^{th} order or harmonics. According to the standard, the maximal allowed frequency is 50 Hz + 1% = 50.5 Hz. Therefore, the 40^{th} harmonic has a frequency of at most 2020 Hz. The signal is band-limited by a lowpass filter to $B = 5000$ Hz. According to the Nyquist sampling theorem, the sample frequency has to be set to at least $R > 2 \cdot B = 10000$ Hz. Hence, the sample frequency of the measurement device used to evaluate the scenario is set to 20000 Hz.

3.1.3 Transformation from Time- to Frequency Domain

Since a great number of important characteristics of the power quality depends on measuring the frequency (and it's components) on a power line, the transformation form a signal recorded in the time domain to it's frequency parts shall be discussed.

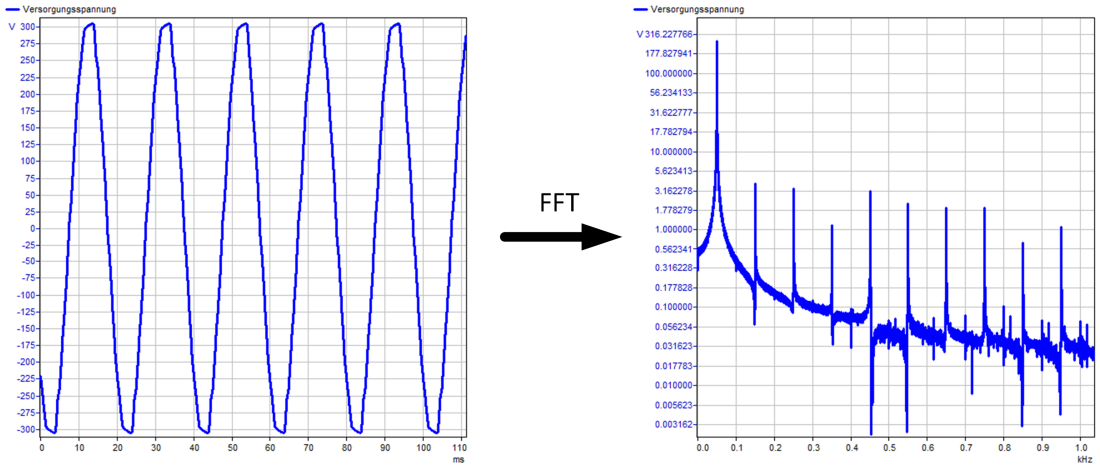

Figure 3.1: Fast Fourier transformation

The most important method used for transformation is the Fourier transform. By using the Fourier analysis, a signal can be represented by its sinusoidal waveforms. In other

words, the output of the Fourier analysis shows the amplitude and phase (cosine and sine components) at every frequency of the original signal. The conversation of a signal from the time- to the frequency-domain can be represented by the Discrete Fourier Transform (DFT). The input for the DFT is a signal of N points and can be calculated using the following formula[**?**]:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk}$$

Implementing the DFT directly results in an algorithm with a complexity of $\mathcal{O}(N^2)$. Therefore, efficient implementations (Fast Fourier Transform - FFT) are available that reduce the complexity to $\mathcal{O}(N \cdot log(N))$. The most common FFT algorithm is the Cooley-Tukey algorithm that uses the divide and conquer approach which can be found in listing 3.1[**?**]. The results of this transformation are 'Bins' representing a sample of the spectrum with the frequency

$$f_i = \frac{R}{N} \cdot i$$

where i is the current Bin-index. Therefore, the output of the FFT has a limited resolution of the frequency. In the proposed scenario, the resolution (or the difference between two Bins) should be at least 0.5 Hz which can be represented as $\Delta f = f_n - f_{n-1} = 0.5$ Hz. Since $n \geq 0$ and $n < N$, we can choose an arbitrary n in this range.

$$n = 2, R = 20000Hz$$

$$\Delta f = f_n - f_{n-1} = f_2 - f_1$$

$$\Delta f = \frac{R}{N} \cdot 2 - \frac{R}{N} \cdot 1 = \frac{2 \cdot R - 1 \cdot R}{N} = \frac{R}{N}$$

$$\Rightarrow N = \frac{R}{\Delta f} = \frac{20000Hz}{0.5Hz} = 40000$$

When looking at the recursive algorithm, it can be seen that the algorithm works efficiently when N is a power of two. Hence we take the next greater power of two for N. $N = 2^{16} = 65536$, which results in a resolution $\Delta f = 0.305$ Hz.

For the sake of completeness, 'Window functions' have to be taken into account. The FFT works under the precondition that the signal is periodic in the examined window (N points). Due to measurement uncertainty and that in practice, the signal is not a perfect sine wave, errors occur in the output. To minimize the error, window function can be applied to the measurement data before FFT, which transforms the input signal to a periodic signal. Different window function (Hanning, Hamming, Blackman, etc.) exists to examine different aspects of the spectrum. Practice has shown that the Hanning-Window is useful in most situations and is also applied in the proposed thesis.

Every sample $x[n]$ in the time domain is multiplied with $w[n]$, which has the following formula[**?**]:

$$w[n] = 0.5 - 0.5 \cdot cos(\frac{2\pi n}{N})$$

A common implementation of the FFT algorithm is the recursive version of Cooley and Tukey. The complex signal input array is splitted in two arrays. One with the even indices, one with the odd indices. After that, the function is called again on this two arrays until the length of the input array is less or equal 1. After the recursion returns, the calculation of the frequency-domain data is made by appliying .... **TODO: cite**

Listing 3.1: Recursive Cooley-Tukey FFT algorithm

```
private void CalculateFFT(ref Complex[] signal)
{
	int n = signal.Length;

	if (n <= 1)
	{
		return;
	}

	//Divide
	Complex[] even = new Complex[n / 2];
	Complex[] odd = new Complex[n / 2];

	for (int i = 0; i < n / 2; i++)
	{
		even[i] = signal[2 * i];
		odd[i] = signal[2 * i + 1];
	}

	//Conquer
	CalculateFFT(ref even);
	CalculateFFT(ref odd);

	//Combine
	for (int i = 0; i < n / 2; i++)
	{
		double kth = -2 * i * Math.PI / n;
		Complex wk = new Complex(Math.Cos(kth), Math.Sin(kth));

		signal[i] = even[i] + wk * odd[i];
		signal[i + n / 2] = even[i] - wk * odd[i];
```

```
        }
}
```

TODO: cite

3.2 Cloud Computing

> "I don't need a hard disk in my computer if I can get to the server faster... carrying around these non-connected computers is byzantine by comparison."
>
> — Steve Jobs, former Apple CEO

3.2.1 Cloud Characteristics

The essential characteristics of cloud computing are as follows[**?**][**?**]:

On-demand self-service

Allows users of the cloud to consume capabilities from it. Common capabilities are network storage, server time or applications. The big difference compared to other computing models is, that the capabilities can be consumed as needed without complex interaction with the cloud provider.

Broad network access

All of the capabilities can be accessed over the network by standard mechanisms. Furthermore, thin and thick client platforms are supported (from mobile devices up to workstations).

Resource pooling

The cloud provider's resources are pooled together to serve multiple consumers. End users of the cloud mostly need not to know the location of the data center hosting the cloud environment (often users can choose the country or a certain datacenter). The different physical and virtual resources are assigned and reassigned dynamically on demand.

Rapid elasticity

For rapid elasticity and scalability, the provisioned capability can be automatically scaled up and down. For the end users, capabilities are often presented as unlimited.

Measured service

For billing purposes and transparency for both the cloud provider ant the end user, measuring capabilities are applied. Furthermore, automatic control and optimization of the resources are performed based on this measures.

3.2.2 Service Models

By definition, the cloud system can be divided into the following three service models[**?**]:

Software as a Service (SaaS)

Software is provided to the users within the cloud infrastructure. To use the software, the services provided by the cloud can be accessed by either a web interface or is integrated into software installed locally. The advantage of this service model is, that the users consuming the services of the cloud rely on the underlying hardware (physical servers or network interfaces) and software (operating systems or databases)[**?**].

Examples: Google Mail, Google Docs[**?**].

Platform as a Service (PaaS)

Software created by users can be deployed to the cloud. For this, the cloud provides libraries, services and tools to support the users by developing a software with a programming language supported by the cloud. Like before, the underlying hard- and software is maintained by the cloud operator[**?**].

Examples: Google App Engine, Windows Azure[**?**].

Infrastructure as a Service (IaaS)

The users can consume computational power from the cloud. This includes the use of servers and operating systems, storage solutions, networking resources or additional capabilities like processing power. Again, the underlying hard- and software is maintained by the cloud operator[**?**].

Examples: Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3)[**?**].

Figure 3.2 shows the three main service models and the responsible entities for the involved services.

TODO: https://blogs.technet.microsoft.com/yungchou/2010/11/15/cloud-comput

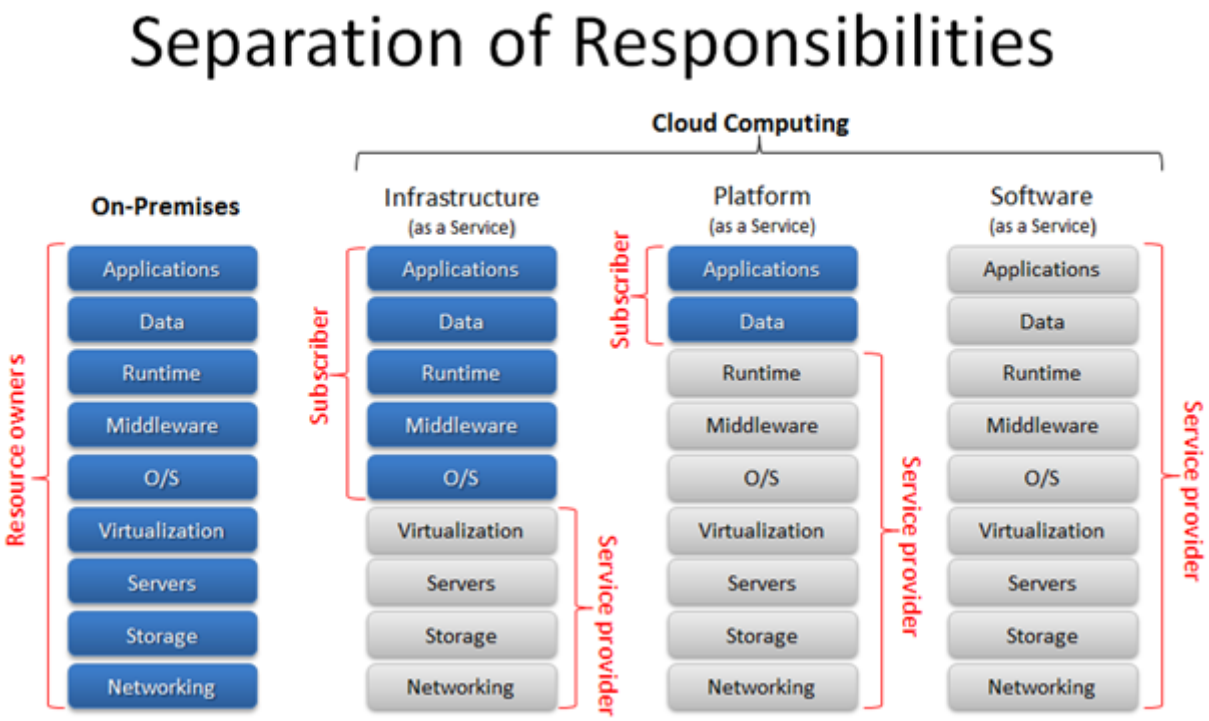

Figure 3.2: Service models[?]

Furthermore, the following subset of service models can be derived from the Everything as a Service (XaaS) aspect:

Data Storage as a Service (DaaS)

DaaS can be seen as enhancement of IaaS. The main idea behind this model is to provide storage solutions to users. Often supported services are databases (RDBMS or other types) and file storage.

Example: Amazon Simple Storage Service (S3)[?].

Human as a Service (HuaaS)

Services, that rely on information generated by a crowd of people. Human intelligence is used to fulfill tasks that are provided as a service to the users. Persons in the crowd use their tools, skills or knowledge to solve a task or a small subset of a task. The cloud systems aggregates the single results and provide as a service to the users.

Example: Amazon Mechanical Turk[?].

Other models

Since there is no strict separation between the stated service models, there exist various combination and modifications of them. The following listing shows some additional classification [**?**].

- Database-as-a-Service
- Security-as-a-Service
- Communication-as-a-Service
- Management/Governance-as-a-Service
- Integration-as-a-Service
- Testing-as-a-Service
- Business Process-as-a-Service

3.3 Internet of Things

When looking at the past, a clear separation of digital and physical industries can be observed. Since 1990, the Internet gave new opportunities to non-digital industries by creating new business models. Around 2005, the term "Web 2.0" came up as a summarization for Social Media, Open Source, Crowd sourcing, etc, allowing users to contribute to content in the Internet. With "Web 3.0" (around 2015), the concept of the "Internet of Things (IoT)" was introduced with new business models like "Digitally Charged Products" or "Sensor as a Service" [**?**].

The Internet of Things introduces hybrid solutions that merge physical objects and digital services, with the vision to connect every object and location to the Internet. To fulfill this task, objects are equipped with computational intelligence (computers) that can be access over the Internet (or other objects) to read and write information from and to a physical object. The object itself is called a "Thing", hence the Name "Internet of Things" and is now a composition of both the physical and digital world [**?**].

By adding computational intelligence to a physical object, the value of the "Thing" is increased in many ways. From an academic view, the value creation can be seen as five layers called the "Value-creation Layers" and will be explained in the next section [**?**].

3.3.1 Value-creation Layers

Figure 3.3 shows the five layers of value-creation within the Internet of Things. To get a better understanding of what the five layers are responsible for, a smart light bulb is used as an example.

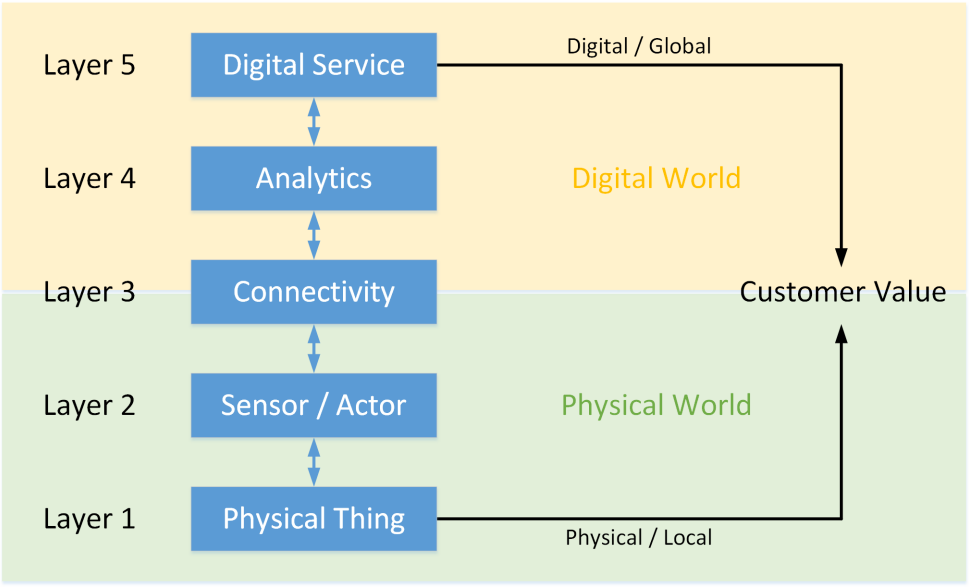

Figure 3.3: Value-creation Layers[?]

Layer 1: Physical thing

This layers describes the physical thing, e.g. a light bulb. A thing is always bound to a location and has a direct benefit for the user (e.g. serving light). Since the location is fixed, it can only act in an immediate environment.

Layer 2: Sensor/Actuator

Layer 2 adds sensor and actuator elements to the thing. By adding a minicomputer to the physical object, the sensors can measure data and the actuators can deliver local services. In the proposed example, the sensor could measure the presence of a person and then switch of the light bulb.

Layer 3: Connectivity

This layer connects the physical thing and the sensors/actuators to the Internet. Within this layer the device gets accessible via the Internet by, for example, a radio module that is connected to the local wireless network. By using modern technologies, this step can be done at marginal costs.

Layer 4: Analytics

In this layer, the collected data from the sensors are collected and stored for further processing (e.g. plausibility checks or classification). Additional web services can be integrated to generate actions for the actuators or add additional information to the

sensor data. Usually this is achieved by using a cloud system. In this layer and additional value is added to the "stupid" thing. In the example this layer could derive motion patterns from the light bulb.

Layer 5: Digital service

This layer combines all the previous layers and offers a functionality as a digital service. As an example, the light bulb could be exposed to users all over the world via a website or a mobile application. On this layers new digital business model patterns can be applied.

By applying the previous layers to a physical object, the value of the thing is more than the sum of its parts. Connectivity to the Internet nowadays can be made at small costs. One important thing is, that the layers are not independent of each other. The flow of information has to pass the five layers always in both directions in order to ensure the additional value of the product.

3.3.2 Business model patterns

By applying the concept of the Internet of Things to non-digital industries, new business model patterns can be derived. One new business model is called "Digitally Charged Products". Within this model, six components can be found that that link digital services to physical objects. An object becomes *charged* with sensors and digital services to generate a new value proposition. This new object is a hybrid bundle of both, the physical and the digital world.

In the next section, the six components of "Digitally Charged Products" are explained:

Physical Freemium

A physical object is equipped with free digital services that a customer can use unlimited. "Physical Freemium" describes that some costumers will select additional fee-based (or premium) services after some time that can be invoiced by the asset supplier.

Digital Add-on

Like "Physical Freemium" this components describes that customers by inexpensive thing and over the time, the customer can purchase or activate other digital services for a higher price.

Digital Lock-in

"Digial Lock-in" describes the principle that only original components are compatible with a already bought system. The manufacturer of a device can decide which add-ons for a product can be applied. This concept limits compatibility but enhances warranties issues.

Product as Point of Sales

This component describes that a physical product can be used for digital advertising. Mechanisms like collecting loyality points or record user activity and tracking of the environment can be applied.

Object Self Service

With "Object Self Service" a thing can place orders on the Internet without any user interaction.

Remote Usage and Condition Monitoring

By submitting information about the environment in real time, things like error prediction can be established. In the non-IoT approach, this was only possible with high costs. With IoT, such prediction mechanisms can be deployed at low cost anywhere in the world.

Beside "Digitally Charged Products", "Sensor as a Server" can be derived as a further business model pattern. This pattern brings the data in the foreground and makes advantage that there is a market for sensor data. Data that measured by things is no longer only processed for this thing, but collected together with data from other things and offered on a market.

3.3.3 Comparision with "Industry 4.0"

On an industrial perspective, the Internet of Things opens the marked for smart, connected and automated systems that can be used in production facilities and manufacturing systems. Because of this, Germany introduced the term "Industry 4.0" together with an initiative of the German Federal Ministry of Education and Research [**?**].

IoT can affect the industry in three different ways: [**?**]

1. Use IoT data to improve internal and external processes
2. Apply the "Digitally Charged Products" business model pattern to update their product portfolio
3. Use IoT technologies to introduce others to the IoT ecosystem

According to experts, IoT will be a game changer in industries that uses B2C[1] and B2B[2]. IoT offers methods and patterns to save costs, improve outputs and enable new technologies. One great German example of such a game change was done with cyber-physical production systems (CPPS) in Germany. Scientific terms like preventive maintenance,

[1] Business-to-Consumer
[2] Business-to-Business

remote control or other analysis and services can now be easily accomplished. As a further example, IoT can be integrated in the whole supply chain to track and trace logistics in a company. [?]

3.3.4 Comparision with "Industrial Internet of Things"

By introducing the concept of "Industry 4.0" and the ongoing trend to use IoT devices in an industrial context, the need for a more reliable and secure concept is shown. The term "Industrial Internet of Things" or "IoT", summarizes methods and concepts to achieve the this goal. IIot covers topics like security-critial or privacy-sensitive data, which have no big impact in class IoT devices, but play a big role in an industrial environment[?]. As an example of such industires aerospace, energy or healthcare can be used. In such industries, a error or failure in a system can yield to dangerous situation or the threat to somebody's life[?].

In industries, an ongoing trend of replacing programmable logic controllers (PLCs) by cyberphysical systems (CPDs) can be observed. A CPD is a programmable device that controls or monitors a physical process and is one of the fundamental building blocks of smart factories. Furthermore, this devices have to have a communication with other parts of the manufacturing process by closed internal networks, or the Internet[?].

By the use of security, privacy and standardization concepts, IIoT provides a solution for smart factories. Further aspects like social or legal issues has to be taken into account by implementing IIoT devices[?].

3.3.5 Security concerns

TODO: chapter really needed?

3.3.6 IoT protocols

While there are stable and grown concepts for building IoT devices by using existing technologies, IoT protocols are still developing. When looking at communication technologies for IoT or IIoT devices, mostly wireless transmission modes are used. Common examples are Zigbee, WiFI or Bluetooth. All this technologies allow smart devices to connect to the Internet for submitting data to web servers or cloud-based services[?].

By choosing the right protocol for an application, the (probably) limited resources of a smart device have to be taken into account. Parameters like computational power, power supply (battery) and network bandwidth have a big impact on the decision[?].

When looking at such IoT endpoints two services can be seen often[?]:

1. APIs that can a user of such an infrastructure can use to read data from the devices.

2. Asynchronous nodes that can be used for communication between devices and end-user applications.

The following sections should give a short overview about the currently common used Internet-of-Things protocols[**?**].

CoAP

The Constrained Application Protocol is a protocol suitable for constrained-devices (devices with limited network access or battery powered devices). It uses a subset of the HTTP methods (GET, POST, PUT and DELETE) for a synchronous and asynchronous request/response communication

As base technology, CoAP uses UDP because of the less overhead compared to TCP. To ensure reliability the protocol has a two bits header field in each packet that defines the desired Quality of Service (QoS) level:

1. Confirmable: A message must be acknowledged either synchronously or asynchronously with an ACK.
2. Non-Confirmable: A message needs not to be acknowledged.
3. Acknowledgment: The acknowledge message for a request has has to be acknowledged.
4. Reset: A message stating that a request could not be processed.

CoAP does not include any build-in security, thus Datagram Transport Layer Security (DTLS) can be used. DTLS for UDP follows the same principle than TLS or SSL for TCP **TODO: ref**. Other ways to secure a CoAP connection are available but not stated here.

MQTT

The Message Queue Telemetry Transport is a protocol really suitable for IoT applications. It supports asynchronous publish/subscribe mechanism over TCP.

In an MQTT environment a broker (server in a traditional context) provides topis that can be consumed by clients. A client can subscribe and unsubscribe from a topic and hence gets updated if a new message for this topic is available. This mechanism makes it useful for devices with limited resources.

MQTT allows three QoS levels to ensure reliability:

1. Fire and forget: A message is sent only once and an acknowledgment is not needed.

2. Deliver at least once: A message is sent at least once, furthermore, an acknowledgment of this message is needed.

3. Deliver exactly once: By using a four-way handshake the delivery of the message exactly one time is ensured.

For security concerns, TLS or SSL on top of TCP can be used to encrypt message.

XMPP

The Extensible Messaging and Presence Protocol, a quite old protocol originally developed for chatting, can also be used for IoT communication. On top of TCP, it provides asynchronous publish/subscribe and also synchronous request/response messaging. XMPP can be used for near real-time communication and can be extended by XMPP Extension Protocols (XEP) which makes it useable for different applications.

Messages are transmitted as XML documents which introduces an additional overhead due to the structure and thus need more computational power on the smart device.

QoS issues can not be captured by the protocol itself, security can be established by using TLS or SSL.

RESTFUL Services

Compared to the previous mentioned protocols, the Representational State Transfer, is an architectural style. REST is a very common approach in the Internet for consuming and providing APIs by using the HTTP methods GET, POST, PUT and DELETE. It uses synchronous request/response message that can be interpreted by nearly every common HTTP server.

Messages are exchanged in the JSON format which makes it more lightweight compared to XML messages. Furthermore, all known HTTP mechanisms like cashing, authentication and content type negotiation can be used by REST.

For security TLS or SSL can be applied. Due to the fact that TLS or SSL introduce additional overhead, unique authentication keys in every HTTP message header can be used to get a level of security.

AMQP

Advanced Message Queuing Protocol can make the use of different transport protocols on top of TCP. It provides asynchronous publish/subscribe messaging with a store-and-forward mechanism that ensure reliability.

Three different message-delivery guarantees can be applied to the messages:

1. At most once: A message is send once, regardless if it is delivered or not.

2. At least once: A message is definitely send once (or more) and is delivered.
3. Exactly once: A message is definitely send once and is delivered.

By using TCP, TLS or SSL can be applied to ensure security.

Websockets

Websockets, based on TCP, establish a session between the server and the client by a handshake. A Websocket connection is a HTTP connection until the handshake is done, after that the HTTP headers are removed and a asynchronous full-duplex connection is established. This mechanism reduces overhead of classic HTTP connection and provides a real-time connection between server and client.

To use publish/subscribe concepts, the Websocket Application Messaging Protocol (WAMP) can be used. Again, TLS or SSL can be used to achieve a level of security.

Comparison

Table 3.2 sums up the relevant points for each previously described protocol[**?**]:

Protocol	Transport	QoS	Architecture	Security
CoAP	UDP	✓	Request/Response	DTLS
MQTT	TCP	✓	Publish/Subscribe	TLS or SSL
XMPP	TCP	-	Request/Response Publish/Subscribe	TLS or SSL
REST	HTTP (TCP)	-	Request/Response	HTTPS (TLS or SSL)
AMQP	TCP	✓	Publish/Subscribe	TLS or SSL
Websockets	TCP	-	Client/Server Publish/Subscribe	TLS or SSL

Table 3.2: IoT proctol comparison

3.3.7 Impact on the thesis

For the proposed thesis, MQTT will be used as protocol for data exchange between the Power Quality Meters and the cloud. The publish/subscribe principle fits into the requirements for the developed system. The PQMs can publish measured data and one or more cloud endpoints can subscribe to this messages. Since MQTT is widely-used in IoT and IIoT environments, tools and libraries can be found for nearly every programming language.

By the use of topics, different types of messages can be used: Status messages from the devices, measurement blocks, single measurements and to a limited amount (due to the protocol itself) streaming measurement data.

Furthermore, security and QoS levels can be applied and examined in the simulation to develop a system that suits the needs for differnent environments (bandwidth, storage, etc.).

CHAPTER 4

System Design

4.1 Requirements

4.2 System Overview

Table 3.1 summarizes the 28 PQM parameters that has to be processed in the proposed system. To monitor this parameters, each PQM system has to have a sample rate of 20 kHz. To reduce data, every ten seconds the power line signal is captured for one second ($t_{n+1} - t_n = 10s$). Figure 4.1 shows this procedure. For visualization reasons, the plotted sine wave is not an actual snapshot of the power grid.

According to this properties, three different types of data processing can be derived:

1. Capture data on device, upload raw signal to the cloud for processing
2. Capture data on device, process data on device, upload results to the cloud
3. Option 2 with *Dynamic Monitoring Frequency Scaling*

Figure 4.2 show the binary representation of the data packets that are transferred from a PQM system to the cloud.

Assuming that every parameter is transferred over the network as IEEE 754 double number (8 Byte) and that every measurement has a prepended timestamp (also double), following assumptions can be derived for the needed bandwidth (without protocol overhead):

1. Timestamp + Signal = 16 Bytes, 20 kHz sample rate
   $\Rightarrow$ 312.5 kB/10s $\Rightarrow$ **2.57 GB/d**

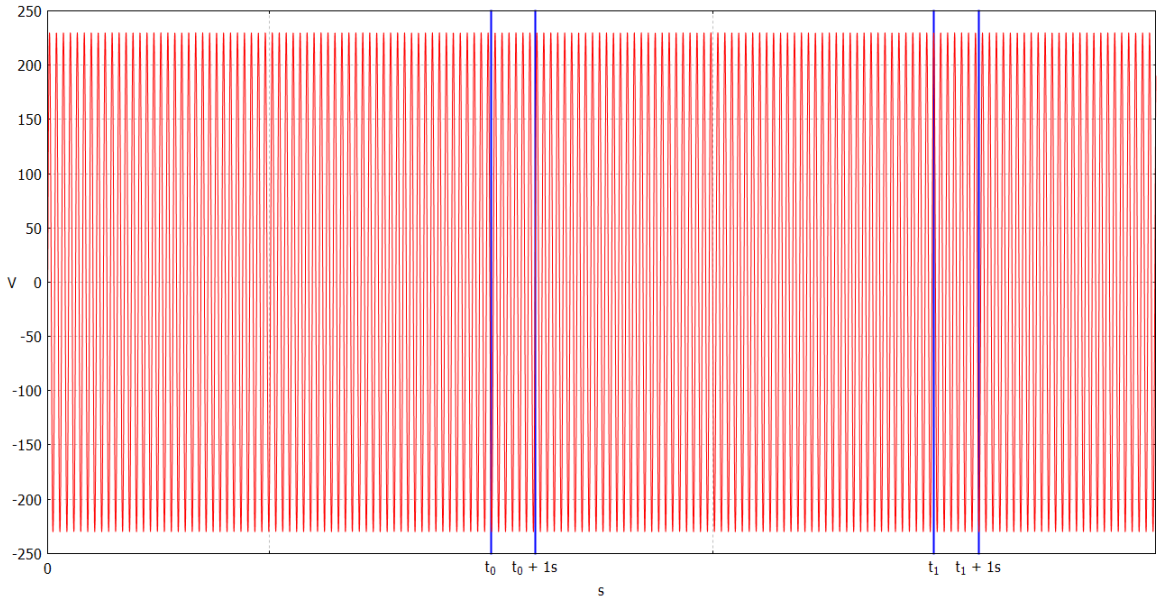

Figure 4.1: Signal capturing

StationID	Timestamp	Value 1	Value 2	...	Value n	CRC-16
2 Byte	8 Byte	8 Byte	8 Byte	...	8 Byte	2 Byte

Figure 4.2: Binary data packet

2. Timestamp + 28 Values = 232 Bytes, 20 kHz sample rate
   $\Rightarrow$ 232 B/10s $\Rightarrow$ **1.91 MB/d**

3. todo ...

4.3 Hardware Architecture

4.4 Software Architecture

4.4.1 Synchronization

Synchronizing a distributed measurement system is essential. Although an not synchronized system delivers correct data that holds true for the point of measure, synchronizing theses systems allow to extract further knowledge out of the captured data. In the use case, selected for this thesis, power quality monitoring provides good arguments for synchronization. If somewhere in the power network an error occurs, it could be possible that this error is propagated throughout the network. Since the electric network and the infrastructure around it consists of resistive, inductive and capacitive load, a cascading

of a fault is possible. With an adequate synchronization, it is possible to track the error propagation with a distributed measurement systems.

The testing equipment used in the proposed scenario support three different synchronization mechanisms. The radio based DCF77 time signal, the satellite based GPS signal and the Ethernet based SNTP protocol. The following section explains these three mechanisms in detail and evaluates which type suits most for this use case.

For the sake of completeness, two further synchronization modes exist. Distributed clock over EtherCAT and Q.sync. The first mode is part of the EtherCAT protocol that is implemented in the used modules, the second mode is a proprietary bus that connects multiple controllers. Since the different power quality measurement systems are spread across a wider geographical area, this two modes can't be used.

DCF77

DCF77 is a radio based time distribution services. It uses a long wave radio transmitter with a carrier frequency of 77.5 kHz. Due to the fact that the transmitter is located in Germany, it covers a maximum range (by using the proper detector) of approximately 2400 km, which is in fact whole Europe. Since DCF77 is driven by atomic clocks, the received radio signal is less accurate than the them[**?**]. Every minute, the time information (and additional information like civil warnings[**?**] and weather data[**?**]) is transmitted amplitude modulated (AM) and phase modulated (PM) in Binary Coded Decimal (BCD). In the current message the information about the next second is transmitted, furthermore, the start of a minute can be detected. It is possible to detect leap seconds and the switch to daylight-saving time [**?**]. Considering accuracy of the DCF77, the decoding mechanism is crucial. For the use in watches only the amplitude modulated signal is decoded which results in accuracy of +5ms up to +150ms. By using better antennas, this values can be enhanced to +5ms up to +15ms. To gain more accuracy, the phase modulated signal has to be decoded as well. This method results in an exactness of $\pm$2ms[**?**].

Since the signal of a DCF77 receiver cannot be connected directly to the measurement devices, the signal has to be converter inside the receiver to a suitable format. As an example of such a format, the "IRIG B003" standard is described in more details here. The Inter Range Instrumentation Group (IRIG) specifies various standards regarding transmitting time information over via a serial time code format. The name of the described standard refers to a format with 100 pulses per second (B), Pulse width code (0), no carrier (0) and Binary Coded Decimal - BCD, Straight Binary Second of Day - SBS (3). The date information is transmitted as a sequence of seconds, minutes, hours, days, years, control functions and time of day [**?**]. Regarding the used testing equipment, DCF77 receivers can provide the IRIG B003 signal either straight as Transistor-Transistor Logic (TTL) signal into the measurement device or over the RS-485 bus.

GPS

The Navstar Global Positioning System (GPS) is a satellite based time distribution service. In normal operation mode, it consists of 24 geometrical space slots with at least one operational satellite in it. Each satellite sends it's time information, generated by an atomic clock, and it's position to the receiver stations down on earth with a frequency of 1.57542 GHz. To control and observer the status of the GPS, the Operational Control System (OCS) is used to communicate with the satellites via ground antennas. The OCS is responsible for various tasks including telemetry, monitoring of different parameters and uploading of navigation data. Considering accuracy, GPS offers two services. On the one hand, the Standard Positioning Service (SPS) that is available to the civil public with less accuracy and on the other hand, the Precise Positioning Service (PPS) available to the military of the USA and it allies with high accuracy.[**?**]. Due to this fact, the proposed system and the used testing equipment can only make use of the SPS.

After receiving the time information from one or more satellites, the data can be passed to the testing equipment. For this, the received information is transformed to the National Marine Electronics Association (NMEA) 0183 format. Since NMEA was designed as interconnection format of different devices used in the marine, the format consists of various sentences representing features of the connected devices. To use NMEA as protocol for transmitting time- and position data, the two sentences $GPRMC (Recommended Minimum Navigation Information) and $GPGGA (Global Positioning System Fix Data. Time, Position and fix related data for a GPS receiver) can be used. The data is transferred as ASCII text and can be decoded directly by knowing the format of the corresponding sentence. NMEA uses, like IRIG, a serial bus for communication[**?**].

Beside NMEA, some GPS receivers are also able to provide information via the previously described IRIG B003 format.

SNTP

The Simple Network Time Protocol (SNTP), as a subset of the Network Time Protocol (NTP), provides time information in a network to clients. If a client wants to synchronize it's internal clock with a clock placed in the network (SNTP server), it has to send a message to this server. In the response, three timestamps are transmitted. The client timestamp when the message was sent, server time when the message was sent and server time when the message was received. With the information in this message and the timestamp when this message was received by the client, the exact time can be applied to the client[**?**]. To synchronize a client's clock, the offset and delay of the client relative to the server is computed. By the used algorithm (especially because of the 64 bit integer arithmetic) a client has to be at least 34 years in the past and at most 34 years in future in order to get synchronized by the SNTP server[**?**].

Conclusion

TODO: add some content here

4.5 Network Architecture

4.6 Security Aspects / Threat-Modeling

TODO: check spelling

CHAPTER 5

Simulation

5.1 Tools

5.2 Simulation Design

Figure 5.1 shows the general simulation setup that is used to test the software components of the proposed system.

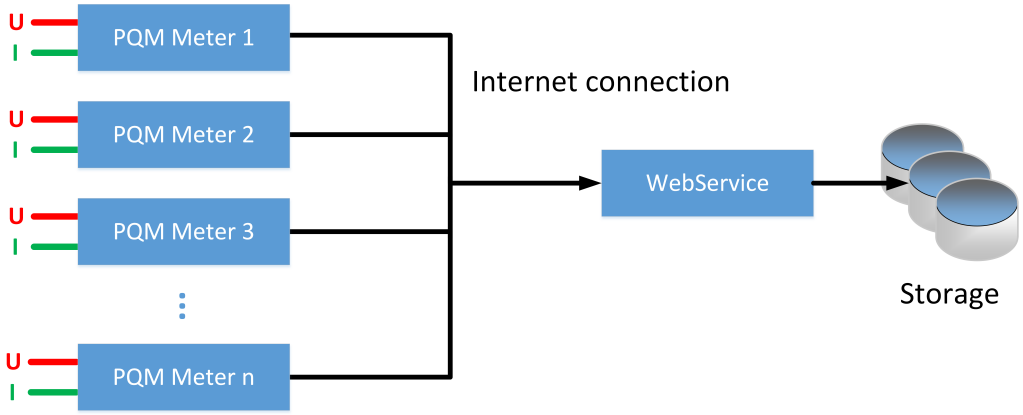

Figure 5.1: Simulation design

Different parameters of the simulation can be changed, such that different environments can be simulated and evaluated against each other. The following enumeration describes the parameters that can be adapted as input data to the simulation:

- Bandwidth of the Internet connection

  - Wired connection: 100 MB/s, 50 MB/s, 20 MB/s, 10 MB/s
  - WiFi connection:
  - Cellphone connection: LTE (), 4G (), 2G (), GPRS ()
- Number of PQM meters
- Number of Webservices
- Transmitted data
  - Raw data
  - Preprocessed data
- Storage
  - File archive
  - SQL database (MySQL)
  - NoSQL database (CrateDB)

Modifications of the above mentioned parameters have an effect to the following parts:

- Server side storage
- Network load, network costs
- Server side throughput

Table 5.1 shows the expected results of the simulation:

5.3 Evaluation

5.4 Impact on System Design

Parameter	Modification	Impact		
		Bandwidth	Server storage	Throughput
Bandwidth				
Number of PQM meters				
Number of web services				
Transmitted Data				
Server side storage	File			
	SQL			
	NoSQL			

Table 5.1: Simulation results

CHAPTER 6

Implementation

CHAPTER 7

Evaluation

List of Figures

List of Tables

List of Algorithms