Blockchain in the European Utilities Market

Frank Blau

29.20.17

*Confidential for Poesis Consulting*

The purpose of this white paper is to identify opportunities for further business and technological exploration for Poesis Consulting in the use of Blockchain technology for the European electrical utilities market. A technology introduction to blockchain topics and assessment of regional initiatives is presented. Finally, a high-level roadmap for Poesis to consider in developing a Blockchain consulting competency is also presented.

A sample contract for an Access Point Identity use case is included in Appendix A.

**What is Blockchain?**

The simple Blockchain design pattern of an encrypted, immutable shared ledger and Smart Contract applications is becoming an interesting opportunity for an array of innovative use cases for electrical, water, natural gas and other utility providers. Developers and architects are investigating and developing capabilities that were previously only imagined within the framework of expensive, monolithic applications. Use cases can now be explored and prototyped at a fraction of the cost of traditional software solutions.

Utility companies are also held to very high regulatory standards for reliability and delivery. The designation as “Critical Infrastructure Components” confers a requirement to provide fully redundant and secure services with 100% uptime for consumers.

Blockchain is also a relatively (4-5 years) new technology. There are still a lot of standards, protocols and software development tools that need to align into a coherent strategic platform to provide the secure and resilient transactional competence that is required of today’s software. But the time to investigate and understand this platform’s potential and patterns is today, as many of the leading software players already are already heavily invested in this adoption.

The biggest challenge for Blockchain in the utilities industry is not a technological challenge. The challenge is that there are so many possible use cases that it is sometimes overwhelming to winnow out the ones that will provide the most business value and fit the design pattern for Blockchain applications. A use case in Appendix A represents just one of several intriguing opportunities for Blockchain-based development.

**Mining Methods (Proof of Work vs. Proof of Stake)**

Today, Blockchain is limited in its ability to execute secure transactions at throughput rates that even come close OLTP platforms. Current Blockchain transaction validation speeds are dictated by both technology (hardware, network speeds, etc…) and Block mining methods. This mining serves two purposes:

* To verify the legitimacy of a transaction;
* To create new digital currencies by rewarding miners for performing the previous task.

The two main methods for block validations are Proof of Work (PoW) and Proof of Stake (PoS). They both accomplish the same function, but by very different methods.

When you want to validate and secure a transaction, the following actions take place in a **Proof of Work** scenario:

* Transactions are bundled together into a block;
* Miners verify that transactions within the block are legitimate;
* To do this, miners solve a mathematical puzzle known as proof-of-work problem;
* A reward is given to the first miner who solves each block’s problem
* Verified transactions are stored in the public blockchain

The newer **Proof of Stake** model being deployed in several Blockchains today, creates a much faster, deterministic approach to Blockchain validation. This is critical, as the PoW model has a fundamental imbalance between the amount of electrical power needed to compute the “Proof” and the mined output value. Proof of Stake remedies this imbalance by implementing a different approach to validation.

When you want to validate and secure a transaction, the following actions take place in a Proof of Stake scenario:

* The creator of the transaction block is chosen deterministically, by the “stake” or wealth of the creator.
* Since there is no “reward” in Proof of Stake, the PoS miner receives the transaction fees associated with the block.

The characteristics and differences between the two methods are best described with the following illustration:

Source: https://blockgeeks.com/

**Public vs. Private Blockchains**

Blockchains can be either private or public networks. The largest public blockchain application is the Bitcoin cryptocurrency with millions of users and transactions. Both public and private share the same design patterns of immutable transactions in a shared ledger, but the main difference is in the composition of the network. In a public network, anyone can join and receive a copy of the transaction ledger. The size and distribution of a public blockchain make it nearly impossible to corrupt or falsify.

The difference in a Private (also known as Permissioned) network is that the participants are only people that have been given access to the network. This results in a smaller, more controlled network, but one that also requires the enterprise security to protect the integrity and functionality of the Smart Contracts and transactions stored on it. An example of a secure permissioned network is IBM’s Hyperledger Fabric service on the Bluemix Cloud network.

Regardless of the type of network for deployment, it is a good practice to use a test network to test your Contracts and methods, prior to publishing on either a public network or a secure permissioned one.

**Smart Contracts**

“The digital codification of a conventional contract, with the same authority and legal status as a written contract.“[[1]](#footnote-1)

Smart Contracts are objects that live on the Blockchain, each with a unique class and instantiated address. These contracts have constructors for new instances, as well as a rich array of functional and operational language features. Smart contracts offer the ability to define immutable contractual objects with secure and encrypted access methods. An example of a Smart Contract is included in Appendix A.

**Blockchain Development Tools**

While Blockchain applications can be coded in variety of different languages, the predominant ones are Solidity, Go, and IBM’s Hyperledger Fabric. These tools are available as free open source development environments for designing and testing Smart Contracts. There are also several simple Blockchain simulators for testing out code before actually publishing them.

**Blockchain Platform Vendors**

After developing prototypes, the next obvious step in the software development lifecycle is the ability to deploy Blockchain applications in a secure production ecosystem. The good news is that the major cloud platforms (Amazon Web Services, Microsoft Azure, IBM Bluemix and Google Cloud) support some version of Blockchain as part of their services suite.

**The European Electric Utilities Market**

The European Parliament resolution “Towards a New Energy Market Design” of 2016 highlighted the need for increased integration of renewable energy sources and prosumers in the market. The resolution also highlighted mobilization of storage as well the supply/demand of electricity resources as key issues facing the industry today.

In the European electricity market, there are both Wholesale and Retail markets, each with their own geography and services. From local offers on the retail level to large multi-national operations across the continent, there are complex markets that cover the spectrum from real-time markets to long-term wholesale contracts. Wholesale markets are generally the generators, suppliers and industrial consumers of electric power. The retail market includes suppliers who purchase electricity from generators and sell it consumers.

The Renewable Energy Directive[[2]](#footnote-2) compels countries to prioritize renewable sources for energy and to make their networks available to these sources. The recommendations in this directive, including statistical transfers of renewable energy, joint renewable energy projects and joint renewable energy support schemes are well suited to the semi-trusted relationship models and highly secure design patterns found in Blockchain.

Some of the challenges in these markets include:

* Increased demand for electricity eg: electric vehicles
* Distributed and variable resources increasing proportionally
* Energy service companies and aggregators innovating markets
* Support for renewable sources, even at a higher cost
* “Prosumers” that both generate and consume power
* Technological developments like smart meters, time-variant pricing, consumer storage
* Increased infrastructure between states

These challenges represent an opportunity for the development of innovative approaches for addressing the use cases that are rapidly appearing in this ecosystem.

Regionally there are strategic plans already adopted for the modernization of renewable energy services in Vorarlberg.[[3]](#footnote-3) Within these plans are many of the same topics that have been addressed at the EU level, such as mobility, security and pricing elasticity.

**Use Cases**

There are a multitude of potential use cases for Blockchain and decentralized applications in the generation, distribution and consumption of electrical power. While a full enumeration would be beyond the scope of this document, a few potential and under development use cases are listed below.

* Peer-to-Peer energy trading
* Identity Management
* Asset Tracking
* Smart Utility Monitoring

In addition, there are several notable Blockchain development efforts being undertaken by companies in the Energy Utility sector:

* UK Startup Electron is working on a Blockchain platform to reduce friction for customers that wish to switch providers quickly.[[4]](#footnote-4)
* Austrian Startup GridSingularity is developing blockchain applications to support energy data analysis, smart grid management, green certificate trading, investment decisions and energy trade valuation. [[5]](#footnote-5)
* Germany-based Ponton GmbH is designing products for the Energy sector using Blockchain in their Enerchain service.[[6]](#footnote-6)
* In the US, the Brooklyn Microgrid projects is one of the first community-powered electric grids run with Blockchain technology.[[7]](#footnote-7)

**Blockchain Competency Roadmap for Poesis Consulting**

So how to incubate and deliver the competencies and tools for world-class Blockchain consulting services for electrical utilities based out of Vorarlberg? While there are still many unknowns, both technical, environmental and economic, there should be some identifiable waypoints to traverse in this journey. Note: most of the items on this list apply generically to a Blockchain product roadmap in any industry.

1. A clear and comprehensive assessment of potential partners, customers and regional players that Poesis would want to engage in projects. While the wholesale supplier market locally is dominated by VKW, there are undoubtedly more actors like Kleinwasserkraft[[8]](#footnote-8) in the retail sector that could be explored for collaboration and marketing.
2. Development of competencies in Blockchain narratives for sales and marketing. This is more than just a good “elevator pitch” (although that is good to have too). This is a slightly deeper dive of technical rhetoric that will allow Poesis have a working knowledge of the current and near-future Blockchain landscape. The customers interested in these technologies tend to be very savvy about the potential and pitfalls of Blockchain.
3. Create working prototypes, with well-designed UI/UX components that can be used to showcase Blockchain decentralized applications (dApps) to potential partners and clients. Using technologies like react.js, solidity, and a test network, it is “easy” to create very functional applications that are easily customized to both marketing identities and customer-specific use cases.
4. Create relationships with a network provider with strong Blockchain services offerings for deployment of Smart Contracts in an enterprise-class infrastructure. IBM’s Bluemix product is the strongest regional platform for these services.
5. Consider the development of a unique framework approach in Blockchain development that can differentiate Poesis from other consulting services exploring this market. This can be as simple as adaptation of a vendor’s approach (such as Monax[[9]](#footnote-9) or Truffle[[10]](#footnote-10)) to demonstrate a branded, repeatable and tested process for delivering Blockchain products.
6. Create a branded “Incubation Team” strategy to pitch to clients that are hesitant to make Blockchain a core IT product right now. There are many successful models for these internal teams and their enablement. Build something that exploits both the experience and agility of boutique consulting within a more traditional client’s operation.
7. Build a brand for Poesis by publishing articles, having outstanding and fresh website content, hosting workshops, and attending conferences to socialize the brand.
8. Partner with the FH Vorarlberg and the University in St. Gallen for grooming and recruiting of software engineers that have the skills to join in the development of Blockchain products. This includes providing internships, and possibly PhD sponsorship for students with these interests.

**Summary**

There are many challenges in the creation, transportation and delivery of electric utilities in the EU today. Traditional IT ecosystems will still play a pivotal role in the delivery of “Critical Infrastructure Services” at the wholesale level. But there will be increased exploration of new techniques, technologies and tools for more agile, elastic and sustainable products and services in both wholesale and retail markets. The decentralized technologies in the Blockchain stack will play an important but disruptive role in this transformation. These new technologies are highly accessible to utility companies and consumers today for the realization of unique use cases.

There are ripe opportunities to create a core consulting competency around Blockchain products in Vorarlberg. Whether at the strategic business consulting, or at the tactical product development level, a nimble boutique firm like Poesis could be well positioned to be a profitable player in the regional Blockchain landscape.

**Appendix A: Sample Solidity Contract for Utility Access Point Management**

To address both mobilization and prosumer empowerment, the concept of access point identity management is demonstrated below. The key design pattern for this prototype is the creation of a Blockchain based identity management contract and an access point interrogation and update contract. The following is a prototype example of a Solidity-based Smart Contract for management of an Access Point assignment and Meter Reading for retail electric utility services and customers.

There are 2 smart contracts included in the code below: 1 for the Customer, and 1 for the Access Point. As each is created the transaction is recorded on the Blockchain and an address is assigned.

The **UtilityCustomer** contract has a function for assigning a Customer to an Access point (which also gets recorded as a Blockchain transaction). Customers can be re-assigned to new access point after being created and still maintain their digital indentity and associated metadata.

The **AccessPoint** contract has functions for reading and writing current and assigning new meter reading values. There is logic to make sure the new meter reading is higher than the current one.

For simplicity’s sake, there is currently no user security demonstrated in this model, but Solidity has methods to validate authorized use of functions by certain users. Future development could include the implementation of financial reconciliation and refunds for both the provider and consumer of the electricity.

Example Code:

pragma solidity ^0.4.14;

/// This is the first UtilityCustomer Contract

/// Author: Frank Blau

/// Date: 29.10.17

/// The only customer attribute is a name string, but others can be added

contract UtilityCustomer {

struct Customer {

string custName;

bool current;

bytes32 streetAddress;

bytes32 postalCode;

address accessPointAddress;

}

/// Event functions are way to log actions to the console for debugging

/// This event echoes back the customer name and the old and new AccessPoint addresses

event moveAccessPoint (string \_custName, address \_oldAccessPoint, address \_accessPointAddress);

address utilityAddr;

address customerAddr;

address meterAddr;

mapping(address => Customer) customers;

/// This is the constructor function for the UtilityCustomer contract

/// This is only called when the Contract is first created.

function UtilityCustomer (string \_custName) {

utilityAddr = msg.sender;

customers[customerAddr].custName = \_custName;

}

/// This function assigns a customer to an access point

function assignAccessPoint (address \_customerAddress, address \_accessPointAddress) {

address oldAccessPoint;

oldAccessPoint = customers[customerAddr].accessPointAddress;

customers[customerAddr].accessPointAddress = \_accessPointAddress;

moveAccessPoint(customers[customerAddr].custName, oldAccessPoint, \_accessPointAddress);

}

}

/// This is the first AccessPoint Contract

/// Author: Frank Blau

/// Date: 29.10.17

contract AccessPoint {

struct smartMeterReading {

uint32 meterReading;

}

/// this event echoes the current reading of an address to the console

event showReading (address \_AccessPoint, uint32 \_currentReading);

address accessPointAddr;

mapping(address => smartMeterReading) meterReadings;

/// this is the constructor for the AccessPoint contract

function AccessPoint () {

meterReadings[accessPointAddr].meterReading= 0;

}

/// this function returns the current meter reading for an AccessPoint address

function getCurrentReading (address \_readMeterAddress) constant returns (uint32 meterReading) {

meterReading = meterReadings[\_readMeterAddress].meterReading;

}

/// this function sets the current meter reading for an AccessPoint address

function setReading (address \_readMeterAddress, uint32 \_newReading){

uint32 currentReading;

currentReading = getCurrentReading(\_readMeterAddress);

if (currentReading < \_newReading) {

meterReadings[\_readMeterAddress].meterReading = \_newReading;

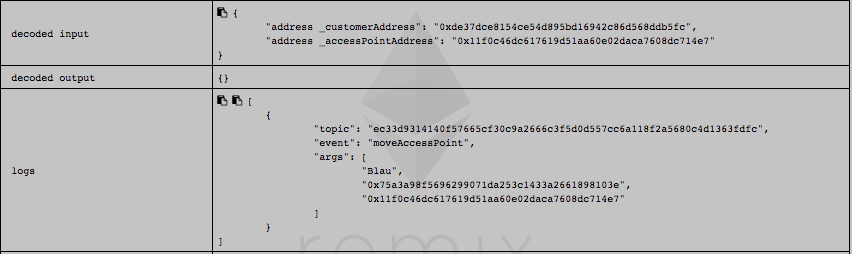
showReading(\_readMeterAddress, \_newReading);

}

}

}

Here is a screenshot of the log showing the assignAccessPoint function transaction:



1. https://www.bayshorenetworks.com/blog/can-blockchain-be-used-to-create-secure-industrial-iot-networks [↑](#footnote-ref-1)
2. https://ec.europa.eu/energy/en/topics/renewable-energy/renewable-energy-directive [↑](#footnote-ref-2)
3. https://www.vorarlberg.at/english/vorarlberg-english/water\_energy/energy/targets.htm [↑](#footnote-ref-3)
4. https://www.smartdcc.co.uk/ [↑](#footnote-ref-4)
5. http://gridsingularity.com/ [↑](#footnote-ref-5)
6. https://enerchain.ponton.de/ [↑](#footnote-ref-6)
7. http://www.brooklynmicrogrid.com/ [↑](#footnote-ref-7)
8. http://www.kleinwasserkraft.at/en/hydropower-vorarlberg [↑](#footnote-ref-8)
9. https://monax.io/ [↑](#footnote-ref-9)
10. http://truffleframework.com/ [↑](#footnote-ref-10)