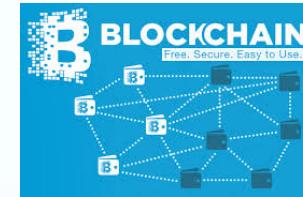


Proof of Understanding - Babelchain



How Machines learn to communicate

Benedikt Herudek (benedikt.herudek@gmail.com)
June 2016
Blockchain Meetup Berlin

Abstract

- Imagine we could extend and generalize the bitcoin model into a digital information and message exchange framework allowing parties in a distributed network to exchange information with a consensus on dataexchange formats. This approach wants to investigate how this could be done in a bitcoin way. A.o. adjustments we would suggest replacing the hashing work miners do with machine learning algorithms so called translators would do.
- This presentation will suggest to :
 - Participants in a distributed Network can 'translate' messages from one to another protocol with the help of Machine Learning Algorithms
 - Successful Translators will receive awards in a cryptocurrency like Miners in a Bitcoin Network get rewards for Executing Proof of Work
 - (partially) replace a Bitcoin Proof of Work by a proposed Consensus Mechanism Proof of Understanding
- The resulting System will:
 - translate Messages from different protocols into each other
 - arrange immutable translated Transactions into a Blockchain
- This 'Thought Experiment' is to be understood as a proposal to implement a prototype for a new kind of Integration Layer. One of several open question is if we the resulting Blockchain will have the same level immutability feature as the Bitcoin Blockchain

Agenda

check

Introduction

- The Problem: Disparate Machine Protocols can't communicate without translation
- Bitcoin versus 'classical' Integration Approach

Understanding

- Format Handshake
- Content Handshake
- Action Handshake

Translator Machine Learning (as equivalent to Mining Hash Power in Bitcoin)

- picking common data exchange formats
- Generating common data exchange formats
- Mapping different data exchange formats

Implementing

- Smart Contract: leave proof of X as is
- Pre req for Proof of Work: partially replace proof of work
- Replace Proof of Work

Immutability Proof of Understanding

- Signing Immutability
- Working Immutability

Summary

- Comparing Bitcoin Blockchain and Proof of Understanding Blockchain ('Babelchain')
- Once again: What we claim & Why it all matters ...

check

Introduction

Insert short description (the problem, why it is a ‘Blockchain’ problem also

Machines can't (hardly) communicate !

che

The Enterprise Integration Version

- Consider the case of a CRM and Billing System and a Provisioning System in a Telecommunications enterprise.
 - Client order are taken in with the CRM Sy
 - stem, the order is send to the Provisioning System, The Billing System will be responsible to generate a bill to the
 - All three Systems (in fact there are many more) know the concept of a Client and of Products, but they have different ways of talking about them and their internal Data Models differ.

The IoT Version

- Devices have different message Formats but need to be able to communicate, e.g. The iPhone needs to be able to take over from all remote controls and talk to the Samsung TV
- Devices can get connected via Software Solution in a centralized or distributed Cloud Solution
- Problem is increasing with number of devices and exposure to the end users who will not like large enterprises be able to account for the Integration effort between disparate protocols

The 'Broken Trusted Middle Man' Solution

Suggest an Industry Standard or follow a Standard set by a large player

But:

- Large players dont stick to industry standards
- With the number of usecases, applications and devices, innovations and the globalized character of the internet industry standards are impossible to keep up

The gated community version

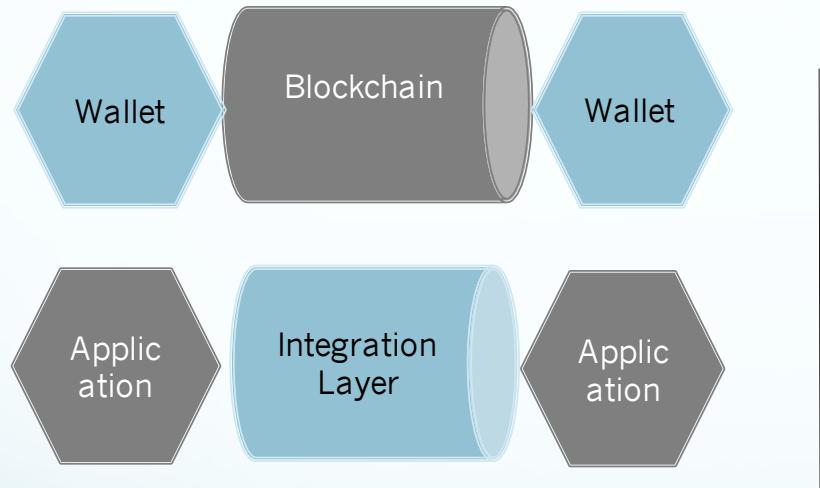
- Facebook / Apple closing networks
- They deliver a service ... one language to allow them all to communicate

The Blockchain (to be !) version

- What data Standard will the Blockchain expect and how can connected end points communicate ?
- Will there be different standard for different Blockchains ?
- Any **Transaction System** has to solve the question of a **common data format** either by **enforcing** (Bitcoin and e.g. most Integrations patterns like SOA and alikes) or **negotiating** (Babelchain)

The ‘naïve’ way of copying Bitcoins Approach to integrate Endpoints

Bitcoin is able to integrate endpoints (wallets, miners) in a large distributed network seemlessly without any integration effort



Blockchain:

- Transactions (Blockchain) are **permanent & primary**
- Endpoints (Wallets) are **derived & ephemeral**

Conventional Systems

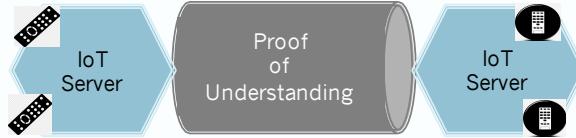
- Transactions (EBS) are **derived & ephemeral**
- Endpoints (Applications) are **permanent & primary**

- Following Bitcoin would mean to create a standard for handling transactions
- Even though Existing Systems often do not keep Transactions but the end points permanent, they use a similar approach
 - existing SOA Style (Microservices, API Industry) follow the same pattern in dictating a message exchange format.
 - often, these are Industry, Technollgy Vendor or Enterprise Standards

What can we learn from Bitcoin and Blockchains?

Bitcoin

How can machines securely come to an agreement about the status of transactions without resorting to a trusted third party ?



Miner's Hashing Power

Rewards for Hashing Blockheaders

Immutable Transactions

Banks, Credit Cards

Computing Power

Use the CPU power of a large distributed System

Incentives
Cryptocurrency reward chance on fulfilling certain

Consensus
Establish Consensus in a distributed (unmuteable) System

Middleman
Mediating Instance, that oversees both sides

And permanently stick between parties

Internet of Things

How can machines come to an agreement about the meaning of a message without resorting to a trusted third party ?

Predictive Machine Learning Capabilities

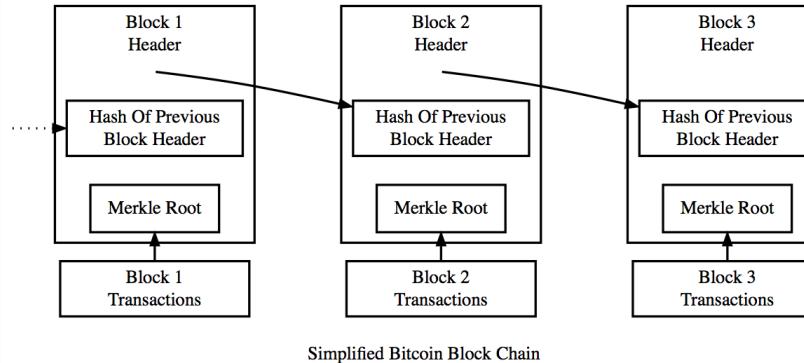
Rewards for Translating Messages

Common Language

Standards (technology, vendor, industry)

Proof of Work: create a shared & immutable bitcoin transaction ledger

- The “**previous block hash**” field is **inside the block header**
- When the parent is modified in any way (eg fraud), the parent’s hash changes and hence the child hash changes and so on
- The **cascade effect** would require that much computing power that deeper layers are practically **immutable**



With courtesy from the
bitcoin book of
Andreas Antonopoulos
and documentation from
bitcoin.org

4.Independent **selection**, by every node, of the chain with **the most cumulative computation demonstrated through proof of work**

Mining & Fees align Miner’s (financial) with Network (immutable transaction ledger) interest

1.Independent **verification** of each **transaction**, by every full node, based on a comprehensive list of criteria

Mining via ‘hash puzzles’

- process of hashing the block header repeatedly, changing one parameter (‘nonce’), until the resulting hash matches a specific target
- The hash function’s result cannot be determined in advance, nor can a pattern be created that will produce a specific hash value
- Hence, the only way to produce a hash result matching a specific target is to randomly modifying the input until the desired hash result appears by chance.

3.Independent **verification** of the new **blocks** by every node and assembly into a chain

2. Independent **aggregation** of transactions into new blocks by mining nodes, coupled with **demonstrated computation through a proof of work algorithm**

check

Understanding

Insert Short description (the idea)

'Meaning' for Machines: Syntax, Semantic, Pragmatics

Meaning is defined as a key value pair of **Format** & **Content** potentially triggering an **Action**:

```
<ZAP_TV>
  <to>
    <receiver>TV in my chinese hotel
      room</receiver>
    <path> TV Cloud Server</path>
  </to>
  <from>
    <sender>my smart phone remote app
    </sender>
    <path> smart phone Cloud Server</path>
  </from>
  <body>
    <device> TV <device>
    <command> change channel
    </command>
    <channel> 12</value>
  </body>
</ ZAP_TV >
```

Message **Format** like file with fields, XML messages, csv files, binary files

```
<body>
  <device> ...<device>
  <command> ...
    </command>
  <channel> .</value>
</body>
```

There are variable slots in which values with **Content** will go

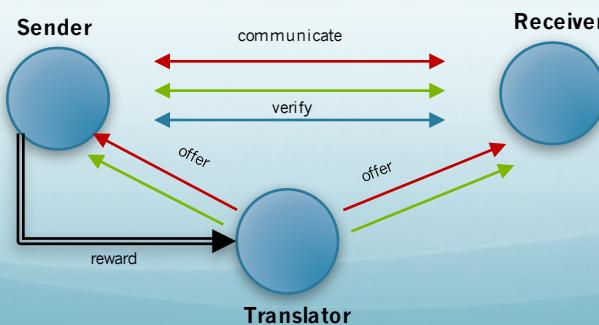
TV
change
channel
12

Action in the real world or a digital form (photo, fingerprint of a machine state)

It can be verified by **humans** or **advanced machine algorithms**

Examples:

- (Digital fingerprint of) changed state in TV
- (picture of) Temperature measuring room on 20 degree



There is an assumption here a
Translator can make format
suggestions ! Hang on a couple
of slides !

Sender

1. A sender creates a message with content (key value pairs in potentially a hierarchical structure) and sends it to the Receiver

...
<device> TV<device>
 <command> change channel
 </command>
 <channel> 12</value>
...

3. Translators offer different data formats and offer those to sender and receiver

...
<thing> ... <thing>
 <action> ...
 </action>
 <what_action>
...</what_action>
...

2. Unless the Receiver ('believes to') understand the message, the message gets offered to the Babelchain Translators with a bounty

...
<device> TV<device>
 <command> change channel
 </command>
 <channel> 12</value>
...

Receiver

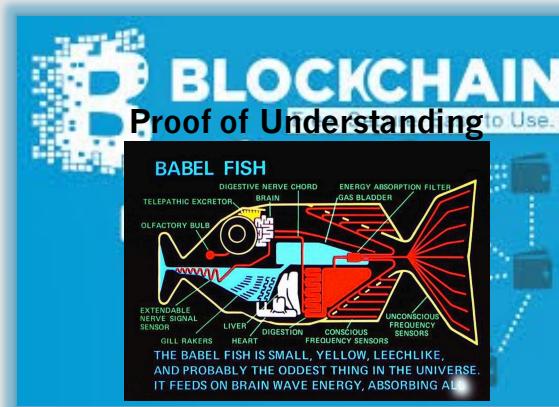
Format Handshake

4. Sender & Receiver pick one message formats they 'believe' they would be able to understand.

Repeat
Format Handshake

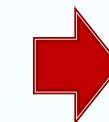


Failure



5. The Translator that generated the format that is first picked by both sender and receiver is marked for the the part of the reward for the format translation.

Agreed Format



4. Sender & Receiver pick one message formats they 'believe' they would be able to understand.

Success

Action Handshake



See this as a confirmation that the agreed message format is understood by both in the same way

Content Handshake

Sender:

3. The Sender will have to agree to the solution, if he doesn't another message format has to be found and the format handshake round of the quiz will have to start

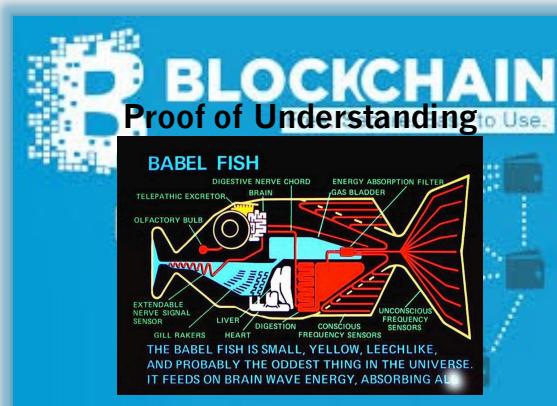
4. If Sender and Receiver agree on the correct assignment, then whoever of the Translators offered the correct solution will get marked for the Content part of the reward..



Receiver:

1. Translators will compile all possible combinations of value assignments to the format.

...
<thing> TV<thing>
 <action> change channel
 </action>
 <what_action>
 12</what_action>
...
...



2. The Receiver will pick the combination, that 'works for him'

4. If Sender and Receiver agree on the correct assignment, then whoever of the Translators offered the correct solution will get marked for the Content part of the reward..

Repeat Format Handshake



Failure

Agreed Content for Format



5. If no Action Handshake is mandated

- both sign the hashed messaged header
- rewards will get paid out

...
<thing> TV<thing>
 <action> change channel
 </action>
 <what_action>
 12</what_action>
...
...

Success



Transaction / Action Handshake (optionally)

See this is a test, that there really was no misunderstanding during format & content handshake Humans could do it, if simple and we good, we automate it

Action Handshake

Sender:

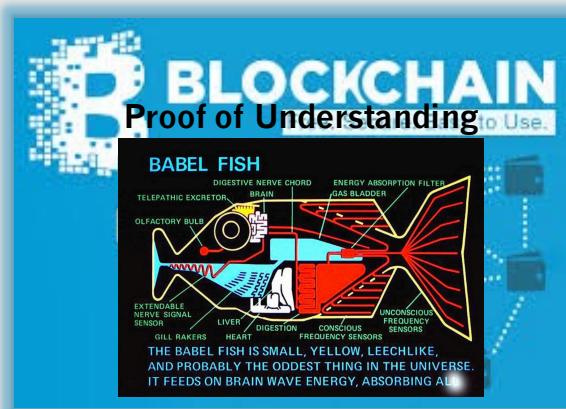
2. Receiver upon his or Senders initiative has to offer evidence of the action. Digital picture of a If the handshake is confirmed

Repeat Format / Content Handshake



Failure

1. Sender or Receiver can choose to open this round before continuing.



Receiver

- Actions handshake can reconfirm the format & Content Handshake but can be time-consuming (if humans check) or difficult for machines to undertake.
- It can typically be useful 1st time two systems handshake and messages can trigger significant and sensitive actions, like opening a banksafe or launching a rocket
- Choosing for this handshake will depend on probability to fail (determined by sender or receiver) and the risk implied upon a misunderstanding

Agreed Action on Content for Format



3. Evidence will be checked by machines or humans, upon success:
 - both sign the hashed messaged header,
 - rewards will get paid out

Success



Transaction

Proof of Understanding kicks on only on request !

- A lot of Message exchanges in the form of Transactions between Machines will work ‘just fine’. After all, we have communicating Machines already for a while without any of the Mechanisms described here. Neither Sender nor Receiver might see a need to explicitly agree with handshakes on terminology nor might they think their information exchange is relevant enough to write them into an immutable ledger. They might be connected to a communication partner for a while and be sure the communication is working fine or messages they exchange might be based on fire & forget to rather float messages e.g. about statuses of sensors, where not the individual signal but the overall trend is relevant. Such Transactions do not need to be written into the Babelchain nor might a message format need to get clarified. In fact, if all messages would need to go through a complex proof of understanding mechanism and all messages would need to go to the Babelchain Blockchain, the network might quickly get overloaded.
- These observations are somewhat similar to the [lightning network](#) in bitcoin, especially as to the goal to off load irrelevant transaction from the Blockchain and the proof of work / proof of understanding algorithm. For this part of the Babelchain, the ‘Babelchain chatter channel’ the general strategy is, that all messages / transactions go through the peer to peer network, but do not go to the Proof of Understanding and into the Blockchain, unless either Sender or Receiver call upon that action. This could be done like in bitcoin lightning by signing a copy of the message received from the other party, who already has signed the message. In principle, whoever initiates the Proof of Understanding would need to pay the Translator. He might be able claiming back part of the bounty with the Communication Partner, since that partner also might have been interested in the message. One would introduce a more sophisticated punitive System in case all handshakes failed. Imagine the Sender called upon a Proof of Understanding, after Sender & Receiver exchanged messages for a while. The Sender wanted to be sure that there is an immutable understanding on the order the Receiver would accept. But then, the Receiver rejects all handshakes, even though everything looked understood & agreed before the handshakes were initiated. That would mean either the Receiver in reality didn’t ‘understand’ the sender correctly and that becomes apparent only with calling upon handshakes or it could mean that the Receiver actually wasn’t serious in accepting such a large order for this very low price and therefore doesn’t want it logged into an immutable Babelchain. In that case, we would like the Receiver to pay the entire fee of the Proof of Understanding to the Translator. This could be achieved for example by requiring both parties upon starting a conversation to add funds into a smart contract between them and the translator, such that there would be enough money to handshake on every single message exchange. Whoever didn’t take the ‘chat’ before the handshake serious enough, would loose his entire deposit to partially the translator to pay for all his services and the rest (for messages that were not called on proof of understanding but were backed by a deposit) to the sender. The sender had spent unnecessary effort with the ‘untruthful’ conversation and in the overall interest of keeping spam messages off the network, we want ‘chatty & gossiping’ network members punished.
- This topic would need to be reassessed in much greater detail based on a yet-missing specification of transactions and

check

Translators

Insert short description (the problem, why it is a ‘Blockchain’ problem also

Translators I – Picking Messages

- A ‘common sense’ approach to building a Translator would be as follows:
- Build a list of
 - common Internet, Enterprise and other message exchange formats
 - Categorize them along Vendor, Technology Platform, Industry, use-case a.o.
- Ensure that Sender & Receiver exchange any information that could be relevant for finding a common message:
 - Which parts of the message are variable (content) and which are fixed (format)
 - Categories like Industry, used Technology Vendor they fall under as listed in (2)
 - Relevant contextual Information, e.g. smart phone owner trying to change temperature is located in a hot country with current temperature x amount Fahrenheit and so on
- Ensure the Translator has a message parsing mechanism, a.o. allowing to:
 - differentiate format from content
 - categorize format along the categories described in (1b)
- Ensure the Translator has message creation mechanism, a.o. allowing to
 - Accept information shared by sender and receiver (2) as input
 - pick a message from common message exchange patterns from table (1a) with the help of the categories listed in (1b)
 - adjust these messages where necessary with module (3)
- Ensure the Translator has a content distribution mechanism, allowing to take all values from the message initially shared by sender and placing them in all possible combinations in the value slots provided in the message format suggested by the sender
- Ensure the Translator keeps a log of all handshakes and
 - updates the logs after each (un-)successful handshake with as much information as possible
 - for any new handshakes reads the log and tries to deduct in a smart way what handshakes could work this time

Translators II – Generating Messages

- The Babelchain Data structure will be designed such that it offers a **training set** of **success-** and **unsuccessful handshakes** with all relevant **features**.
- Over such a training set Translators can create a **supervised logical or probabilistic regression learning algorithm** trying to **predict** message formats, contents. Statistical Machine Learning Algorithms can Machine learning algorithms (eg [swiftspear](#)) can create shakespeare sonnets when fed with existing one's
- Participants typically would feed their Translators with all kinds of xml, csv and other technology and industry message exchange standards
- There would need to be a proper **balance** between what is made **public** on the blockchain to allow the network to learn and what successful Translators keep **private** for themselves. ‘Public’ data on the blockchain serving as training data would be masked and or permissioned
- Translating will get easier over time** with the network and translators learning, hence Translators will need to receive other rewards eg for infrastructure services like delivering messages within time, similar to the bitcoin shift from miners benefiting from Mining rewards towards transaction fees

Message	Location Start Signal	Identity Sender	Identity Receiver	Feature n	S ACK Format	R ACK Format	S ACK Content	R ACK Content	S ACK Action	R ACK Action	Format Handshake	Content Handshake	Action Handshake
...<device> TV</device><command> change channel </command><channel> 12</value>... 	Hotel, front of TV	GUEST	TV Hotel	...	yes	yes	yes	yes	yes	yes	<>;<>;<>	THERMOSTAT;ADJUST_CHANNEL:12	-
...<device> Thermostat</device><command> adjust temparaturechannel </command><channel> 20</value>... 	Hotel, front of thermostat	GUEST	Thermostat Hotel	...	yes	no	-	-	-	-	THERMOSTAT;ADJUST_CHANNEL:20	-	-
...safe; OPEN ... 	Hotel, front of sage	UNKNOWN	Safe Hotel	...	?	?					?	-	-

Translators III – Mapping Messages

- In case we can't find one message format with both approaches, we need to pick two message formats generated and map, the translator will do that
- ABC

describe

check

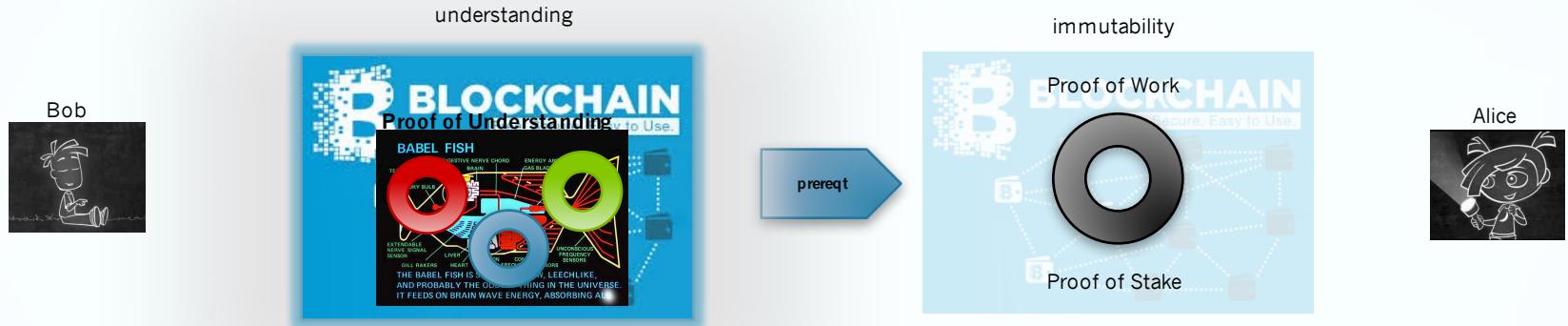
Implementing

Insert short description (the idea)

Immutable Proof of Understanding

	Description	Energy Waste	Goods & Bads
Smart Contract	<ul style="list-style-type: none"> write out a competition, a bounty smart contract where anyone solving the translation task will get a cryptocurrency award like ethereum paid out 	High: <ul style="list-style-type: none"> as high as the hosting blockchain e.g. ethereum No smart reuse of the work translators do 	+ Ideal for prototyping - Low scalability
Prereq for Proof of Work	<ul style="list-style-type: none"> whoever generated a valid proof of understanding (there can be several) can start the proof of work Hashing puzzle and finding a nonce will include the agreed message, hence proof of work can be started only after proof of understanding Agreed Message will be signed by sender and receiver, so anyone in the network can check 	Smarter, not lower: <ul style="list-style-type: none"> Machine work will be used for a useful task Combined work overall would be similar to bitcoin blockchain 	+ Re-use proven Bitcoin approach to make the Blockchain immutable + high scalability - Energy consumption and waste as high as Bitcoin - <i>To be clarified, how the effort of proof of understanding can be measured and how proof of work difficulty target could be adjusted dynamically</i>
Replace Proof of Work	<ul style="list-style-type: none"> In any transaction system between connected parties having a common terminology is a pre req to make useful transactions. With the size and amounts of usecases in IoT the 'one agreed transaction data format' strategy of Bitcoin will not be feasible. 	Smarter and lower: <ul style="list-style-type: none"> Machine work will be used for a useful task Effort for Translation will decrease with the learning effect 	+ high scalability + use of machine work to cover 2 goals: translation & immutability - <i>To be clarified how Blockchain Immutability will be achieved, even more with learning effect and decreased work</i> - <i>Potentially smart combination of machine work with consent sender & receiver to message reverts</i>

Proof of Understanding as prereq for existing Proof of X Mechanisms



Combine with '**Proof of Work**' style:

- have Translator not only find Translations but also hash them against a difficult target
 - Proof of Understanding Message will be signed by sender and receiver
 - Only who generated a valid proof of understanding can start the proof of work,
 - fingerprint of the agreed message is part of the blockheader for which to find a nonce to meet the difficulty target
- Revising a handshake will require re-do of computationally expensive Hash – against – difficulty - targets like in bitcoin

Combine with **Reputation**, **Trustability** and '**Proof of Stake**' Systems:

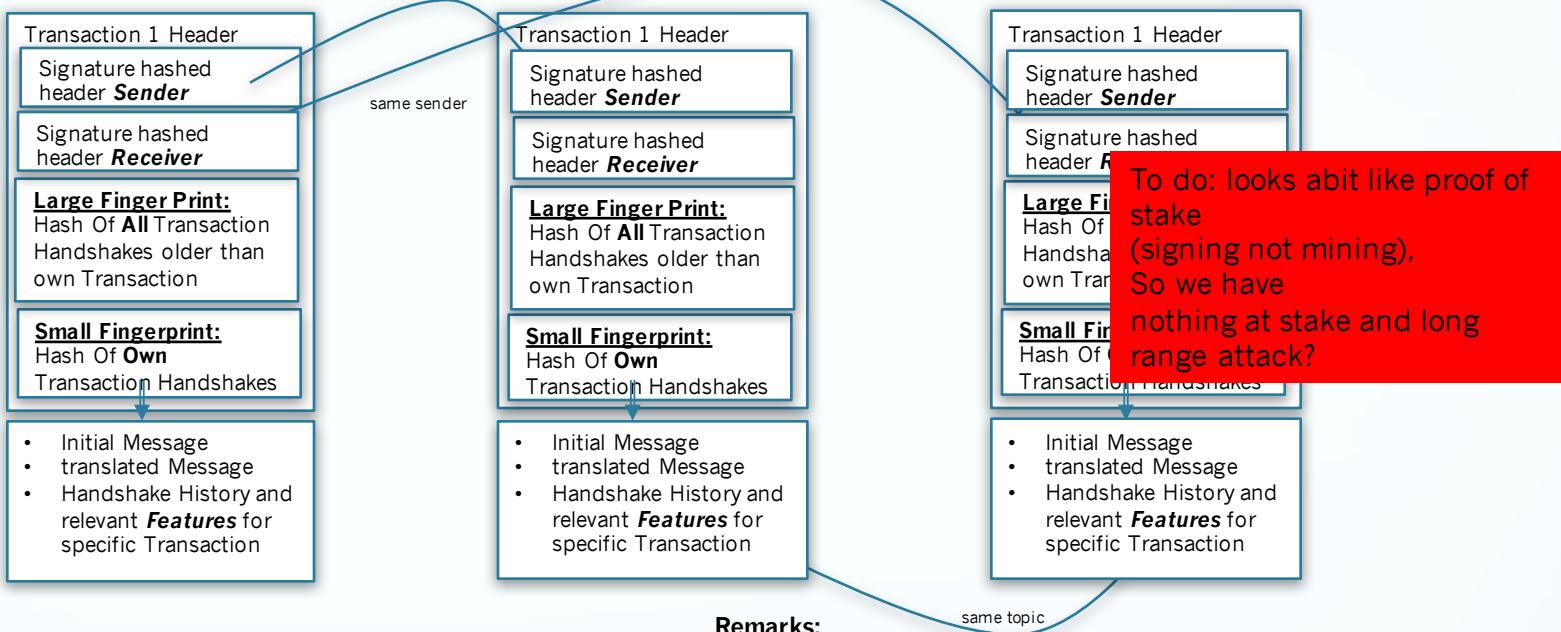
- Have not only sender and receiver agree to handshakes but have also other 'trustable' network members like eg very successful **Translators sign and approve translations**
- **Trustability or Reputation** is defined via the amount of **currency won** as it indicates that a member is very knowledgeable about correct Translations
- Members additionally **holding a large amount of the cryptocurrency** hold a **financial stake** in the network and have interest to avoid fraud

check

Immutability

Insert short description (the idea)

Proof of Understanding – Signing Immutability



Cascade Effect:

A Blockchain representing ‘Conversations’ between Senders & Receivers would take:

- **small fingerprint:** hash fingerprints of their own Handshake Negotiations
- **large fingerprint:** Handshake Negotiation history existing at the time of their handshake

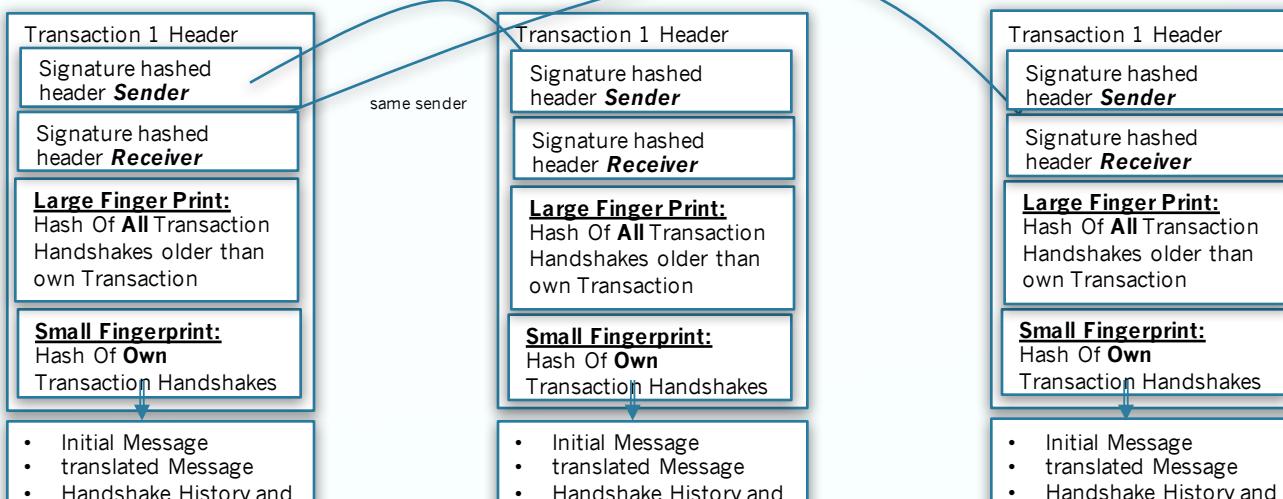
Changing a transaction, handshake will trigger:

- **weak immutability:**
 - Re-do **translation** work, unless same format / content / action accepted
 - Re-ask **consent** from senders and receivers if handshakes are adjusted
 - Header needs to get **hashed**, optionally against a difficulty target
- **strong immutability:** Change of Handshake will change the Large Fingerprint of all younger Transactions
 - Re-do **all translation** work, unless same format / content / action accepted
 - **all those Senders Receivers** will need to sign the Header again
 - **All their Headers** need to get hashed, optionally against a difficulty target

Remarks:

- Usage of a hash **against a difficulty target** can be necessary to rule out the possibility that (all affected) sender & (all affected) senders conspire to revert transactions on expense of a 3rd party affected in the real world but not part of the digital transaction
- The larger the network, the less likely it is all will ‘cheat’. Usage of a hash against a difficulty target can therefore be useful especially in the **network startup phase** when the network is small and the cascade effect hence negligible
- Note **neither** Blocks of Transactions **nor** Transactions (C pays B) **depend** on previous Transactions (B pays A)
- There is an **indirect link** via the **history of handshakes** (the training set), which can be used to create a cascade effect (see below)
- Transactions in a Blockchain representing Conversations between Machines are **loosely coupled** along **Senders & Receivers** or **topics** (training set **features**) like in a social network

Proof of Understanding – Working Immutability



To do

We prefer
A proof
via work
you did!!!
Not just
signing !

What about this, when reverting:

- You have to re-play the handshake history with sender & receiver such that you achieve the same (lower ?) target than before BUT you must add the nonce with the value of the previous transaction header

Note that you play with the machine learning algorithm, not the hash

First time (not redoing) this mechanism means:

Changing a transaction, handshake will trigger:

- **weak immutability:**
 - Re-do **translation** work, unless same format / content / action accepted
 - Re-ask **consent** from senders and receivers if handshakes are adjusted
 - Header needs to get **hashed**, optionally against a difficulty target
- **strong immutability:** Change of Handshake will change the Large Fingerprint of all younger Transactions
 - Re-do **all translation** work, unless same format / content / action accepted
 - **all those Senders Receivers** will need to sign the Header again
 - **All their Headers** need to get hashed, optionally against a difficulty target

a hash against a difficulty target can therefore be useful especially in the **network startup phase** when the network is small and the cascade effect hence neglectable

- Note **neither** Blocks of Transactions **nor Transactions** (**C** pays **B**) **depend** on previous Transactions (**B** pays **A**)
- There is an **indirect link** via the **history of handshakes** (the training set), which can be used to create a cascade effect (see below)
- Transactions in a Blockchain representing Conversations between Machines are **loosely coupled** along **Senders & Receivers** or **topics** (training set **features**) like in a social network

Summary

Comparing Proof of Work and Proof of Understanding

	Bitcoin Proof of Work	Babelchain Proof of Understanding
Purpose	Create an immutable Blockchain for Financial Transactions	Create common message formats to allow machines to communicate
Quiz	Find a value for the nonce that results in a block header hash that is less than the difficulty target	Find a format (content, Action) which sender and receiver approve
Deterministic	Yes, for any input (arbitrary length) egSHA 256 will produce always the same fixed length output	No, sender and receiver could agree on different formats for the same messages asked on different times as long as they both believe the format will work for them
Predictable timeframe	Yes, statistically	<i>For many messages probably yes statistically, but there will be non-translatable messages</i>
Can it fail ?	Statistically not	Yes, sender and receiver could never reach an agreed message format (content / action)
Money supply	decreasing	constant, never ending
Hard to find solutions	Yes, depending on difficulty target	Case by Case. Will generally get easier with the learning effect of the network. There can always be very hard or untranslatable messages.
Easy to verify solution	Yes, feature of hash function	Yes, sender & receiver have to agree and sign the message, which can be verified
Quiz Difficulty adjusting over time	no	Yes, network learning effect implies that translation will get easier. There can always be very hard or untranslatable messages.
Difficulty of winning reward	Increasing (money supply decreasing and quiz difficulty unchanged)	<i>Decreasing (learning effect and constant money supply). This raises the question, how to reward Translators for (too) easy translations over time</i>
Payer	Mining Reward: Bitcoin network Transaction Fee: the Sender of Bitcoin	Translation Reward: Sender (try claiming part from Receiver) Transaction Fee: Sender (try claiming part from Receiver)
Cheating possible	no	Yes, senders and receiver might pass on more features outside the network to preferred translators
Immutability	Yes	'weak immutability' per default as sender and receiver need to agree to changes 'strong immutability' depending on either combination with existing proof of work / proof of stake or smart usage of training sets to cause a

Why ?

- Computers are ‘no good’ at understanding concepts and translating them into each other. This is a problem in many areas of IT like Cloud API’s, Enterprise Integration
- With the size of IoT and the exposure of devices to normal consumers (Baskets of Remotes) the problem will get larger

Why Blockchains?

- Bitcoin shows us that Transactions can get safely handled without any central ‘trusted Middle man’ solutions
- Blockchains promise economical advantages for large volume, long living, low margin devices over centralized cloud solutions

Why Proof of Understanding ?

- Any Transaction System has to solve the question of a common data format either by enforcing (Bitcoin and e.g. most Integrations patterns like SOA and alikes) or negotiating (Babelchain)
- Proof of Understanding can make more efficient usage of Machine work by (partially) replacing Proof of Work, while keeping the Blockchain unmutability feature

Why not some Middleware ?

- Middleware & ESB’s are permissioned with Identity & Access Management. They typically have a centralized data exchange format
- Blockchains with Consensus Mechanisms can be permissionless (open & distributed) and negotiate Message Exchange Formats.