A NEW PARADIGM

Bitlattice - advanced decentralized technology

Vision and design goals

HIBRYDA AND TEAM BITLATTICE

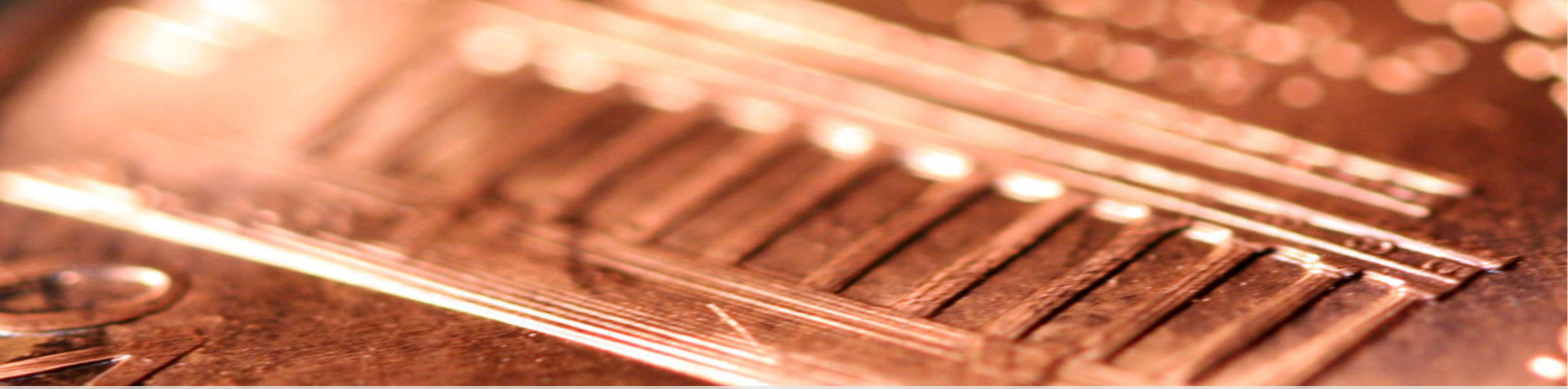
19TH NOVEMBER, 2019

Publisher
First published by Hibryda and Team Bitlattice in 2019.

Abstract

This document presents the assumptions and possibilities of Bitlattice, its superiority of concept and the innovative way distributed data storage and processing is organized and performs.
It shows the approach that led us to the right but counterintuitive solution, describes the natural consequences of such arrangement and set of new tools we had to invent.
It explores general categories to which Bitlattice can apply such as financial operations, governance frameworks, sensing networks, Internet of Things, Artificial Intelligence and others.
Before presenting a long list of use cases, it shows the prominent features of Bitlattice innovation, addressing the most important matters.
Potential use cases listed in chapter 4 show a wide range of applications and highlights innovation aspects related to specific domains.
The part about funding outlines our view on certain financial matters.
There are also legal aspects of the project that needed some clarifications and they have been explained in chapter 6 of this document.
The motivation behind Bitlattice's conception, presented in appendix, deserves a closer look, as it indirectly shows its design goals and potential uses.
The article about the Interface Problem closes the entire document covering the seldom mentioned problem of computation.

Table of contents



1 Introduction

When the first ever distributed ledger, Bitcoin, came into existence, its raison d'être stated in the introduction of the Bitcoin white-paper was to eliminate the human trust element from economic transactions. That attempt, while extremely successful at popularization of an alternative view on economy, enabling secure transactions, managing and securely storing value, couldn't deliver the expected outcome in full. There were multiple reasons for that. First and foremost the world's economy is too scripted to be cured by a single, ingenious, but not very important in terms of absolute numbers, solution. There are also multiple issues with blockchains and derived technologies that stem from their design. Those issues make full scale adoption very difficult as they render the technology hard to implement on a global scale.

The above factors led to the idea behind Bitlattice. The idea itself emerged very early, shortly after the idea of first altcoins in blockchain emerged, though its technical basis needed time to evolve. At the time of writing the idea has had about 4 years to mature.

If you want to learn more about motivations, please head to Chapter A (*Motivations – in depth view*) on page 33 in the Appendix.

1.1 What is Bitlattice?

There are both short and long answers. The short one is:

> **Bitlattice** is a novel form of distributed ledger, built from the ground up with the goal of achieving what blockchains and related technologies cannot – self-contained data storage that does not require any form of trust in an operator or a set of operators to account for changes.
> **Bitlattice** solves inherent deficiencies of blockchains and general data storage solutions that limit their flexibility and areas of possible adoption.

The long answer is more complex and requires some background knowledge. We'll first focus on the main difference in logical organization between current data management solutions and **Bitlattice** . One thing that should be stressed here is that distributed ledgers (including blockchains) and Ǝitlattice are transactional databases[1] . Transactions in such databases should, to perform properly, satisfy ACID[2] requirements (**A**tomicity, **C**onsistency, **I**solation, **D**urability). Distributed ledgers deal properly with first two properties, actually excelling over regular database systems when it comes to consistency. However, with respect to isolation and durability they suffer from inherent design related issues. Those issues, while possible to mitigate to a certain extent, are persistent. Parallel computation and swarm data validation are in fact reasons why distributed ledgers cannot compete with regular databases in most areas. They must, to keep isolation and ensure proper storage, artificially enforce sequencing of operations and perform additional, computationally intensive operations.

1: Block and similar control structures can be regarded as atomic transactions

2: To read more visit this Wikipedia entry on the subject.

Please take a look at the ideogram the change in paradigm Ǝitlattice introduces.

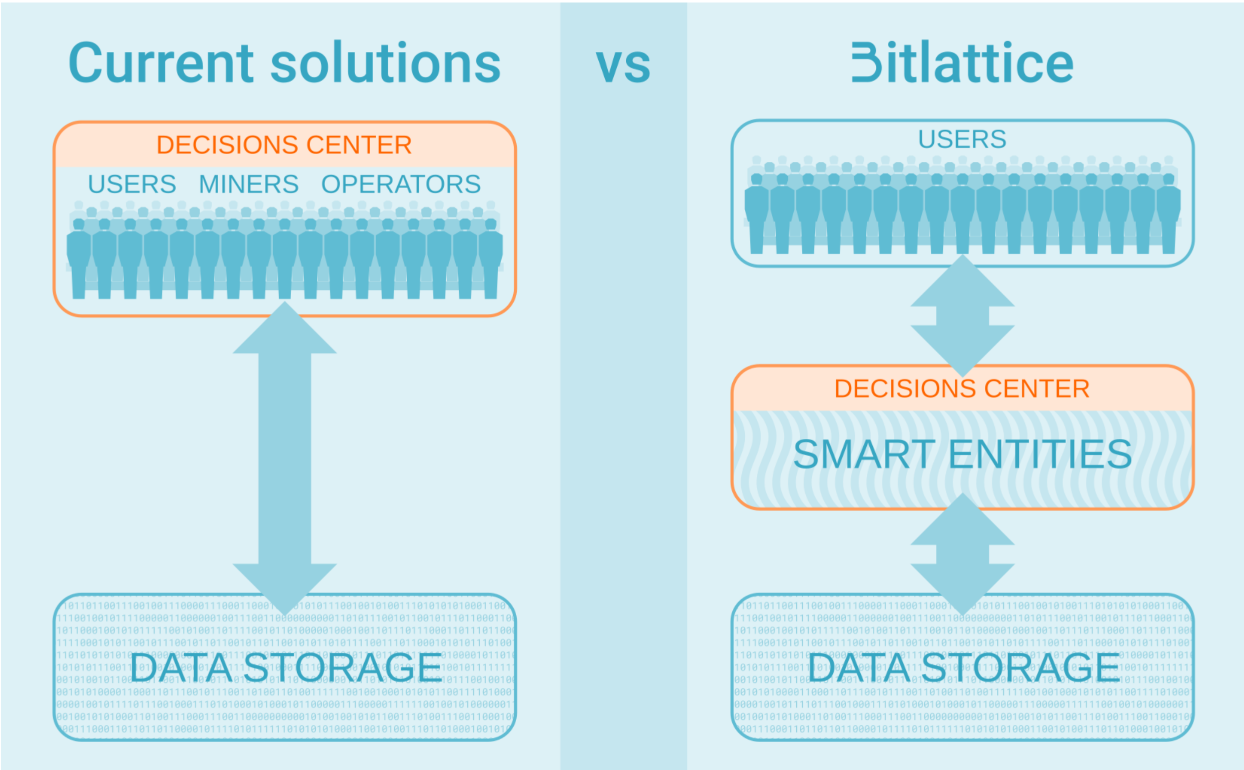


Figure 1.1: Difference between current solutions and Ǝitlattice

In Ǝitlattice we approached the issues mentioned above differently – when a solution doesn't provide ways to mitigate issues there must be another that does. We identified what are roots of those problems and concluded that whatever can go wrong in distributed ledgers, will at the processing layer. Processing means making decisions, either by humans or by machines.
Then, we realized that if we could move that layer inside the database and protect it from the outside world that would provide us with expected solution. That additional buffer layer could benefit from having direct access to the ledger, could ascertain proper sequencing and could tame parallel computations and schedule their outputs inter-

nally. That way all actual decisions are taken there, with external actors stating only their intents[3] . Real assembly of transactions takes place there.

While remotely possible with available tools when the idea was conceived, the arrangement proposed couldn't be efficient or even viable. Therefore, we had to invent a set of new tools. It was natural to leverage fully homomorphic encryption[4], however even with recent developments it's still computationally intensive. The solution we found is counterintuitive – **Ɔitlattice** introduces a different architecture[5], where data is arranged in a 5 dimensional lattice (with 3 Cartesian degrees of freedom and two specifically constrained). It's counter intuitive as benefits of such solution aren't immediately obvious. The lattice is a storage schematic that doesn't directly enable FHE (however has it's share in binding FHE shielded processing to the data), in turn it allows for easier management of distributed storage giving an elegant solution to the scalability issue that plagues most distributed systems.

There are further consequences of the arrangement mentioned above (some of them being side effects, some being further extensions) – they are presented in the coming chapters. To mention some of the most important capabilities:

FULL NODES
There isn't a need to have full nodes (with whole structure stored) at all. All nodes could be light, internal governance will attempt to achieve a level of data duplicity set upon network creation. In practice, there will be some full nodes, but their existence isn't critical

COMPUTATION EFFECTIVENESS
The network itself governs how much computation power is needed and when required spawns new smart entities of the middle layer. It does that only on devices that reported being capable to handle such costly computations. The number of computation heavy nodes is limited, always significantly lower than the total number of nodes

SCALABILITY
Due to its inherent capabilities (like the ones in two points above) the network is easily scalable

DECENTRALIZED CENTRALIZATION
The network is decentralized in physical sense, however functionally speaking it's fully centralized – all decisions are made in the middle layer and there is no authority that could affect them

SMART CONTRACTS
They are obviously available, more, there are two layers of them. The first, internally trusted operates on the middle layer. The second, external, can be processed by light clients and refer to the internal one only when required. That distinction is

3: Some may find it unclear and state, that in present solutions of distributed ledgers users state an intent and a network decides whether the intent is valid via some consensus mechanism. That's true. However, users able to organize to reach **51%** threshold, able to plant altered versions of client applications, able to exploit them can affect the overall state of a network in question.
Conversely, **Ɔitlattice** moves all processing out of reach of any, even malicious actors.

4: To read more visit this Wikipedia entry on the subject.

5: The technical details are scheduled to be released in Technical White-Paper here to properly balance the computation load of the network. That also allows for different VMs being plugged into external layers enabling greater flexibility.

To sum up the above – **Bitlattice** is a new paradigm in distributed ledgers. It changes the way distributed data storage and processing is organized and performs.

1.2 What Bitlattice isn't?

It's necessary to explain what **Bitlattice** isn't, as we are often compared to other solutions in the field. Therefore, **Bitlattice** is not a:

BLOCKCHAIN
While **Bitlattice** retains some properties, like for instance keeping the data consistent and immutable, these are superficial similarities. **Bitlattice** differs substantially from blockchains in the way it operates, in internal structure and in offered capabilities.

DIGITAL CURRENCY
Use of **Bitlattice** as a currency is one of many potential uses and will likely be its first, however there's a plethora of other applications that can benefit from **Bitlattice**.

1.3 What is it for?

While more detailed, potential use cases are listed in Chapter 4 (*Potential use cases*) on page 16, here we are going to name some general categories to which **Bitlattice** can be used.

Financial operations

Digital currency is the most obvious financial use case. However, financial operations aren't limited to this specific use. Every operation involving proper accountancy of some assets can be performed with **Bitlattice**, while benefiting from its inherent properties. Therefore, automated trading, assets management, processing financial transactions can be successfully implemented with use of **Bitlattice**. One important factor that should be mentioned here is that due to all decisions being taken internally such systems, if properly set, could end up being zero maintenance which means no operators needed to manage them. The idea of a data driven language was conceived due to investigating of potential application of **Bitlattice** in high frequency trading. While the actual trading algorithms cannot be replaced by **Bitlattice** due to computation efficiency, they can be fed with inputs (most likely via a machine learning interpreter) from **Bitlattice** based global network collecting market state indicators. With properly defined language the data can be initially processed streamlining further operations.

Governance framework

Governance related tasks are a natural choice for **Bitlattice** as a consequence of its isolated decisions center. Every chain of decisions that can be observed in real life, be it bureaucratic machines, company management, political systems, legal systems, can be to the extent required by those who wish to implement it ported onto **Bitlattice**.
When such chain is sufficiently[6] formalized, internal logic can deal with input sanitation, while in less formalized cases normalization via machine learning or artificial intelligence should be applied. This is related to the interface problem which in short happens every time one system meets the other and must interpret messages between system communicating. It's explained further in Chapter B (*Interface problem*) on page 35 in Appendix.

6: "Sufficiently" here has no strict meaning, architecture must be evaluated on per case basis, as the amount of computation on the middle layer should be adjusted to computation power available.

Sensing networks and Internet of Things

One of the fastest growing trends in the IT industry is data collection. There are however recurring issues in that field. To be cost effective, nodes should be cheap and mass manufactured. On the other hand this requires large data centers to prepare the collected data before processing it further. Another approach is to use middle scale devices (in terms of computation power) like smartphones for instance. Yet, that approach limits both range of data that is possible to collect (even if it's fairly broad) and users' experience (battery life, reaction speed) as those devices are supposed to process data before sending it further.
Use of **Bitlattice** can be beneficial in the above and similar strategies. **Bitlattice** allows for use of different clients, ranging from a low level ones that just sends data to the network and advertises its computation capabilities to complex ones equipped with multiple external level preprocessing VMs and advertise extensive capabilities to the network.
Moreover, **Bitlattice** self-balances, at least when it comes to the middle layer, as smart entities are spawned only when needed[7] . Next, operations that take place in a specific region of data storage (that way not changing the overall state of that region) don't require costly operations to update the rest of network. The region can be defined based on numerous factors, like geographic position, specific data, etc. Last, but possibly most important, data can be processed on untrusted nodes without the risk of leaks. The flexibility offered by **Bitlattice** enables other approaches extending the ones above.
Potential use cases in Chapter 4 (*Potential use cases*) on page 16 shed more light on the different strategies.

7: need is determined by a number of operations to assemble into a transaction, if an entity "overflows" it spawns another one to take care of excess.

Artificial intelligence and inference automation methods

Artificial intelligence is an umbrella term that in short encompasses technologies that are either similar in effect to how natural intelligence (human and animal) works or extend it. Therefore, while the term is catchy it's hard to address matters related jointly to it and Bitlattice without going into many fine details. To narrow deliberations down we'll refer to a set of inference automation methods, as while Bitlattice can be used, for instance, in effectors, covering too wide area isn't a purpose of this document. That set intersects with AI techniques, spans beyond and contains methods that basically allow for inference from the set prerequisites.
These methods can be further sub-categorized into inference from complete and incomplete information[8] . AI solutions fall into the latter category. That and often stochastic nature of those solutions has consequences manifesting in specific deficiencies – lack of precision, unpredictability, unreliability. For a casual observer they aren't apparent, being obscured by their nearly magical capabilities. However, the IT industry started noticing them.
Bad news are that these are not flaws that can be fixed as they are inherent properties of these solutions. Some of them stem from the interface problem sketched in Chapter B (*Interface problem*) on page 35. Good news are that more deterministic solutions coupled to AIs can minimize negative impact of these properties. Bitlattice can interact with AIs in many different ways, depending on the type of inference method used and specificity of application. Though, we can name three main modes:

8: To visualize the difference (simplifying a little), we, humans infer from incomplete information, as our senses never provide us with exact depiction of reality. It's always more or less an approximation.
On the other hand, if we consider a computer program that accepts only two numbers as input and is supposed to tell whether the first number is bigger than the second we have a case of inference from complete information.

TRAINING DATA ACQUISITION AND PROCESSING

In this mode Bitlattice performs an initial gathering and processing of data that is later fed to an AI solution. Primarily in the ones based on neural networks several steps must be accomplished before the actual training can begin. Simplifying, those steps can contain normalization (making data homogenous), feature extraction (gathering prominent patterns) and classification (assigning meanings to patterns). Bitlattice can perform them in one pass while processing incoming data. The added values of using Bitlattice apart from preparation, are potentially large data gathering network and possibility to decouple actual data from symbolic description passed to AI. Globally accessible training dataset is one of interesting potential applications.

OUTPUT STORAGE AND PROCESSING

This is a rather obvious consequence of Bitlattice being a distributed ledger. Actually distributed ledgers at the moment do a similar job, but using mostly natural (human) intelligence as input. Bitlattice in this mode could accept input from multiple AIs and offer an aggregated and/or processed output. At that point the latter can further deterministically classify, the data provided by AIs and produce a conclusion. Given a large scale and a large number of AIs that could minimize risk of residual poisoning of training data and other forms of manipulation. That mode could also serve to combine different observations using strict criteria.

ACTIVE COMPONENT

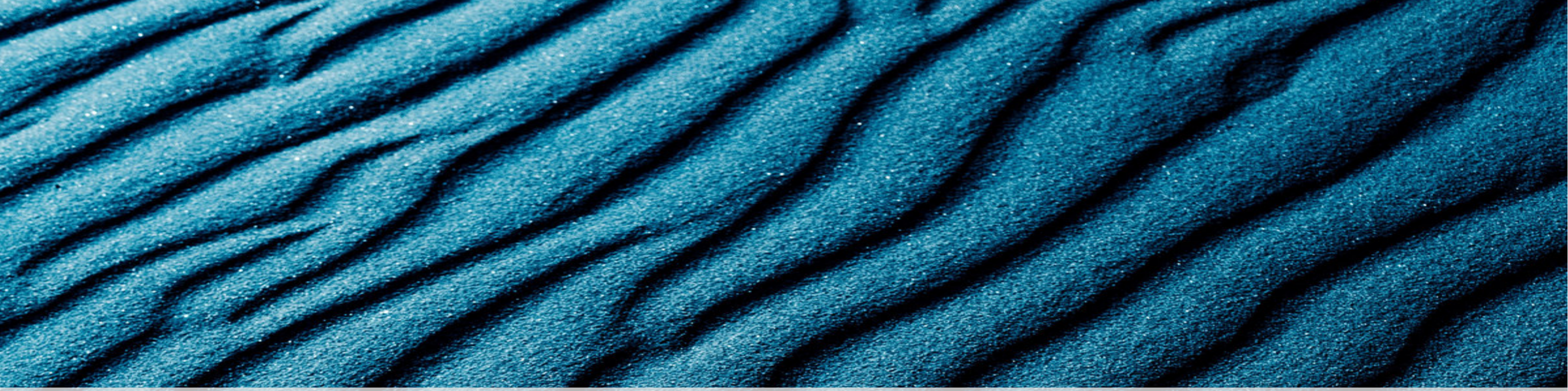
The compartments that Bitlattice stands on, expose a set of metrics that facilitate operations. Those metrics can also be used as an input for AIs. As they are uniform, there is no need to normalize them. They carry a general picture of the data stored in a compartment and therefore can be used by AIs to infer about the data structure and dynamics.

ARTIFICIAL INTELLIGENCE ITSELF

This is probably the most interesting mode. While Bitlattice's middle layer works like a decision tree, other strategies can be implemented as well. Simplifying again, distributed ledgers are networks of peers, but to become neural networks they would need several additional properties, like layered organization, persistent neurons, defined excitation levels, etc.

Bitlattice has or can have implemented instrumentation needed to act as a neural network. That idea is wild, but ultimately possible and potentially beneficial. While the global wide network in this mode won't be fast (due to physical limitations of signals speed and delays of network) the fact that the middle layer contains far less nodes than actual number of participating devices makes that idea at least possible to implement. The practical aspect here could be, for instance, making a "feeling planet" like project.

The above modes can also be mixed and extended, as the domain of AI is very wide and provides plenty of different tools to experiment with.



2 Features

Presenting specific features, when it comes to a network with multiple potential applications is a task that requires selecting the most prominent ones which have a general impact on usability and applicability.
Therefore, we are not going to delve into every little detail, instead we will be presenting features addressing the most important areas.

2.1 Scalability

Scalability is a problem that has haunted the IT industry from its beginning. It started with the question "How many vacuum tubes can be packed in a 167 sq. metre room without boiling all human operators?"[10] . That was obviously a case of hardware scalability and when transistors replaced vacuum tubes, metres scaled down to nanometers, and the issue of boiling humans turned into an issue of boiling a chip, but principles remained the same. We want more computing power, but the physical limits don't disappear. Instead, we must walk around them or find solutions that better utilize the resources we have at hand.

10: It probably started with an even weirder question regarding brass gears, possibly asked by Mr.Babbage. However, as his idea never took its final shape we referred to the latter, successful realization of the first general-purpose computer ENIAC. Read more about Babbage engine or ENIAC .

Scalability is a problem that always boils down to hardware, however in present times manifests more prominently on the software layer. Its essence can be summarized in a simple question – "What amount of additional resources must be used to make a system more efficient by a certain quantity". A real world use case to illustrate that point could be a situation where a cloud provider wants to accommodate twice as many clients as it has now, so must expand its data center by a certain amount of additional machines, employ additional maintenance staff, be prepared for bigger fixed costs. The problem with scaling anything up is that while the cost in money can be cumulatively lesser (as big companies can negotiate better deals) actual resources invested usually grow with every unit of additional output of a system. Those resources don't have to be consumed only by the entity that scales the system. For instance, any wasteful system implemented on mobiles can affect overall energy consumption worldwide.

Nevertheless, some basic scalability issues can be illustrated with the simple infographic below:

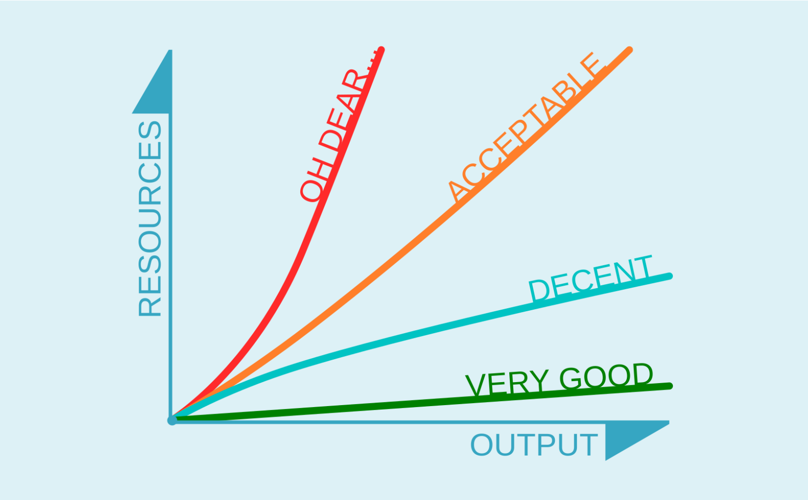


Figure 2.1: Relation between output of a system and resources invested – conceptual depiction.

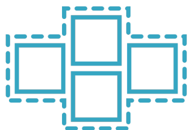
The green line in the above picture exists only in the alternate reality marketing professionals operate in. The teal one is how well designed system should behave. **Bitlattice** fits close to that line. The orange line is a border that should not be crossed – still manageable, but wasteful. The red line is a situation when every little improvement leads to sky-rocketing resource consumption. The current top blockchains reside between red and orange.

Bitlattice is scalable and the resources required to expand it grow slowly thanks to several factors:

REDUCED DATA SATURATION

Distributed ledgers used to store data must replicate it far more often than centralized systems, so that data availability can be retained in the event that a large portion of the network is attacked. There are multiple strategies of replication, from replication on every node to some form of sharding. Nevertheless, they are seldom adaptive and mostly wasteful.
Bitlattice reduces data saturation thanks to its internal structure and the middle layer (mentioned before) that acts as a hive mind being central authority governing storage distribution. Governance can either depend on a statically set replication level or can adaptively force storage on nodes depending on set factors and circumstances. In a simplified case a factor of 12 seems to be a minimal viable replication multiplier. For real applications that factor must be evaluated on a per case basis.

DATA COMPARTMENTALIZATION

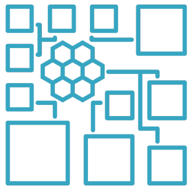


Contrary to most solutions of distributed ledgers, **Bitlattice**'s storage architecture employs compartmentalization of logical regions of data. Depending on use case it can be performed based on different criteria, like operations' density, network response distance, type of operations, etc.
This is possible due to the localized impact of data change and therefore no need to frequently update the data storage as a whole. This in turn leads to reduced operations related to data turnover.

While superficial reception could be that we just use another name for sharding[11] there are enough differences between what **Bitlattice** proposes and what is presented by sharding implementations to justify a different name.

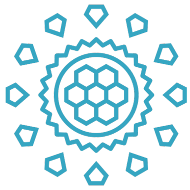
11: which would make sense as sharding implies that something is broken (pun intended).

GRANLULAR GOVERNANCE



The hive mind that manages all operations on the data and has all knowledge about it can perform many optimizations to balance resource usage across the network.
While already mentioned before, it should be emphasized here, that the central authority provides means for boosting efficiency via using multiple types of interfaces that can differ in complexity, permissions, functions, via shifting tasks to areas where resources were reported to be available and via spawning additional smart entities only when needed.
All these strategies ensure flat resource usage while retaining stable performance.

AUTONOMOUS NATURE



Bitlattice, on its middle layer, operates independently of any external actors. That ensures that all mechanisms mentioned above are executed to keep the network in homeostasis regardless of changes in external conditions. That in turn, given a large enough network, translates into a system that is highly stable and resistant to many forms of manipulation.

2.2 Quantum resistance

Quantum resistance and quantum readiness are concepts that recently became very trendy due to the emergence of functional quantum computers.
Bitlattice is quantum resistant and quantum ready, but reasons behind that aren't as obvious as it seems and are twofold:

SECURITY

Quantum computers can theoretically solve certain classes of problems amazingly fast, faster by orders of magnitude than any regular computer. Many encryption algorithms currently commonly used, also in distributed ledgers, are theoretically able to be cracked by quantum computers. The risk exists, but contrary to popular belief, isn't just around the corner. The already delayed success of Google to deliver quantum supremacy means that it will probably take several years before we are able to attempt breaking currently used ciphers.
Nevertheless, we at **Bitlattice** treat this fact seriously and use algorithms proven resistant to quantum cracking.

DESIGN CONSIDERATIONS

The most prominent reason to use quantum resistant methods of encryption is that they perfectly fit the structure of **Bitlattice**.

Or, to stress it even further, without specific computational problems those algorithms make use of, whole instrumentation of **Bitlattice** wouldn't make much sense. Hence, that was the natural and only choice.

2.3 Trustlessness

Building a trustless system was already mentioned as a motivation that stood behind creating **Bitlattice** (read more in Chapter A (*Motivations – in depth view*) on page 33 in Appendix).
There are prominent reasons why trust is detrimental to economy and social relations and, in general, to sustainability. Hibryda wrote an article[12] explaining the mutual relation of trust and sustainability in detail.
In short, trust is a tool that we use to conserve resources and compensate our deficiencies when it comes to proper and detailed perception of reality. The phenomenon of trust is also strongly linked to the interface problem discussed in Chapter B (*Interface problem*) on page 35 in Appendix. While trust cannot be eliminated completely from human society eliminating it from certain areas like economy and widely understood governance could be beneficial. As a self contained tool, **Bitlattice** provides trustless response to human queries about its internal data storage. Trustless here means that the answer (or action) is always correct and cannot be altered by external, malicious or not, actions. **Bitlattice** is therefore a kind of real life oracle machine (even if it differs from the theoretical abstract description).

12: read it on Medium.

2.4 Ability to decide

The ability to make decisions is a consequence of previously mentioned self-containment, as **Bitlattice** has perfect data about the internals of itself. Therefore, as human influence cannot alter the state of the network and all data management is performed internally, every answer to a query is logically valid with respect to that data with absolute certainty.
The quality of data is an external issue, **Bitlattice** always properly infers from what it contains.

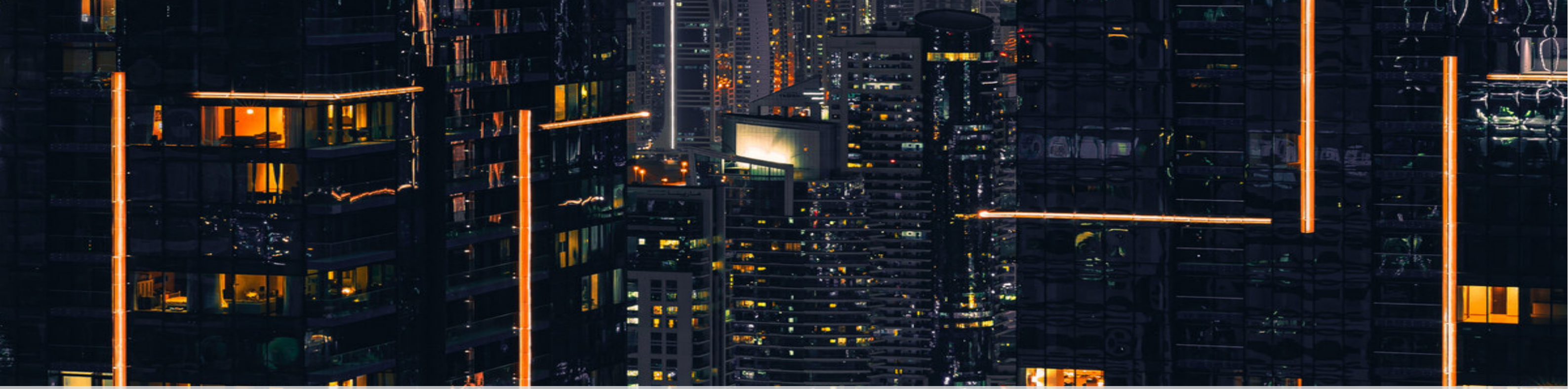
2.5 Hardware integration

Bitlattice is meant to be easy to integrate into hardware solutions thanks to an ASIC[13] chip that is planned to be released together with the software implementation.
Apart from forcing use of reference implementation of the protocol, which is further elaborated in Section 6.3 (ASIC chip) on page 30 in Chapter 6 (Legal), there are also other motives behind that move, which are probably even more important.

13: **A**pplication-**S**pecific **I**ntegrated **C**ircuit

The first one is of technical nature. While **Bitlattice** can have different types of nodes including very thin ones, the more computing power in the network, the better. When it comes, for instance, to IoT devices, while they can contribute to processing, they are usually very limited by a plethora of factors including memory, stack, limited set of instructions, energy, etc. They are also not usually suited for any complex operations. On the other hand, as they are mass manufactured and becoming more and more popular in many applications they could greatly enlarge the computation pool of **Bitlattice**ȦSIC chip, being a specialized circuit aimed at providing fast and energy efficient, but limited in scope, computations is a solution. While it's mentioned later in Chapter 4 (Potential use cases) it's worth visualizing here why that can matter. Let's take for instance energy grids, they already make use of IoT devices and machine learning. With **Bitlattice** they could provide even more stability for a negligible price of additional chips and electricity.

However, while use of **Bitlattice** on IoTs is probably more a domain of industrial applications, mobiles are closer to the consumer market. Implementation on mobiles apart from extending the computation pool can also contribute to popularization and mass adoption of **Bitlattice** and any derivative project that is built upon it. It boils down to ease of use, if something is natively supported and using it is just a few taps away. We are realists, so we don't assume that mobile manufacturers will adopt the chip just because we offer it, but possible benefits stemming from further expanding abilities of their products and opportunity to make money on, for instance, smart microtransactions, can attract them.



3 Performance

One of the most prominent "selling factors" of distributed ledgers is their performance measured in number of operations possible to be completed in a unit of time.
Bitlattice is fast, but before we proceed to show how fast and why, we'll shortly focus on measuring speed of distributed, dynamic systems.
The first thing that needs to be stressed here is that such measures are both of negligible value when it comes to evaluating a product and rarely reliable in themselves. However, this categorical in its form statement, requires a more detailed explanation. To maintain these considerations concise, we can identify only the three most prominent factors that affect processing speed of a distributed system, ignoring nuances:

Physics

Distributed systems can span large areas and can even encompass the whole planet. Whenever there is need for bidirectional communication (for instance: action and confirmation) that affects the total number of operations when at least some operations require the state of network being up to date. Speed of a signal depends on transmitting medium and the medium a waveguide is made of. It's always less than the speed of light in vacuum, sometimes substantially less. A fair approximation is 80% of that speed. There are additional delays when the signal switches between mediums (for instance when the electric signal that travels through a copper wire is converted to light pulse that travels through optical fiber). When a signal is meant to travel back and forth along half of the big circle of Earth that alone introduces a delay of 0.17s. That is time that an operation which execution depends on execution of a previous operation must wait until being executed. In reality there are more steps to make the operation being executed and that time grows into seconds (sometimes really many of them). This limitation is a property of our reality and most likely cannot be addressed otherwise than by resorting to tricks like executing many operations at once if possible, avoiding too many state coordinations, etc.

Environment

Environment can be under control of an operator of a system or not. The best situation is the former. That's why high frequency trading businesses invest into infrastructure. When every fraction of second matters, own, predictable, possible to evaluate infrastructure provides means to calculate close to exact time thresholds. But that is a niche. In most of these cases infrastructure is getting out control, either totally or in major part. The situation is entirely out of control when an enterprise must be operated on public internet and/or long distances. Then nothing can be evaluated for sure and discrepancy between minimum and maximum time thresholds grows to values eliminating any close to reliable inference of speeds achievable.

Processing

Every operation must be processed and this consumes time. This must be added to the whole equation. A natural thought here is that to save some time we can run many programs on many devices in parallel each servicing another operation. Which is a good improvement, but with some caveats. Ledgers, to provide functionality they are designed for, need to make sure that no two operations are performed at the same time on the same chunk of data. Let's assume that our ledger is forbidden to store negative numbers and subtraction operations leading to making the data negative are meant to be discarded without processing. Then, we have multiple instances of a program running in parallel that write into the ledger. Among operations there are two that subtract a certain value from some data at exactly the same location in the ledger. The subtracted value in both cases is bigger then a half of current value of ledger's entry. We don't know which is executed first. Therefore, we cannot predict the final value, however we can be sure that one operation is going to be rejected. That's how it works in ledgers that consist of a central storage and processing and clients connected over the network sending requests to change some data.

All becomes complicated when our ledger is distributed (which means there is no central unit that could immediately have information about the change being already written) because there can be situations when two operations will attempt to change the same data and will succeed as propagation of each change will fail to reach the other processing unit. It's known in blockchain world as double spending problem.

There are multiple strategies to mitigate this issue all boiling down to linearizing the parallel stream of operations effectively nullifying its benefits.

From the above reason it should be clear why delays directly translate into number of operations that can be performed. Furthermore, it should also be clear why figures produced to advertise a product that state a certain number of operations shouldn't be treated seriously. Simply stating, the number of factors involved and dynamic nature of processes taking place make any such measure very volatile.

Therefore, instead of giving practically meaningless figures, we will answer the question of why Bitlattice is fast and how it's achieved.

There are basically three elements that allow **Bitlattice** to achieve results comparable to performance of centralized systems:

COMPARTMENTALIZATION
: This feature, mentioned already before, enables parallel operations on the ledger, as long as operations are performed inside a compartment. They do not have to have a fixed size – this matter depends on implementation only. For instance, in networks with large amounts of micropayments sizes should be bigger. Creation of compartments can depend on different factors like geolocation, subject of operation, servicing a certain DApp.

HOMEOSTASIS
: To ensure optimal performance **Bitlattice** makes use of several, very simple, yet powerful mechanisms based on combination of negative feedback with implementation specific constraints. They allow **Bitlattice** to keep a safe balance between resources at hand and processing demand. That ensures that there is always some reserve kept.

DECENTRALIZED CENTRALIZATION
: **Bitlattice** is, as already mentioned before, an internally centralized system consisting of externally decentralized peers. That allows for behavior that is close to that of centralized systems – the middle layer has all information of state of the storage. That information isn't precise, it's rather a fingerprint of data, but precise enough to decide on a tasks' distribution and allow or disallow certain operations. The analogy to centralized systems is close – indexes are usually more readily available than the data itself. The latter must be read to be verified. The middle layer with its smart entities takes care about discovery of conflicts, this time based on positive feedback, which is needed to limit effort entities invest in verifying data, by narrowing their scope to the lowest number of compartments possible. Ideally it is the single compartment, if computing power is abundant.

Presented mechanisms make **Bitlattice** adaptively optimal in terms of performance. That also means that regardless of circumstances **Bitlattice** will achieve the highest reasonable processing speed. Reasonable, as there there will always be a reserve capacity to trigger a response to unforeseen circumstances.

4 Potential use cases

A fundamental goal is to make Bitlattice an open-source platform for innovation and human progress enabling contributors from around the globe to build a better future. We anticipate interest in a vast spectra of real world use cases: instantaneous transactions, machine learning, data marketplace, 100 percent secure voting systems, smart contracts, AI, distributed applications, infrastructure orchestration and Internet of Things to mention just a few of them.

We decided to list potential use cases in an ordered manner to facilitate understanding in what areas Bitlattice can be applied. While we only covered select general cases, they provide a practical perspective on issues Bitlattice is able to solve.

4.1 General properties

Before we present specific use cases themselves, it's worth mentioning properties all of them share when built upon Bitlattice.

NO HUMAN INTERFERENCE

The whole idea of Bitlattice was designed around excluding human factor from core processing at all. Therefore, while Bitlattice can accept input from humans (and any other input as well, from AI or sensors, for instance) that input isn't an order to perform an operation, but rather a wish that may or may not be granted. At the same time, the middle layer of Bitlattice is firewalled from the outside world thanks to strong encryption that only enables very constrained in form communication. That approach guarantees that the effect of processing cannot be manipulated.

FLEXIBLE COMPUTING

Computing can be provided in a wide range of scales, from very basic IoT devices to powerful server clusters. Bitlattice is designed to adjust performance to available power and keep it on a certain, homeostatic level.

BUSINESS CONTINUITY

Self balancing plays also a major role in ensuring proper reaction to disastrous circumstances. Bitlattice being a distributed system, is resistant to loss of parts of infrastructure, even if that loss is not recoverable. Properly adjusted data multiplication can ensure no data loss at all. But even if the data is lost due to a very wide malfunction the rest of network can still function due to compartmentalization.

The above properties apply to any solution that uses Bitlattice for whatever purpose.

4.2 Finance

The financial world depends is all about operating on ledgers that ideally should be tamper-proof, confidential and exact. Bitlattice provides new opportunities due to its properties that match the above requirements.

Coin

Probably the first use of Bitlattice will be as a coin cryptocurrency. Reasons are threefold: coin is easy to implement, smart contracts provide means to expand the ecosystem very fast, coins provide fast capital accumulation in the system. Main assumptions with respect to the first implementation as a coin are:

- Flat, balanced earnings for coin users – earnings aren't generated from mining, but from capabilities of devices connected and their actual computation contribution. That doesn't exclude large actors providing vast resources, only attempts to equalize internal economy;
- Open ledger – visible transactions;
- Smart contracts enabled;

This document isn't intended to present the coin aspect in detail – a separate document that will address this particular implementation, when it will be nearing deployment.

Digital currency

Global and country wide digital, officially issued, currencies start to be a subject widely discussed and first attempts to implement them are already in place.

While there is little doubts that first attempts will differ substantially from presumptions behind cryptocurrencies, as in a hand operated, scripted economy, the medium to mediate a value must allow for hand cranking as well, we deeply believe that people ultimately will resort to rational solutions when all other possibilities are exhausted. The most rational solution is to use a not manipulable and highly redundant medium – Bitlattice.

To name the most prominent prerequisites, why that solution is the most rational:

- International relations depend on a huge number of more or less important factors. Humans inability to comprehend too many stimuli leads to the mess we have since always. While we develop tools and institutions that are meant to tackle that mix, they seldom address actual issues.

Probably the most prominent factor that shapes our civilization is money. It enables exerting power, battering rivals, grabbing resources, controlling population and just by accident enable value exchange. The amount of games it enables to be played is enormous. We could probably play them indefinitely, but our time for such childish entertainment is limited. Just around the corner there can be a black swan event that can send us into oblivion if we are not properly prepared. A step of Great Filter. Sustainability[16] , understood as our determination to sustain the existence of our species, requires actions that could maximize our chances.

16: read more in an article by Hibryda.

Therefore, if possible, we should rework faulty elements of our society, instead of patching them indefinitely. Money, in the global scope, is an obvious candidate to start with. It's an artificial concept that we have all tools to give it shape. At the moment it's unreliable and hand cranked. To make it reliable it's enough to remove cranks. **Bitlattice** is able to isolate internal ledger from external attempts to influence its form.

- As mentioned above, there can be black swan events that could strike our civilization at any moment. While we don't know their nature and possible conseqences, the more we are prepared for unknown threats the better. The financial system behind our economy isn't adaptive. That can be proven by historical examples of crises that plagued us both globally and locally. All of that without any major, planet wide threat – just some minor, when it comes to the scale of our civilization, events. If we only could build in our financial system tools that attempt to retain homeostasis and avoid larger spikes and drops in available value... **Bitlattice** can do that.
- The ideal (or close to ideal) solution for value exchange on the global scale should be scalable and resistant to local loss of parts of the system. **Bitlattice** is build around the idea of being able to scale and operate on smaller chunks (compartments). When there is no access to the rest of the network, remaining part can still operate independently as long as operations that are meant to be performed are local as well.

  In global implementations, compartments could be arranged geographically (the way they are arranged depends only on chosen factors – for instance in retail aimed solutions, compartments can be isolated based on types of products) therefore providing a resiliency mechanism built directly into the medium of exchange.

To sum the above – designing a tool for global exchange of value shouldn't be about casting existing, flawed and deficient, system on a digital platform, as that way the only outcome is getting a similarly dysfunctional product, that just processes faster and degrades quicker.

Instead, a correct tool mitigating most damning deficiencies –

Ʒitlattice – should be applied.

Financial Institutions

Tasks that financial institutions (banks, insurers, etc.) must process and execute are usually fairly constrained and not particularly complex in themselves. Therefore, this industry already leads in implementation of smart algorithms to cut costs, earn more and minimize human intervention. That however, touches only select areas of activity, as there is no commercially available versatile base, that could combine outputs of separate algorithms, fed data to others, provide confidentiality and security, centralize the management, be resistant to manipulation and have built-in disaster recovery ability. **Ʒitlattice** can address those issues.

As **Ʒitlattice** can be internally nested via use of task specific interfaces, realization of a whole structure under a single monolithic system, covering all tasks that doesn't require human intervention, is possible. The field of financial institutions is very broad, thus it's not possible in such short form to address different possible situations that can be managed via **Ʒitlattice** based system, without resorting to oversimplification.

Fairshares, Coops, Commons Management, Tokenomics

Ʒitlattice can also manage a sidetrack of economy – socially oriented, purpose built systems. Their implementation is possible on two levels – either via smart contracts interpreted and sanitized before processing or directly in the middle layer. The first way provides an interface dependent on a selected interpreter with all security mechanisms engaged, the second one provides more flexibility for a cost of having to design safeguards independently.

4.3 Governance & Society

The transformative journey toward an e-State and e-Community including managing social relations, facilitating decision making, automation of bureaucratic processes and data management is a complex challenge. **Ʒitlattice** with its inherent ability to combine governance with AI mechanisms enables connected systems to achieve continuous evolution. Participation allows to build an inclusive society. Below we present a limited in its scope list of potential society and individual-in-society oriented applications.

Global personhood

In interconnected world there is always a persistent tension between anonymity and accountability that stems from our inability to deal with those two, seemingly exclusive phenomenons. Pseudonym Parties'[17] extension – Pseudonym Pairs[18] – can provide a solution enabling creation of globally recognizable digital personhood allowing persons to remain anonymous while being identifiable and accountable. Connecting such system to

17: see this article

18: see this article

commercial and public services would glue together a right to remain private with ability to interact socially. Properties of **Bitlattice** enable realization of such project on a global scale with regionalization based on compartmentalization. Added value would be also a resistance to manipulation that could be exerted otherwise by private and state actors.

VOTING

Voting is a common way to crowd decide about certain issues, either local or state wide. To make voting reliable several rules must be applied and strictly obeyed:

- The ledger of votes cannot be mutable (manipulated);
- Voter cannot vote multiple times;
- Voter should be able to vote in secrecy;

Bitlattice is the best available solution to implement voting accounting for several reasons. Once it's implemented and the middle layer starts processing there is neither insight into how it's performed, nor any practical way to change it. That satisfies manipulation related caveat above. Inability to manipulate is independent of scale – no matter whether a malicious actor is a person or a state agency – they cannot alter the state of the ledger.

Provided voters have individual identification keys, any subsequent input after the first one will be discarded, which satisfies the second caveat. Voting can be performed from any place with network connectivity. That satisfies the third caveat. Even if temporarily the large scale connectivity is lost, **Bitlattice** is still able to store results in local compartment if there are some capable devices in local network.

Bitlattice can also be set to be Markovian[19] , making it a dynamic, memory-less oracle and effectively erasing (or rather not storing) all traces of voters actions leaving only the global result. In some situations such solutions could be beneficial, specifically when a voter could be subject of some adverse effects if information of how voted is accessed by a third party, as even given the provision that a voter has own private key, there can be situations when this key is taken from the person/entity. Besides addressing those main rules, **Bitlattice** provides also additional functions, like enabling voters to test their vote, optional correction of vote in certain time threshold, exposing data about regional and global results and many more.

19: which means having properties of Markov chain – such chain doesn't store previous data, only the current state

LAW – STRUCTURE AND APPLICATION

Law is a perfect field to apply artificial inference mechanisms on all stages of its existence due to its constrained and imperative nature. First stage **Bitlattice** could be applied is law structuring. With advances in natural language processing transformation of textual representation into first order logic[20] becomes possible and fairly faithful. **Bitlattice** can further enable use of such preprepared data in many ways, to name a few:

20: or other specific formal system or metalanguage enabling further lossless transformation into logic processable by deterministic algorithms

- Facilitating creation of regulations that are globally consistent with the whole body of law in a jurisdiction;

- Enabling translation of regulations between jurisdictions;
- If coupled with global personhood access to **Bitlattice** based legal oracle could facilitate local and international, commercial and private relations compliant with respectable legal systems.

Second stage can cover the area of dispute settlements. It is worth mentioning here an important aspect of already applied, although still immature, solutions based on machine learning. As mentioned before, AI-based solutions are prone to errors resulting directly from learning from highly processed and therefore incomplete data. Humans, having built-in natural intelligence, are susceptible to the same problems.

In applied law, these deficiencies were identified long time ago and mitigating strategies are in place. Minimum two instances of proceeding, in some jurisdictions multiple professional judges, jurors – all these solutions are aimed at accumulating the input and producing the most credible verdict possible. Here, transition from intelligence based, incomplete information dependent system, into deterministic, quantity based one (quantity based ratios to decide, or true+true situation) is most prominent.

Current known solutions perform very simple tasks in which AIs excel. With added complexity they will become more erroneous. **Bitlattice** provides here a solution, by accumulating inputs of possibly hundreds of different AIs (in globe wide implementation) and deterministically assessing their veracity based on formal rules provided at implementation. Then, it acts as a legal oracle issuing a verdict.

Third stage is a subject side of law. The **Bitlattice** based global legal network, given it's designed along the ways described above, can provide a subject of law with reliable assistance again being a result of fetching and assessing inputs of multiple AIs that are taught from a dataset stored in **Bitlattice**.

Access to public information

Accountability of public authorities plays a major role in every advanced society. While legal systems often contain specific provisions as to disclosure of different types of information, that process remains controlled by authorities that are subject to being accounted for their actions. **Bitlattice**, thanks to its ability to self sign data and process logic, can abstract that process creating a middle sphere between the controlled and controllers where data is initially stored in encrypted form, until a decision is made to disclose it, a certain amount of time passes, or other specific circumstances happen. Then, the data is disclosed without direct human intervention, just based on logical premises or valid input. Either the key to data is published, or the data gets decrypted.

UBI – Universal Basic Income

UBI gained recently a lot of traction and support, probably due to dysfunctional economy we live with and real risk of automa-

tion induced unemployment and poverty. While the idea itself may sound humane, as its main aim is to provide a basis for regular existence to everyone, realization of it can generate many risks, starting from bureaucracy eating a large chunk of funds for the system, doubtful positive activation effect, economy collapse, and ending at practical serfdom.
Nevertheless, no matter how that idea can corrupt in the future, it's most likely going to be applied. Therefore, we think that it should be applied in a practically best way possible, way that minimizes at least some risks.
Our proposal consists of three layers – **Bitlattice** being a base that ensures bureaucracy free functioning, proper accountancy and distributed storage, ValueInstrument[21] logic built in the middle layer of **Bitlattice** and users and suppliers (retailers, service providers, etc.) being external. ValueInstrument is a form of token economy, where tokens carry a diminishing with time, continously or in steps, value. At the same time the value cannot drop below zero, which makes the system debt free. Our proposal is to apply such system to UBI to ensure the following:

- Users have motivation to earn money outside the system, as tokens provide them means only to secure their basic needs;
- Users must spend their tokens with suppliers in a certain time since they appear in their wallets. That ensures that preplaned distribution strategies can be formulated and applied with lower costs due to bulk orders, prior noticing and sure income;
- Suppliers are certain that tokens they receive are automatically changed into non diminishing money;

All logic and actions (like for instance, signing contracts with suppliers, managing supply chain) are done singlehandedly by **Bitlattice**.

21: see ValueInstrument website

Due to space constraints, many use cases that should be mentioned in this section aren't. Therefore, we only signal select important ones below in which **Bitlattice** can be implemented with beneficial results:

- Digital sovereignty
- Civic engagement
- Incentivized participation
- Social accounting
- Creation of commons
- Reliable reporting and journalism

4.4 Energy, Resources & Industry

Industry is what provides benefits of technological progress to billions of inhabitants of our planet. Every optimization introduced can directly translate into well-being of millions. **Bitlattice** can provide many such optimizations.

Industrial Products

Digital twins[22] (DT) of machines and industrial products can let **Bitlattice** identify the best solutions even in large and complex contexts. Thanks to the middle decision layer or (a central internal authority) **Bitlattice** can either decide about steps needed to be taken in order to enforce the best strategy, or provide already processed and sanitized data to machine learning algorithms that can analyze the situation and find the best solution. Moreover, as mentioned in the Chapter 1 (Introduction) integration with AI mechanisms (like for instance said machine learning algorithms) can vary in depth, so in certain solutions both AI and **Bitlattice** can be a single system. Therefore, cost of investments for machines plant optimization can be significantly reduced. Having direct mirroring of physical infrastructure via DTs, **Bitlattice** based system can also process maintenance tasks, automatically order payments for needed new or replacement hardware or products. All potentially with no human factor involved with nearly negligible, post implementation costs.

22: digital twin is a ledger entry that follows the state of a particular physical thing. Depending on application, linking can be declarative or depend on RFID, GPS trackers or other technologies.

Supply chains

Bitlattice enables traceability of commodities like minerals, gas, oil, metals, food and other supply chains. This case is analogical to the one mentioned above with respect to added value provided by **Bitlattice** – optimization of supply, contract based workflows for participating entities in the chain and analysis of available scenarios can be performed by a uniform system with no human operator needed. Additionally however, **Bitlattice** can also manage material rights and other rights and their derivatives enabling direct cast between commodities and market operations. With provided set of rules, either legal or internal to the implementing organization **Bitlattice** can issue certificates and other formal documents required signing it by itself with own electronic key. Such system can also expose certain data allowing for transparency of, for instance, origin of a commodity.

Power and Its Distibution

Power generation and distribution is an ideal area for **Bitlattice**'s application due to several factors like ready data exchange network, current use of light computation-able devices on many stages of process and need for balanced operation and prompt disaster recovery. The same mechanism, **Bitlattice** uses to attain homeostasis can be used to detect problems across the grid, either as a stand-alone solution or a supplementary one. As **Bitlattice** can operate on a compartment's level when not connected to the rest of network it can provide fast local switching in case of emergency. Given that **Bitlattice** can, as mentioned earlier, incorporate AI analytic tools and be set with all such components to form a monolithic system, its decisions would likely be faster and more robust than those made by either human operators or composite systems. Meanwhile, apart from ensuring business continuity **Bitlattice** can collect data, take care about billing and settlement, payment clearances and many other administrative tasks, provided relevant interfaces are plugged into its external layer. All of the above makes such implementations efficient and cost effective.

4.5 Health

Medical sciences and healthcare are areas extremely sensitive when it comes to data confidentiality and reliability of systems – **Bitlattice** design goals contain these properties.

HEALTHCARE

Bitlattice-based data integrity with smart contracts managing data leads to faster, more efficient, and more secure medical data management with extended capabilities of drug and medical device tracking. New data and control models can help facilitate advancements in new medical discoveries. It also provides means to ensure the authenticity of drugs circulating global markets. All this have the potential to significantly improve patient care and global health status. Moreover, with its structure decentralized from external perspective, but centralized internally complex access permissions can be applied globally, allowing for easy access to medical data all around the globe, restricting it to professionals, patient and other authorized entities at the same time.

MEDICAL AND LIFE SCIENCES

There are many possible **Bitlattice**'s implementations in these fields. From organizing and storing research results to active analysis of empirical data with via AI inference mechanisms. **Bitlattice** can extend data availability across the area while keeping it confidential whenever required, for instance with respect to test subjects personal information and other sensitive data. Moreover, **Bitlattice** can facilitate comparison of results using preset criteria and prepare aggregates using externally provided algorithms, without actually exposing any original data to analyzing party.

4.6 Technology

Technology is what powers progress and mirrors its advancement. It's also a common denominator of all, already mentioned cases. As broadly understood technology allows for multitude of applications leveraging benefits offered by **Bitlattice**, listed are only select general cases not covered before or not covered in general terms.

DISTRIBUTED DATA MARKETPLACE

Bitlattice decentralized fabric enables disconnected and connected data access and services offering secure transactions and automated smart contracts to enterprise data streams and devices. Marketplaces can be used to stream many data from any source, IoT devices, physical assets, autonomous cars, drones, and many more. **Bitlattice** can preprocess data either via AI layer or other, plugged, mechanisms.

DAPPS – DISTRIBUTED APPLICATIONS

Bitlattice can fully use existing blockchain based applications minimizing the need to overhaul and re-engineering of existing solutions thanks to pluggable interfaces. Furthermore, **Bitlattice** have the potential to enhance current blockchain based applications, offering new capabilities with performance and scalability being inherent features. DApps can be built and deployed on the foundation of a true trustless network.

TELECOMMUNICATIONS

P2P distributed mobile ecosystems or mesh networks in combination with **Bitlattice** could provide an alternative to centralized telecoms, providing resistant to blocking, self managing networks.

DIGITAL IDENTITY

Next generation digital identity management systems have the potential to provide a unified, interoperable, and tamper-proof infrastructure enabling enterprises, users, and IoT management systems to transform into new levels of operation and business models.

MEDIA

Crowdsourced news/journalism and fact-checking in the network will help to create real feedback loops. Personal data marketplace will force advertisers to rethink their business models.

INTELLECTUAL PROPERTY RIGHTS MANAGEMENT

Intellectual property of data can be secured with full tracking of the data and life cycle of any content. This can effectively reduce abuses of rights, protect digital content, and facilitate the distribution of authentic digital data.

5 Funding

Bitlattice has been a privately funded enterprise since its very beginning. This chapter outlines our views on certain financial matters.

5.1 ICO

Our views on ICOs[24] we already expressed publicly. However, to reiterate, we don't consider that way of funding either viable or honest before the product is almost ready to go live. We may organize a limited scale ICO just before **Bitlattice** is made publicly available. Doing it any earlier isn't an option due to:

ADDITIONAL WORKLOAD
Such events requires managing a workforce to take care of many issues. We prefer to focus on important matters first.

COMMUNITY BUILDING
Expanding a community is crucial when it comes to proper coverage of an ICO event. However, it requires time and the mentioned workforce to achieve proper results.

LEGAL ISSUES
Organizing an ICO requires careful legal planning and preparation. The rules are constantly changing and vary wildly between government jurisdictions. Also, it's viable to have the backing of a certified financial institution (more on that in Chapter 6 (*Legal*) on page 29), as that way certain operations impossible otherwise are doable and compliant with regulations.

24: **I**nitial **C**oin **O**ffering

5.2 Other forms

We often receive inquires about the possibility of investing. We are happy about that as it means that we draw some attention. However, as we have some funds secured we are picky when it comes to partners. We have certain expectations and a hard set rules – you can learn them by contacting us.

We are open to welcoming people that have a vision and understanding of the potential gravity of Ɜitlattice. Also, understanding that investment in new technologies (and investment in general) is about gambling. There's a reason behind game theory having such enormous impact on modeling economies. So, if we decide to partner with an entity, it will certainly not be one that bets on blacks or reds, but one that bets on a single number[25] .

25: reference to roulette rules

We often receive inquires regarding partnership with us, mainly from institutions, and are often asked about the business plan and the roadmap.
As to the roadmap, it's here in Chapter 7 (*Roadmap*) on page 31.
As to the business plan – we always wonder what for? Seriously, what is a purpose of such document when it comes to a potentially revolutionary product like Ɜitlattice? Business plans are about presenting a product (which this document does), presenting a spending plan (which requires prior discussion and can be summed up in a short email; anything longer is usually an expression of the writer's imagination and seldom meets future facts) and predictions as to the future returns. Those predictions make sense when it comes to a hot-dog stand, and even then not always. When it comes to a new, innovative product, future is a great unknown. And we know no reliable fortune-teller.
We've heard various explanations as to why a business plan is crucial. One of them being that it must be presented to accountants. While that doesn't affect the lack of rationale behind such a paper, it's at least an explanation. We've been told that accountants like hockey stick curves[26] – here we can help. The below formula is rather self-explanatory.

26: while there are really nice hockey stick curves here and here accountants aren't rather famous for their sense of humor.

$$\sqrt{\frac{(x-l)^p}{r-x}} * m + 1$$

$$x, l, r, m \in \mathbb{R} \quad and \quad l < r \quad and$$
$$for\ convenience\ p \in \left\{ n \in \mathbb{N}_* : \bigvee_{k \in \mathbb{N}_0} n = 2k+1 \right\}^*$$

*: :D

We assigned below values as they best depict our predictions as to dynamics of ROI:

$$\sqrt{\frac{(x-2)^5}{5-x}} * 30 + 1$$

Which produces a curve on the next page:

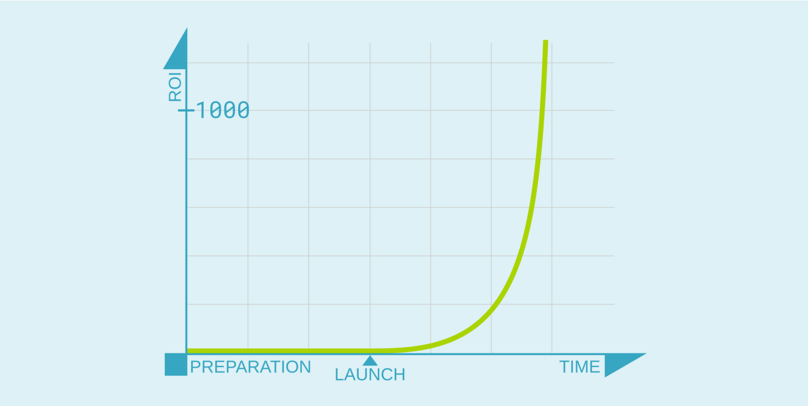


Figure 5.1: Return on investment, with vertical units being based on the assumption that initial investment is equal to 1

That should be enough to please accountants, at least for a while.

That was a test, if you went thus far without feeling offended and if you read that you probably lack that huge, sharp corncob down back there that would certainly render our cooperation impossible. That said you can be a person that we could find a common language with and you should email us.

6 Legal

There are certain legal aspects of our project that certainly need some clarifications.

6.1 Open source

Bitlattice is meant to be released as open-source and free of charge to all willing to build upon it. That however doesn't mean that its license won't include certain restrictions, as open source software isn't synonymous to free software. Practically speaking, there is no strict definition stating what the open source model is, some interpret it extensively, some narrowly. For us, open source means exactly what those words together mean in English – that the source is open (to public). Richard Stallman wrote an extensive article[28] about differences between free and open-source software. While we respect values that stood behind emergence of the free software movement, we are also critical about the current state of matters as it seems that really grand values can become gravely corrupted. Therefore, we decided to pick our own way and follow what is the best for our project, ignoring big words and focusing on rational solutions.

28: Read it here

6.2 Organizational structure

The whole enterprise is planned to be eventually composed of three main legal entities listed below:

COMPANY

Its main function is to bind all other components together, coordinate them, create and supervise daughter entities, distribute tasks; The company already exists.

FOUNDATION

Foundation is meant to take care of delivering a reference client and base software. Its main goal is to provide transparency of the standard and enable community participation and oversight over shaping a form of the ecosystem;

FINANCIAL AGENCY

Its aim is to make possible and facilitate operations on financial markets;

6.3 ASIC chip

Apart from the already mentioned technical benefits from having a specialized chip serving the **Bitlattice** protocol, there is also an important legal matter that affects both the financial and the IP rights related to the enterprise.
Our aim is to order manufacturing of a reference chip to be able to patent it worldwide. That way, we can enforce a standard for **Bitlattice**'s implementation as entities willing to manufacture the chip will be bound by that implementation and any change in design will have to be approved by the **Bitlattice**'s foundation. Apart from the above, income from licenses to manufacture chips is meant to ensure stable financing for the enterprise for many years. The inner structure of the chip will be made public as open-hardware (with the same caveats as those that apply to previously mentioned open-source).

6.4 Intellectual property rights

All IP rights related to **Bitlattice** are the property of Hibryda. The enterprise is given a license to both make use of those rights and manage them. Hibryda, before releasing the first implementation, will make sure that licenses provide users and the enterprise with unrevocable rights to use and modify **Bitlattice**.

7 Roadmap

Our roadmap presents the tasks needed to complete each phase. See the note on the margin for explanation of time badges.

The two numbers that are listed aside of the flowchart are respectively the low (larger) and high (smaller) estimations of time needed to complete a phase given in months.
The low estimation is an optimistic variant that assumes that the Universe likes us, all goes smooth and no unexpected circumstances happen.
The high estimation is realistic and conservative.
The time taken for each phase will most likely fall between the two.

8
22

PREPARATORY PHASE

Preparation of the ecosystem of Bitlattice:

- core programs for all target architectures;
- finishing specifications;
- providing detailed documentation;

Dynamic testing

Setting all elements of formal infrastructure

Preparation of ASIC implementation and patenting

3
8

PRE-DEPLOYMENT PHASE

Deployment of a prototype network

Presentation of full technical details

Crowd testing

Potential ICO

2
5

DEPLOYMENT

Deployment of a Bitlattice coin

Deployment of a other closely related projects

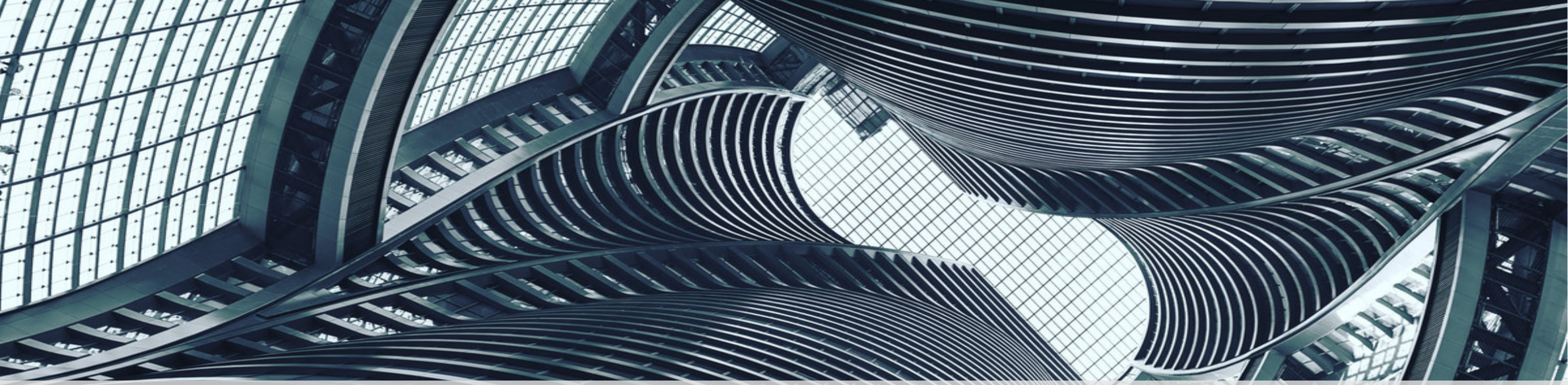
POST-DEPLOYMENT

Maintenance of code base

Development of further solutions based on Bitlattice

Training and popularization

APPENDIX



A Motivations – in depth view

The motivation behind Bitlattice's conception deserves a closer look, as it indirectly shows its design goals and potential uses.

A.1 Scripted economy

The economy, in any advanced society, is subject to regulations and limitations that have nothing to do with actual value exchange. There's nothing odd in having it skewed toward centers of power. There's nothing odd that together with scarcity of resources come attempts to squeeze a last bit of value or to make it out of thin air, so maybe no one will notice. It's a recurrent motif in history, kings chasing money forgers while forging it on their own on a regular basis.

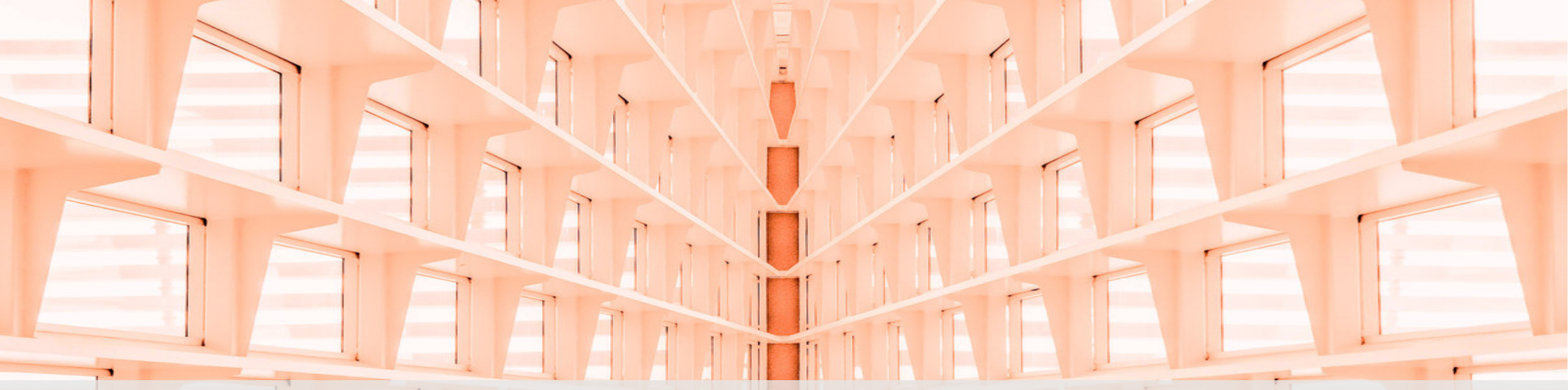
The only issue is scale. Since the early seventies, as a result of the Bretton Woods conference and the fiat currencies of attending countries being detached from any real world backing, the worldwide economy has become a scheme that allows for any arbitrary value creation, practically without limits. The only limit being that in some countries, other than Zimbabwe, politicians and bankers abusing the system too aggressively could end up on lampposts. Nevertheless, thanks to a regular practice of creating money from nothing over the last half century, the world's economy became a monstrous creature impossible to save. Recent emergence of dynamic economies, China in particular, complicated matters. The Bretton Woods agreement died without an issue. Meanwhile, China, Russia, India and Middle Eastern countries find ways to exchange value based on local currencies, leaving a dying monster to its fate. The problem here is that without a common system of value exchange particular economies will have to gamble whether to trust the other party or not. As history shows, such gambling is a risky business.

Providing a system of value exchange that eliminates both the need to trust the other party and the possibility of altering the ledger could solve at least that issue.
We are perfectly aware that before interested parties even notice its existence, the world will suffer multiple further economic crises. Though, ultimately, the only way to bring some sanity to the economy is by providing a tool that abstracts value exchange from any external, human influence. Because humans cannot be trusted.

Besides, great ideas must start with great aims.

A.2 Issues in distributed ledgers

The whitepaper of Bitcoin states the most important issue of blockchains in its introduction – the 51% problem. Since inception of Bitcoin that particular issue was addressed in numerous ways. Nevertheless, the issue is inherent to blockchains as there is no practical way for a client to tell the difference between two valid sets of data. While the issue applies to all distributed systems, it may have adverse effects only when a malicious actor is able to modify the dataset. **Ɔit-lattice** introduces a set of encrypted middle agents (called entities) that abstract the ledger from interacting clients and serve as a swarm of provers enabling zero knowledge proofs. As their processing is hidden from clients, there is no way to overwrite information in a ledger while retaining its validity. There are more issues in distributed ledgers, the two prominent ones, related to each other, are scalability and linearity. However, it's not the number of issues that server as a motivation, but rather the fact that promises this technology made could not come true.



B Interface problem

There are two profound and permanent, paramount problems[32] of computation – interface problem and physical limits of computation. While the latter is well covered in many papers and articles[33] the former is seldom mentioned in a form described below and even its name was coined by us ad hoc. While problems found in binding different interfaces is often investigated, a more general approach is seldom touched, be it due to being seemingly obvious, trivial or considered not so important. No matter the reason we were unable to find any theoretic considerations on that subject, nor we intend to present such here as this document attempts to present Bitlattice and related subjects to a larger audience.

32: jokingly referred to as PPP class of problems.

33: For instance this article on the subject.

That said we should start from useful analogies that make the matter less abstract.

B.1 Cognition in humans

Human cognition, when it comes to data gathering, depends on a multitude of sensors that provide information about surrounding environment. That is the first of a series of interfaces that are present in our cognitive chain. "Interface", because the particular sensor doesn't transfer a real state of a measured quantity, instead presents only a time delayed, space constrained, initially preprocessed chunk of data to higher layers.

Speaking on higher layers, those who never touched that subject in detail often tend to perceive our nervous system as a more or less monolithic structure. In reality it's highly modular and flexible setup, comprises of logically (and often morphologically) separate structures, fact that is proven by both studies of pathological cases and recently studies of simulated equivalents [34]

34: Sensory deprivation is particularly interesting, as we know that humans (and other animals as well) tend to produce fake sensations with absence of a real ones. It seems that, according to this paper, the mechanism behind that phenomenon is very low level.

One of most apparent examples of senses that distort the actual input is our visual perception. There are many interesting researches on that subject[35] . They offer insight into how really our perception works and how much data we actually collect. Our eyes present to higher cognitive areas a distorted, inverted, narrow and limited when it comes to wavelengths depiction of environment. Only at that stage the amount of data that successfully crossed the interface is several

35: For instance: this paper or this paper orders of magnitude less descriptive than the data transmitted to our eyes via photons. Yet, even with that narrowed dataset our brain cannot effectively deal as the amount of data to process is still too vast for the computation unit inside our skull. Therefore, to save computation, the data is transformed again. It's normalized and subjected to features extraction to enable fast inference from observed factors. Very similar workflow to the one in some artificial inference systems. However, at this moment the data no longer bear any resemblance to the initial image that arrived to retina.

Our brains are computationally limited. While not so pathetically primitive as the devices we make and use, thanks to millions of years of optimization and balancing tradeoffs, they still cannot process too many signals at once[36] . Therefore, limiting the amount of data is crucial to ensure proper functions. However, that also leads to humans being prone to errors, distraction, deception, biases as the representation of environment always differs from reality.

36: This paper attempts to verify the hyperlearning hypothesis and shows that neural networks that are subjected to overflow of information consistently classified as important can develop symptoms comparable to those of schizofrenia in humans

B.2 Machines

Machines are subject to the same issue as data transfer is either lossless or lossy, never gainy. Crossing an interface results in either additional computation or loss of data or both. Examples of interfaces are numerous, to name just the most proficient ones: analog to digital converters, network interfaces, digital sensors. All either retain the data unchanged for a cost of additional computation or discard a part of it. What matters here is scale of data loss and/or effort engaged to manage transfer. At the moment we are in an analogous situation to that of energy crisis in 1970s. Our solutions are big, inefficient, wasteful, but we already realize that some problems, that we weren't aware of, exist and should be solved. We're still deluding ourselves that there are some magical solutions that are going to pop anytime soon. Artificial inference mechanisms (AI, ML, etc.) and quantum computers will change the landscape, though most likely less than all expect, as both those paths suffer from own deficiencies and, what is very important, are greatly impacted by the interface problem. Artificial inference mimics and extends how our brains work[37] . It also exhibits similar traits that plague biological brains. Being dependent on highly processed input they are prone to the same issues as mentioned above in relation to humans. Next installment of hacker wars will most likely concentrate on polluting datasets with hard to detect residue that may affect inference long term and render some solutions unreliable. Again, interface problem. Quantum computing, is incredibly performant in select and narrow range of applications, in others can perform comparably or even less efficiently than regular computers. It's also very hard to tame and, of course, has a prominent problem with interfacing the outside world as most data is lost in this process. The two promising branches of technology listed above doesn't give an exhaustive picture of solutions that may potentially boost progress. Multistate gates on graphene ribbons and similar promising solutions

37: It's a simplification, though not very far from truth.

should also be mentioned, though this document isn't intended to investigate why progress in IT so abruptly halted and what potential solutions can move it forward.

B.3 Conclusion

The interface problem is about a conclusion that data is either lost or retained for a cost when crossing an interface. There is a question that arises here – is it possible to evaluate how much data must be retained, given a certain computation effort, to effectively infer about reality with a specific degree of confidence. That question is seldom answered while designing computerized systems, despite the fact that information theory contains tools that could deliver sufficiently accurate estimations in most cases. The result is obvious, we collect extensive amounts of data, but benefits from this collection aren't growing in parallel. And no matter how fantastic the new gizmo implemented is, feeding garbage results in getting garbage[38] . Always.

38: GIGO – **G**arbage **I**n **G**arbage **O**ut.

That's also one of reasons behind rather low adoption of blockchain derived solutions philosophy behind immutable ledgers substantially differs from the one behind current data collection and management. Blockchains et consortes can accept whatever gets pushed into them, but that data cannot be later removed or altered. The computation effort to run such storage grows until it's no longer viable to maintain it.

Bitlattice while also being subject to the same issues caused by the interface problem addresses them instead of pretending they do not exist. Multilayered processing allows for flexible data preprocessing, minimizing and distributing computation effort. While immutable by default, the central internal authority can perform data sanitation and apply retention limits if it's designed with them in mind. Distributed storage with balanced saturation provides means to store far larger amounts of data without affecting performance. The central internal authority can act as a reliable oracle if provided with sufficiently reliable input.

Referenced resources

Here are the references arranged in the order they are mentioned in this document.

[1] *ACID*.
last accessed 16 Nov. 2019;
link: en.wikipedia.org/wiki/ACID. Nov. 2019.

[2] *Fully homomorphic encryption*.
last accessed 16 Nov. 2019;
link: en.wikipedia.org/wiki/Homomorphic_encryption. Nov. 2019.

[3] *The Babbage Engine*.
last accessed 16 Nov. 2019;
link: www.computerhistory.org/babbage/.

[4] *Celebrating Penn Engineering History: ENIAC*.
last accessed 16 Nov. 2019;
link: www.seas.upenn.edu/about/history-heritage/eniac/.

[5] Hibryda. 'Sustainability and trust'. In: *Medium* (Sept. 2019).
last accessed 16 Nov. 2019;
link: medium.com/@Hibryda/sustainability-and-trust-7e340e151600.

[6] Richard Stallman. *Why Open Source misses the point of Free Software*.
last accessed 16 Nov. 2019;
link: www.gnu.org/philosophy/open-source-misses-the-point.html. Apr. 2019.

[7] Seth Lloyd. 'Ultimate physical limits to computation'. In: *Nature* 406.6799 (Aug. 2000).
available online; last accessed 16 Nov. 2019;
link: arxiv.org/abs/quant-ph/9908043, pp. 1047–1054. DOI: 10.1038/35023282.

[8] Michael Deistler et al. 'Tactile Hallucinations on Artificial Skin Induced by Homeostasis in a Deep Boltzmann Machine'. In: *arXiv e-prints*, arXiv:1906.10592 (June 2019).
available online; last accessed 16 Nov. 2019;
link: arxiv.org/abs/1906.10592, arXiv:1906.10592.

[9] Daniel C Dennett, Michael Cohen, and Nancy Kanwisher. 'What is the Bandwidth of Perceptual Experience?' In: *Trends in Cognitive Sciences* (Apr. 2016).
available online; last accessed 16 Nov. 2019;
link: hdl.handle.net/1721.1/112190. DOI: 10.1016/J.TICS.2016.03.006.

[10] Ruth Rosenholtz. 'What modern vision science reveals about the awareness puzzle: Summary-statistic encoding plus decision limits underlie the richness of visual perception and its quirky failures'. In: *arXiv e-prints*, arXiv:1706.02764 (June 2017).
available online; last accessed 16 Nov. 2019;
link: arxiv.org/abs/1706.02764, arXiv:1706.02764.

[11] R. E. Hoffman et al. 'Using computational patients to evaluate illness mechanisms in schizophrenia'. In: *Biol. Psychiatry* 69.10 (May 2011).
available online; last accessed 16 Nov. 2019;
link: www.ncbi.nlm.nih.gov/pubmed/21397213, pp. 997–1005. DOI: 10.1016/j.biopsych.2010.12.036.

Contact us via...

email:

bitlattice@bitlattice.org

or

bitlattice@protonmail.com

Twitter:

@bitlattice

Or visit our website at

bitlattice.org

Revisions history

Version	Description	Date
1.01	Release-ready first version with some fixes	19 Nov 2019
1.0	Release-ready first version	19 Nov 2019
0.3	Pre-release version	17 Nov 2019
0.2	Structure of the document set	29 Oct 2019
0.1	Document created	21 Oct 2019