
**PUBLIC WHITE PAPER VERSION 3.0**

# An Introduction to the Self-Improving Security Node

***The Cybersecurity Guardians that Protect Blockchain***

JAROONA CHAIN OÜ

Contact:
Anna Bacher, CTO
anna@jaroona.com

December 6th, 2018

# Legal Notice

The ownership of Jaroona tokens (JSN) does not represent any participation in Jaroona capital nor any rights of payment, remuneration, profit distribution or money reward of any kind. This Whitepaper has been prepared in good faith to provide a comprehensive overview of the Jaroona Project and JSN Token Crowdsale and is for information purposes only. With the development of Jaroona Software and its services, it may be amended in the following. Please also note that the Jaroona Project itself may be redesigned/reshaped in future, if that would be required for any material reasons (including, but not limited to: commercial considerations, technical possibilities, or the need to ensure compliance with any (existing or future) applicable laws and regulations, or any other material reasons). JSN tokens are not intended to constitute securities in any jurisdiction. This Whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities or a solicitation for investments in securities in any jurisdiction. The contents of this Whitepaper are not a financial promotion. Therefore, none of the contents of this Whitepaper serves as an invitation or inducement to engage in any sort of investment activity.

Do not contribute any money that you can't afford to lose in the Jaroona Token Sale. Make sure you read and understand this Whitepaper and TERMS AND CONDITIONS FOR PARTICIPATING IN THE JAROONA TOKEN SALE (including all warnings regarding possible token value, technical, regulatory and any other risks; as well as all disclaimers contained therein), as published on our website **www.jaroona.com** (and as they may be amended from time to time). For any questions regarding Jaroona Token Sale or Jaroona Software services please contact us via e-mail at address **office@jaroona.com.**

# Introduction

Blockchain will benefit society in profound and diverse ways. By ensuring that value is correctly and safely exchanged between parties, blockchain technology will transform the industries that rely on centralized assurance mechanisms, acting as a powerful, transparent force against corruption.

Before this potential is realized, we must address and overcome the security limitations that exist within our current blockchains—this includes modifications to consensus algorithms and the typical patchwork of fixes to potential threats (most of which are unsuccessful).

This approach cannot and will not deliver the performance, reliability or adaptability that is required for the global adoption of blockchain technology.

At Jaroona, we envision a decentralized and secure Web, where businesses and users can communicate freely, distribute value and be protected from the centralized threats that aim to steal assets and data from our blockchains.

To ensure safety and security for everyone, we're introducing a new, revolutionary type of blockchain node—the Jaroona Security Node (JSN). The JSN acts as an automated, distributed cybersecurity guardian with the power to protect entire blockchains, DAPPs and your own personal assets and data.

Based on machine learning algorithms that automatically improve and adapt to new security threats, the JSN will work with both new and existing blockchains, ensuring its security benefits are available to everyone.

Picture a world where every mobile phone or desktop computer has the power to act as a JSN, earning income for protecting our blockchains, whilst simultaneously protecting what matters most.

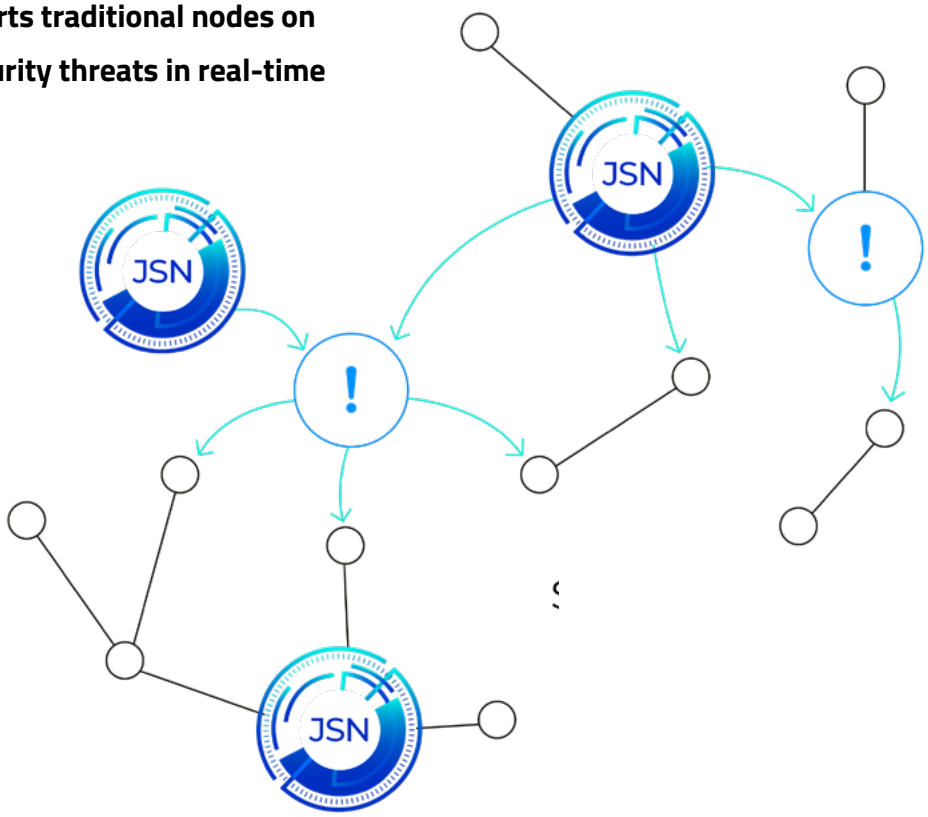
Existing blockchain nodes will also be able to act as Security Nodes, with each community able to establish their own fee and reward structure. Regardless of location or income, anyone and everyone will be able to protect our decentralized world by running a JSN.
Without the JSN, the blockchain ecosystem is incomplete, but with the JSN, we are able to protect the decentralized Web at scale, using JSN to protect our community, data, friends and blockchains.

Christian Bacher, CEO

# The Jaroona Security Node (JSN)

**Self-learns and adapts to emerging security threats automatically**

**Alerts traditional nodes on security threats in real-time**




**Runs on any type of hardware**

# Contents



# Blockchain and DAPPS

Distributed Ledger Technology (DLT) will transform the way people and organizations handle identity, transactions and the exchange of information. It provides an electronic, public record of integrity without central ownership[1].

DLT has already experienced several important developments since, and the scope for potential use-cases has expanded with each evolution. Initially, DLT was only used for simple quasi-monetary transactions between parties. It solved previously unsolved computing problem, known as the *Byzantine Generals Problem*, but the utility of blockchain beyond its impact on the financial industry was limited.

The next important development was the integration of a programming language, allowing for creation of smart contracts, which enabled sophisticated interaction between users beyond the transfer of virtual money[2].

This enabled the development of a new breed of decentralized applications (dApps), connected to the blockchain, allowing them to expand beyond traditional distributed applications like BitTorrent, Popcorn Time, BitMessage, and Tor[3].

A number of blockchain protocols and networks have emerged since 2015, and over 1,450 dApps can now be found on the curated list of decentralized apps on Ethereum alone[4].

While there are several exciting dApps within the blockchain ecosystem, industry adoption is severely lagging. Total count of daily active users for all dApps is barely over 10,000[5] users, which is negligible when compared to centralized applications, like eFacebook, YouTube and WhatsApp (which have over 5,000,000,000 daily active users collectively[6]).

# Barriers to Adoption

Distributed Ledger Technology promises secure and transparent peer-to-peer transfer of digital goods, including money and intellectual property, assuring the value exchanged between parties, without the requirement for a central authority to validate said transactions.

While the Blockchain networks utilizing Distributed Ledger Technology generally solve the problem of peer-to-peer transactions and transparency, none of them are able to simultaneously deliver on the promises of decentralization, speed (measured in transactions per second), and safety without sacrificing one for the others.

While many blockchain projects focus on increasing the speed of transactions, an effective and scalable security solution is yet to be designed and deployed.

# Security

Blockchain technology relies on a ledger to keep track of all transactions. Most transactions represent a transfer of digital assets *"tokens"* or *"cryptocurrencies"* between parties. Since digital assets tend to hold financial value, which makes it vital to ensure the security of such transactions.

A real-world example of a smart contract attack that gained mainstream coverage is the DAO exploit. The DAO (decentralised autonomous organisation) was hacked and drained of 3,689,577 ETH over the course of three weeks, and the total list of all cryptocurrency hacks is growing as quickly as the public interest for this new technology and its applications.

Crypto Aware, a decentralized token investment-management community has found that over $1.7 billion-worth of cryptocurrencies were lost to hacks and frauds between 2011 and 2018. Such exploits present another obstacle to mass adoption of the DLT powered distributed applications. Ensuring security of blockchain networks requires a different approach that goes beyond hard forks, soft forks, and network politics.
The surface for potential attacks is growing rapidly as new networks and applications are deployed. At the same time, the raising market caps offer attackers an ever-increasing potential for monetary gains, presenting additional motivation for malicious behaviour.

Current security approaches rely on modifications to consensus algorithms and a patchwork of fixes to known threats, which cannot deliver the performance, reliability, or adaptability that is required. The blockchain community is not currently developing solutions to the most important security threats, such as identifying malware in smart contracts and network attacks.

Patching vulnerabilities discovered through past attacks is costly and may result in forks. More than that, new vulnerabilities will continue to emerge from multiple sources, including existing and emerging programming languages, third-party applications, smart contracts and the consensus economy.

To be effective and sustainable, any method of combating malicious behaviour must be at the same scale as the potential attack surface. Identification of malicious players and vulnerabilities must be a continuous online process performed by a scalable number of nodes in an automated self-improving manner.

# Between scalability, security and decentralization

At the time of this writing, a blockchain network is considered as secure as its consensus model. The blockchain system safeguards the transaction and block order thereby safeguarding all the key properties of blockchain, such as immutability and auditability.

As long as security depends on the power of a consensus protocol, there is a trade-off between the consensus protocol used, scalability and network openness. Each community is required to prioritize one to the detriment of the other and this trade-off results in differentiation between various types of blockchain networks.

Public blockchains (those opened to all users) are not scalable, have high latency and computation costs, but are still considered relatively secure. Private blockchains (closed systems with permissions) can be more scalable, but are biased, unequal and go against the main idea of blockchain—decentralization.

A compromise between the two is found in the grey zone of public blockchains that use more scalable Proof of Stake consensus protocols (such as Ethereum Casper, EOS DPOS). Proof of Stake require miners to purchase tokens in order to get the opportunity to produce a block and this is perceived as creating inequality for potential users. As Vitalik Buterin put it, fees are especially bad for *"the poor, who are not interested in putting the entirety of their often very low savings into a funky new crypto-asset in order to be able to use a blockchain."*

Simply put, the trade-off between **scalability, decentralization and security** results in the following **three types of blockchain networks:**

| 1st TYPE | 2nd TYPE | 3rd TYPE |
|---|---|---|
| **A pure permission-less public network with limited scalability;** | **A semi-public network based on POS/DPOS with better scalability, but with inequality among users;** | **A permissioned network with better scalability designed for big players or consortiums, with a walled garden approach only available only to the select few.** |

Neither of these options are optimal, as each of them sacrifices one aspect in favour of another. In addition, experience shows that a consensus algorithm alone (as mathematically and economically secure as it may be) is not 100% reliable safeguard when it comes to blockchain security.

Consensus protocols secure only the blockchain network itself, but not the underlying smart contracts. At the same time, the public nature of blockchain, prevents securing network perimeters on an individual level.

A number of different mitigation solutions have been proposed by blockchain developers and the community, but all of them are either blockchain network specific or consensus algorithm specific. There is no blockchain security solution at scale that **has four major distinctive qualities:**

- **IS "AGNOSTIC" REGARDING NETWORK;**
- **CAN RESPOND EFFICIENTLY TO CURRENT AND EMERGING THREAT MODELS;**
- **CAN DEAL WITH BUGS AND VULNERABILITIES IN SMART CONTRACTS BY AUTOMATED DISCOVERY AND DEBUGGING PERFORMED AT SCALE;**
- **CAN ADD SELF-IMPROVING SECURITY WHICH IS DECOUPLED FROM THE CONSENSUS ALGORITHM;**

# Inefficiency of existing security measures

A number of security measures and solutions are being developed to improve security without sacrificing decentralization and scalability by Ethereum and alternative blockchain networks including EOS and Polkadot. New solutions are being proposed by various auditing firms and startups, but most of them cannot promise automation, cross-network security, or scalability.

# Ethereum

Ethereum is working to close notorious vulnerabilities within its code base with the support of multiple researchers and white-hat hacker organizations[8]. Vitalik Buterin is inventing new algorithms, such as Slasher[9] and Casper[10], which aim to scale Ethereum without compromising security. We are yet to see their deployment in action, and while they may not compromise existing levels of security, they do not offer any solutions to move beyond modifications to consensus algorithms and the patch work of fixes against identified bugs.

# Eos

EOS utilizes Delegated Proof-of-Stake protocol (DPOS) and in-built algorithms to handle potential forks. EOS argues in a Technical White Paper that "under normal conditions a DPOS blockchain doesn't experience any forks because the block producers cooperate to produce blocks rather than compete. In the event there is a fork, consensus will automatically switch to the longest chain." [11] Dan Larimer (CTO of EOS) says, *"I fully recognize that voting is not ideal, but it is currently the best approach when factoring in all risks, attack vectors, and recovery options."* [12] EOS.io also introduces accounts and permission management to combat potential security issues.

# Polkadot

Polkadot introduces a high-level concept of a Fisherman node. As Polkadot white paper states: *"...Fishermen get their reward through a timely proof that at least one bonded party acted illegally. Illegal actions include signing two blocks each with the same ratified parent or, in the case of parachains, helping*

*ratify an invalid block.*" [13] Polkadot relies on the Fisherman's reward concept to combat security threats, but doesn't exactly specify which types of threats will be handled by Fisherman— *"...due to the existence of fishermen, we expect events of misbehavior to happen seldom, and when they do only due to the bonded party being careless with secret key security, rather than through malicious intent.*"

# Other Solutions

There are a number of blockchain startups working on a different approach to ensure the security of blockchain networks and smart contracts. Most of them focus on providing developers with compilers to identify vulnerabilities in the smart contract code. Some of them developed solutions, which can detect batchOverflow (and similar vulnerabilities) in Ethereum Smart Contracts. Such tools can be effective, but their implementation is fully dependent on individual developers, and therefore lacking cross-network scalability and efficiency.

An individual may be able to examine the smart contract's source code before performing transactions and interacting with new parties, for Ethereum smart contracts, they may use Etherscan during the process. This is not an automated process and most users lack the technical capabilities to analyze the source code and identify malware. In addition, this is a post-deployment process, so the contract may have already been executed and already caused financial damage. This approach is not scalable and is fully dependent on the deliberate action of individuals who are striving to protect their own smart contracts. Such approach also fails to address one of the major cybersecurity problems—identifying whether a smart contract is itself malware.

There are a number of smart contract auditing firms that certifies that smart contract contains no vulnerabilities and performs actions stated in its specification. The certification process happens before smart contract is deployed on the blockchain. This approach only verifies the smart contract state at a certain moment of time. It does not guarantee that bad actors will not find new vulnerabilities in the future after the contract is deployed. Therefore, offline smart contract auditing is a good stepping stone towards protecting a smart contract, but can't be considered as a robust end-to end solution to ensure funds, reputation and personal data is secure on a blockchain.

None of the security measures introduced so far are sustainable in the long term. To address these shortcomings, Jaroona Software implements a new approach based on blockchain role separation.

# Jaroona Blockchain Security Solution

Jaroona Software will address the issue of blockchain network security, without the need to sacrifice decentralization or scalability.

This takes shapes in a new type of node—the Jaroona Security Node (JSN), which uses machine learning algorithms, to allow for quick detection and response to security threats, including hackers, malicious actors, spammers and malicious software.

The solution is designed to enable commercialization and mass adoption of decentralized applications ("dApps") across various blockchain networks.

Individuals will be able to run their JSN on any available hardware device, earning fees for the transactions they secure on a specific blockchain. Each blockchain community will be able to establish its own fee structure and set the balance between security and performance.

The Security Nodes will be connected in a meta-network, allowing Jaroona Software to adapt to new threats automatically across different blockchain networks. The software itself is designed to work in conjunction with major existing and emerging blockchains, as well as any hardware, making the benefits of security and scalability available to various networks and their participants.

**With this approach, JSN has four major distinctive qualities:**

- **IS "AGNOSTIC" REGARDING NETWORK;**
- **CAN RESPOND EFFICIENTLY TO CURRENT AND EMERGING THREAT MODELS;**
- **CAN DEAL WITH BUGS AND VULNERABILITIES IN THE SMART CONTRACTS BY AUTOMATED DISCOVERY AND DEBUGGING PERFORMED AT SCALE;**
- **ENABLES SELF-IMPROVING SECURITY MEASURES, DECOUPLED FROM THE NATIVE NETWORK CONSENSUS ALGORITHM.**

The JSN will thus improve security in any consensus scenario, whilst freeing the blockchain to improve the transaction processing speed.

# Main concepts and features

To enable 'secure blockchain scalability', the combination of self-improving security and horizontal scaling—**Jaroona Software introduces two new revolutionary blockchain concepts:**

### JAROONA SECURITY NODE (JSN)

Specialized node within network, that utilizes artificial intelligence algorithms and neural networks for self-improving security features.

It self-learns from previously detected security threats and breaches, to identify good and malicious behaviour/code patterns, alerting traditional blockchain nodes when the probability of a breach is above the consensus threshold. Security Nodes act in conjunction with blockchain consensus algorithms to enable 'security-at-scale.'

### ADJUSTABLE ALLOCATION OF HARDWARE RESOURCES

Will address blockchain security at scale and attract an ever-growing number of Security Nodes. Acting as a Security Node will be affordable for all types of hardware on which Security Nodes could be deployed. We will introduce algorithms that adapt Security Nodes software based on deep learning networks against a given resource budget such as latency or energy consumption. The goal is to allow various devices, such as mobile phones, IoT devices and desktop computers to contribute and earn digital assets by performing the Security Node function.

The JSN addresses the major security bottlenecks that exist at the core of blockchain technology—a large surface for hacker attacks[15] and lack of real time protection.

# Jaroona Security Node

The introduction of the Security Node is a revolutionary concept in the blockchain environment. Since the introduction of Bitcoin (2008[16]), Ethereum (2014[17]), Corda (2016[18]) and other blockchains, a blockchain node has been defined as a computer connected to the blockchain network that performs the task of validating and relaying transactions using a copy of the blockchain, competing to win cryptocurrency by solving computational puzzles. Solving the computational puzzles enables reaching a consensus required to confirm a block of transactions, such that no transaction conflicts with any other.

Securing the blockchain ledger with cryptographic algorithms consumes a lot of energy. As economically and mathematically efficient as they are, the existing blockchains still have many security issues.

With eclipse attacks and vulnerabilities in smart contracts, blockchain code, wallets and third-party applications connected to blockchains, both the number of attacks and the financial consequences are growing rapidly. For example, at the time of this writing, there are about 25,000 nodes in the Ethereum network[19]. Assuming growth continues at current rates, the number of nodes will reach 100,000 within 3 years. The number (and scale) of hacker attacks will also likely increase accordingly.

As previously explained, Jaroona Software differentiates between two types of nodes: a traditional blockchain node and a Security Node. The goal of a Security Node is to self-learn from security threats and breaches, alerting other nodes when the probability of a breach is above the consented threshold.

A Security Node analyzes each incoming contract, transaction, network traffic and underlying blockchain code to identify 'malicious' and 'good' behaviour patterns, alerting traditional nodes, using AI technology deployed as part of the node software.

The AI technology is a black box for the node itself and the underlying algorithms are based on supervised and unsupervised deep learning networks.

**Security Nodes have two primary functions:**

- **TO PROTECT A BLOCKCHAIN NETWORK IN REAL TIME;**
- **TO HELP BLOCKCHAIN DEVELOPERS FIND VULNERABILITIES BEFORE THEY ARE DEPLOYED INTO A BLOCKCHAIN.**

Security Nodes performs an 8-class classification in two stages. Firstly, the model performs a coarse classification, which preliminarily classifies the data into classes of good flows or malicious flows. Secondly, the model executes a fine-grained classification on the malicious flows, which classifies them into 7 different categories according to the attack behaviour (Bot, Exploit, Trojan, Malspam, Ransomware, Eclipse, Unclassified). Jaroona saves all behaviour patterns, including unclassified pattenrs, for further self-learning.

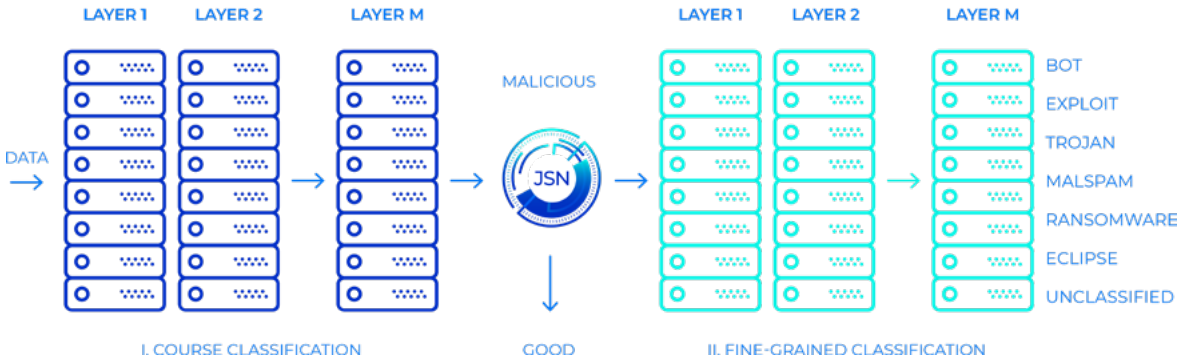



Figure 1: *The Jaroona 8-class Classification Architecture*

## BASED ON DEEP LEARNING NETWORKS

In the first phase of development, Security Node alerts will be based on coarse classification and fine-grained classification will be performed after alerts are sent out. In later stages, fine-grained classification will be included in alerts and will be linked closely to Security Node compensation (in digital assets—e.g. JSN tokens).

Compensation will be correlated with the financial benefit generated from the alert. For example, if a Security Node detects an attack that it classifies as Ransomware and that puts $250 million at risk, that Security Node will receive higher compensation than one that averts a lower-impact Denial of Service (DDOS) attack.

Security Nodes will also help the blockchain community scan the smart contract code base for vulnerabilities before deploying a contract to a blockchain network. The Security Node solution is highly efficient for this task, as they will learn from a wide network of

similar vulnerabilities from multiple smart contracts across multiple blockchains.

A Security Node deployed in the test environment of a blockchain company could act as a malicious player based on the learned threats and enable a blockchain developer to discover open vulnerabilities, allowing developers to fix vulnerabilities before deploying a smart contract.

# Self Improving Security

We equip Jaroona Security Nodes with powerful deep learning network algorithms, as deep learning is resilient to changes, mutations and input variables, making it incredibly efficient at catching cybersecurity threats and mutations. This is critical for any effective security solution because the number of cybersecurity mutations constitutes for 80% of all malware.

**The choice of deep learning networks is based on multiple factors:**

- Large varieties of data types, including network events, smart contracts, transaction events inside smart contracts, base code underlying the blockchain, and third-party protocols requires a robust machine learning approach to act at scale and with the speed required to predict a security threat;
- The requirement for near linear scalability of processing time vs. volume of data, as the amount of data grows, processing time must not degrade;
- The availability of processing power (GPU) to pre-train the security models. Deep learning technology requires the same GPU-powered computers that are used for mining, therefore there is an easy adoption path.
- The availability of ARM-based chips designed to provide machine learning capabilities on a mobile processor, powering AI systems and software on devices locally (without an internet connection or with low bandwidth). This is incredibly important, as it allows Security Nodes to be deployed on a large variety of mobile devices and enable exponential growth and adoption.

# Training the deep learning network

In Phase 1 of development, Security Nodes will be able to detect malware from network traffic and smart contracts.

## SMART CONTRACT ANALYSIS

By analyzing binary and source code of the smart contract, they will be able to determine whether the smart contract itself is malware, identifying vulnerabilities or bugs in the smart contract code.

Security Nodes are then trained using different blockchain binary executables, to make the Security Node deployable and effective on all major blockchain networks.

Deep learning networks have been successfully used to learn features from raw inputs for image, signal, and text use cases. It has recently been demonstrated that deep learning networks can learn to identify malware when trained on just 300 bytes from the PE-header of each file[20].

Based on these discoveries, our Security Node software extends this approach by training networks on other executable files written in blockchain programming language—starting with Ethereum (Solidity).

We initiate the training with publicly available data on known malware. For further fine-tuning, we will rely on data provided by anti-virus industry partners, where both the benign and malicious programs are representative of files seen on real machines.

With a deep learning model called a memory network, the Security Node software is trained to detect bugs resulting in buffer overflows, format string attacks, general memory corruption and other vulnerabilities in source code.

## NETWORK TRAFFIC ANALYSIS

Security Nodes also analyze network traffic to predict potential attacks. To train the underlying deep learning networks, we use several publicly available datasets, with each record containing features that are labeled as either normal or as a specific type of attack. This includes basic features derived directly from a TCP/IP connection, such as

traffic features accumulated in a window interval (either a time window or a number of connections) and content features. The training data contains traffic classes that include classes of attack and one normal class. All these attacks are grouped into categories based on purpose, such as Exploit (DoS, Probing) and others.

# Adapting to hardware resource budgets

To enable security-at-scale using Security Nodes, we optimize the software for different types of underlying hardware with limited resources to encourage the number of Security Nodes to grow exponentially. To achieve this, Jaroona Software utilizes the latest hardware resource optimization algorithms.

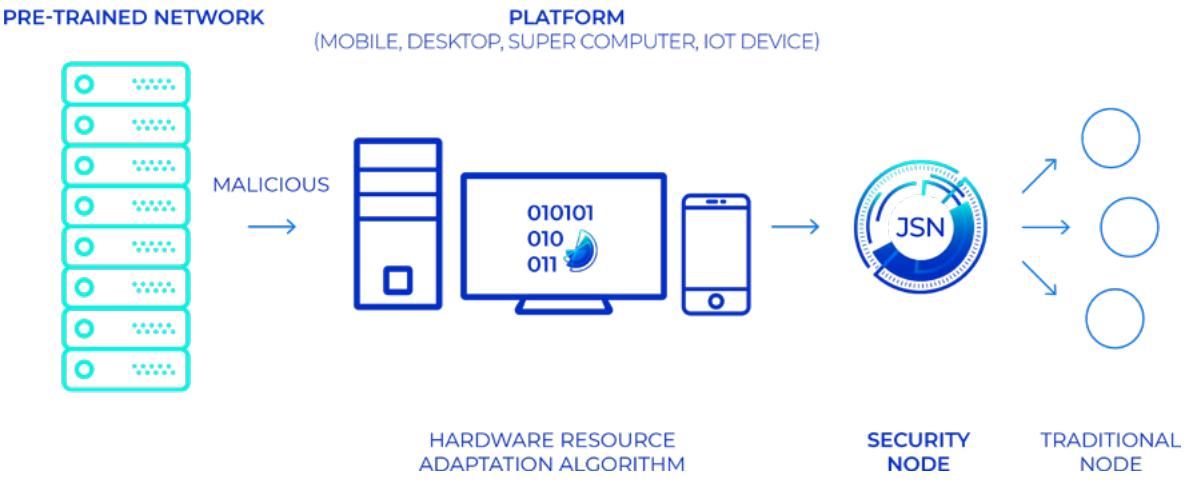



Figure 2: *Pre-trained Network is Optimized for Security Node Hardware, Sending 'Malicious' alerts to Traditional Nodes*

Such optimization results in expansion of different devices, that can be converted into Security Nodes, beyond powerful mining hardware. This will include mobile CPUs, mobile GPUs, typical business or home CPUs and gaming desktops with GPU chips.

The economic incentive for converting a device into a Security Node is a reward in the form of a percentage of the transaction fees or tokens mined for every transaction in the Proof of Work blockchain, or a percentage of tokens paid for storage, voting, or other services in the Proof of Stake blockchain.

# Jaroona cybersecurity protocol

The network of JSNs accompanied with the network p2p protocol and blockchain integration framework forms the Jaroona cybersecurity protocol (hereafter "the JSN protocol").

The JSN protocol is a highly scalable off-chain layer 2 solution for securing a public or private blockchains in real time against malicious attacks.

The JSN protocol is effectively a security layer running on top of a public or private blockchain. The JSN is blockchain and programming language agnostic, making it scalable, with minimum dependency on expertise in the particular blockchain programming language being used on the networks it secures.

**The goal of the JSN protocol is to perform:**

- **Real time threat risk analysis** on a specific blockchain and for a specific client on a specific blockchain;
- **Real time incident response** and providing warning alerts to traditional nodes, miners, and block producers;
- **Real time attack prevention** on a specific blockchain (e.g. stopping or re-routing malicious network traffic, placing malware into virtual containers or quarantine);
- **Smart Contract Auditing** (initial offline + ongoing in real time after deployment on a blockchain);
- **Endpoint protection** for traditional blockchain nodes, Dapps ,computers connected to a specific blockchain and any users connected to a blockchain.

**How JSN work with blockchains**

The basic process works as follows:

- JSN nodes analyze each incoming contract, transaction, event, and all network traffic to identify "malicious" and "good" behavior patterns as alerts for traditional nodes

- JSN nodes alert traditional nodes on an attack in real time.

- Traditional nodes can act upon the attack and vulnerabilities alert based on their governance, constitution and, or implementation logic agreed between the specific blockchain and Jaroona.

- JSN offers to the affected blockchain a number of options upon warning alert:
  - JSN stops or re -routes network traffic from malicious nodes;
  - JSN places malware into a virtual container (quarantine);
  - JSN records the attacker's details on the blockchain's in question;
  - JSN provides a blockchain (miners) in question with the detailed report on attacker's details and attack strategy;
  - JSN doesn't perform any further action after sending "malicious" behavior alert to miners. In this scenario miners shall act in accordance with the Blockchain's constitution or governance model.

In some blockchain implementations block producers are empowered to block or freeze smart contracts and accounts in the scenario that a vulnerability/exploit is found. Block producers can also decide which transactions are included in blocks.

Based on this function, JSN nodes can be used in an attack /vulnerability finding capacity in real time and as real time alerts to block producers than those can effectively decide on further steps.

Some blockchains enable a peer-to-peer terms of service agreement or a binding contract among those users who sign it, referred to as a "constitution".

Under such a constitution, JSN nodes can be contracted directly by block producers or other peers to provide cyber cecurity as a service in real time.

In some implementations, for example Polkadot, the creators designed a security frontman, such as Fisherman in Polkadot. The goal of this security guard is to identify other nodes' misbehavior. JSN can enhance the Fisherman capabilities with the cybersecurity knowledge base built on a multitude of different blockchains.

**Special use case to protect blockchain at scale in cooperation with multichain solutions**

As scalability, interoperability and security are seen as pivotal to the future of blockchain, multichain solutions like Polkadot, Block Collider and others are on the cutting edge.
We designed a special use case to improve security for the entire blockchain community.

We envision that JSN nodes will be running on a growing number of different blockchains protecting them in real time. At the time of writing, there are estimated 699 public blockchains and a substantial number (not publicly shared) of private blockchains.

If a specific blockchain is under a malicious attack, as identified by JSN, JSN networking protocol can efficiently re- route an affected transaction or a smart contract to a multichain solution. In its own turn a multichain solution can further re-route a transaction to a safe blockchain network using its internal capabilities.

By fostering co-operation between projects, we believe that enterprise will adapt blockchain solutions rapidly due to the reassurance that transactions will be concluded securely and in a timely manner, regardless of the growing number of malicious actors and the delays caused by their actions.

# Participation in the Jaroona cybersecurity protocol

There are three basic roles in the upkeep of the JSN protocol: Light Nodes, Full Nodes and Master Nodes.

## JSN Light Node

Light Nodes acts as a cybersecurity field agent with main goal of assessing the incoming network traffic and smart contracts in real time and alerting blockchain nodes and 3rd party customers about malicious behavior in real time.
Light Nodes use the pre-trained deep learning network results delivered by Full Nodes

to make an educated judgement about whether incoming traffic and /or smart contracts contain malicious behavior or not.

Light Nodes are able to run on low resource underlying hardware, such as a mobile phone or desktop, based on the hardware resource optimization algorithm discussed in the „Adapting to hardware resource budgets „ section of the present public white paper.

A network of Light Nodes achieve low-level consensus over a set of mutually agreed cybersecurity alert messages through an asynchronous Byzantine fault tolerant (BFT) algorithm. The algorithm will be inspired by the HoneyBadgerBFT. It provides an efficient and fault-tolerant consensus over an arbitrarily defective network infrastructure, given a set of mostly benign authorities or Light Nodes.

For permissioned networks, this alone would be sufficient, however the JSN cybersecurity protocol will be also deployable in a fully open and public situation without any particular organisation or trusted authority being required to maintain it.

As such we will introduce a final incentive authority – Master Node – with a means of monitoring a set of Light nodes and incentivising them to be honest. For Master Nodes we utilise PoS based selection criteria.

For a Light Node to act honestly means not to manufacture a false bug report or attack situation in order to receive an unearned reward.

**Communication with other Nodes:**

**JSN Light Node > Traditional Blockchain Nodes:**

After reaching a consensus with a quorum of 33%+1 Light Nodes on sending „malicious" alert, JSN Light Node may perform the following actions:

- Real time alert on malicious behavior
- Real time alert to a smart contract holder
- Block network traffic from a malicious node (this action is only performed if desired by a particular blockchain)
- Place the vulnerable contract / malware into quarantine /virtual container (this action is only performed if desired by a particular blockchain)

**JSN Light Node > JSN Full Node:**

- Sends frequent updates on all incoming network traffic for further Full Node training
- After reaching a consensus on malicious behavior, sends updates on traffic identified as malicious and contracts identified as vulnerable

**JSN Light Node > JSN Master Node:**

- Reports on online availability
- Sends alerts on malicious behavior

## JSN Full Node

Full Node acts as a cybersecurity lab with three main goals:

- Training the JSN underlying software on existing and emerging vulnerabilities and complex attacks strategies;
- Validating, and verifying the alerts which have achieved consensus amongst the Light Nodes;
- Working as a distributed cyber security server for those JSN Light Nodes operating as end point cyber security agents on the machines of traditional nodes.

Full nodes maintain a "full-node" copy of a particular blockchain as well as the latest version of the cybersecurity knowledge database; meaning that they retain all necessary information to be able to author a new version of attacks and vulnerabilities specification to a Light Node.

The underlying technology is a deep learning network. The choice of a deep learning network is discussed in the Self Improving Security Section of the present public white paper.

Full nodes do not use a consensus mechanism to receive training, as it is simply a neural network learning function.

However, Full Nodes do require a PoS based consensus on deciding whether malicious behavior consented by Light Nodes actually happened or not.

This is critical for a fair reward distribution function (for example, in the case of a sybil attack, an attacker can attack malicious Light nodes that are part of an attacker's network) and those nodes will alert the traditional nodes to receive an undeserved incentive.

JSN Full Nodes hardware requirements are similar to "full nodes" in present-day blockchain systems (GPU enabled infrastructure).

JSN Full Nodes must post only a small bond. This bond prevents sybil attacks from wasting other JSN Full Nodes' time and compute resources. It is immediately withdrawable after reaching consensus on whether malicious behavior reported by Light Node is confirmed.

**Communication with other Nodes:**

**JSN Full Node > JSN Light Node:**

- Sends frequent updates on cybersecurity knowledge base learned by deep learning networks in a format of the pre-trained model results

**JSN Full Node > JSN Master Node:**

- After reaching consensus on whether or not malicious behavior reported by Light Nodes actually happened, sends proof as a basis for rewards.
- Provides latest version of self-learned cybersecurity knowledge base

**JSN Master Node**

A JSN Master Node is the highest charge and mastermind behind JSN cybersecurity network. A JSN Master Node coordinates in emergencies, organizes the recovery of the JSN cybersecurity network from the attack, and acts as the main authority for reward distribution. The Master Node's role is contingent upon a sufficiently high bond being deposited, though we allow other bonded parties to nominate one or more Master Nodes to act for them and as such some portion of the Master Node's bond may not necessarily be owned by the Master Node itself but rather by JSN token stakeholders. A Master Node must run a full copy of a specific blockchain plus latest copy of the cybersecurity knowledge base delivered by JSN Full Nodes with high availability and bandwidth.

A Master Node monitors availability of Light and Full nodes, checks alerts and decides on recovery operations.

A Master Node is also a final authority on deciding whether a group of Light Nodes acted honestly and therefore, deserve a reward.

Master Nodes will be elected through a Delegated Proof-of-Stake (DPoS) scheme.

**Communication with other Nodes:**

**JSN Master Node >Traditional Blockchain:**

- Writes attackers' data on traditional blockchains (only basic detail about an attacker without attacker's strategy which will be available upon request by reputable 3rd parties)

**JSN Master Node > JSN Light Node:**

- Monitors Light Nodes availability
- Decides on an emergency plan and notifies Light Nodes on new peers available
- Distributes rewards

**JSN Master Node > JSN Full Node:**

- Monitors Full Nodes availability
- Decides on an emergency plan and notifies Full Nodes on new peers available
- Distributes rewards

Communication scheme is presented on Figure 3.

JSN Nodes will be communicating with each other via the custom JSN protocol. To ensure an efficient transport and peer communication, the JSN network protocol will include an extensible peer selection and discovery mechanism accompanied with peer availability monitoring and peer planning to ensure a necessary and sufficient number of JSN nodes of different types are connected at the right time. The JSN network protocol will take into consideration and extend already existing peer to peer protocols, such as libp2p and GNU's GNUnet.

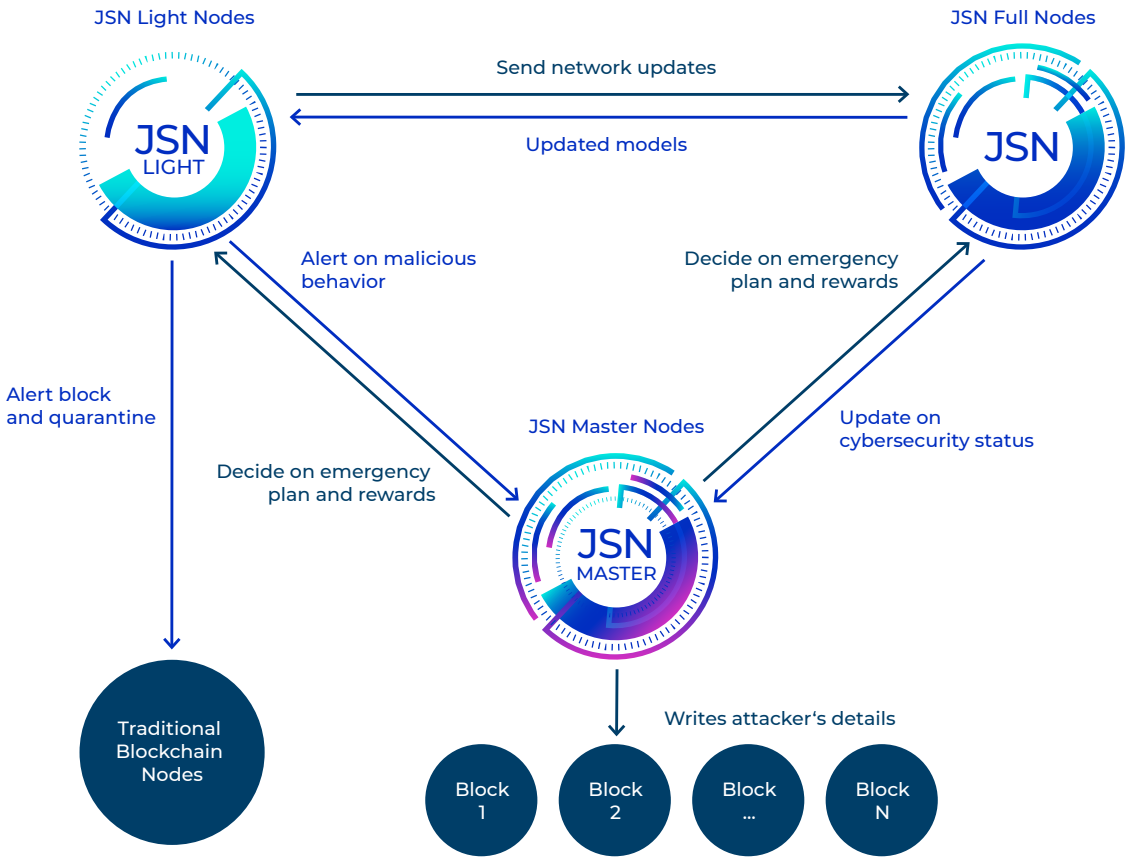



Figure 3: *Nodes communication scheme*

# Scalability and latency considerations

The JSN cybersecurity protocol is designed to work with existing and emerging blockchains.

Taking into consideration existing blockchains' scalability and throughput limitations, the JSN cybersecurity protocol is designed to work as a low bandwith / low latency off-chain layer 2 solution on top of existing public blockchains.

Due to the off-chain layer 2 nature of the JSN protocol, it is completely decoupled from any scalability issues experienced currently by existing blockchains.

The implementation of an attack alerting system from JSN Light Nodes to a particular blockchain will utilize blockchain native networking protocols, such as devp2p protocol on Ethereum or JSON based p2p protocol on Lisk.

For example, Ethereum devp2p protocol allows for many subprotocols to be multiplexed over a single peer connection that is more than sufficient for the JSN alert system to work efficiently.

The estimated latency and bandwidth requirements are minimal, as an attack alerts messages typically is about 256KB.

JSN cybersecurity protocol is designed not to affect the existing blockchain throughput, as JSN Master Nodes write the minimal nominal attack data on the main chain several times per week. Master Node smart contracts with attackers data do not contain any computation steps and therefore, doesn't require substantial mining computational resources.

# JAROONA Security Node Use Cases

1 USE CASE 1: **POS NETWORK PROTECTED BY A NETWORK OF JSNS.**

JSN follow sharding concepts to protect PoS/shards in real time at scale. Figure 4 below depicts networks of JSNs protecting different subgroups.

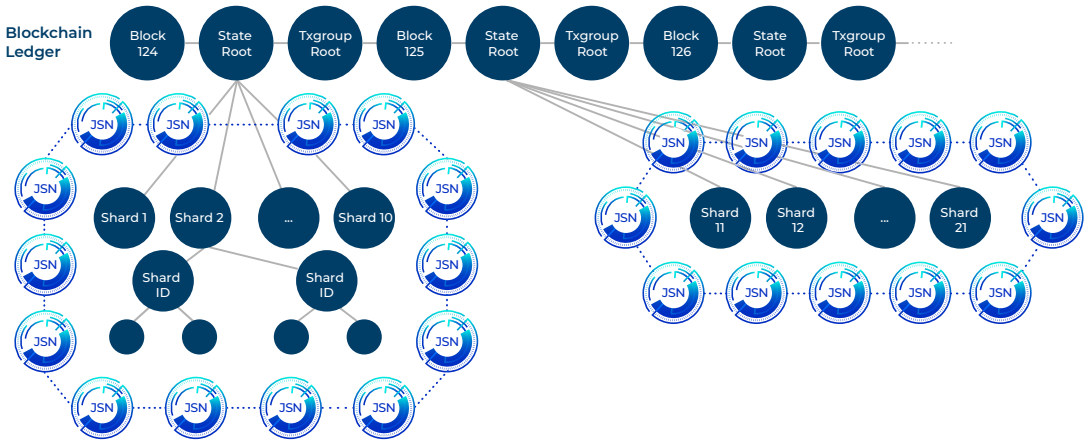



Figure 4. *Networks of JSNs Protecting Shards Subgroups*

2 USE CASE 2: **JSNS PROTECTING BLOCKCHAIN NETWORKS AT SCALE BY IDENTIFYING MALICIOUS NODES IN DIFFERENT ATTACK SCENARIOS.**

Figures 5a and 5b below depict JSN identifying malicious nodes.

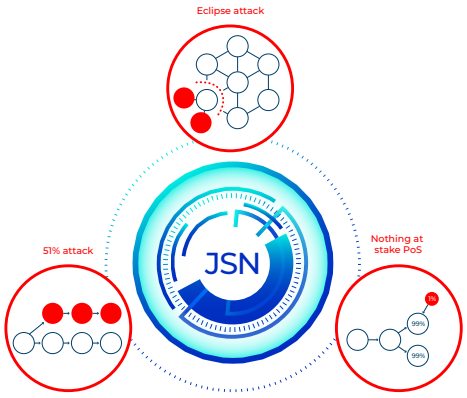



Figure 5a. *JSN identifies malicious nodes*

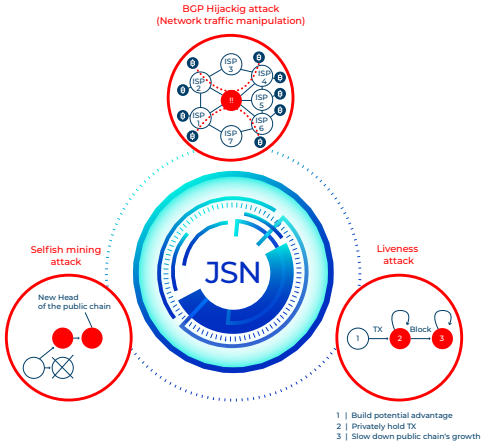



Figure 5b. *JSN identifies malicious nodes*

3

USE CASE 3: **JSNS SHARE CYBERSECURITY KNOWLEDGE BASE**

Meta-network of Jaroona Security Nodes across blockchains will learn on attacks and vulnerabilities, sharing this data with blockchain partners. Figure 6 (right) depicts Security Nodes leaning across different networks, sharing knowledge accordingly.

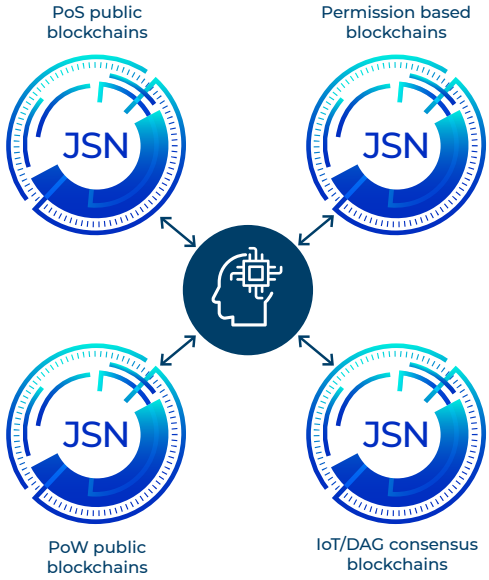



Figure 6. *JSNs learn across networks and share knowledge*

4

USE CASE 4: **JSN CERTIFIES SMART CONTRACTS AUTOMATICALLY**

Smart contract certification by JSN means robust protection against vulnerabilities that have been learned by JSN nodes across multiple blockchains, different programming languages and multiple industries. Figure 7 below depicts JSN used to certify smart contracts in multiple industries.


Figure 7. *JSNs Certify Smart Contracts*

# Jaroona Token System

The JSN token is the economic engine that fuels the implementation and expansion of the JSN Software across multiple blockchain networks, by creating immense value for the users, who convert their devices into Security Nodes.

Users are rewarded with JSN tokens, which represent payment for their service of aiding and securing a particular blockchain network. In addition, holders of the JSN tokens gain access to the JSN cybersecurity knowledge base, allowing them to run their smart contracts against potential vulnerabilities.

The JSN tokens (JSN) are cryptographic tokens, conforming to the ERC-20 standard, distributed by an ERC-20 compliant smart contract on the Ethereum blockchain network.

**JAROONA TOKEN (JSN) UTILITIES**

- **Incentivize users to run Jaroona Security Node as a service on the blockchains and/or for individual clients that would be looking to secure their operations on any specific blockchain;**
- **Provide a means of value and exchange (closed loop currency).**

The first function of the JSN tokens will be to entitle holders to run a Jaroona Security Node, as a service on a specific blockchain or for individuals looking to secure their operations and smart contracts.

The second function of JSN tokens will be to provide a value mechanism within the Jaroona Security Node as rewards for adding valuable development and scientific research—with higher value contributions being rewarded with a higher number of JSNs.

# Roadmap

**Spring 2019**

Security Node beta version release

**Spring / Summer 2019**

Security Node hardware adaptation mechanism

**Summer 2019**

Minimal viable testing environment

**Fall 2019**

Security Node benchmarking on a variety of security threats

**Fall 2019**

Minimal viable network testing environment

**Fall 2019**

Integration with blockchain partners

**Fall 2019 / Winter 2020**

Testing & Audits

# Team

The Jaroona Software project is founded by four core team members with more than 80 years of combined experience in creating and scaling IT architectures and software development.

Previous successes of the team members including founding The Smart Engine Group *(www.smartengine.solutions)* in 2011.

Smart Engine now operates globally, providing predictive analytics and AI technologies for financial services, retailing and advertising. Clients include 25 banks, 500 retailers, and 4 million consumers, growing at over 90% annual rate with 8-digit revenues achieved in 2017.

***20 years experience in creating scaling secure architectures***
***Founded two companies in the past decade***
***Cybersecurity, AI and Deep learning Network Experts***

**Christian Bacher, CEO**

**Former IBM, Raiffeisenbank,**
**CEO Smart Engine Group,**
**MBA Henley Business School**
**30 years in Software Dev and complex IT services**

**Anna Bacher, CTO**

**Former IBM (Security solutions), Accenture**
**CTO Smart Engine Group**
**20 years in Software Dev**
**7 years in AI + Deep Learning Algorithms**

**Dr. Erich Gstrein,CSO**

**PhD in Computer Science**
**30 years in AI algorithms**
**Winner of German INNOVATIONPRICE-IT 2011,**
**Innovation Price 2009 - Jury Award from Austrian Ministry of Economy**

**Heimo Tomann, CFO**

**CFO Smart Engine Group –**
**30 years in complex finances for software companies and other industries**

**Zlata Baldekova, CMO**
**Co-Founder**

**5+ years experience in strategy and management consulting Deloitte and KPMG**

**Adrian Procopenco,**
**Lead Solidity,**
**JAVA Developer**

**2 years Solidity,**
**9 years Java experience**

**Ivan Chilienco**
**Lead Full stack Developer**

**10 years experience in Java, C++, JavaScript, JSP, Maven, CSS /HTML5, react.js**

# Company Information

**CORPORATE NAME:**

JAROONA CHAIN OÜ

**REGISTERED SEAT:**

Harju maakond, T allinn, Kesklinna linnaosa,

Ahtri tn 12-Ahtri tn 12-200, 10151, Estonia

**REGISTRATION NUMBER:**

14501208

**REPRESENTED BY:**

Christian Bacher, director

# Notes

- [1] Mainelli, Michael and Smith, Mike, Sharing Ledgers for Sharing Economies: An Exploration of Mutual Distributed Ledgers (Aka Blockchain Technology) (November 7, 2015). Journal of Financial Perspectives, Vol. 3, No. 3, 2015. Available at SSRN: https://ssrn.com/abstract=3083963
- [2] Olleros, F. Xavier, and Majlinda Zhegu. Research Handbook on Digital Transformations. Cheltenham, Gloucestershire, UK: Edward Elgar Publishing, 2016.
- [3] "What Is a DApp? Decentralized Application on the Blockchain." BlockchainHub. Accessed June 01, 2018. https://blockchainhub.net/decentralized-applications-dapps/.
- [4] "Explore Decentralized Applications (projects Built on Ethereum)." State of the dApps - 1450 Projects Built on Ethereum. Accessed June 01, 2018. https://www.stateofthedapps.com/
- [5] McCann, Chris. "State of the DApps: 5 Observations From Usage Data (April 2018)." Medium. April 11, 2018. Accessed June 01, 2018. https://medium.com/@mccannatron/state-of-the-dapps-5-observations-from-usage-data-april-2018-a3e9da01bc22.
- [6] "Leading Global Social Networks 2018 | Statistic." Statista. Accessed June 01, 2018. https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/
- [7] For more details on proposed security solutions refer to subsequent sections in this document.
- [8] One example of closed vulnarability - https://arstechnica.com/information-technology/2018/03/ethereum-fixes-serious-eclipse-flaw-that-could-be-exploited-by-any-kid/
- [9] "Slasher: A Punitive Proof-of-Stake Algorithm." Ethereum Blog. July 22, 2014. Accessed June 09, 2018. https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/.
- [10] Casper the Friendly Finality Gadget, Vitalik Buterin and Virgil Griffith Ethereum Foundation, 17 of October, 2017
- [11] EOS.IO Technical White Paper, June 26,2017 – paragraph Consensus Algorithm (DPOS)

- [12] Fisch, and Jacquelyne. "Reponse to Vitalik's Written Remarks - Steemit." - Steemit. Accessed June 09, 2018. https://steemit.com/eos/@dan/reponse-to-vitalik-s-written-remarks.
- [13] Polkadot: Vision for a heterogeneous multi-chain framework, Dr. Gavin Wood , Founder, Ethereum & Parity Gavin@Parity.io
- [14] Mueller, Bernhard. "Detecting Integer Overflows in Ethereum Smart Contracts." ConsenSys Media. April 30, 2018. Accessed June 09, 2018. https://media.consensys.net/detecting-batchoverflow-and-similar-flaws-in-ethereum-smart-contracts-93cf5a5aaac8.
- [15] Wikipedia - The attack surface of a software environment is the sum of the different points (the "attack vectors") where an unauthorized user (the "attacker") can try to enter data to or extract data from an environment.
- [16] Satoshi Nakamoto, „Bitcon: A Peer-to-Peer Electronic Cash System" 2008
- [17] Gavin Woods, „Ethereum: A Secure Decentralized Generalised Transaction Ledger", 2014
- [18] Richard Brown, James Carlyle, Ian Grigg, Mike Hearn, „Corda: an Introduction" 2016
- [19] Number of Ethereum nodes is taken from https://ethernodes.org/ - Ethereum's node explorer. The estimation is based on an active crawling process that recursively connects to a node and asks for its known peers.
- [20] Raff, E.; Sylvester, J.; and Nicholas, C. 2017. Learning the PE Header, Malware Detection with Minimal Domain Knowledge. arXiv preprint arXiv:1709.01471.


# References


- "Persistence (computer Science)." Wikipedia. May 28, 2018. Accessed June 09, 2018. https://en.wikipedia.org/wiki/Persistence (computer_science).
- In probability theory, an elementary event (also called an atomic event or simple event) is an event, which contains only a single outcome in the sample space.
- Massachusets Institute of Technology, Google Inc., Tien-Ju Yang, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Vivienne Sze, and Hartwig Adam, submitted on 9 Apr 2018 arXiv preprint arXiv: 1707.01083 (2017), Zhang, X., Zhou, X., Lin, M., Sun,J.
- "Search UNB." University of New Brunswick Est.1785. Accessed June 09, 2018. http://www.unb.ca/cic/datasets/index.html.
- "CSIC 2010 HTTP Dataset in CSV Format (for Weka Analysis)." Peter Scully PhD. June 02, 2018. Accessed June 09, 2018. https://pmdscully.wordpress.com/research/csic-2010-http-dataset-in-csv-format-for-weka-analysis/.
- "Analyzing Web Traffic ECML/PKDD 2007 Discovery Challenge September 17-21, 2007, Warsaw, Poland." Attack Challenge - ECML/PKDD Workshop. Accessed June 09, 2018. http://www.lirmm.fr/pkdd2007-challenge/#dataset.
- "Search UNB." University of New Brunswick Est.1785. Accessed June 09, 2018. http://www.unb.ca/cic/datasets/index.html.
- An Ngoc Lam, Anh Tuan Nguyen, Hoan Anh Nguyen, and Tien N. Nguyen. Combining deep learning with information retrieval to localize buggy files for bug reports. In Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015, pages 476–481. IEEE, 2016.
- Sahil Bhatia and Rishabh Singh. Automated correction for syntax errors in programming assignments using recurrent neural networks. CoRR, abs/1603.06129, 2016.
- Martin White, Christopher Vendome, Mario Linares-Vásquez, and Denys Poshyvanyk. Toward deep learning software repositories. In Mining Software Repositories (MSR), 2015 IEEE/ACM 12th Working Conference on, pages 334–345. IEEE, 2015.
- "Public PCAP Files for Download." Netresec. Accessed June 09, 2018. http://www.netresec.com/?page=PcapFiles.
- "Cyber Research Center - DataSets." Department of History - Vietnam War. Accessed June 09, 2018. https://www.westpoint.edu/crc/SitePages/DataSets.aspx.
- "Search UNB." University of New Brunswick Est.1785. Accessed June 09, 2018. http://


www.unb.ca/cic/datasets/index.html.

- Raff, E.; Zak, R.; Cox, R.; Sylvester, J.; Yacci, P.; Ward, R.; Tracy, A.; McLean, M.; and Nicholas, C. 2016. An investigation of byte n-gram features for malware classification. Journal of Computer Virology and Hacking Techniques.
- Kolter, J. Z., and Maloof, M. A. 2006. Learning to Detect and Classify Malicious Executables in the Wild. Journal of Machine Learning Research 7:2721–2744.
- Raff, E.; Sylvester, J.; and Nicholas, C. 2017. Learning the PE Header, Malware Detection with Minimal Domain Knowledge. arXiv preprint arXiv:1709.01471.
- Bacina, Michael. "$1B Lost: The 5 Biggest Cryptocurrency Fails of 2017." Medium. January 06, 2018. Accessed June 09, 2018. https://medium.com/@MikeBacina/1b-lost-the-5-biggest-cryptocurrency-fails-of-2017-9862131e2bf7.
- "Jorge Rodriguez on LinkedIn: "EOS Blockchain Smart Contract Architecture..." LinkedIn. Accessed June 09, 2018. https://www.linkedin.com/feed/update/urn:li:activity:6396830444918312960.
- "Network Number 1Last Updated a Few Seconds Ago." The Ethereum Node Explorer. Accessed June 09, 2018. https://ethernodes.org/.
- "Crypto Aware: $670 Million in Crypto Hacks and Scams in 2018." NewsBTC. April 04, 2018. Accessed June 01, 2018. https://www.newsbtc.com/2018/04/04/670-million-worth-of-hacks-and-scams-in-2018-says-crypto-aware/.
- "Vbuterin." Reddit. Accessed June 09, 2018. https://www.reddit.com/user/vbuterin/gilded/.
- Goodin, Dan. "Ethereum Fixes Serious "eclipse" Flaw That Could Be Exploited by Any Kid." Ars Technica. March 03, 2018. Accessed June 09, 2018. https://arstechnica.com/information-technology/2018/03/ethereum-fixes-serious-eclipse-flaw-that-could-be-exploited-by-any-kid/."Slasher: A Punitive Proof-of-Stake Algorithm." Ethereum Blog. July 22, 2014. Accessed June 09, 2018. https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/.
- Casper the Friendly Finality Gadget, Vitalik Buterin and Virgil Griffith Ethereum Foundation, 17 of October, 2017
- EOS.IO Technical White Paper, June 26,2017 – paragraph Consensus Algorithm (DPOS)
- Sheffield, Cuy. "Cuy Sheffield (@csheffield3)." Twitter. February 21, 2018. Accessed June 09, 2018. https://twitter.com/csheffield3.
- Mueller, Bernhard. "Detecting Integer Overflows in Ethereum Smart Contracts." ConsenSys Media. April 30, 2018. Accessed June 09, 2018. https://media.consensys.net/detecting-batchoverflow-and-similar-flaws-in-ethereum-smart-contracts-93cf5a5aaac8.


JAROONA