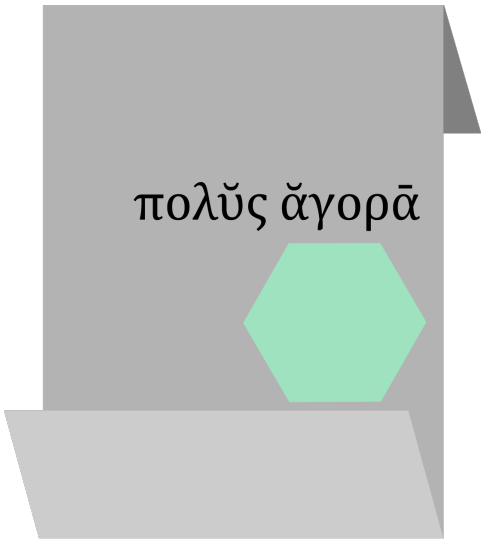
Polygora
Distributed SSL/TLS via Blockchain

2017



The internet is fundamentally broken.

Every secure connection, from banking to Facebook, is at risk... presented as a feature, instead of a flaw.

HTTPS is used to keep your data safe, but the protocol below it, *TLS*, is not. Third party intermediators control the certificates that prove websites are who they say they are, third parties who inherently cannot be trusted. This is the first rule of security - *never trust third parties*.

Polygora is a secure, decentralised, open-source alternative to the traditional certificate authorities. It's a blockchain like that of *Bitcoin* or *Ripple*, except it works **with** them, not against them. Polygora is used to help secure banking transactions and bitcoin payment requests in a cryptographically secure manner. Polygora too can be used for payments and trade, but it's main goal, is ensuring the security of the internet and all it's encrypted traffic, safe from fraudsters and prying governmental eyes.

The distributed network means anybody can sync a node and get involved mining, and trade, distribute, and use them for generating secure TLS certificates without the involvement of shady third parties and malicious actors.

Contents

Abstract

Today, millions of websites across the internet use HTTPS (*HTTP Secure*) to encrypt and secure communications between their servers and individual clients. The HTTPS protocol uses cryptographically verified certificates to encrypt communications - without it, there would be no online banking, logging in to websites would be insecure, passwords would be stolen constantly. The protocol makes a majority of malicious interference infeasible[1]. The HTTPS protocol itself encapsulates another protocol, TLS (*Transport Later Security*), which exists to verify the integrity, security and privacy of client-server communication.

However, the existing methods of certificate issuance and distribution is fundamentally flawed. TLS requires a trusted third party to issue a certificate, a very unfavourable situation. The industry of issuance is controlled by an oligopoly of untrustworthy yet massively influential Certificate Authorities, of which have been demonstrated to be malicious actors, time and time again.

The solution about to be presented to you is an application of the blockchain into this sphere - a distributed, trust-less, open solution, that takes control from the oligarchs and puts the users in control. Security and transparency, bound together by mathematical proof and practically proven by demonstration (see *Bitcoin*, or *Ethereum*).

[1](when properly configured, as with any software)

Part I
The problem

1 The TLS protocol

The TLS protocol, as with it's predecessor SSL (*Secure Sockets Layer*), is a cryptographic protocol designed to secure the privacy and integrity of communications between computers on a network. For example, in the case of HTTPS, the protocol exists to ensure a client's web browser is sending and receiving the data the server intends to receive and serve, and nobody is interfering with or monitoring those communications. Without TLS, a malicious actor can listen-in on communications to retrieve usernames, passwords, and all other potentially private data, utilising a type of *Man-in-the-middle* attack.

TLS' fundamental flaw is that of requiring a trusted third party to issue signed digital certificates (X.509 certificates). In a perfect world, this would be a non-issue - unfortunately, we do not live in such a world. The issuer of these certificates in the case of HTTPS are called *Certificate Authorities*, or CAs. This third party, if it chose, could act maliciously, and compromise the integrity or privacy of a website's certificate. If an attacker with access to the CA's computer network could create their own certificate under another website's name, they would appear to be the intended site, and trust is assured by the success of the TLS interaction, and yet this would not be the case.

2 The failures of the Certificate Authorities

There are multiple examples of CAs violating user's trust and being infiltrated by malicious actors[2][3][4]. There is also the case of malevolent Government agencies interfering with the process, which too, has been documented[5]. This state of affairs is not reliable, and is not suitable for client-server security, an integral part of the backbone of the modern internet.

[2]https://www.bankinfosecurity.com/using-symantecs-tlsssl-certs-start-replacing-them-now-a-10283

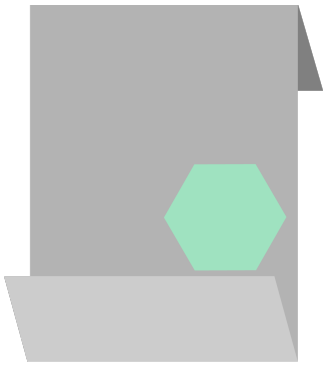
[3]http://www.flyertalk.com/forum/travel-technology/1643089-gogo-issuing-fake-ssl-tls-certs-facilitate-traffic-interception.html

[4]https://www.computerworld.com/article/2510951/cybercrime-hacking/hackers-spied-on-300-000-iranians-using-fake-google-certificate.html

[5]https://www.schneier.com/blog/archives/2013/09/new_nsa_leak_sh.html

Part II
The solution: Polygora

3 What is Polygora



The Polygora logo, a carbon steel certificate - immutable and unforgeable.

From the Greek:
πολύς • (polús)

- large, great
- a lot of, much of
- mighty

ἄγορᾱ́ • (agorā)

- assembly, especially an assembly of the people (as opposed to a council, βουλή (boulē))
- the place of assembly
- speech
- market

The Polygora (a portmanteau of both *poly* and agora, to mean ”many markets”) network is the first *distributed and decentralised* TLS certificate authority, designed to transparently and autonomously issue X.509 certificates for secure client-server communication. The network's fuel is the cryptocurrency token *Aguris (AGS)*, distributed to users initially as tokens on the Ethereum network during the presales, but which will be transferable into AGS on the Polygora blockchain once released.

The Polygora blockchain will at first generate blocks through a *proof-of-work* scheme (similar to that of *Monero* or *Zcash*). The current plan is to switch to a *proof-of-stake* work scheme once the Ethereum project releases details on the future ”Casper” protocol. A *proof-of-stake* protocol allows users to generate wealth within the network, by ”staking” AGS tokens they already own.

Breaking away from the centralised cartels, the Polygora project aims to provide *low-cost, reliable, and trustworthy* digital certificates without any chance of third-party or malicious interference. This is achieved through the use of a blockchain on which transactions are not made to other clients, but paid into the network as incentive for continued staking, mining, and redistribution.

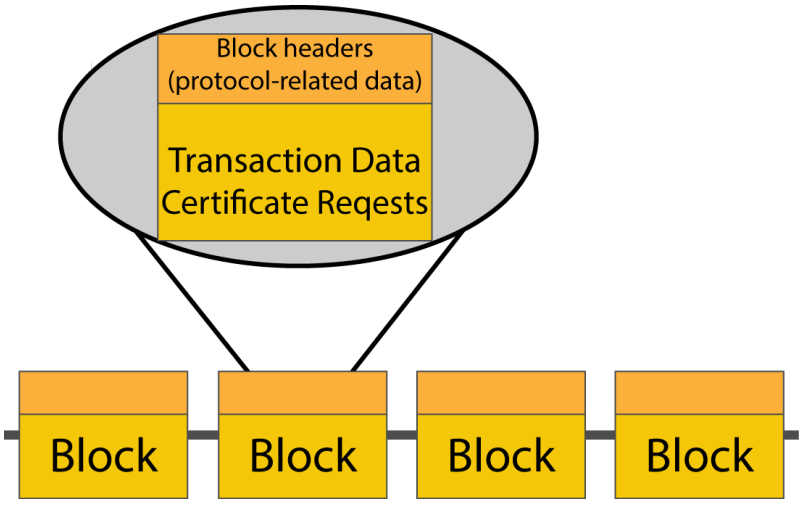
In a similar fashion to *Monero*, Polygora will utilise separate wallet and daemon programs to help users manage their wallets and nodes.

4 The blockchain

4.1 Blocks

At the core of the Polygora network is a revolutionary technology called *the blockchain*, introduced by *Satoshi Nakamoto*'s Bitcoin. The blockchain is a series of *blocks*, which contain a header stating some technical information about the block, some kind of data (in our case, transaction and certificate data), and a cryptographic hash of the previous block. Every node in the network has some copy of the blockchain, with each copy being completely identical. The blockchain therefore contains a complete history of all *transactions* made using the Polygora network, stored in a publicly viewable and transparent manner. Every Aguris token has a history that can be traced through transactions back to the block it was mined from.

A new block is aimed to be produced every n time periods, in the case of Bitcoin 10 minutes, which Polygora will be adjusting to a time of 1 minute - this is to increase the amount of transactions the network can handle. A transaction is a transfer of value from one wallet (an address to which is ascribed some number of tokens) to another. Network participants verify the transaction algorithmically, and if the transaction is valid, it is written to a block. Miners "confirm" transactions by mining the block that contains said transaction - the rule-of-thumb is that a transaction is not truly confirmed until it is considered *six blocks deep*, i.e. six blocks have been mined since the transaction was made.



Because of the unique architecture of Polygora's blockchain-based network, transactions are both *transparent* (anyone can view all transactions), and *immutable*, that is, all transactions are final and cannot be altered post-sending. This allows for unmatched security and integrity compared to traditional systems: 100% impervious to fraud, 100% transparent, and 100% verifiable. Polygora is a network *by the people, for the people*.

4.2 The Aguris token

Traditional *Bitcoin*-style blockchains require a currency or token to operate, and Polygora is no different. In this case, the network's token is Aguris, which we hope to be known on exchanges as AGS. Whenever a new block is mined, the miner of the block receives an amount of AGS, which is paid into the wallet associated with their mining software. This can be considered as the *minting* of new tokens. Since the Polygora network is consuming tokens to generate certificates, the current solution being considered for block rewards is described as follows.

The network maintains a distributed wallet at say, address *0x000000000000* (an arbitrary number chosen for this example), which users requesting certificate issuance or renewal send a transaction to, with the required data. The funds in this wallet are used as block rewards. When a block is mined, some portion of the funds in this wallet are what are used to pay the miner of the block, which provides incentive for continued mining above that of pure transaction fees.

Tokens do not deteriorate in relative value and can not be moved to another wallet without the wallet owner's consent. This means a wallet that holds 500AGS will continue to hold 500AGS for as long as the wallet owner decides to keep them in reserve for. When the wallet owner decides to make a transaction they will have to pay a nominal transaction fee (as an incentive, to encourage miners to include the transaction in an upcoming block).

5 Applications on the blockchain (e.g. Smart Contracts)

With *Bitcoin*, the network has a programming language built into the protocol, which allows for a primitive implementation of contracts - this was expanded upon by projects such as *Ethereum*, which provide a blockchain-based virtual machine to interpret *smart contracts*, programmed in a special language and compiled into bytecode. With the implementation of a similar system into the Polygora network, features such as automated certificate renewal, or escrow-based domain transfer, can be implemented and executed on some sort of schedule as opposed to requiring user interaction with a live full node.

6 Centralisation vs Distribution

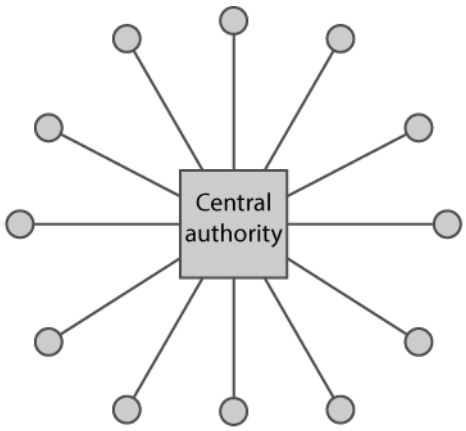
This section aims to detail the differences between centralised, and distributed architectures, and why the distributed architecture is superior, with respect to the context of certificate issuance.

6.1 The centralised model

Centralisation is defined by the Oxford dictionary as:

> *"The concentration of control of an activity or organization under a single authority.".*

In our context, an example would be a singular body that society has explicitly or implicitly given full authority to issue and revoke certificates, where only this organisation's certificates would be generally accepted. This is only marginally far from the current system, as certain certificate authorities issue a disproportionate amount of certs - for example, as of 2015 Comodo was the issuer for over 1/3rd of the top *ten million* TLS-enabled websites examined[6]. This model has shown time[7] and time[8] again to be an unreliable solution to the TLS problem.



6.2 The distributed model

Distribution is defined by the Oxford dictionary as:

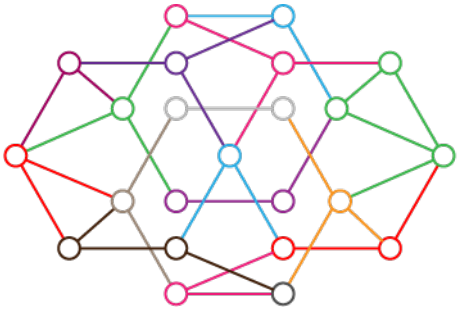
> *"The way in which something is shared out among a group or spread over an area.".*

[6] https://w3techs.com/technologies/overview/ssl_certificate/all

[7] https://www.bankinfosecurity.com/using-symantecs-tlsssl-certs-start-replacing-them-now-a-10283

[8] http://www.flyertalk.com/forum/travel-technology/1643089-gogo-issuing-fake-ssl-tls-certs-facilitate-traffic-interception.html

This definition does not help us gain much of an insight into Polygora's model of a distributed certificate authority. As we have seen, a blockchain network in Polygora's case is divided into nodes. The *"something"* of the definition are the blocks of the blockchain, which is distributed to individual nodes for verification. Each node individually stores a copy of the Polygora blockchain and shares that among it's peers, cryptographically verifying integrity of transactions. Some nodes may choose to *mine* new blocks, but no node is obliged to. With a distributed system, there is no central authority to the model, and the network can continue to run indefinitely, with little to no chance of some sort of failure. Nodes can leave and re-join the network at any time. The more distributed the network is, the more resistant it is to various form of attack. This model puts the power into *the people's* hands.



7 Certificate issuance protocol

The method through which Polygora certificates will be deployed is known as the ACME protocol[9], ACME standing for *Automatic Certificate Management Environment*. The protocol is the brainchild of the *Internet Security Research Group*[10], ISRG, a Californian public benefit corporation. The protocol allows for certificate-related interactions between CAs and clients to be automated and therefore technically secure from third party interference.

It does this through JSON-formatted messages transferred over a HTTPS connection, during which a user requests a certificate by generating an account with the CA (in this case the Polygora network), and provides proof they own the domain which their certificate request is for. The process is succinctly described in the draft IETF document as follows:

1. The client submits an order for a certificate to be issued
2. The client proves control over the domains requested the in the certificate
3. The server confirms the client has control and issues the requested certificate

[9] https://tools.ietf.org/html/draft-ietf-acme-acme-07
[10] *https://letsencrypt.org/isrg/*

Within the framework of the Polygora network, this can be done through the Polygora client software, which will process the requests into the network in a format suitable for block data. The current system for certificate issuance is up for debate - it will either be a fixed price, or depending on the generation process, a variable price set by the network, calculated by some algorithm. At launch, 90-day certificates will be definitely be supported, other validity periods are too, still up for debate. The design of the ACME protocol means wildcard domains will not be initially supported by Polygora, however this is potentially a feature to be included in a future release.

8 Research initiative

After the ICO, there will be a period where the Polygora team will research blockchain technology and the solutions it can provide, and how to best implement them for the network. This is due to the unique solution the network provides, and so the optimal solution needs to be determined through mathematical proof and programmatic experimentation. All research gathered over this period will be published in a research paper for review by the blockchain community at large, so as to share knowledge and perspectives within this emerging field. Some example topics shall be; VM implementation and capabilities versus that of expanded transaction scripting, network topology and the potential requirement of more important nodes ("master nodes"), blockchain storage optimisation, networking capabilities, and similar such topics. The reason for this, is implementing multiple blockchain solutions designed to interface with the outside world appears to be more complicated than a blockchain that only communicates internally. Malicious nodes provide a much larger threat when it is infeasible for the entire blockchain to do a single interaction. Two months is the maximum extent of this period, we believe that once full time development is commenced, it should be resolved much sooner.

9 Example network description

> **Important! The Polygora network is not aimed to replace TLS.**

Although the project has been described in places throughout this document, it appears to be appropriate to group all such information together into a single, more concise section, with an expansion on details. What follows is a description of the network, an explanation of the process of a certificate request, and a short description of the mining process.

The Polygora network is a distributed blockchain, consisting of individual nodes running the Polygora server software. Each node conforms to the same specifications, relaying transactions throughout the network on Polygora's chosen port, and, if mining, collecting transactions into blocks. The Polygora implementation will be similar to that of Satoshi Nakamoto's vision for

Bitcoin, i.e. immutable, transparent, fair (as fair as proof-of-work is). Users may individually have wallets, which are accessed and interacted with through software known as a client. To interface with the network in some regard, such as to make a transaction, a client must be connected to a server.

There can be two types of transaction on the Polygora network - a regular transaction (known as a payment), and an issuance or renewal request (simply known as a request). A payment may be of the types P2PKH or P2SH. A request may only be of a specific form of P2SH, which contains a request to initiate a certificate issuance-related interaction. The network's internal cryptographic token is known as *Aguris* - this is the equivalent to Bitcoin's *Bitcoin*. All transactions in the Polygora network are made with Aguris.

A payment consists of one non-zero amount of Aguris being transferred from the *output* of one or more transactions to the user's wallet, to the *input* of one or more transactions out of the user's wallet (ignoring change, which is unspent AGS being transferred back to the spender - all inputs have to have outputs or the tokens are lost). Sending a payment requires the sender the also pay a small transaction fee, distributed to the miners.

A request is similar to a payment, however the request is to a predefined address, a *central wallet* on the network, *for example*, 0x0000... . A request will then be negotiated with the client - this will follow closeley the ACME protocol specification, however implemented with the Polygora blockchain. The current messaging solution is likely to be adding messages to small back-and-forth transactions, to keep all communication in the chain. Polygora certificates will only be domain-validated, so the network will provide a requester with a challenge to prove they own the domain. This will then be checked by n randomly selected nodes and confirmed - if a node cannot confirm the challenge is complete, the network will select a new group of nodes which can be used to check the success of a challenge. Retries will be both limited in amount, and rate-limited. If the challenge is successful, the network will generate signed certificate and distribute that to the requester. During these interactions, as much as possible is kept in-chain as to provide maximum transparency.

The mining process consists of the same proof-of-work method used by most other cryptocurrencies. The block limit will (at time of writing) be 18.3 million AGS. When that is reached, tokens from the central wallet will be used as block rewards. This ensures there is almost always a place for mining, and provides the network a level of self-sufficiency not found in other cryptocurrencies. (just like Monero, block time will be adaptive, with a similar inflation rate.)

10 Project roadmap

An outline of our internal timeline for the project's progression is presented here. Dates are presented in DDMMYY format.

January 2018

- Start of Aguris token ICO (Initial Coin Offering).

Early-to-Mid Feburary 2018

- End of Aguris token ICO.
  Internal reorganisation relative to income received.
  Enter talks for token listing on major exchanges.

Mid-Feburary 2018

- Start of network topology research initiative, to determine optimal network configuration.

April 2018

- End of research initiative - best solution to be put into development.
  Initialisation of project, under Free and Open Source licensing.
  Enter talks with internet-relevant agencies and organisations.

May 2018

- Release of Polygora client and server software alpha.
  Enable Aguris token exchange with Polygora network.
  Initial token value relative to USD should start to stabilise at equillibrium point.

June-July 2018

- Release of Polygora client and server software beta.

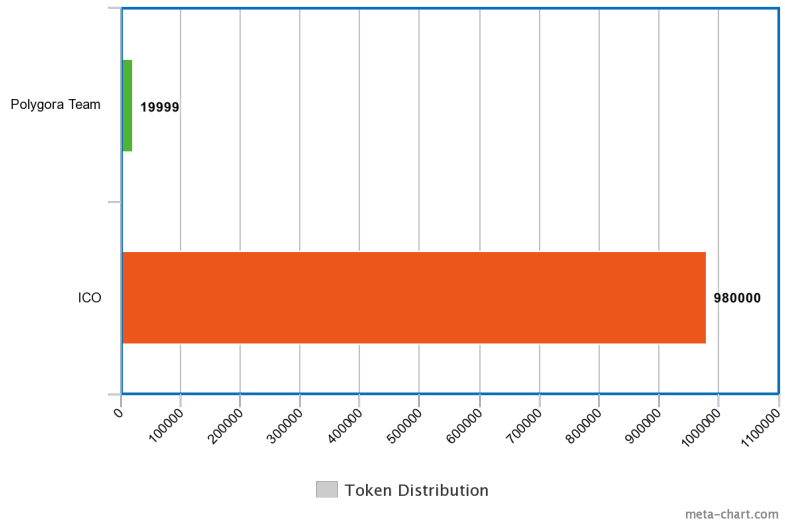
July-August 2018

- Release of Polygora client and server release version.

11 Project funding and presale distributions

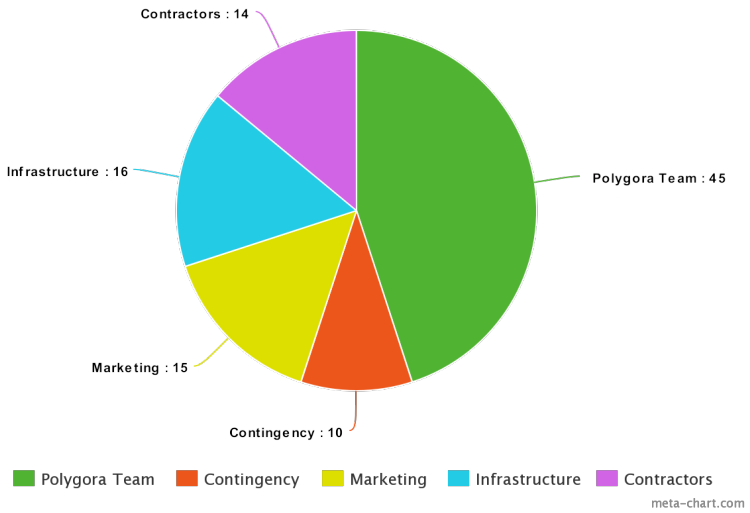
To fund the project and help distribute tokens for block rewards, there will be an ICO (*Initial coin offering*) for Polygora network tokens, with the aim of stimulating the Polygora economy and allowing the market to decide on the value of the project as development ensues, as well as to help fund development of the project. The tokens will first be distributed as *placeholder tokens* on the *Ethereum* network. This will be until a preliminary version of the Polygora software stack has been released, after which ICO tokens will be able to be exchanged 1-for-1 with Polygora network tokens. Double the required tokens will be generated to ensure security of the contract, which will all be burned after the sale period if possible. Aguris tokens *can* be purchased in denominations of less than 1ETH, up to 18 decimal places as allowed by the token smart contract. Conversion is achieved automatically through a smart contract, which will be accepting tokens in the form of a crowd-funding effort.

> The ICO will have a conversion rate of 1ETH to 600AGS, starting approximately early January, and will last until approximately late-January, or mid-Feburary, this is yet to be decided. This sale has an upper limit of 1633.3ETH, or 980,000AGS.



When the and ICO is over, and the development period begins. At the end of each period the total ETH received will be distributed to the project's main wallet, which will further distribute ETH to wallets allocated to various working parts of the project. The distribution will be approximately

as follows:



The following segments of the pie are detailed like so:

Polygora team These are the funds that will be kept to pay Polygora project employee salary. Assuming 100,000 presale tokens are sold, this can fund the development team for at least two years.

Contingency fund This is a fund storing 10% of the project's total ETH at the end of the ICO, kept for a potentially catastrophic emergency event.

Contractors fund Kept on hand to help pay for contracted development work, for example, bounties on added Polygora network functionality.

Marketing To pay for marketing costs, advertising, community involvement schemes and funding project growth initiatives. The presale is partially to fund marketing for the ICO phase.

Project infrastructure Server and domain costs, organisational costs (office space, employment-related costs). This will also be used to help similar projects and support the ISRG on further developing the ACME protocol.

Part III
About Us

12 Our team

The Polygora team are a diverse group of entrepreneurs and business experts, from all over the world, who put their brains together to come up with a unique application of blockchain technology. Our number one aim was that our project be *useful*, something worth putting work into, something that could help improve people's lives and the world. There are enough *in-game token* and *sports betting*-related uses of blockchains, and we really wanted to make a change. Hopefully you too will support our mission of bring security and privacy back to being the core pillars of the internet.

Dakota Harris CEO and Lead Developer. Currently studying for an undergraduate degree in Robotics, in his final year, Dakota has always been inspired to help bring about change. A supporter of Free and Open Source Software (FOSS) since 2010, he is always thinking of solutions and applications for revolutionary engineering in a way that can help everybody, not just those involved. Experienced in software development (C, Go) and hardware design, he gathered the best team members he could find to help design and put together the Polygora project. In his spare time he reads Rothbard and studies financial analysis.

Tadeusz Fabianczyk Lead Developer. Growing up in Poland, frugality was a part of his traditional lifestyle, and that shows in his code. He too studies Robotics, but truly has a passion for developing algorithmic software such as neural networks, and when asked to help participate in the Polygora project he rushed to study up on blockchains and how to build them. Also experienced in programming (C, Go, and C++), he has an unmatched grasp of taking ideas off of paper and turning them into magic. On weekends, he trawls the web, searching for new ideas to implement into his simulations.

Colin Cantwell Business Management. A success in the construction industry, Colin is unmatched when it comes to making on-the-spot business decisions under any amount of pressure. Long time friends with Dakota, he actually convinced him to start the Polygora project - without Colin, Polygora would still be an idea on a notepad. An integral part of the team, he handles decisions with impressive brevity and calm. When unoccupied with work, Colin prefers to sleep in his recliner and smoke big, big cigars.

James Kirkby Web Development and UX. A first-class graduate in Information Technology, James has been a magnificent developer for the past 5 years, working on projects for clients such as the Greater London Authority, and Relendex. With heaps of experience working on back-ends and frontpages, and an endless pit of inspiration, when contacted about supporting the Polygora project he did more than that - he proposed he join as a team member, and we are incredibly thankful for that, as none of us knew web development in any sense.

David Clift Financial Advisor and Business Advisor. A 30-year plus insurance industry veteran, David has spent most of his life calculating the risks of important financial decisions for both his own business and the businesses, and assets, of others. Working in 6-figure and above ranges, David knows what to do when any situation arises, and is able to communicate with corporate clients like nobody else. With an attitude for success, David is a staunch reminder that age brings experience and skill you can't acquire otherwise.

13 Disclaimers

THIS IS NOT A FINANCIAL NOR GAMING-RELATED PRODUCT.
THIS IS A SOFTWARE PROJECT DESIGNED TO FACILITATE RESEARCH INTO PEER-TO-PEER SOFTWARE FOR SECURE NETWORK COMMUNICATION.

The Polygora project and the associated under-development Polygora network are not intended as investment vehicles. The Polygora project is a crowd-funded research effort with no full guarantee of fruition. Polygora tokens are not stocks, shares, securities, or any related product, they are exclusively a digital asset. The purchase price of Polygora network tokens (AGS) is and will be presented in and/or relative to ETH.

None of the information or analyses presented within this document are intended to form the basis for any investment decision nor advise the reader on making any form of investment decision. This document is for informational purposes only and does not constitute an offer or solicitation to sell shares or securities in the Polygora project. This document is subject to change at any point, and the final project may or may not conform to the general specification laid out in the document. The Polygora Project disclaim any responsibility for any direct or indirect loss or damage of any kind resulting from:

1. Reliance of information contained within this document,
2. any error, omission or inaccuracy contained within the information within this document,
3. any action resulting therefrom.

The Polygora token, or AGS, is not a cryptocurrency, but a cryptographic token. At the current time it cannot be exchanged for goods nor services. The value of AGS may fluctuate, and indeed the value of your purchase may decrease as well as increase, and may not increase at all.

The Polygora project will refund all ETH received if it does not return an appropriate research report by the dates stated on the Polygora roadmap. Any other refunds will be at the sole discretion of the Polygora project.

AGS tokens purchased during the presale period will be issued once the presale period has concluded. AGS tokens purchased during the ICO period will be issued once the ICO period has concluded.

It is the sole responsibility of the purchaser to ensure they have lawful permission to exchange ETH for AGS tokens, within the borders of the country which they reside. The Polygora project stores no information about token purchasers except their Ethereum wallet address.

This is a living document and all content within the document may be subject to change, without prior notice.