Gladius

Technical Paper

December 2017

Section 1
Overview

1.1 Background

DDoS attacks are becoming an increasing reality of operating a business with an internet presence. These attacks can cost an incredible amount of money, not just in lost uptime, but also a loss of consumer trust and crisis PR. Last year alone DDoS attacks amounted to a $150 billion loss, while on average corporations can spend over $5000 a month on protection from other DDoS protection services, even if they never suffer an attack. This happens while a majority of computing power and bandwidth in the world goes unused. Gladius envisions a future where these vast resources can be utilized to mitigate attacks. (Incapsula and SecureList)

By leveraging the blockchain to allow communication between computers, and a pay as you go marketplace, Gladius can facilitate the creation of the extremely fault tolerant and inexpensive pools tailored to a client's specific needs. Aside from just filtering traffic, these pools will be able to accelerate content as well.

By decentralizing and removing the middleman, the whole nature of protecting a website will change. Business that previously couldn't afford adequate protection can purchase services that perfectly fit their budget and operating requirements. Those paying hundreds of thousands of dollars for a service they never fully utilize can start paying for only exactly what they use - and for much less. In addition, customers who are tired of paying for a high speed internet connection (that is only used for a few hours a day) can finally put that connection to good use.

1.2 Platform overview

Gladius' goal is to create a fully decentralized, peer to peer, serverless node network to connect bandwidth and storage pools to websites looking for DDoS protection and expedited content delivery. Anyone with a computer can download and run the Gladius peer client in the background to rent out their unused bandwidth and storage space and earn Gladius Tokens (GLA). Large pools with hundreds, if not thousands, of nodes will then be able to handle a continuous stream of requests to validate website connections and block malicious activity.

Client nodes can be started up on any Linux, Windows, or MacOS machine and will automatically run in the background when you choose. Users will be part of localized pools where they contribute their bandwidth and small amounts of storage and processing power. Every website request a user processes will earn them Gladius Tokens. In turn, users can sell these tokens back to websites to create an economic cycle that promotes the growth of the Gladius Network.

Websites looking for protection will simply be able to create an account on Gladius' website, acquire Gladius Tokens, and then request services in a matter of clicks. Once it is part of the Gladius network, a live request graph will be available to monitor connections, protection, and speed deltas.

The entire platform will progress in 3 steps which can be seen in the roadmap. First we have the proof of concept stage that will showcase all of the basic functions of the network while not actually going into official production mode. Then, after our token sale, we will have a closed production period where anyone can download and run the client node.

However, only approved websites can purchase protection since the network and pools will still be in early growth stages. Once the Gladius Network can support a large attack and the core tech is improved, we will enter the next stage. The Gladius Network will then allow anyone to purchase services from the various pools.

The process will be streamlined for both the clients and the requesters. People who want to purchase web services will simply be able to create an account with Gladius, purchase Gladius Tokens, choose a website they wish to synchronize, set a base and max price for how much they are willing to spend, and then request and monitor all connections to their website.

Nodes will easily be able to create a local account and wallet, configure the network settings to open any necessary ports, join any pools that they are best suited for, and start earning GLA. Users will be able to toggle when they are renting their spare bandwidth and space in a matter of seconds. There will also be additional configuration settings to automatically toggle the service based upon time of day, other programs running, and even more in-depth user configurable parameters.

1.3 Tokens

For our platform we will be releasing the Gladius Token, or GLA. A fixed supply of tokens will be issued during the Token Creation and no more tokens will ever be created. The tokens will immediately be available to be used on our network system we are launching prior to the public sale.

GLA will be a key component to the Gladius Network in that the token will be used by websites to buy DDoS protection and CDN services. The majority of the fees will go to the node owner (the individual renting out their spare bandwidth and storage space) and the pool manager, with a small portion (less than 1%) going back to protocol development and support. All fees will be denominated in GLA, which is subject to change based on supply and demand.

The previously mentioned node owners who are part of protection pools will in essence act as miners and be incentivized with GLA for their network support. Node operators will be paid for their individual work in these pools, and will be rewarded for sharing bandwidth and storage space.

Section 2
Technical Details

2.1 Gladius Architectural Description

Gladius works similarly to traditional CDN and DDoS protection companies by creating a custom proxy which sits between a website's server and the open internet. However, unlike traditional networks the layer that sits between the website and the internet is made up of small clients that split up the traffic verification and cached files/ content into thousands of tiny parts that are able to communicate with each other in fractions of seconds.

Pools of nodes exist to group people into demographic (geographical, pricing, SLAs) groupings to provide a better and faster network experience. These pools can be seen and accessed through a marketplace where their information on geographical location, pool size, and reputation can be viewed. Pools can additionally consist of only one individual's or organization's resources, allowing them to run their own instance of a Gladius pool by not approving outside nodes. Using our contracts, anyone can distribute their content and combat DDoS attacks.

A critical component of any DDoS protection system is keeping the IP address of the server hidden, the Gladius network accomplishes by having a final proxy (the master node) mask the IP from the nodes in the pool. The network will also have a built in reputation system to prevent malicious pools. Pools also have the ability to approve individual nodes entering the pool, allowing for a secure experience.

To create a fully functioning CDN, each node will be capable of caching key content, and then delivering that content to nearby clients. A client will be directed to closest node to them by using a location based DNS server, guaranteeing they will always be sent to the nearest available node.

2.2 Technical Component Overview

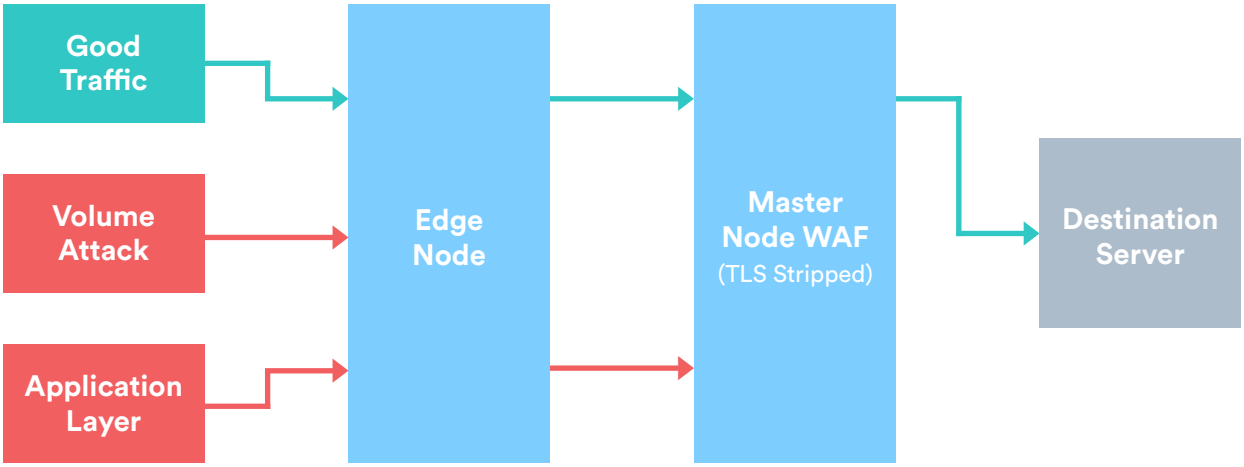
Storage and Marketplace - Ethereum Blockchain
Acts as a decentralized database for storing the pools and their associated service providers, as well as a marketplace where these services are sold. To discourage spam there is a GLA stake required to add to the database and list on the market, which can be dynamically adjusted to account for changes in the token price.

Node - Proxy and Cache
A series of clients act as a distributed traffic validator while keeping the protected service hidden from the rest of the internet. These nodes will primarily serve as a cache for site content and be part of the Gladius CDN, they will also serve as a primary barrier to filter traffic based on IP address information.

Master node - WAF
Acts as a source of content for local nodes, strips TLS for the Web Application Firewall, and acts as a barrier between untrusted nodes and the end server's IP address.

Pool Manager - DNS Anycast system
The manager will be responsible for building and maintaining a DNS Anycast system that is highly distributed and can control bandwidth flow to nodes using a low TTL. They will also be responsible for content verification servers, uptime checking, and load testing the nodes before entry. The main goal of this system is to lower the barrier of entry and make this process as easy to deploy as possible.

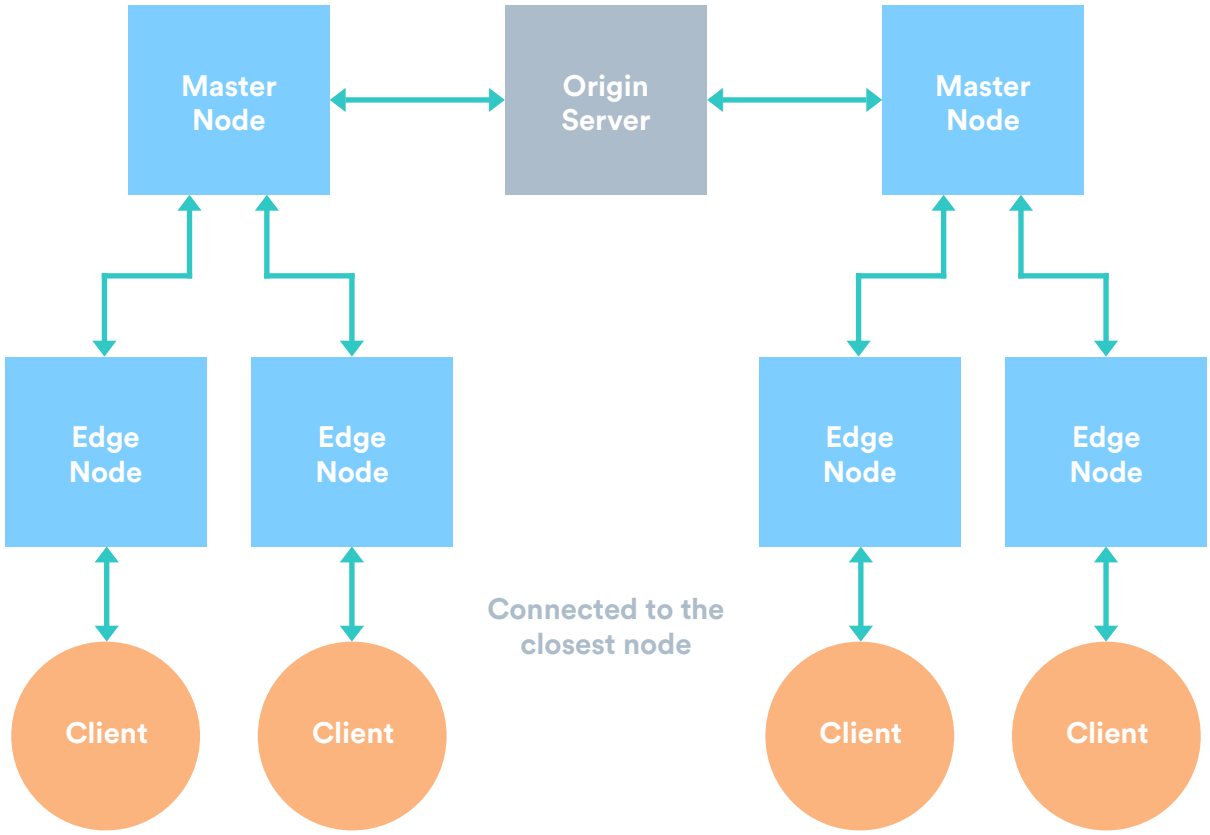
General diagram of traffic flow



2.3 Content Delivery Network

Overview

The Gladius network can more effectively deliver content faster than existing architecture because of the decentralized nature of it. By having many relatively low bandwidth nodes spread over a large area, a potential client will be connected to a node that is extremely close to them. In the ideal scenario, instead of having a datacenter far away but with high capacity, a client will connect to a node with less capacity but that is very topographically close to them. Every node will have static file caching so that the bulk of the data that a client requests will come from a nearby node, while API calls and other dynamic data will be passed through to the end server.



Security

To verify that files have not been tampered with by a malicious node, each pool will be able to send verification requests from the master node through another node to the node being tested. Because the blockchain ensures that each node has no knowledge of the IP information of other nodes in the network ensuring that they would treat a verification request like any other. If the files do not match (i.e. their hashes are not the same), the pool will have the option to immediately remove the node, or damage their reputation. There is a potential that pools could also do this with an independent auditor that was unknown to the nodes and randomizes the requests and the time it makes them.

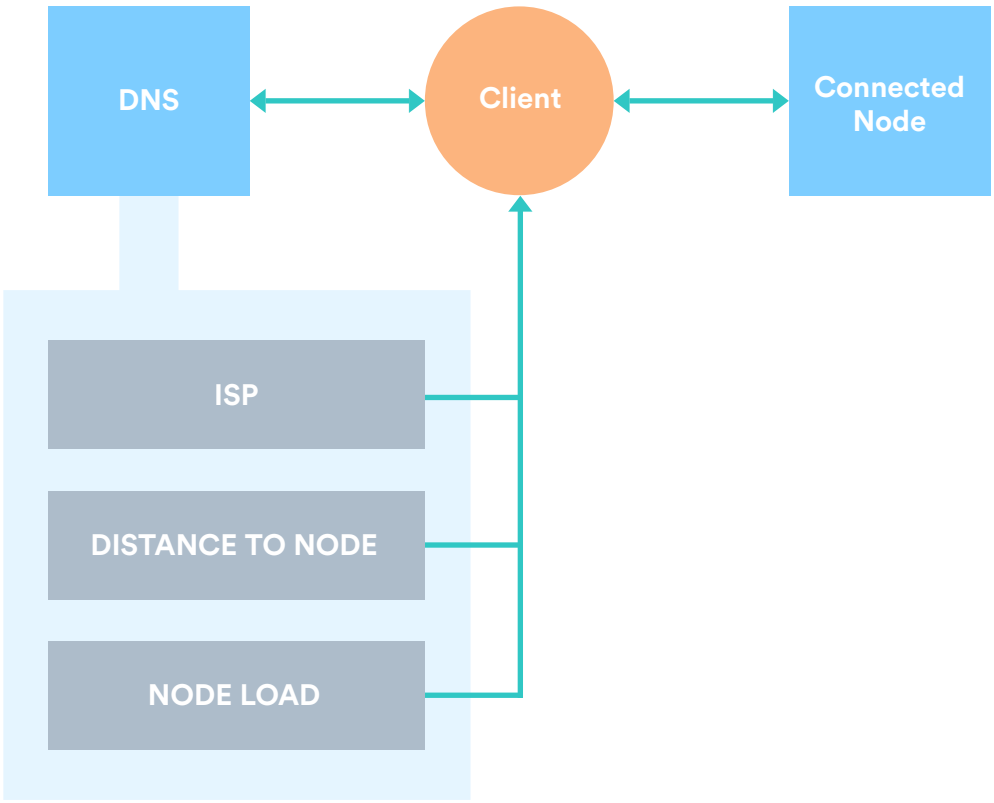
Reputation of nodes will also be calculated based on random uptime requests initiated by the pool. Reputation will be treated as a multiplier on top of the profit earned from the number of requests, so that the total amount the pool pays out never changes, but shifts from poorly performing nodes to better performing ones. This system creates incentives to have high uptime. Additionally content delivery optimizations can and will be added as the platform matures. From dynamic sampling of website metadata in order to intelligently update caches, to better and more custom-tailorable load balancing techniques, there are dozens of additional features to be added after the initial release of the Gladius Network.

DNS Solution

To implement an easy to deploy, GeoIP based, and bandwidth limiting DNS Anycast system, Gladius will need to develop a custom solution. Our anycast solution will be quite resilient to attack because of the inherent property of distributing load over many nodes throughout the system.

By lowering the TTL (the time it takes for a client to ask for a new IP from DNS) we should be able to effectively redirect bandwidth to connections over 40 megabit/s without issue, lower numbers could be achieved however that would require significantly higher resource allocation from the pool manager. The low TTL also makes it relatively easy to quickly remove offline nodes when they are unable to show uptime to the pool manager.

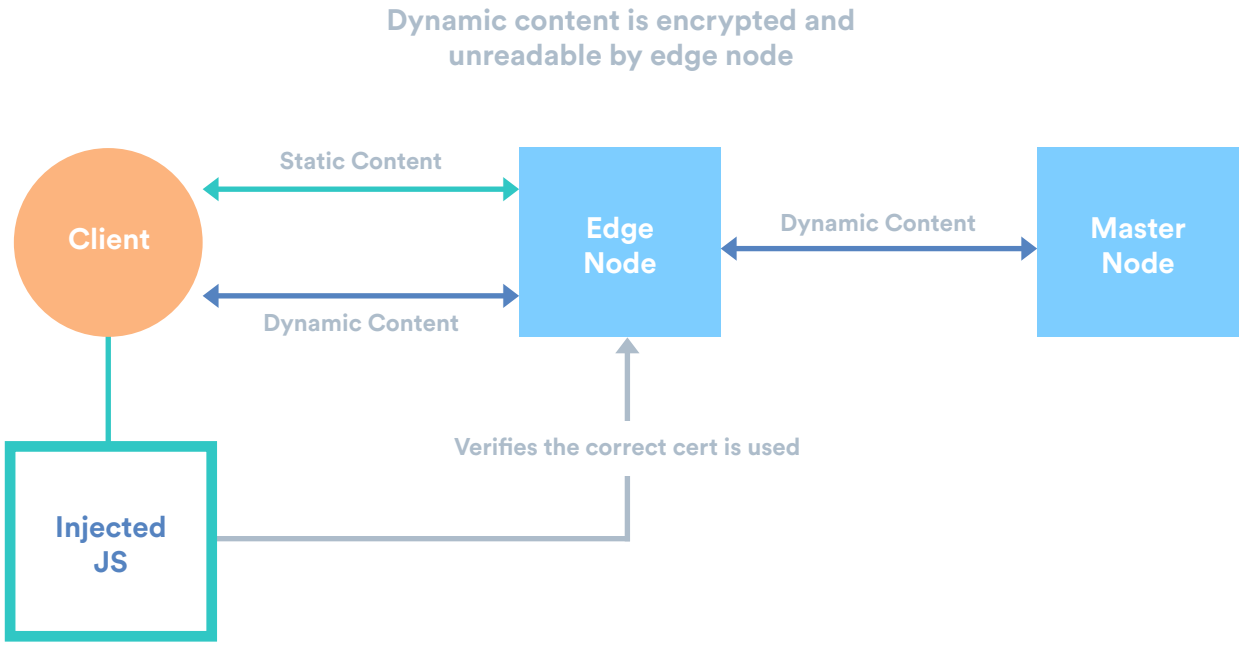
The priority for how traffic is directed will be like this



Aside from the general goal of ease of deployment, we will also be implementing horizontal scaling facilitated by the blockchain, being able to on the fly update zone files without any necessity of IP communication between servers, or manual updating.

TLS/SSL

To ensure that nodes are properly forwarding dynamic content and can't listen in on user's information, we use two separate certificates: one held by the nodes to serve static content, and one held by the master nodes to forward dynamic content. Because our master nodes are the source of static content for other nodes, we can inject JavaScript into the served content. As stated earlier, the pool performs static content checks, and we use this to our advantage by ensuring that we have injected JavaScript that does the certificate validation in the browser. Normally, a browser would not care which certificate is used because both certificates will be valid for the domain, so we need a way to ensure the correct one is used. We can verify this by injecting code that makes random calls at random times to endpoints in the destination application and checking the certificate that is used.



This JavaScript would then be able to stop page rendering, or warn the user if the incorrect certificate was used as this could indicate snooping or malicious intent. It would then notify a service run by the pool manager that the node they connected to was performing abnormally, depending on the pool policy they could be immediately suspended or have significant reputational damage.

Heartbeat

Every node will have a "Heartbeat" that it broadcasts to the pool manager. This heartbeat allows the pool to know what nodes are online and dynamically adjust where traffic flows in the network. Because of the design of the network and the surface area that is presented to a potential attacker, this means that even if a portion of the network is taken offline the system should be able to recover relatively quickly and move traffic to unaffected areas.

2.4 DDoS Mitigation Techniques

To combat DDoS attacks Gladius nodes inside the pool will perform a number of services and computations to ensure requests are not malicious. By breaking up the work into our two main components we can more effectively target resources.

Node Level - IP Based
Because nodes do not have access to the actual traffic data, they are responsible for handling only IP based DDoS mitigation.

Rate Limiting
By identifying IP addresses that are constantly making website requests, Gladius can block these requests from getting access to the website. Once the threshold of requests is hit (which is set by the service requestor) the IP will no longer be able to access the site.

IP Address Matching
Similar to rate-limiting but smarter, IP address matching will be able to group similar IP address together that have known associations with each other. This will take shared information from the pool to block threats ahead of time.

Intelligent Geo Matching
Additionally Gladius can analyze requests and find geographic anomalies to detect and block attacks by region.

Master Node - WAF Based
Master nodes strip the SSL, and are able to view the raw data. This allows them to do analysis on all the traffic before it is sent to the end server. Here we can look for common attack vectors such as XSS and SQL injection, as well as more advanced rules that will continually evolve to meet the industry requirements. Gladius will also actively be researching and acquiring new exploits before they are used maliciously.

Overall a large amount of the protection utilizes anomaly detection to stop malicious attacks in their tracks. A key feature of the Gladius Network is that pools over time will learn about common attackers and block them preemptively for all other sites and services that are being protected.

2.5 Smart Contracts

General communication
All communication on chain happens in an encrypted way. Every entity in the network (node, pool, client) will store their public key on chain. This allows off chain encryption in our applications and subsequent communication between contracts to be entirely encrypted.

Pool Contracts
A pool contract is instantiated in a factory pattern from the marketplace and stored in the mapping that maps pools to owners.

The database of pools will be maintained on the blockchain, and because of how Ethereum functions, this has an inherent cost associated with it, along with a fixed cost implemented by the marketplace to incentivize pool owners to be serious as well as honest. Joining a pool is initiated by a node with an Ethereum smart contract. The pool can deny this request if it believes the node would not be beneficial based on the user's general demographic information such as location and available bandwidth and storage space.

In this database there will be information about reputation, bandwidth, maximum cache size, and location. Reputation will be based on key information such as user reports and subsequent investigations, protection provided over time, total pool age, and total pools size. Bandwidth is a self reported quantity and will be based on the aggregate of all of the node's available bandwidth as well as the maximum bandwidth the pool receives. Because of the off blockchain communication it will allow evidence other than just what is stored on chain. Location will be viewable on a per node basis. Maximum cache size is based on the aggregate of storage space made available by nodes. Having all of this information allows for websites to choose pools that would be best suited for their needs. For instance, websites could pay more for a very trusted pool that has many nodes close to their target audience.

Node

When a node joins a pool they execute a function on the pool's smart contract which adds them to the pending nodes. Their "application" information is encrypted off blockchain using the pool's public key, and then added to their application. This includes contact info, bandwidth information, IP address etc.

Marketplace

The marketplace is the core of the Gladius Network, it allows websites to browse and pay pools for their services, and allows nodes to broadcast their services directly to pools. When a pool or a node lists themselves on the market, it will require a certain amount of GLA tokens to spend to show stake in the ecosystem and lower spam (value could be changed from default by pool manager). The data on each pool would contain information about number of clients, number of nodes, mitigation capacity, pricing, and most importantly the pool's reputation. Pools can be removed from the marketplace by consensus of the market, or very poor reputation from clients (clients can only review after paying a certain amount to discourage false reports by malicious actors).

From the client's perspective, there will be a single page application that connects them to the blockchain using web3. When they request protection from a pool they submit an encrypted application much like the node. Multiple applications could be submitted as well, however in the initial release only one pool will work at a time. This encryption is done off chain in the browser, and it includes all the relevant information that the pool manager needs to onboard them, like domain, contact information, estimated traffic, the amount of GLA the client has, and any notes they would like the pool manager to receive. After this transaction has taken place and the pool manager has accepted the client, they provide the website with nameservers that the website must update in the same encrypted way as before. Finally to pay the pool, a client can put an amount of tokens (and provide daily maximums) in the marketplace to be transferred to the pool as they provide services.

2.6 Roadmap

Pre Token sale

The goal of this stage is to complete a minimum viable product to showcase the core system architecture of the Gladius Network. We will have a functional CDN that users can participate in and earn tokens for their bandwidth and storage space. DDoS protection will be done through firewall rules.

Development Goals | Pre Token Sale
(Released on GitHub: github.com/gladiusio)

- Smart Contracts V1.0
- Payment flow from websites to pools to nodes
- Gladius Client V1.0 - Ability to earn GLA (when tradable) renting bandwidth and storage space
- Gladius Web Portal
- Ability to pay pools for website CDN service

Note - This stage will be immediately usable by purchased tokens after the public sale. This is our core product for the Gladius Token. Additional features are scheduled to be added to this version.

Phase 1 - ETA March 2018

The goal of this stage is to complete a working first iteration of a complete CDN and DDoS protection network capable of handling several Gbs of connections per second. This will include multi-level protection and a private rollout to select partnered websites.

Development Goals | Phase 1
(ETA: March 2018)

- Smart Contract V2.0 -Gladius Client V2.0
- Full pool integration, headless client mode, and improved blockchain integration
- Gladius Node Pools V2.0
- Improved blockchain integration, and the start of a vetting process for new nodes
- Fully Encrypted communications (TLS and trustless nodes)

Phase 2 - ETA August 2018

The goal of this stage is to completely finalize the network so that it will be commercially viable on a large scale. This means that we will have a network large enough to take on hundreds, if not thousands, of websites. Additionally we will potentially open the marketplace and allow for more pools to be created, each being rated and validated to ensure they are performing up to standards.

Development Goals | Phase 2
(ETA: August 2018)

- Remove centralized server (DNS scalability improvements)
- Smart contracts for discovery and identification services (Marketplace for pools and nodes)
- Interface implementation for add-on modules injected into static content.
- Full vetting and rating process in the marketplace
- Complete auto-payment and bid/ask system for the marketplace
- process for new nodes
- Fully Encrypted communications (TLS and trustless nodes)

Phase 3 - ETA December 2018

The goal of this stage is to add several additional features to the Gladius Network to add "polish".

Development Goals | Phase 3
(ETA: December 2018)

- Complete multi-pool support for protection purchasers in the interface (data aggregation etc.)
- Add novel CDN techniques to further increase load speeds
- Stretch Goals