The Golem Project

Crowdfunding Whitepaper



final version

CRYP2NOVA

November 2016

Archived copy · cryp2nova.com

Table of contents

- Overview of the Golem Project
  - Grand vision and core features
  - Golem as an Ecosystem
    - Supply of Infrastructure
    - Demand for Computing Resources
    - Software & Microservices
    - The first use case: CGI rendering
    - Long term vision: Golem as a building block of Web 3.0
  - Golem Network Token (GNT)
  - Application Registry
  - Transaction Framework
  - Resilience
- Roadmap
  - Brass Golem
  - Clay Golem
  - Stone Golem
  - Iron Golem
  - Future integrations
- Crowdfunding
    - Crowdfunding summary
    - Budget and levels of funding
- Golem Team
  - Managers
  - Key developers
  - Developers
  - Business development and communication

CRYP2NOVA

Archived copy · cryp2nova.com

Overview of the Golem Project

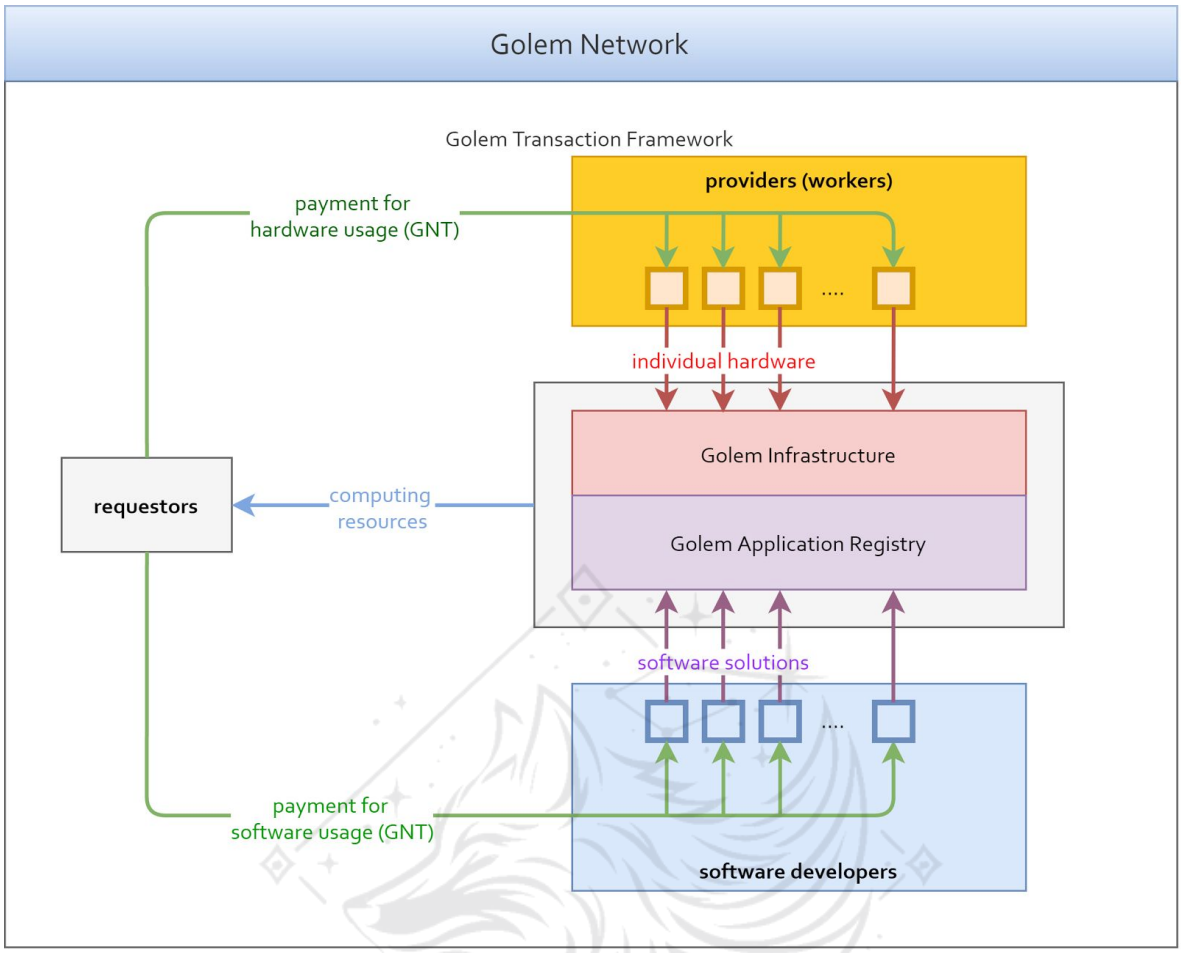
Grand vision and core features

- Golem is the first truly *decentralized* supercomputer, creating a global market for computing power. Combined with flexible tools to aid developers in securely distributing and monetizing their software, Golem altogether changes the way compute tasks are organized and executed. By powering decentralized microservices and asynchronous task execution, Golem is set to become a key building block for future Internet service providers and software development. And, by substantially lowering the price of computations, complex applications such as CGI rendering, scientific calculation, and machine learning become more accessible to everyone.

- Golem connects computers in a peer-to-peer network, enabling both application owners and individual users ("requestors") to rent resources of other users' ("providers") machines. These resources can be used to complete tasks requiring any amount of computation time and capacity. Today, such resources are supplied by centralized cloud providers which, are constrained by closed networks, proprietary payment systems, and hard-coded provisioning operations. Also core to Golem's built-in feature set is a dedicated Ethereum-based transaction system, which enables direct payments between requestors, providers, and software developers.

- The function of Golem as the backbone of a decentralized market for computing power can be considered both Infrastructure-as-a-Service (IaaS), as well as Platform-as-a-Service (PaaS). However, Golem reveals its true potential by adding dedicated software integrations to the equation. Any interested party is free to create and deploy software to the Golem network by publishing it to the Application Registry. Together with the Transaction Framework, developers can also extend and customize the payment mechanism resulting in unique mechanisms for monetizing software.

Archived copy · cryp2nova.com



Golem as an Ecosystem

Golem's business case boils down to the fact that, due to relatively recent technological advances, the market for computing resources can be organized according to entirely new principles. In contrast, the compute market today is dominated by heavyweight players such as Amazon, Google, Microsoft and IBM, who leverage their market power and assets to ensure hefty margins, resulting in inefficiently priced compute services. Luckily, the market is not doomed to function this way forever. With Golem the supply of computing resources is based on contributions of individual and professional providers, combined with an array of dedicated software solutions via Golem's Application Registry – itself operating on a single and competitive market with nearly complete information market.

Scaling the compute market enabled by Golem requires onboarding three groups: suppliers of computing resources ("providers"), task creators ("requestors") who submit their tasks to be computed by the network, and of course, software developers. These three groups comprise Golem's unique, interdependent ecosystem.

Archived copy · cryp2nova.com

Group	Golem features	Incentive to participate
Requestors	Golem offers tools to execute compute-intensive tasks.	Requestors get access to affordable and scalable solutions, which combine hardware and software.
Providers	Golem combines and utilizes (almost) any kind of existing computing hardware.	Hardware providers get paid for renting out their hardware.
Software Developers	Golem is a flexible platform to deploy and monetize software.	Software developers use Golem as a distribution channel, associated with access to hardware.

Supply of Infrastructure

The supply of computing power to the network comes from providers. This could be anyone, from an individual renting out idle CPU cycles of a gaming PC, to a large datacenter contributing their entire capacity. Providers have the incentive to join Golem because they receive payments from requestors for the completed tasks. Of course, Golem's user interface will be easy to use, giving providers a clear way to set prices and decide what fraction of their own idle resources they are willing to rent out.

Demand for Computing Resources

In order to reward providers for contributing their resources, Golem needs to attract requestors seeking additional computing resources. The market Golem creates will be highly competitive due to nearly complete information, and the ease of deploying tasks on any hardware. This will not only make using Golem simple, which will attract requestors - highly competitive setup will also increase efficiency of the market, very likely resulting in much more comprehensive and advantageous pricing compared to the existing cloud computing platforms.

One important feature is that Golem's marketplace will enable requestors to become providers: because most of them will need additional resources only occasionally. They will be able to rent out their own hardware and earn extra fees. In addition, financial aspects are not going to be the sole incentive to use Golem: one of its key features will be the availability of diverse software running on the Golem network, accessible from the Application Registry.

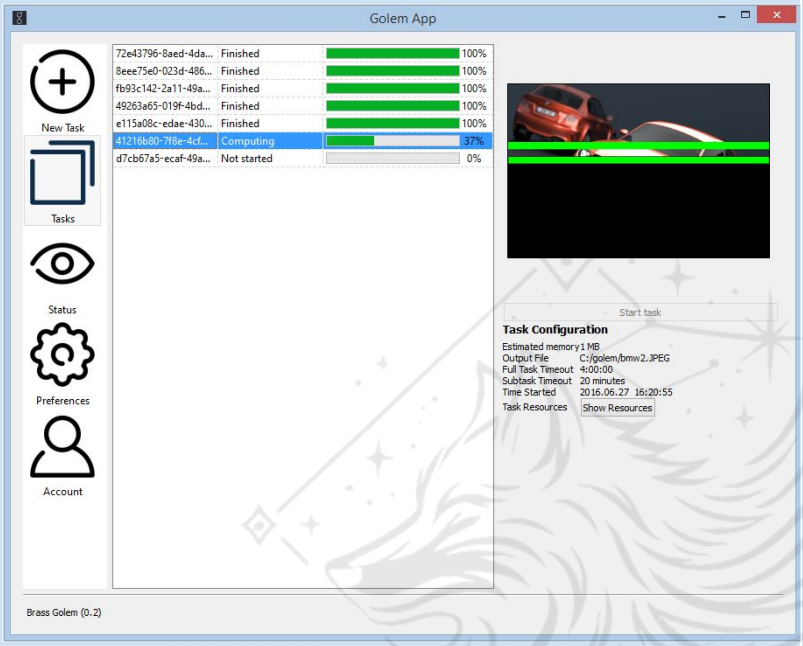
Software & Microservices

Although some initial use cases (such as CGI rendering) are going to be developed and implemented by the Golem team, it is essential to engage other software developers to develop their own ideas for Golem applications. The number and quality of such applications is one of the key factors in Golem's future success. For this reason, the Application Registry and Transaction Framework are among the most important features of the entire ecosystem,

Archived copy · cryp2nova.com

and will be front and central to the development process. Once introduced, they will provide developers with flexible and efficient tools to deploy, distribute, and monetize software running on Golem. This is going to be a perfect solution for microservices and some of the forthcoming decentralized applications (DApps), but could also become an interesting way to distribute existing proprietary and open source software.

The first use case: CGI rendering



Golem Alpha release: CGI rendering using Blender
It is public, follow [the link] to test Golem.

CGI rendering is the first and very illustrative case of real Golem usage. Rather than using costly cloud-based services or waiting ages for one's own machine to complete the task, CGI artists can now rent compute resources from other users to render an image quickly. The payment from a requestor (in this case, a CGI artist) is sent directly to providers who made their resources available. In addition, when the artist's machine is idle, it can itself accept tasks from other users.

Long term vision: Golem as a building block of Web 3.0

We believe the future Internet will be a truly decentralized network, enabling users to securely and directly exchange content, without sharing it with corporations or other middlemen. Accordingly, Golem will be useful not only to compute specific tasks, but also to bulk-rent machines in order to perform operations within a self-organizing network. Of course, this will require the simultaneous development of other technologies, many of which have gained significant traction in recent years.

Better data-sharing technologies are necessary, but taking into account the ongoing development of IPFS/Filecoin and Swarm, the appropriate solutions seem to be within reach. Eventually, the Ethereum network will become more scalable, more efficient, and include a fully functional network of micropayment channels. Once these technologies become available, it is easy to imagine Golem primarily as a platform for microservices, allowing users to run both small (e.g. a note-taking app) and large (e.g. a streaming service) applications in a completely decentralized way. Although ambitious, this vision seems to be the ultimate argument for Golem's long-term potential.

Archived copy · cryp2nova.com

Golem Network Token (GNT)

The Golem Network Token ("GNT") account is a core component of Golem and is designed to ensure flexibility and control over the future evolution of the project. GNT is created during the crowdfunding period (described in this whitepaper) and, following the first major release of Golem, GNT will be attributed a variety of functions in the Golem network.

- Payments from requestors to providers for resource usage, and remuneration for software developers is going to be exclusively conducted in GNT.
- Once the Application Registry and Transaction Framework are implemented, GNT will be necessary for other interactions with Golem, such as submitting deposits by providers and software developers or participation in the process of software validation and certification (as described in the Application Registry section).
- The general conditions for using GNT will be set in the Transaction Framework, but specific parameters of these interactions will be possible to define within each software integration.

The supply of GNT will be limited to the pool of tokens created during crowdfunding period.

Creation of the GNT and initial GNT account functionalities

- The Golem Network Token is a token on Ethereum platform. Its design follows widely adopted token implementation standards. This makes it easy to manage using existing solutions including Ethereum Wallet.
- Maximum number of tokens created during crowdfunding period:
  - Total: 1 000 000 000 (100%)
  - Crowdfunding participants: 820 000 000 (82%)
  - Golem team 60 000 000 (6%)
  - Golem Factory GmbH 120 000 000 (12%)
- Sending 1 ether to the GNT account will create 1 000 GNT
- No token creation, minting or mining after the crowdfunding period.
- Tokens will be transferable once the crowdfunding is successfully completed.

Go to the crowdfunding section to learn the details and see how to support the Golem Project via crowdfunding.

Archived copy · cryp2nova.com

Application Registry

The Application Registry is an Ethereum smart contract, to which anyone can publish their own applications that are ready to run on Golem network. The goal of the Application Registry is to:

- Give developers a way to publish their integrations and reach out to users in a decentralized manner;
- Give requestors a place to look for specific tools fitting their needs;
- Give providers full control over the code they run because of the security concerns.

Since the Golem network is fully decentralized, we also want the Application Registry to be driven by the community.

Golem allows requestors to execute the code of an application on someone else's computer. This code is sandboxed and executed with the minimal required privileges. But software bugs are everywhere, and once in a while people defeat sandboxes, manage to execute malicious code on a host machine, and sometimes even take it over. That's why we can't rely only on sandboxing. We could try to automatically evaluate whether or not the code is safe, but this is literally impossible (*vide* halting problem). The process of code review and validation cannot be fully automated and left to the autonomous network. On the other hand, it is impossible to assume that no one will ever publish malicious software to run on top of Golem network.

We solve these problems by splitting Application Registry users into three categories: authors, validators and providers. Authors publish applications, validators review and certify applications as safe and trustworthy by adding them to their own whitelist. Validators may also mark applications as malicious by adding them to their own blacklists. Providers are also given the right to choose whom to trust by selecting validators whose lists are used by the particular instance of Golem running on their nodes. Apart from that, providers may maintain their own whitelists or blacklists. This gives each provider a lot of flexibility in deciding exactly what software to run, and what amount of work to put into software curation. What is more, this system does not exclude any party, and there is always room for new validators to emerge.

By default Golem runs using a whitelist of trusted applications. Since an empty whitelist is a problem for someone just trying Golem out for the first time, we will add a number of verified entries to the whitelist as a part of the initial distribution. A provider can take advantage of this mechanism, managing her own whitelist, or simply using whitelists of validators she trusts.

On the other hand, a provider running a computing farm may wish to rely entirely on blacklists. This is an option tailored for administrators of dedicated machines, who want to maximize their profits and who are willing to ride the bleeding-edge. In this scenario, a blacklist is used to banish any known troublemaking applications. Again, the provider can maintain her own blacklist, or use the blacklists of validators she trusts.

Archived copy · cryp2nova.com

Transaction Framework

When creating something new and exciting, it's hard if not impossible to predict all the opportunities which the new artifact will suddenly make possible. Golem is a generalized global supercomputer, and as such, it will no doubt find its niche with vastly varied applications. They might need very diverse remuneration models. We are not able to design a one-size-fits-all payment system for Golem, nor will we attempt to force one upon application authors.

When a developer integrates her application with Golem, she has the freedom to decide which transaction model she implements, as long as it is compliant with Golem's Transaction Framework. The Transaction Framework will take the form of a set of requirements to follow; basic requirements may include:

- Entry in the Application Registry;
- Use of open source and/or deterministic environment, such as EVM;
- Community approval or rating of transaction model;
- Use of GNT for remunerating software and resource providers.

We are building the transaction framework on top of Ethereum. Ethereum gives us expressive power, which is much-needed when implementing advanced, trustless schemes.

Example transaction framework components:

- Diverse payout schemes such as nanopayments, batching
- Off-chain payment channels
- Custom receipts
- Payment to software developer
- Per-unit use of software (per-node, per-hour, etc.)

In the future, this might evolve into community-reviewed template code to be used in the implementation of custom transaction models.

It is also possible to introduce more sophisticated components into the transaction model design, to meet specific goals not related to payments. For example:

- Requestor escrow for tasks where a higher level of commitment is required (high price because of specialized hardware or long running subtasks); the requestor may create a two-party escrow account and require providers to take part in it.
- Provider deposit: the requestor may require to be in control of some amount of timelocked GNT.
- Requestor deposit: the provider may accept tasks *only* from requestors who are in control of some amount of timelocked GNT.
- Registration of a task as an anchor for a TrueBit-style conflict resolution fallback mechanism.

Archived copy · cryp2nova.com

Resilience

Golem is a fully decentralized, open source P2P network which is resilient to censorship, and free from a single point-of-failure.

Consensus is important for resilience and is another reason why Ethereum is utilized for transactions, as well as replicating some shared state and metadata. Golem tasks rely on integrity and consensus mechanisms for deployment, task execution, validation, and transactions. Naturally, Golem inherits Byzantine fault tolerance properties from Ethereum. For task execution and validation, Golem will initially rely on redundant validation and Ethereum state. As research and development progresses, optimizations will be implemented to reduce costs, increase throughput, and improve resilience. Optimisations will result in improvements of integrity and consensus validation schemes, P2P network formation, asynchronous transactions, and off-chain state transitions.

The Golem P2P network will be built on a continuation of the devp2p protocol series, which was spearheaded at the Ethereum Foundation, and which will see improvements in confidentiality, robustness, latency, modularity, and the inclusion of pertinent libp2p and IPFS standards.

Additional means to achieve resilience:

- Signed and encrypted messages inside the Golem network ensure authenticity, which protects against man-in-the-middle attacks and passive data collection.
- Computation takes place in isolated environments with minimal privileges and lack of external network connectivity.
- Whitelist and blacklist mechanisms allow providers to build trust networks and run *only* applications prepared by trusted developers.
- Reputation system helps to detect malicious node and mitigate them and additionally evaluates metrics to assist in secure, efficient and correct task routing.
- Together, the application registry and transaction framework mitigates Sybil and whitewashing attacks by providing an incentive to participate, introducing an economic and computational cost to participation, and providing a metric for reputation in order to maintain optimal connectivity.
- The Ethereum integration and transaction framework makes possible custom payment-based security mechanisms, eg. escrows, deposits, insurance and audit proofs.
- Security audits will be conducted for every release, performed by external contractors.

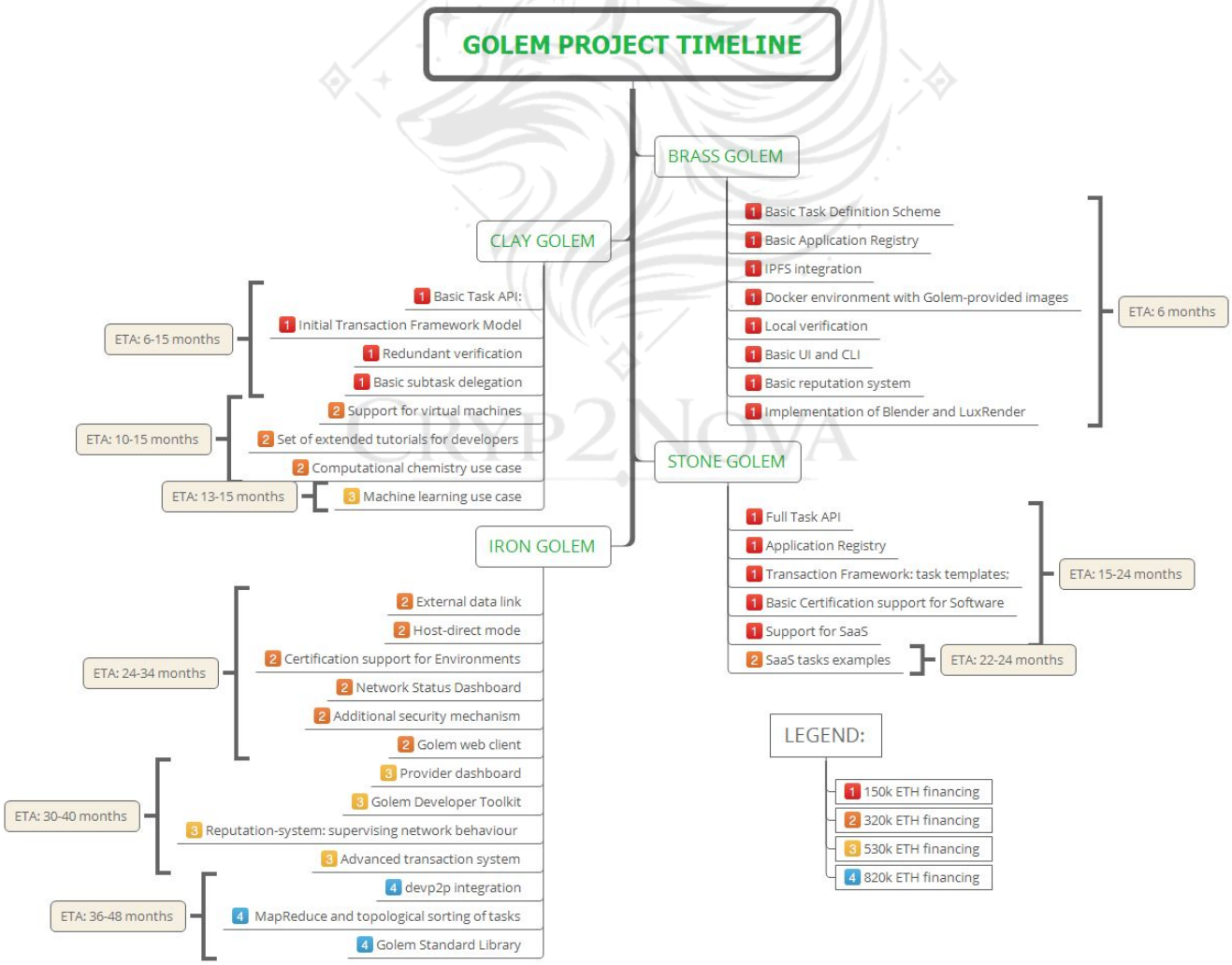
Archived copy · cryp2nova.com

Roadmap

In this section we present planned milestones for Golem development. You will find a nontechnical description of Golem's architecture in this post on our blog, some thoughts about challenges ahead here and of course you can examine the code on GitHub.

The successive versions of Golem's software are split into milestones. This plan should be considered preliminary, as Golem is using bleeding-edge technologies, and is itself a very complex project. Each milestone should be preceded with research, with results described in technical whitepapers. At every stage new functionalities are added, but the scope of deliverables at every milestone depends on the level of funding raised. In the description of milestones presented below, these functionalities have been assigned to four indicative funding scenarios: (1) are implemented regardless of the level of funding, while with (2), (3) and (4) will be implemented if the adequate level of funding is reached.

We have codenamed concurrent versions of Golem using descriptions from the long list of D&D's golems. The analogs might not be perfect, but these are e-golems after all.



Archived copy · cryp2nova.com

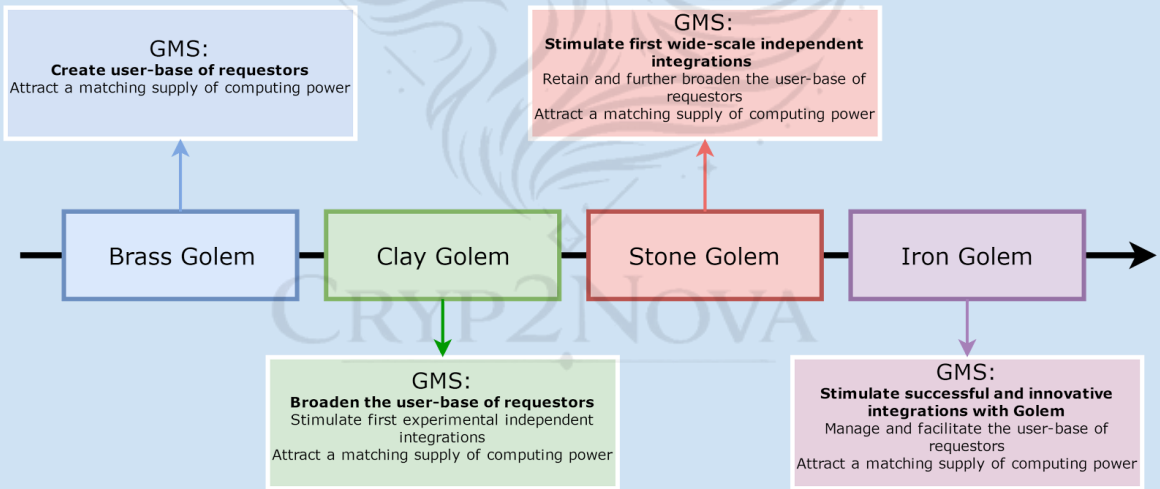
Go-to-Market Strategy

For **every milestone** there is a **corresponding Go-to-Market Strategy** (GMS). Each of them is focused on value propositions developed in respective milestones and specific needs of Golem's ecosystem: software developers, requestors and providers.

- **In the long run, software developers are pivotal elements of the GMS, especially starting from the Stone release onwards**. With Golem architecture being mature enough to allow developers to integrate their own solutions, further growth of the entire ecosystem will depend on innovative and widely-adopted software running on Golem. Apart from creating state-of-the art technical infrastructure for such third-party integrations, the GMS include appropriate marketing activities and incentives.

- **Requestors are the main focus of the GMS in Brass and Clay releases**. The efforts will include facilitating integrations of widely-needed and commercially viable use cases, combined with dedicated marketing activities and eradicating barriers for potential requestors to join the network.

- **Providers are most likely to respond to economic incentives offered by Golem and simply join the network in order to earn tokens in exchange for their computing resources**. Nevertheless, the GMS assume communication with this target group, so that at every stage the demand is matched with sufficient supply.



Note: Each of the GMS assumes the full level of funding (i.e. the crowdfunding cap is reached). In other scenarios, the GMS will be adjusted to deliverables of a particular funding level.

Archived copy · cryp2nova.com

Brass Golem

...these are created to fulfill one goal, set at the time of their creation, and wait with absolute patience until activated to perform this task.

Brass Golem is where we are at the moment with our proof-of-concept, in alpha testing now. This current version of Golem is only focused on rendering in Blender and LuxRender, and although it will be useful to CGI artists, we consider CGI rendering to be a proof of concept primarily, and also a training ground. Brass Golem should be frozen within 6 months after end of crowdfunding period and a full battery of tests. Even though we do not expect that Blender CGI rendering will create enough turnover to justify all the work we have put into the project, this will be the first decentralised compute market.

List of proposed functionalities:

- (1) Basic Task Definition Scheme that allows to prepare first task definition;
- (1) Basic Application Registry - first version of Ethereum-based Application Registry which allows to save tasks defined with basic task definition scheme;
- (1) IPFS integration for coordinating task data and content delivery, e.g. deliver files needed to compute a task, deliver the results back to the requester;
- (1) Docker environment with Golem-provided images for sandboxing the computations;
- (1) Local verification: a probabilistic verification system based on the calculation of a fraction of the task on the requestor's machine;
- (1) Basic UI and CLI;
- (1) Basic reputation system;
- (1) Implementation of Blender and LuxRender tasks.

Brass Golem: Go-to-Market Strategy

Objectives:

1. Create user-base of requestors by attracting customers interested in applying Golem in CGI rendering using supported open-source software. [target group: **requestors**]
2. Attract a matching supply of computing power. [target group: **providers**]

Value proposition:

- CGI rendering solution based on LuxRenderer/Blender, which is cheaper, more reliable and offer user experience superior to existing cloud-based solutions, notably render farms.

requestors	providers	software developers
CGI artists and other users of open source rendering software: ● Targeted marketing: artists' portals, social media etc. ● Incentives for initial purchases of computing resources in Golem (subsidies, vouchers, etc.).	First pool of providers: Golem community, alpha/beta testers and first requestors (prosumer model). Targeted marketing: owners of data centers, mining rigs etc. (if needed) Broader marketing aimed	Software developers not targeted in this release.

Archived copy · cryp2nova.com

• Referral schemes.	at users of high-end PCs.	

Clay Golem

There is a chance (...) that a Clay Golem will be possessed by a chaotic evil spirit. If this happens control is lost and the Golem attacks the closest living creature.

Clay Golem is a big leap from the Brass milestone. Clay milestone introduces the Task API and the Application Registry, which together are going to make Golem a multi-purpose, generalized distributed computation solution. Developers now have the means to integrate with Golem. This advance, however, may come at the cost of compromised stability and security, so this version should be considered an experiment for early adopters and tech enthusiasts. Prototype your new ideas and solutions on Clay. Clay Golem should be delivered within 15 months after end of crowdfunding period.

List of proposed functionalities:

- (1) Basic Task API: an interface that allows a user to define simple tasks;
- (1) Initial Transaction Framework Model with hard-coded payments schemes;
- (1) Redundant verification: a verification scheme based on the comparison of redundant computation results;
- (1) Basic subtask delegation: a mechanism for more fined grained subtasks distribution (e.g. can be used to help with creation of an ad-hoc proxy delegating tasks in a more efficient manner);
- (1) Basic tutorials for software developers;
- (2) Support for virtual machines as a sandbox for computation;
- (2) Set of extended tutorials for developers explaining how to implement their own tasks for Golem network;
- (2) Example computational chemistry use case implementation;
- (3) Example machine learning use case implementation.

Clay Golem: Go-to-Market Strategy

Objectives:

1. Broaden the user-base of requestors by extending the rendering use case (integration of commercial rendering engines) and implementing a new use cases (computational chemistry, machine learning). [target group: **requestors**]
2. Stimulate first experimental independent integrations with Golem. [target group: **developers**]
3. Attract and retain a matching supply of computing power. [target group: **providers**]

Value proposition:

- CGI rendering use case extended to popular commercial software solutions.
- Additional use cases implemented by Golem team (computational chemistry, machine learning).

Archived copy · cryp2nova.com

- A basic rudimentary platform for software developers to implement first experimental integrations with Golem.

requestors	providers	software developers
CGI artists and other users of open source and commercial rendering software: • Targeted marketing: artists' portals, social media etc. • Economic incentives for initial purchases of computing resources in Golem (subsidies, vouchers, ect.). • Referral schemes. Similar means applied to attracting users for the computational chemistry and machine learning use cases.	Targeted marketing: owners of data centers, mining rigs etc. Broader marketing aimed at users of high-end PCs.	Tutorials for software developers (experimental integrations with Golem API). Small-scale communication with different communities of developers.

Stone Golem

Stone Golems do not revoke their creators control like (...) Clay Golems.

Stone Golem will add more security and stability, but also enhance the functionalities implemented in Clay. An advanced version of the Task API will be introduced. The Application Registry will be complemented by the Certification Mechanism that will create a community-driven trust network for applications. Also, the Transaction Framework will create an environment that will allow Golem to be used in a SaaS model. Clay Golem should be delivered within 24 months after end of crowdfunding period.

List of proposed functionalities:

- (1) Full Task API: an interface that allows users to define tasks;
- (1) Application Registry: where developers publish applications ready to run on Golem;
- (1) Transaction Framework that allows a choice of remuneration models for task templates;
- (1) Basic Certification support for Software: A mechanism that allows users to whitelist and blacklist applications, building a decentralized trust network;
- (1) Support for SaaS: the possibility to add support for proprietary software which can be used in tasks. Payments for task creators should also be implemented in the application;
- (1) Application Registry and Transaction Framework tutorials for developers;

Archived copy · cryp2nova.com

- (2) SaaS tasks examples - example use cases that shows developers how to create tasks available in SaaS model;

Stone Golem: Go-to-Market Strategy

Objectives:

1. Stimulate first wide-scale independent integrations with Golem. [target group: **developers**]
2. Retain and further broaden the user-base of requestors by facilitating marketing efforts of independent requestors and continuing promotion of existing use cases (rendering, computational chemistry, machine learning). [target group: **requestors**]
3. Attract and retain a matching supply of computing power. [target group: **providers**]

Value proposition:

- A much more advanced platform for developers, including a fully functional API, Application Registry and Transaction Framework as well as a basic version of the Certification mechanism and support for SaaS.

requestors	providers	software developers
Customer retention and further acquisition: continuous development and marketing of rendering, computational chemistry, machine learning use cases. First wide-scale third-party integrations: independent marketing attracting additional groups of requestors, however Golem team will facilitate and coordinate these marketing efforts.	Targeted marketing: owners of data centers, mining rigs etc. Broader marketing aimed at users of high-end PCs.	Tutorials for software developers (Application Registry, Transaction Framework). Demonstration of SaaS implementations in Golem. Economic incentives for the most successful and innovative third-party integrations.

Iron Golem

Iron Golems are made of iron and are among the strongest type of Golem. They never revoke the control of the wizard that created them.

Iron is a deeply tested Golem that gives more freedom to developers, allowing them to create applications that use an Internet connection or applications that run outside the sandbox. Of course, the decision to accept higher-risk applications will still belong to the providers renting their compute power. Iron Golem should be robust, highly resistant to attacks, stable and scalable. Iron will also introduce various tools for developers that will make application creation far easier. Finally, the Golem Standard Library will be implemented. Assuming that maximum financing will be reached, Iron Golem should be delivered within 48 months after end of crowdfunding period.

List of proposed functionalities:

Archived copy · cryp2nova.com

- (2) External data link: enables Golem to use resources and interface with software outside of the Golem network;
- (2) Host-direct mode: a trusted mode for explicitly whitelisted applications or invulnerable environments, where Golem runs computation outside the Docker/VM;
- (2) Certification support for Environments;
- (2) Network Status Dashboard - public website displaying basic stats about Golem network;
- (2) Additional security mechanism - tasks that uses public data link or host-direct mode are particularly challenging for security. Additional means may be necessary to make running those tasks safer for providers (eg. central audit oracles, agreements contracts or code-execution observers may be implemented);
- (2) Golem web client: a web interface for Golem nodes as an alternative to the native GUI / console interface;
- (3) Golem Developer Toolkit: a set of diagnostic and test tools to make creation process of applications for Golem even easier;
- (3) Reputation-system: reputation protocol that allows the node to effectively supervise network behaviour;
- (3) Advanced transaction system: a system that automatically tries to match requestors with providers in a way that is most profitable to all participants;
- (3) Golem Developer Toolkit tutorial
- (3) Provider dashboard - providing stats, graphs and more advance settings management for providers;
- (4) devp2p integration - changes in p2p and network protocols using new version of devp2p;
- (4) MapReduce and topological sorting of tasks: add the next abstraction layer, allowing users to define more generic tasks that are interdependent;
- (4) Golem Standard Library (Golem STD): language agnostic functionality providing access to the low level core components required to interact with Golem from within a programming language. Special attention will be paid to I/O functions exposed to tasks and subtasks related functionalities. Each supported programming language will have bindings to Golem STD. These bindings will serve as a means of extending the default standard library of the language in question (custom extensions provided by developers of programming languages will also be possible). With Golem STD an automatic task definition, independent from the operating system, will be possible. Golem STD will allow users to create Golem applications using different programming languages, which shall significantly increase the number of potential use cases and simplify task creation process.
- (4) Golem STD tutorial for developers

Archived copy · cryp2nova.com

Iron Golem: Go-to-Market Strategy

Objectives:

1. Stimulate successful and innovative integrations with Golem. [target group: **developers**]
2. Manage and facilitate the user-base of requestor. [**providers**]
3. Attract and retain a matching supply of computing power. [target group: **providers**]

Value proposition:

- A fully-functional for developers, including the API, Application Registry, Transaction Framework, support for SaaS, Golem Development Toolkit, devp2p as well as an array of functionalities to interact with Golem from with a programming language (Golem Standard Library).

requestors	providers	software developers
Customer retention and further acquisition: continuous development and marketing of existing use cases. Marketing of the entire Golem network, coordination of marketing efforts of independent software developers.	Targeted marketing: owners of data centers, mining rigs etc. Broader marketing aimed at users of high-end PCs.	Tutorials for software developers (Golem Development Toolkit, devp2p, Golem Standard Library). Targeted marketing: software developers from outside of the crypto-world (in order to attract ideas with the largest potential for wide-scale adoption).

Future integrations

There are numerous Ethereum dapps and future platforms currently under development or in alpha release. We see great opportunities in this environment, not to mention solutions that could potentially be used as a part of Golem's ecosystem, either directly or as extensions. The following systems will be considered for integration and their implementation will be dependent upon the release of production code and complexity of integration:

- Payment channel solutions based on P2P routing and transactions, eg. Raiden or multi-party payment channels;
- External decentralized identity services, e.g. uPort;
- External solutions for task verification or reputation, eg. TrueBit;
- External solutions for storage, eg. FileCoin, Swarm.

Archived copy · cryp2nova.com

Crowdfunding

The crowdfunding of Golem and the corresponding token creation process are organised around smart contracts running on Ethereum. Participants willing to support development of the Golem Project can do so by sending ether to the designated address. By doing so they create Golem Network Tokens (GNT) at the rate of 1 000 GNT per 1 ETH.

A participant must send ether to the account after the start of the crowdfunding period (specified as the block number). Crowdfunding ends when the end block is created, or when the amount of ether sent to the account reaches the maximum.

Crowdfunding summary

GNT created per 1 ether	1 000 GNT
Minimum ether	150 000 ETH
Maximum ether	820 000 ETH
% of tokens generated to Golem team	6%
% of tokens generated to Golem Factory GmbH	12%
Approximate date of start (StartBlock)	11th November 2016, 3:00 pm GMT
Approximate date of end (EndBlock)	2nd December 2016, 3:00 pm GMT
Maximum number of GNT generated	1 000 000 000 GNT
of which crowdfunding participants	820 000 000 GNT
of which Golem team and Golem Factory GmbH	180 000 000 GNT

The crowdfunding address will be announced at the crowdfunding start through the following channels:

- Project webpage: golem.network
- Official Twitter: twitter.com/golemproject
- Official Slack: golemproject.slack.com (you can join here)
- Official Blog: blog.golemproject.net
- Reddit: reddit.com/r/golemproject

Please, double-check the address before sending ETH. For security reasons, we advise to confirm the address using at least two different sources above.

Archived copy · cryp2nova.com

On the project webpage, you will also find a detailed guide on how to participate in the crowdfunding using popular Ethereum wallets.

Crowdfunding is implemented as a smart contract with a few simple parameters:

- Golem Factory GmbH: controls the contract and the address to which gathered ether will be sent (implemented as a multisig address);
- Percent of pre-allocated tokens is 18% (6% - Golem team, 12% - Golem Factory GmbH);
- StartBlock, EndBlock: these block numbers indicate the start and the end of the crowdfunding process;
- minTarget: minimum cap for this crowdfunding, crowdfunding fails if not reached;
- maxCap: maximum cap for this crowdfunding, denominated in GNT;
- GNT creation rate, denominated in ETH.

The crowdfunding contract conforms to a few important rules:

- Before the crowdfunding starts, no ether can be sent to the crowdfunding contract;
- After the crowdfunding (either maxCap was reached or the crowdfunding deadline passed), no ether can be sent to the contract;
- During the crowdfunding, participants simply send ether to the crowdfunding contract which results in GNT creation;
- All created tokens are **non-transferable** during the crowdfunding;
- Only after the crowdfunding period has ended:
    - Any user can initiate the transfer of ether to the specified address of Golem Factory GmbH;
    - The crowdfunding contract creates an 18% endowment of tokens such that crowdfunding participants' tokens constitute 82% of supply, regardless of the level of funding;
    - The crowdfunding contract finalizes funding which results in an allocation of founders' tokens and unlocking the created GNT.

If minimum financing is not reached, then after the crowdfunding period, participants can claim their ethers back from the contract.

Crowdfunding process leads to creation of GNT, a backbone token for the Golem network. GNT implementation follows widely adopted token implementation standards with two additional functionalities core to the crowdfunding process and future upgrades, namely, token creation and token migration:

- Create token - during the crowdfunding process, the crowdfunding contract can issue new GNT (based on the amount of sent ETH).
    - By default, created GNT is locked (nontransferable). Only when crowdfunding is finalized are tokens unlocked and participants able to transfer them.
    - The creation function is enabled only during the crowdfunding phase; it does not allow the creation of tokens after the crowdfunding phase is over (token supply is constant ever after).
- Migrate token - a function which implements GNT migration to another contract.

Archived copy · cryp2nova.com

- Does nothing by default, but if at some point a GNT upgrade is required, a separate migration contract can be specified and recommended by Golem Factory GmbH to be used to transfer tokens to the new contract.
- Technically speaking, if a GNT upgrade is required, a contract implementing the *MigrationAgent* interface is created and set by Golem Factory GmbH in the GNT contract (for security reasons this can be done only once). Following that, each GNT holder can decide whether to call *MigrationAgent.migrateTokens* to transfer GNT to the new contract, or not.
- *MigrationAgent* can be implemented only after the new token is implemented and deployed. That's why only an interface is provided right now.

Migration is to be used if it turns out at some point that token upgrade is needed for whatever reason (e.g. changes in Ethereum, or changes in Golem's design). The upgrade will need action from token holders and cannot be imposed by Golem Factory GmbH.

Budget and levels of funding

The ether raised during crowdfunding will be used by Golem Factory GmbH in accordance with the roadmap presented above. Crowdfunding code implies that level of project financing might be anything between minimum financing and the maximum financing (cap). The roadmap is a full vision to be completed if the cap is reached.

Golem should be considered an R&D project involving bleeding-edge technologies. The progress we have already made while working on the Brass Golem alpha proves the validity of our general assumptions presented in this whitepaper, but we are also well aware of the huge amount of work ahead. The commitment of Golem team with respect to the technologies presented in this whitepaper is full, but still ultimately depends on the level of success of the crowdfunding.

In the 'minimum financing' scenario, the ultimate deliverable is a working Stone Golem with functionality enabling the creation of a decentralized market for computing power, as well as a rudimentary toolbox for developers to integrate their own software with Golem. In particular, the minimum financing will be sufficient to introduce a basic version of both the Application Registry and the Transaction Framework.

In the 'maximum financing' scenario, we are making a commitment to deliver Iron Golem, which is not only gunning for total disruption of the market for computing power, but also dives head-first into the development of some important components of Web 3.0. In particular, this level of financing will make it possible to create a flexible platform to distribute and monetize innovative software solutions, notably dapps and microservices. Assuming higher thresholds of funding are reached, the Golem team commits to create integrations useful to the entire community in earlier releases as well to provide software developers with more support in integrating their solutions with Golem and bringing them into the market. The extended marketing effort outlined in the Go-to-Market Strategy is one of the crucial factors behind Golem's ultimate commercial success, but it will not be possible to put it into action without sufficient financial resources.

Archived copy · cryp2nova.com

Functionality vs funding

	Min financing (1)	additional features (2, 3, 4)
Core Functionality	• [B] Basic Task Definition Scheme • [B] Basic Application Registry • [B] Local verification • [C] Basic Task API • [C] Redundant verification • [C] Subtask delegation • [S] Full Task API • [S] Application Registry • [S] Certification support for Software	• [I] Certification support for Environments (2) • [I] External data-link (2) • [I] Host-direct mode (2) • [I] Additional security mechanism (2) • [I] MapReduce (4) • [I] Topological sorting (4)
Reputation & security	• [B] Basic reputation	• [I] Additional security mechanism for host-direct mode (2) • [I] Reputation system (3)
Transaction system	• [C] Initial Transaction Framework Model • [S] Transaction Framework	• [I] Advanced transaction system (3)
Integrations	• [B] IPFS • [B] Docker as sandbox	• [C] Virtual machine as sandbox (2) • [I] devp2p (4)
UX/UI	• [B] Basic GUI and CLI	• [I] Web client (2) • [I] Provider dashboard (3) • [I] Golem Developer Toolkit IDE (3)
Use cases	• [B] CGI Rendering (Blender, LuxRender)	• [C] Computational chemistry use case (2) • [C] Machine-learning use case (3) • [S] Examples of SaaS integration (2)

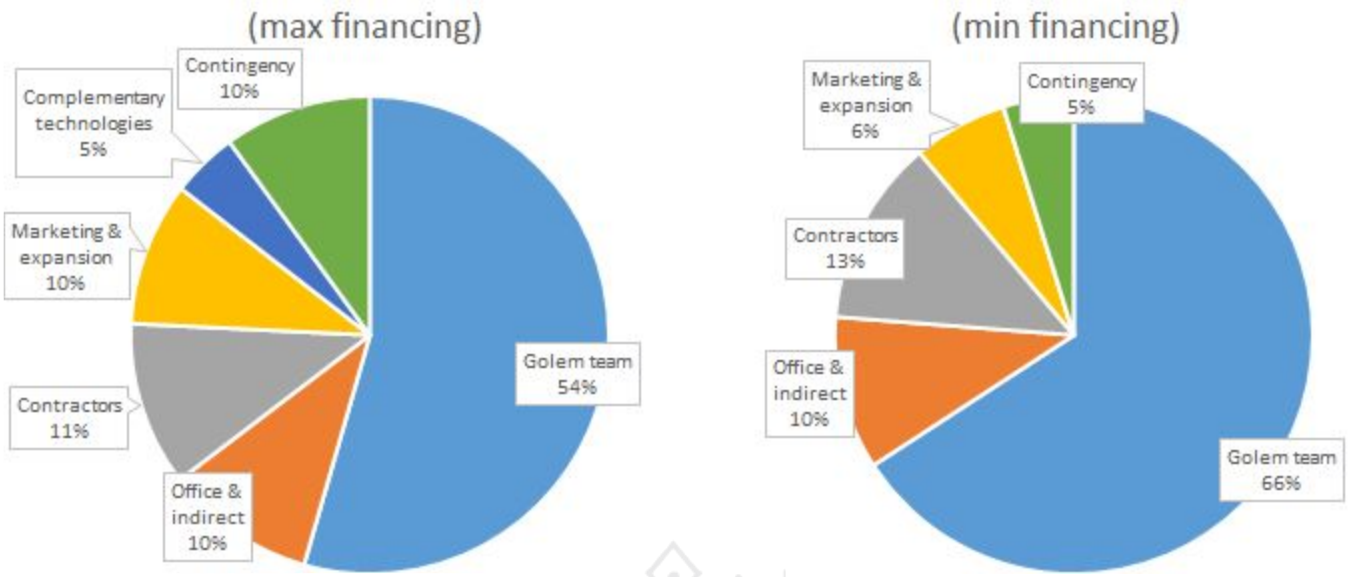
Tools for developers	• [C] Basic tutorial for task developers
	• [S] Application registry and Transaction Framework tutorial
		• [C] Enhance tutorials for task developers (2)
		• [I] Golem Developer Toolkit (3)
		• [I] Golem Developer Toolkit tutorial (3)
		• [I] Golem Standard Library (Golem STD) (4)
		  ○ [I] Golem STD integration with GDT (4)
		  ○ [I] Golem STD support for programming languages (4)
		• [I] Golem STD tutorial (4)

Go-to-Market Strategy	• [B, C, S] Essential elements of the Go-to-Market Strategy	• [B, C, S, I] All elements of the Go-To-Market Strategy focused on value propositions of additional features and broader expansion.

Levels of funding (indicative thresholds):	Milestones:
• (1) 150k ETH • (2) 320k ETH • (3) 530k ETH • (4) 820k ETH	• [B] Brass Golem • [C] Clay Golem • [S] Stone Golem • [I] Iron Golem

CRYP2NOVA

Archived copy · cryp2nova.com



Budget structure for maximum and minimum financing

Golem team consists solely of employment costs. We assume that with maximum financing we will be able to finance team of 20 people (most of them developers) for a period of 4 years.

Office and indirect costs includes costs of offices in both Zug and Warszawa, as well as other indirect, employment-related costs.

Contractors covers all third parties we are willing to work with. The number here is high largely because of security audits (four in case of maximum financing: one for every release). Legal and accounting services are also included in this category.

Marketing, community animation and expansion activities are strictly related to Golem's Go-to-Market Strategy This includes both communication and marketing efforts to get new communities on board, as well as supporting (financing or co-financing) third party integrations with Golem. Activities here will be mostly requestor-oriented, especially in Brass and Clay releases, to ensure that there are a growing number of use cases integrated, with users actively using them on the Golem network.

The **Complementary technologies** category covers expenditures on external technologies Golem is dependent on. This will most likely take the form of financing original efforts to introduce modifications needed by Golem.

Contingency fund is calculated as 10% of the total budget (5% for minimum financing).

Archived copy · cryp2nova.com

Golem Team

Julian (CEO) and Andrzej (COO) have worked together on a variety of consulting projects and business endeavours since 2008. At the time, Julian was developing a consulting branch at IBS, a Polish economic think-tank, where he served as vice president of the board between 2006 and 2012. In 2013, both Julian and Andrzej left IBS to create imapp, a consulting and software development company. As a team, they complement each others' competences very well: Julian might be described as an extraverted leader: passionate , ready to enlist in a Mars no-return-trip at the drop of a hat (if not for his beloved family and children). Andrzej on the other hand is more of the introverted perfectionist, paying attention to every detail and voicing criticisms loud-and-clear if he considers the business to be heading in the wrong direction. Together as imapp, they have proven over the last 10 years that they are able to run a profitable consulting and software development company that delivers a high-quality product.

A brief history of imapp is described in this blog post. Among the numerous consulting and software development projects we have completed over the last years, the following deserve special mention:

- BlackVision is a real-time TV broadcasting rendering engine, used internally by BlackBurst, one of the leading Polish broadcasting production companies. You can see an example of its use here.
  Real-time rendering proficiency was a good starting point for working with the photorealistic CGI rendering that we are implementing in Brass Golem. BlackVision is a work in progress and will become standalone product.
- We work as an IT contractor for GSI on Chematica, which gives us great insight into how computing chemistry is done, why this is relevant to business, and how to perform tremendously demanding computing in a cloud environment.
- Since September 2014, we work for the Ethereum Foundation - now this is mostly Pawel's work on the EVM.
- We are also a contractor to Omise and its Blockchain Lab, which gave us the opportunity to work with (and in some cases, contribute to) many exciting blockchain technologies, including Factom, Hydrachain, and Raiden.

We are not a random collection of people, but a strong team with a proven track record of delivering. Most of us have worked together on many projects over the last couple of years (Andrzej, Julian, Piotr 'Viggith', Aleksandra, Paweł 'chfast', Radek). Alex and Wendell heard about the project long time ago, and were always ready to support us, but only recently have they begun working on the project as part of the team. And, to be able to ensure our delivery, we have enlisted over the last couple of months Marek, Pawel 'pepesza', Magdalena, and Adam. They have become Golem believers as well, fascinated as we are by the work ahead of us.

Archived copy · cryp2nova.com

Managers



Julian Zawistowski, CEO, founder
MA economics (Warsaw School of Economics)
Leader and entrepreneur, willing to change the world for the better. His first attempt to do so was as an economist running a think-tank and policy analysts for government, but he at some point realized that technology is what can really make the difference. Since then, Julian has been running imapp, with an agenda to build software projects which will advance all of us beyond 140 characters. With this catchy agenda, he was able to gather top-class team around him, to complement skills he is definitely missing (such as software development).



Piotr 'Viggith' Janiuk, CTO, co-founder
MSc mathematics, MSc computer science (University of Warsaw)
Piotr is an experienced computer programmer interested in bleeding edge technologies, and keen on bringing innovative, ground breaking (in the field) projects which bring these technologies to life. He worked on a secure P2P communicator when P2P was used primarily for file sharing, implemented highly optimized CUDA code just after the technology appeared, adding it to production-ready compositing software (the first in the world), implemented the fastest-at-the-time software DCP encoder, including a fast software jpeg2000 codec, assisted in optimization of particle renderer for The Witcher 2, and the father of Black Vision (see above).
Piotr implemented core components of Golem prototype and leads the design of Golem protocol and its preliminary use cases.



Andrzej Regulski, COO, co-founder
MA economics (Warsaw School of Economics)
Andrzej is an experienced manager and consultant. He devoted his professional life to running a successful consulting company focused on advising public authorities in their attempts to support innovative enterprises. With this experience, he is aware of most pitfalls, and the "dos" and "don'ts" which tech startups are often facing. He loves managing people who are smarter than him, and is obsessed with increasing the efficiency of organizations beyond the scope of imagination.

Archived copy · cryp2nova.com

Key developers



Aleksandra Skrzypczak, Lead Software Engineer, co-founder
MSc mathematics, BSc computer science (University of Warsaw)
Hungry for new challenges, Aleksandra decided to join imapp more than two years ago, and was the first to work full-time on the Golem Project, doing R&D and generally planning the project. She's not easily disheartened by difficulties, and still knows every line of code in the Golem repository. Prior to joining imapp, Aleksandra was an intern at ICM, building training algorithms for automatic terrain type detection, and for detecting aptamers in DNA and RNA code. She was also an employee of Red Ocean, a software development and deployments group operating under the Ministry of Finance and Custom Service.



Alex Leverington, P2P Network, Adviser
Alex has been involved with Ethereum from a very early stage. He famously worked at ETHDEV, where he architected and coded devp2p, Ethereum's underlying P2P protocol layer. Alex also made key contributions in Ethereum encryption and security, primarily touching the various communications protocols. He remains active in the protocol steering group, and is presently interested in developing a decentralized, secure, and authenticated P2P messaging system, which will be one of essential technologies in Golem.



Paweł 'chfast' Bylica, Lead Ethereum Engineer
MSc computer science (Wroclaw University of Science and Technology)
Paweł is an experienced programmer who specializes in C++, but uses other languages as well, sometimes even designing new ones. Currently he is focused on Ethereum and its Ethereum Virtual Machine (EVM). Pawel is the creator of EVMJIT project, and a still active member of the Ethereum C++ Team. He is also a contributor to the LLVM and C++ Guidelines Support Library.

Archived copy · cryp2nova.com

Developers



Marek Franciszkiewicz, developer
MSc Electronics and telecommunication (Warsaw University of Technology)
Marek touches all of Golem's internals, but mostly works on networking. He was previously the lead back-end developer of a few commercial web services, where he built and handled daemons, and he also left his mark on Temptonik's real-time vocal assessment software. During his university years, Marek worked on orchestration protocols for wireless sensor networks. For his Engineer's thesis, he developed an IP packet mangling engine with a focus on network steganography.



Paweł 'pepesza' Peregud, developer
An experienced programmer, crypto and security enthusiast, and console warrior, Paweł's largest deployment so far was a system handling 1.5 millions of concurrent websockets with 99.997% uptime, since 2012. A gentleman who sometimes can't help to brag about property-testing distributed protocols.



Adam Banasiak, developer
MSc cryptology (Military University of Technology)
This bonafide Linux and security enthusiast joined the Golem team to find new challenges and to work with newest technologies. In the past, he was a developer in Samsung R&D, responsible for the Tizen platform (mobile, wearable and TV profiles). Adam is also the winner of the Polish finals of the Imagine Cup 2013, game category.

Archived copy · cryp2nova.com



Magdalena Stasiewicz, developer

As a student of computer science, astronomy and physics at the University of Warsaw, Magda fit right in, and now has almost one and a half years of experience with imapp during which she was primarily working on Golem. Besides programming, Magda also helps our testers with technical problems of various severity. She loves to learn new things, and one of the main reasons she is excited about Golem, is because there is *tons* of research to be done.



Radosław Zagórowicz, developer

MSc mathematics (University of Warsaw)

With 9 years of an experience as a C++/Java/Python programmer, Radek is head of imapp's IT section, and led development of the Chematica project. Along with Julian and Piotr, Radek is one of the three who came up with the original idea of Golem more than two years ago. After long day and night of discussions he started implementing the first prototype of a Golem client.

Business development and communication



Wendell Davis, BDM

As a technology visionary and founder/entrepreneur both in and out of the blockchain space, Wendell serves an important role in Golem as connective tissue between Ethereum and the larger movement of decentralization.

Some of his past endeavors include Hive, one of the earliest easy-to-use Bitcoin wallets, Vizor, an open source WebVR development platform, Splice, an online music-making/mashup community, and the now-sadly-defunct Kitchensurfing, which connected mobile chefs with buyers of their trade.

Archived copy · cryp2nova.com