SYMBIOSES

A decentralized computing ecosystem

hello@symbioses.io

v1.6.9

Abstract

This paper describes Symbioses, a decentralized ecosystem for monetizing the spare computing resources of any device with a processor. It combines the available resources into accessible Software-as-a-Service (SaaS) solutions available in a pay-as-you-go manner.

Fully compatible with existing cloud solutions and advanced distributed computing technologies, Symbioses empowers connected devices to forge their computing resources into turn-key SaaS solutions, determining their price, negotiating terms, executing agreements and gathering the rewards. Its main pillars are a multi-platform, multi-OS application that takes care of SaaS execution based on the availability of the device; integration with non-interoperable computing networks; a tokenized incentive program for end users to offer their resources to the network; an off-chain, abstraction layer to maximize the response time of the network and minimize the transaction fees; and a decentralized consensus engine over a blockchain payment ledger.

The economic model built into the Symbioses Ecosystem creates an open market for the monetization of valuable assets, e.g., computing power, by hosting readily available SaaS solutions for end consumers. It introduces key incentives and dynamics that not only support the SYM token, but also promote its price and usage, also correlating them with the cryptocurrency market.

By providing a free and automated application, Symbioses aims at lowering the entry point barriers of state-of-the-art blockchain technologies, thus allowing the general public to seamlessly exploit and monetize the spare resources of computing devices.

Note: Symbioses is a work in progress in an area of active research. New versions of this paper will continue to appear at `https://www.symbioses.io`. Please read carefully the Legal section at the end of this document as it contains important information about the project. For comments and suggestions, do not hesitate to contact us at `hello@symbioses.io`.

1 Introduction

In recent years, there has been an explosion of data creation like never before in human history. Several studies have analyzed this trend and current estimates suggest that the total amount of digital data in the world, including things like books, images, e-mails, music, and video, is doubling every two to three years [19] [25].

Big Data is a rather popular term describing the application of new tools and techniques to digital information on a size and scale well beyond what was possible with traditional approaches [25]. Moreover, Big Data tools and techniques have not only delivered value in the social media sector and retail industries [15], but also led to major breakthroughs in a range of other contexts, including scientific research [10] [22] and healthcare [5], among others.

Big Data is not a result of a unique technology, but rather the convergence of several innovations and ideas working in a highly complementary way. Two of these technological developments are particularly significant:

- A decline in data storage costs. The cost of storing digital information has been exponentially falling for a long time. It is no longer the case for data to be discarded when their initial use had passed, as the economic expense associated with archiving the data afterwards has dropped considerably [23].

- The development of software platforms that make it possible to process large data sets in structured way. Such developments have significantly helped Big Data practitioners see through the messiness of these new data sets to find useful information and relationships.

This last point is noteworthy, since it is usually neglected: data have no value without the relevant software and required computing power to sustain their analysis. Imagine a pile of books waiting for someone to read them: books are worthless without the reader's brain power to interpret them. In the context of Big-Data analysis, computing power is used to understand the stored data and ultimately making better decisions.

1.1 Distributed computing

Computing power provides the means for dealing with activities involving data sets that are so large and complex that they require advanced data storage, management, analysis, and visualization technologies [5]. Whereas traditional data sets have historically needed to be fairly structured and static, digital information in the era of Big Data and Deep Learning is frequently noisy, messy, unstructured, and dynamic [8]. However, considerable progress has been made in representing massive data sets as networks of nodes and edges, getting structured information out of unstructured data so that machine-learning algorithms can be applied to them.

Google MapReduce and Apache Hadoop probably represent the most prominent examples of modern software applied to big-data analysis. They make it possible to break large data sets into smaller chunks that can be delegated to several computing devices. The results of the calculations arising from each of the smaller chunks can then be reintegrated at the end of the process. This divide-and-conquer approach is not only driven by the large amount of data being analyzed, but also by the smoothing of the continued growth in the processing speeds of computing devices: Moore's law is coming to an end [21]. After decades of a constant increase rate, the number of transistors that can be printed on a surface unit is reaching its limit due to diverse physical phenomena.

Since single processing units are not getting any faster, the same workload is undertaken by a number of them, each of which takes care of solving a piece of a puzzle. The technique of tackling problems requiring large amounts of resources (mainly computing power and memory) by dividing them in smaller problems and delegating to several computing devices is known as distributed computing [35]. Distributed (parallel) computing is playing an increasingly important role in the way big problems are dealt with. This approach, which is also prominent in other areas of science, technology, engineering and mathematics, frequently uses cloud computing infrastructure as a platform for transferring these data chunks to different computing devices, and finally bringing back the results.

With blockchain technology reaching further than the crypto-currency world, a new paradigm of distributed computing is emerging for solving large-scale problems [13] [16] [34]. Built on top of the Ethereum blockchain and smart contracts, these platforms promise to enable the decentralized connection of computing entities for sharing and aggregating resources geographically distributed across different countries, organizations and administrative domains.

It is clear that decentralized computing will be playing a key role in the way computing resources are accessed and made available to end users and enterprises. For now, however, it has been available to only a selected group of technology-knowledgeable users and members of tech-savvy communities. Symbioses is going to change that by bringing key technological advances together in an easy-to-use and seamless way, in order for the general public to play the main role in this area.

1.2 Mission Statement

Symbioses aims at further extending the decentralized computing segment by delivering readily available Software-as-a-Service (SaaS) solutions accessible by anyone: like Uber changed taxis and Airbnb changed room renting. Symbioses provides the general public with a one-size-fits-all tool for real-time monetization of existing devices, the available resources of which are shared by hosting turn-key SaaS solutions. The term *general public* is defined as end users who do not have a deep understanding about technology in general and blockchain in particular. For this reason, the main goal is to deliver a simple and automated application for the general public to quickly and seamlessly start earning their rewards in exchange for the computing resources they share. The Symbioses mission is to allow such users to capitalize on the unused resources of every computing device, whether it is a server farm, a laptop, a smartphone or an Internet-of-Things (IoT) device, by delivering SaaS solutions to home users and enterprises. Considering there are millions of underutilized devices that can offer their computing resources to the Symbioses Ecosystem, the potential is enormous. Symbioses has the technology to accomplish this feature.

1.3 Vision

The Symbioses vision is to unbundle and democratize the traditional cloud market. The creation of a *public cloud* to offer SaaS solutions in a scalable and secure way at competitive prices compared to centralized services currently run by a few big players. This will allow anyone instant access to the computing resources required to work, create and research, giving them a central role in the monetization of their devices by hosting SaaS solutions from a blockchain-driven network. These networks guarantee the highest level of transparency, resilience and security. Symbioses provides a complete Ecosystem that lowers the entry barriers of state-of-the-art blockchain technologies, democratizing this new era of technology, making it accessible to the general public.

2 Motivation

2.1 Background

Traditional methods of distributed computing can roughly be divided into two groups: grid computing [9] and cloud computing [1].

Over a decade ago, grid computing became a real alternative to traditional supercomputing environments for developing parallel applications that harness massive computational resources. Indeed, the term grid computing suggests protocols offer shared computation and storage over long distances, but those protocols did not lead to the use of generalized software environments. Mainly due the complexity incurred in building such parallel grid-aware applications, it never grew beyond its core community and certainly did not replace high-performance computing (HPC) environments [18].

Cloud computing, on the other hand, provides a simpler environment where the user can operate a virtual rack, with reduced capital expense, and the ability to scale up as demand requires, as well as to support brief surges in capacity [29]. Some authors claim that the construction and operation of extremely large-scale, commodity-computer data centers at low-cost locations was the necessary enabler of cloud computing [1].

The Symbioses Ecosystem delivers tools and services built on top of key decentralized technologies that provide these and other services of grid, HPC and cloud computing, while seamlessly allowing the general public to monetize the resources of commodity devices.

2.2 Cloud computing limitations

Cloud computing still has some disadvantages and has not succeeded in taking over the high demand of the computing market. Since centralized cloud services are remote, they suffer from latency and bandwidth-related issues associated with any remote application. In order to mitigate such effects, only the largest centralized cloud providers have deployed data centers in specific locations around the world, so that the latency is minimized and the user's perception (or QoS) is improved (see Figure 1). Clearly, the high costs in terms of human capital and servers means that only the largest corporations can afford such deployments, leading to more service centralization and a higher prices for customers.

End users and small businesses often lack the expertise to operate cloud-based platforms. Also, many scientific projects still rely on grid computing for developing their science forward since the pricing models of centralized cloud computing providers are prohibitive. Indeed, it has even been argued that some provider holds a monopolistic position in the cloud market [27], i.e., a single service provider dominates the market and is therefore the only provider of a particular service. According to economic theory [33] users cannot influence the prices of services and have to choose the service at the price given by the provider who monopolizes the marketplace.

By providing a decentralized public cloud, the Symbioses Ecosystem has the potential to unwrap this situation and to drastically lower the cost of infrastructure usage, while making it readily available on demand.

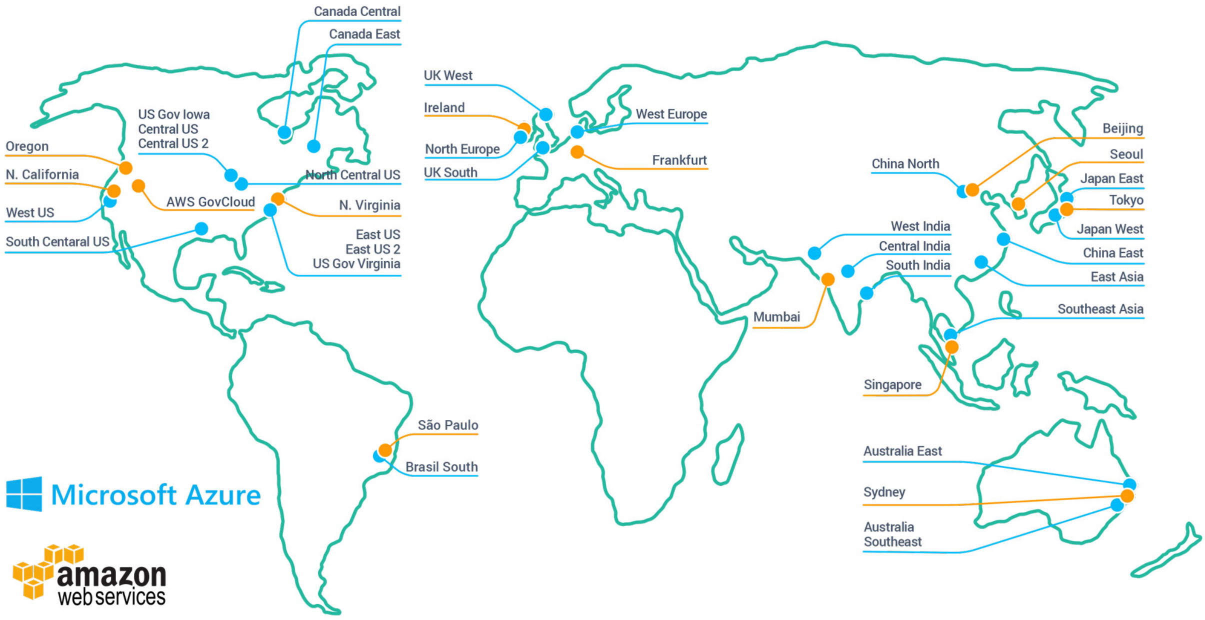

Figure 1: Map of the Azure and AWS regional offerings as of 2016 [6].

2.3 The latency issue

It is well known that the data-transport latency is the Achilles' heel of distributed and HPC applications [14]. Indeed, recent paradigms are based on services located closer to the end user in order to improve latency concerns and data access, e.g., edge computing. That is, instead of storing information in sites far away from the end user (e.g., traditional cloud computing), these approaches ensure direct proximity of the data to the customer in order to support low-latency and scalability [3].

However, as of late 2017 existing decentralized-computing efforts do not focus on bridging the gap with the data: they consistently fail in bringing the computing power and the data closer together, or rely on a third-party service to do so.

The Symbioses Ecosystem provides a coordination layer on top of decentralized computing resources, thus regulating its behaviour to minimize the latency impact between compute and data. In other words, it makes sure the requested SaaS instances are served by the Producers with the minimum available latency. This also avoids the potential increased fragmentation to data allocation introduced by storage services working in an algorithmic-market fashion.

2.4 Blockchains: compatibility, scalability and transactions

Almost all existing and emerging decentralized technologies for distributed computing rely on Ethereum smart contracts.

Ethereum smart contracts allow programmers to write code that is executed on the blockchain virtual machine. Despite the many unique use cases this technology can enable, blockchains offer very limited computing capacities to run decentralized applications: few kilobytes of storage, very inefficient virtual machine and a very high latency protocol [16]. Moreover, the code included in a smart contract has to be executed on all the nodes of the network before being accepted, creating a potentially harmful delay to the dynamic market of on-demand computing services. Just imagine waiting several hours before a Producer is found to run your application of choice, not because of availability, but due to the smart-contract execution being delayed.

Another emerging issue is the lack of interoperability between different blockchains in general. Indeed, Symbioses connects technologies that are not interoperable with each other, bringing down yet another hurdle in the way of massive adoption of decentralized technologies (see Section 6). For this reason, the core component of the Symbioses Ecosystem is an agnostic, off-chain protocol that encapsulates its functional logic in distinctive operational modules. This abstracts away from potential scalability issues by keeping the execution of time-critical segments off-chain (see Section 5), only registering the finalized transactions on-chain in an asynchronous way.

Finally, the transaction prices have dramatically increased in the last few months, again pushing people away from daily use of blockchain-powered solutions. This is particularly problematic if compared with products like Apple Pay or Facebook Messenger, which enable their users to make payments and transfer money to their friends with no transaction costs. The Symbioses Ecosystem aims at minimizing the impact on-chain transaction fees have on the final price payed for a SaaS solution, as explained later in Section 4.5.

2.5 Blockchains and the general public

The awareness of blockchain already reached most social layers in developed countries, mainly driven by the Bitcoin price euphoria of late 2017 [24]. Symbioses will focus on closing the gap so that this technology is truly available to anyone, without requiring advanced knowledge of IT or blockchain. We feel the blockchain technology is mature enough for the general public to take advantage of it in an easy and simple-to-use way.

Despite its simplicity, Symbioses emphasizes the absolute transparency towards the end users. By providing detailed access to all created transactions, end users can verify that the received rewards and payed fees precisely match a particular series of events. This provides a validation mechanism for everyone to employ, giving transparency the place it deserves in an open economy.

3 A decentralized public cloud

The Symbioses public cloud is a complete ecosystem for accessing SaaS solutions on demand and in a decentralized way. By employing an easy-to-use, thin application, the general public can monetize the computing resources of any device with a processor.

Symbioses makes SaaS solutions available in a pay-as-you-go manner, making it functionally equivalent to what it is known as a public cloud. Figure 2 shows an overview of the elements and actions taken in order for the Symbioses Ecosystem to be accessed in such a way.

Additionally, Symbioses is available for the general public seeking a reward in exchange for sharing their underutilized devices through the network. Moreover, small and medium-sized data centers may also provide their computing resources to the Symbioses Ecosystem and redeem their infrastructure in case of having available resources. This principle holds even for underutilized resources on cloud instances of traditional cloud providers like Azure or Amazon.

Figure 3 shows the role of Symbioses when used as a platform for sharing and monetizing computing resources.

The notion of SaaS [12] is central to the Symbioses Ecosystem, since it delivers readily-available solutions for end users and enterprises using the available resources provided by the users themselves. An alternative to the standard model where a user has to build the server, install the application and configure it, SaaS does not require any technological knowledge other than operating the chosen software itself. Also, the user does not own a software, but leases it instead. They have the authorization to use it for a period of time and pay just for the SaaS that they are using.

Users of the Symbioses Ecosystem may have two roles: Consumers or Producers.

Consumers are end users and organizations seeking for reliable SaaS solutions.

Producers are end users and organizations offering the resources of their devices to the network.

In some cases, the same user can play both roles. For instance, Consumers may reach out to Symbioses in case of an exceptional event (project deadline, peak of users, etc), and regularly share their underutilized devices to the Ecosystem otherwise, e.g., outside business hours.

From a Producers' point of view, this means that sharing their resources on Symbioses allows them to keep using these devices as before, since the SaaS instances they host do not interfere with their on-going work or data: it is all managed by the Symbioses Ecosystem.

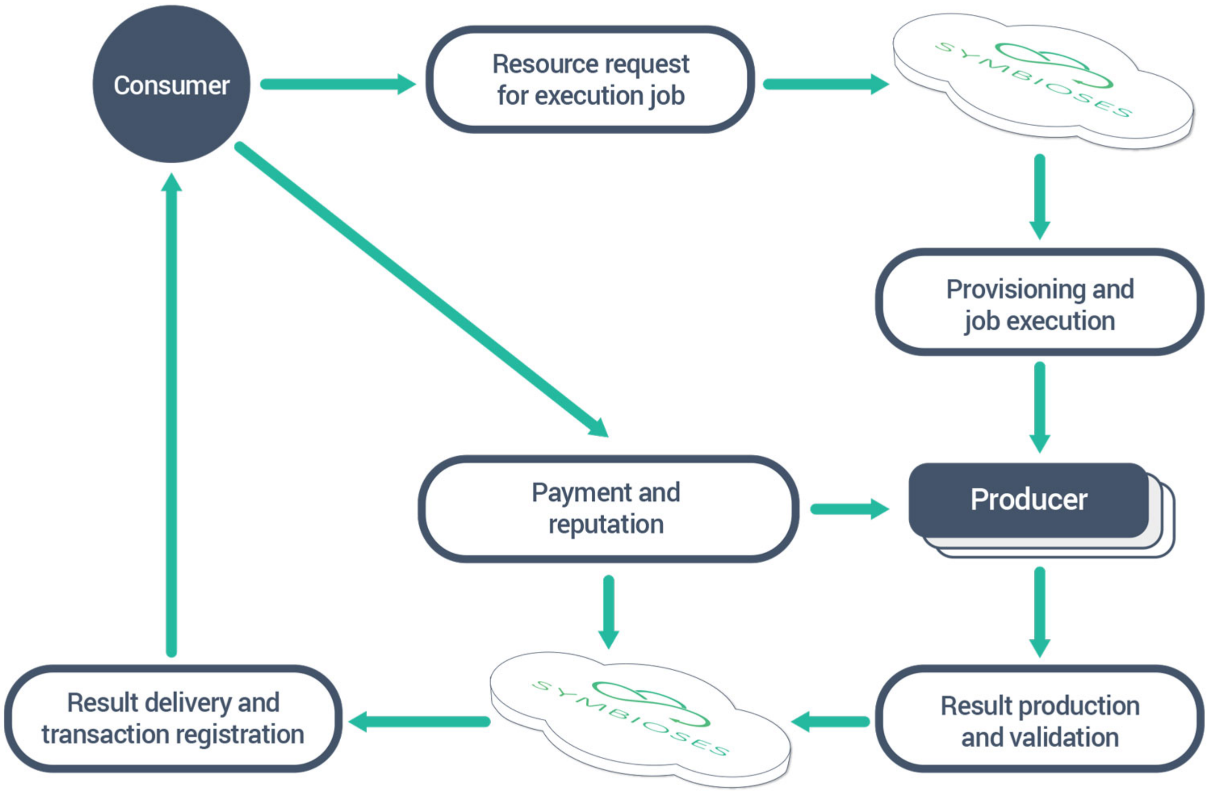

Figure 2: The Symbioses Ecosystem as a public cloud.

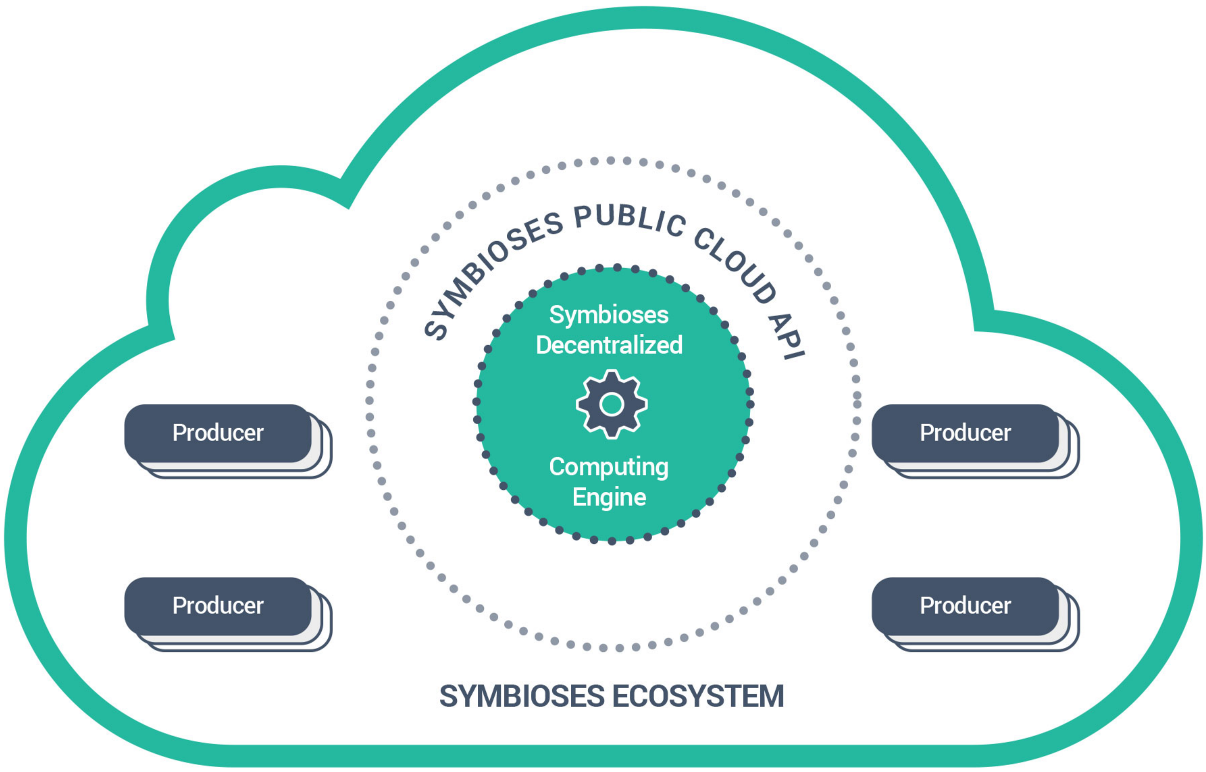

Figure 3: The Symbioses Ecosystem as a monetization platform for Producers' devices.

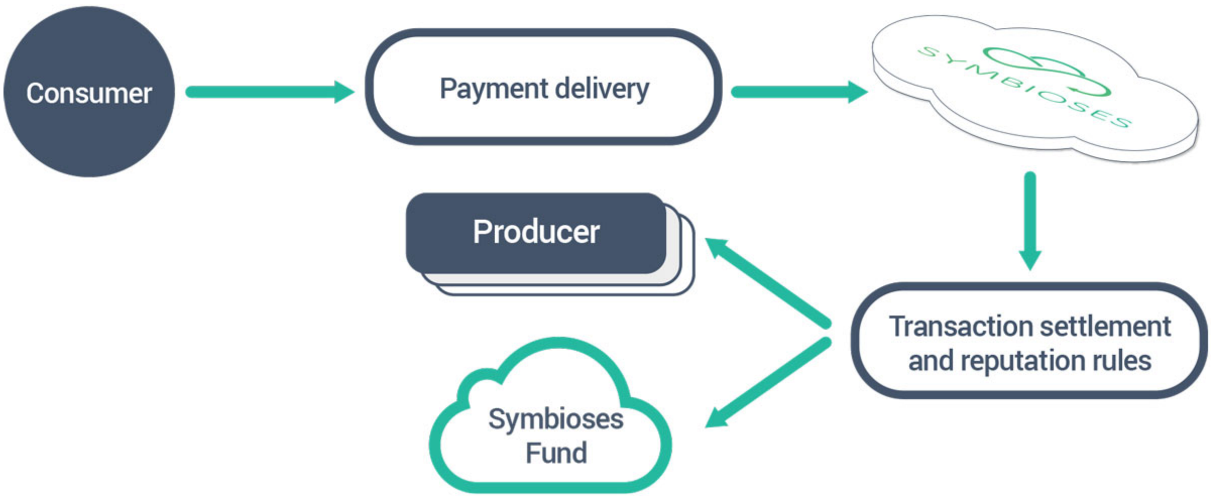

Figure 4: SYM token distribution within the Symbioses Ecosystem.

3.1 The SYM token

Symbioses uses its own ERC20 token[1], providing value to all involved actors in the Ecosystem. Section 7 provides details about the economic model and how the SYM token will increase its value with the functioning of the Symbioses Ecosystem.

The creation of the SYM tokens will occur during the Token Creation event, so the total supply of SYM tokens is fixed.

SYM tokens are the exchange unit within the Symbioses Ecosystem. Consumers will be charged fees for their usage of different SaaS solutions. As Figure 4 shows, the fees are distributed among the Producers providing the computing resources to host those solutions. A small percentage is dedicated to supporting the continuous development of existing and new SaaS solutions, i.e., the Symbioses Ecosystem fund. However, this fee is not fixed and it will remain flexible in the future in order to protect the service Symbioses provides to its user. This is directly related to the fundamental goal of targeting the general public: the services Symbioses offers are designed to reach a large number of users, and affordable prices are key in this regard.

All payments will be denominated in SYM tokens. Assuming the on-going discussions with exchange services to include the SYM token will be favorable, this should allow Consumers to purchase services and redeem tokens in their currency of choice.

[1] The team is also considering using other underlying blockchain providing faster and cheaper block creation.

4 Architecture

As part of the so called Web 3.0, blockchain technology already provides the foundation for a myriad of products and services. However, blockchain as a technology is still relatively new and keeps evolving at a high pace. It is this evolution what will help the blockchain community understand the advantages and drawbacks of several of the existing approaches (see Section 2.4 for a discussion about some of these issues). Thus, it is clear that the blockchain technology available today will go through several radical changes in the following years before converging to a more mature set of established principles and tools.

Within this context, the modular architecture of the Symbioses Ecosystem, including its interface to external systems, is the critical pillar that will hold the effort up within this constantly evolving environment. Such modular approach features modern and well known software-engineering principles, the advantages of which are summarized below [2]:

- Interface stability.
- Data, workflow and user-interface integration.
- Security and reliability concerns.
- Adding new functionality.
- Evolving while slimming the platform down.

The alternative of pursuing a monolithic approach would instead cause serious mistakes from a software architecture perspective, potentially leading to abandonment among the users due to its restrictive nature.

4.1 A modular architecture

Figure 5 shows a diagram depicting the different modules (layers) in the architecture.

The essence of the Symbioses Ecosystem is a scalable Decentralized Cloud Engine (DCE), the core of which is powered by a decentralized and encrypted protocol (see Section 5 for an in-depth discussion). On its edges, a set of well-defined layers provide Application Programming Interfaces (APIs) which allow a clean interaction with external protocols and systems. The Symbioses Public Cloud API provides a common dialect to communicate with external and potentially incompatible decentralized networks, the communication to which is encapsulated in individual Clients. On the lower end, the Symbioses Transaction Settlement API takes care of creating and signing the resulting transaction in the relevant blockchain.

Undoubtedly, this clear separation of concerns enables the continuous adaptation of the interfaces used for interacting with external systems without introducing disrupting changes within the Symbioses Ecosystem itself. The advantages of this design is many fold. First, it ensures Symbioses will stay compliant and up-to-date with the latest and more

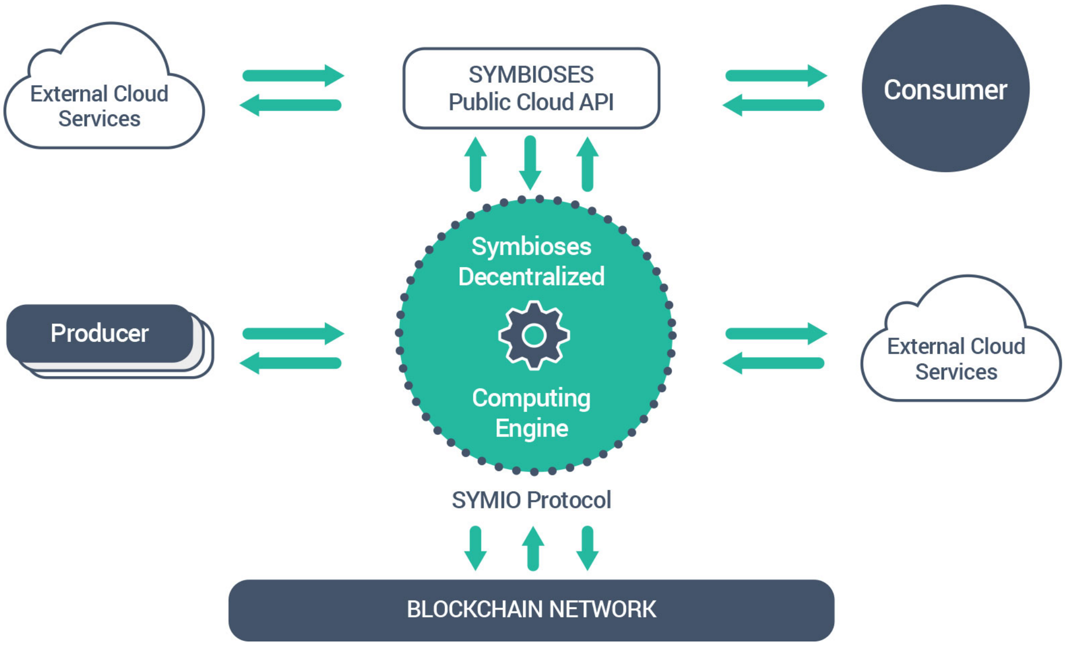

Figure 5: The different components of the Symbioses Ecosystem architecture.

relevant blockchain technologies in general and decentralized computing in particular. Second, by abstracting the Symbioses DCE, cross-protocol compatibility can be achieved by following off-chain calculation principles featured in other renowned projects [31]. Third, the encrypted proprietary protocol adds an additional security layer to the Symbioses Ecosystem without compromising transparency, since all processed transactions are ultimately registered in the corresponding blockchain for the users to inspect and review. Fourth, the scalability of the Ecosystem is independent from the rate at which the transactions can be created on a given blockchain: this protects the dynamics of the Symbioses DCE from the performance fluctuation of certain blockchain protocols.

4.2 Symbioses Client Application

A device becomes part of the Symbioses Ecosystem once the Client application is installed on it (see Figure 6).

The Client Application is very lightweight as it interfaces with the rest of the Symbioses network and maintains very little state of its own. Moreover, this is the only requirement for the general public to turn any device into a Producer of computing resources for the Symbioses Ecosystem. Anyone will be able to start earning right after installing the Client Application in their device of choice.

The Client Application is a process running in the Producer's device which:

- takes care of communications between the Producer's device and the rest of the Symbioses Ecosystem,
- deploys the Consumer's SaaS and data before their execution,

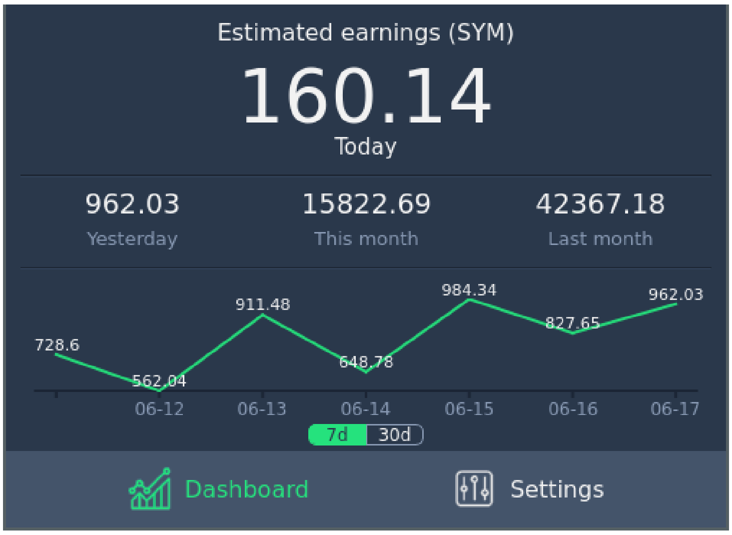

Figure 6: A running instance of the Symbioses Client v0.1.0b11.

- provides logging information to support complete transparency,
- makes sure the SaaS binaries are always up-to-date, and
- schedules the device resources based on its availability and between different SaaS instances.

The logging functionality provides the means for any Producer to verify the transactions created by the Ecosystem. By comparing transaction blocks with a series of events registered by the Client Application, the users can audit the Ecosystem, verifying the fees and rewards for sharing their devices are accurate.

Although the Client Application is capable of downloading applications, it does not automatically update itself: this is due to the potential security risks to the end user. Depending on how the Client Application was installed, it can either run in the background like a daemon, or start when an individual user logs in (thus stopping when the user logs out).

The Client Application communicates with the rest of the Symbioses Ecosystem using the SYMIO protocol (see Section 5), although it does not allow direct connections from other devices. This mechanism increases the security of the Producer's device by avoiding to open extra ports. In order for one Producer to manage several Symbioses installations, the same authentication credentials should be used in all of them. A drawback to do so is that there is some initial work required in order to install the Client Application in each of the devices. This decision, however, is based on the increased security risks remote installation management could cause, since they can be the route through which hackers can intrude upon targeted devices, even if configured to accept connections from the same device only.

4.3 Symbioses Public Cloud API

The objective of the Symbioses Public Cloud API is to interact with the Symbioses Ecosystem independently from the underlying decentralized computing protocol. The API provides a unified library of functions, the usage of which is independent from its source, abstracting the implementation details of the Symbioses DCE. Its use should make it easy from the different Symbioses Clients to build interfaces that work between Symbioses and any other external services.

The resources that can be accessed with the Symbioses Public Cloud API can be divided in the following categories:

- SaaS - services that let you select a software of choice.
- Compute - services that let you deploy the chosen software on multiple devices.

4.4 Symbioses Decentralized Cloud Engine

The purpose of the Symbioses DCE is to find the best possible Producers for hosting a Consumer's SaaS request or instance. This process includes different sources of information as well as the intelligent use of the evolution of data analytics based on the history and reputation of both Producers and Consumers.

The dynamics within the Symbioses DCE are dictated by the SYMIO protocol, a detailed explanation of which is presented in Section 5.

4.5 Symbioses Transaction Settlement API

Apart from the abstraction layer provided by the API itself, the role of the Symbioses Transaction Settlement API is to provide a single point of access for settling the computing deals completed by the SYMIO protocol. This not only eases the development of other components in the Ecosystem, but it also allows asynchronous handling of blockchain-specific smart contracts, preventing such transactions from overloading the SYMIO-powered main chain.

Clearly, blockchain commission payment is inevitable for inserting and withdrawing tokens from the Symbioses Ecosystem. Thus, this layer continuously attempts to pack as many transactions as possible inside one block in order to minimize the impact the transaction fee applied by the target blockchain has on the final price paid by the Consumer.

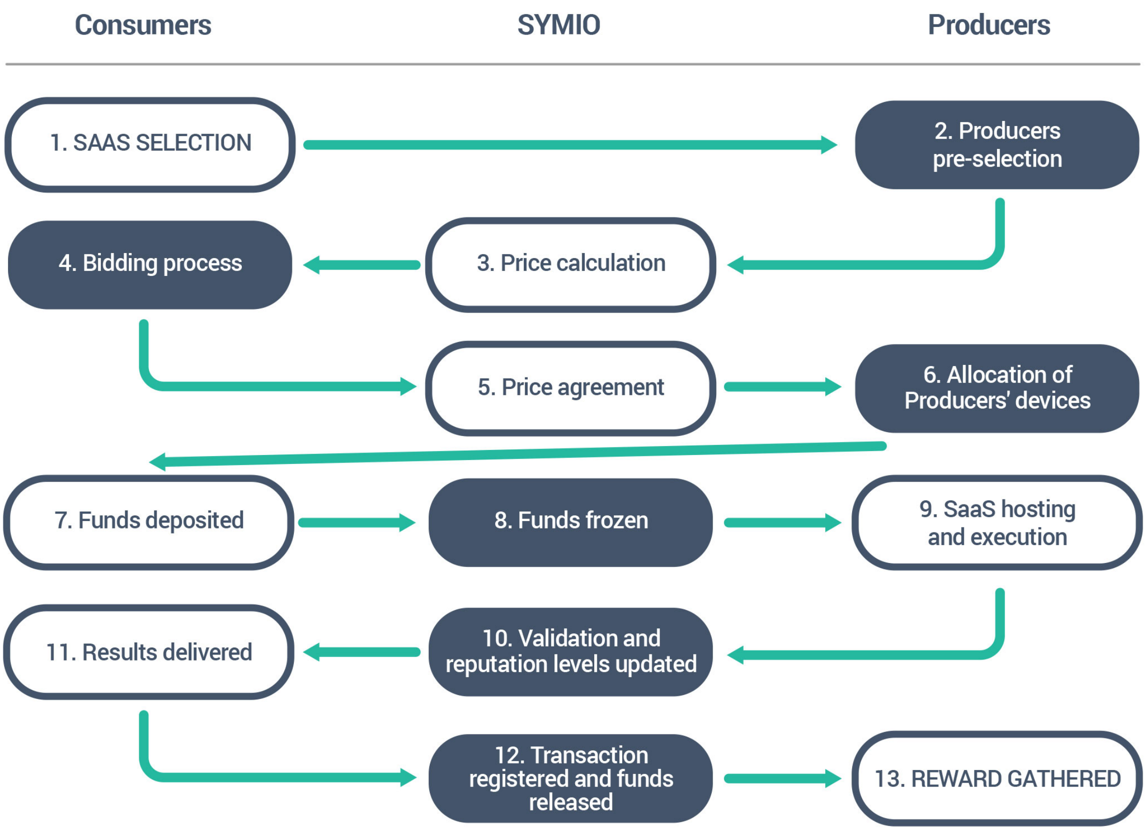

Figure 7: Flow diagram of the SYMIO protocol and the involved entities.

5 SYMIO: a dark-pool matching protocol

The SYMIO protocol connects the different components and actors within the Symbioses DCE, dictating the rules that regulate its consensus, dynamics and security. This internal protocol negotiates computing resources between Consumers and Producers. Since it is kept off chain, it enables a much improved scalability compared with state-of-the-art blockchain networks.

Included in its range of capabilities, the SYMIO protocol creates an incentive for Producers to participate in a competition for hosting a SaaS instance and calculating its results quickly and correctly, thus demonstrating their contribution to the network: the Proof of Work Resources (PoWR). PoWR enables a useful blockchain framework that does not waste computing power on hash discovery, but rather share these resources towards useful tasks required by the Consumers. This scheme provides the basis for earning SYM tokens as a participant of the Symbioses Ecosystem.

SYMIO also captures the dynamics played by network latency for transferring data in combination with different hardware configurations, e.g., better suited hardware will likely host a given SaaS instance better, but it may not be able to transfer the involved data fast enough due to a slow network connection and/or a higher latency when transferring them.

A diagram depicting the system of rules that allow two or more entities agree on a computing-resource sharing session is shown in Figure 7.

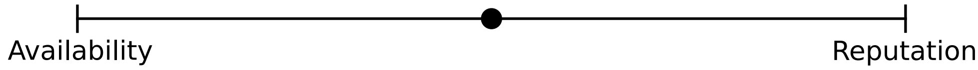

Figure 8: Setting the balance between *Availability* and *Reputation* of execution.

5.1 SYMIO: protocol rules

5.1.1 SaaS selection

The process begins with the Consumer choosing the SaaS to be deployed on the Symbioses Ecosystem. The Consumer also chooses the type of deployment with a sliding bar between *Availability* and *Reputation*. A preference towards *Availability* will favor Producers having a higher availability of their computing resources, i.e., devices that are kept up and running for the required execution period of the SaaS deployment. Similarly, a selection towards *Reputation* increments the significance of Producers' reputation has in the pre-selecting set. The default option is to have a perfect balance between *Availability* and *Reputation*, as depicted in Figure 8.

5.1.2 Producers pre-selection

SYMIO attempts to match a Consumer's request with the existing Producers providing devices that fulfill the requirements to host the selected SaaS instance, like hardware (e.g., GPU or CPU), operating system (e.g., Linux or Windows), etc.

If a suitable subset is found, the protocol continues to filter the Producers who are able to provide their resources for the requested amount of time, i.e., their availability. This information is gathered automatically by SYMIO, which keeps a histogram of the time of the day during which each Producer is available for providing its resources to the network. Figure 9 shows an example of the availability distribution of two Producers during the day. Clearly, the confidence in each Producer's histogram is directly proportional to the time a Producer has been part of the Symbioses Ecosystem. Using this information, SYMIO further categorizes the matching set with a certain confidence level, since it can predict (with a given probability) whether the execution of a SaaS instance is going to be completed within the expected time window. After this last subset is determined, the Producers are organized according to their reputation level: a higher reputation level translates into a higher probability of being selected for hosting the SaaS instance. An example set resulting at the end of a pre-selection process is shown in Table 1.

It is important to note that the described process is performed among registered Producers and not the devices they share. This means that a Producer sharing ten devices does not have a ten-times higher probability of being chosen over another one sharing just one device.

If at any point during the pre-selection process matches cannot be found for a certain amount of time (W_{max} from Equation 3), SYMIO offloads the request to an external network, so it can harvest the required resources from the global marketplace (see Section 6).

Another point worth mentioning is that due to the dark-pool nature of the protocol, the steps taken to fulfill a Consumer's request are not publicly visible. This prevents open requests from being front run as they are being selected.

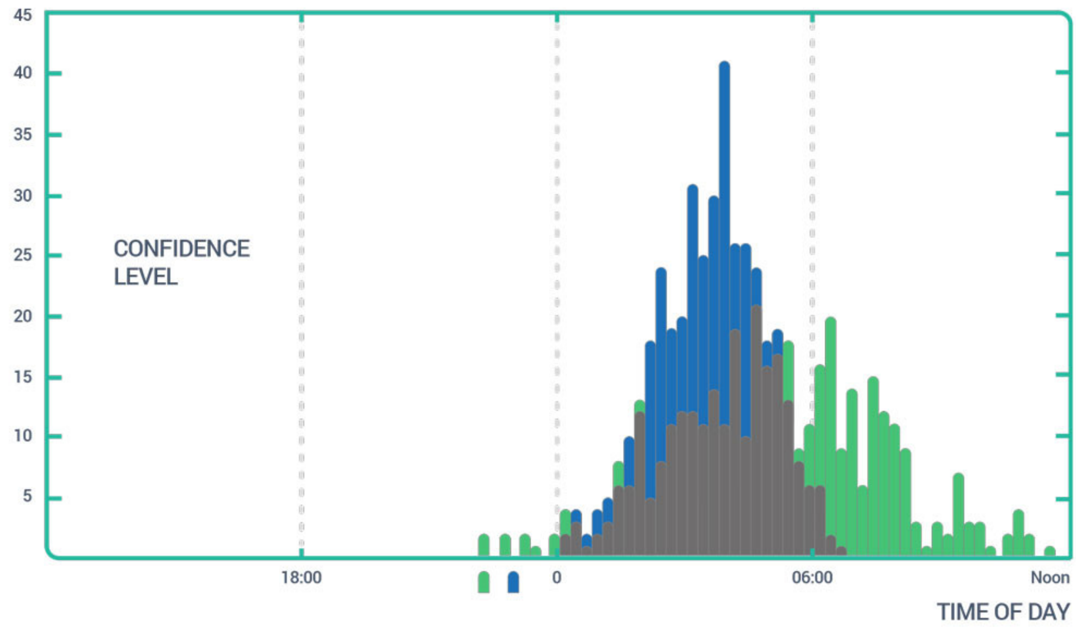

Figure 9: Example of an overlapping histogram showing the resource availability of two Producers during 24 hours. Zero marks midnight on the X axis, and the confidence level is on the Y axis. Overlapping areas are marked in gray.

5.1.3 Price calculation

Once the pre-selection process is completed, SYMIO calculates the price for each of the Producers in the set. The price calculation happens before the deployment of a SaaS instance on the Symbioses public cloud, and it follows the mathematical definition below.

Let H_t be the weighted moving average of SYM tokens paid for hosting a SaaS instance during e-minutes on platform p and hardware h, calculated at time t for the last f days where f is the f-th Fibonacci number:

$$H_t(e, p, h) \tag{1}$$

Let W_t be the normalized weighted moving average (in minutes) for a given platform p and hardware h to become available for execution, calculated at time t for the last f days where f holds the same value as in Equation 1:

$$W_t(p, h) \tag{2}$$

and let W_{max} be the maximum waiting-time threshold before offloading to an external system, where:

$$W_t(p, h) < W_{max} \tag{3}$$

Let now be g the Consumer's balance between *Availability* and *Reputation* as depicted in Figure 8.

Producer	Availability vs. Reputation ratio	Availability	Reputation	Price - $P_t(e,p,h,m)$
p0	0.50	0.80	0.89	**284.2485**
p1	0.50	0.87	0.85	297.8603
p2	0.50	0.76	0.92	294.6576
p3	0.50	0.73	0.55	262.6296
p4	0.50	0.65	0.75	272.2379

Table 1: Example of calculated prices for a pre-selection ser of five Producers. The price suggested to the Consumer, i.e., the median, is marked in **bold**.

Let D be the price for computing resources for hosting a SaaS instance during e minutes on platform p and hardware h is defined as, calculated at time t for Producer pr:

$$D_t(e,p,h) = H_t(e,p,h) \cdot (1 + W_t(p,h) \cdot [(1-g) \cdot pr_{availability} + g \cdot pr_{reputation}] \quad (4)$$

Let S_t be the market price expressed in SYM tokens for the storage of 1 MB of data in a decentralized storage network, calculated at time t:

$$S_t \quad (5)$$

Then P_t is the Producer's price in SYM tokens for hosting a SaaS instance during e minutes on platform p and hardware h, requiring m MB of data storage and calculated at time t is defined as:

$$P_t(e,p,h,m) = D_t(e,p,h) + m \cdot S_t \quad (6)$$

In case of SaaS instances allocating multiple devices for their hosting, the factor $D_t(e,p,h)$ has to be multiplied accordingly.

The suggested price of the execution job presented to the Consumer is the median of all $P_t(e,p,h,m)$ contained in the Producers pre-selection set, an example of which is shown in Table 1. Note that the final suggested price for the Consumer in this case is 284.2485 SYM tokens, which is marked in bold font.

5.1.4 Bidding process

SYMIO implements an adaptation of the double auction model [4]. This model is one of the most commonly used in exchange institutions, the roots of which go back to ancient Egypt and Mesopotamia. In a double auction model, buy orders (bids) and sell orders (asks) may be submitted at anytime during the trading period. If at any time there are open bids and asks that match or are compatible in terms of price and requirements (e.g., quantity of goods or shares), a trade is executed immediately. In this auction model orders are ranked highest to lowest to generate demand and supply profiles. From the profiles, the maximum quantity exchanged can be determined by matching asks (starting with lowest price and moving up) with demand bids (starting with highest price and moving down).

Symbioses incorporates a software-agent mechanism that automates the double-auction method for trading computing resources, i.e., the execution begins as soon as there is a match between the offered price and the cost of the computing resources. However, the Consumers may alter their offer while waiting for a Producer to become available. Recall that the initial price is suggested by the SYMIO protocol itself, as explained in Section 5.1.3. This means that it is up to the Consumer to decide how much a certain execution is worth. Instead of choosing an offer from a pre-defined price list, the Consumers' bidding together with the Producers availability shape the pricing of the computing resources. It follows that if the offered price is too low, the SaaS deployment may never begin, because the Producers will rather choose to host other, better rewarded executions. On the other hand, if the offered price is higher, the Consumer will most likely receive the computing resources with the desired level of availability and reputation. Clearly, changes in the price will influence certain aspects of the computing service, e.g., the waiting time in queue, the reliability of the selected Producers, and others.

Once an execution begins, the price it was agreed upon cannot be changed for the duration of the SaaS deployment. Although this may be an open point for further discussion, we believe that the fluctuation price of the SYM token while the job is ongoing is a shared risk between the Consumer and the Producer, since the token price may change either way due to market dynamics.

The negotiated prices for computing resources are saved and later used as the initial seed for calculating subsequent suggested prices of similar executions, i.e., H_t from Equation 1. This factor plays a key role should the required amount of SYM tokens to access computing resources on the Symbioses network become too expensive (to the point where using the network becomes unattractive due to pricing). Since H_t is the weighted moving average of the agreed prices during the last f days, its value will gradually converge towards zero if no deals are closed, consequently lowering the price for hosting SaaS instances on the network.

It is at this point that the race between the Producers matching the offered price begins.

5.2 SYMIO: the race

In the Symbioses Ecosystem, Producers compete with each other to be the first to deliver the results of a SaaS instance. Clearly, the Producers having the lower latency in terms of data transfer and the most powerful hardware will most likely deliver ahead of the other ones. However, just having the best hardware available does not necessarily guarantee a Producer to come out as the Winner of the race. Depending on the details of particular a SaaS instance, such configurations could easily be overtaken by slower hardware on a faster network link providing lower latency. When a Producer delivers the results ahead of others, SYMIO declares a Winner and the deployment begins it closing phase. This includes releasing the Consumer's funds, the Winner receiving the payment for the provided resources, and the reputation levels being updated based on the gathered SYM tokens. The Consumer's reputation level also increases, but with the number of SYM tokens spent for each requested and successfully completed SaaS instance in the Symbioses Ecosystem.

5.3 SYMIO: verification

Sometimes, however, the race ends with several potential Winners. If the Consumer requested the verification of the delivered results, the second Producer delivering the results is recognized by SYMIO as a Verifier. At this point it should be clear that result verification requires additional Producers' resources, which directly influence the price payed by the Consumer for the requested SaaS (see Section 7 for a detailed description about pricing). Once the results delivered by the Verifier are available, SYMIO compares them with the results delivered by the Winner. If they match, everyone is happy: the Consumer's funds are unlocked, the Winner and the Verifier proportionally receive their payment for the provided resources, thus gathering SYM tokens that will enable them to increase their reputation levels accordingly. As for the Consumer, the mechanism is the same as before: the reputation level improvement is directly proportional to the number of SYM tokens spent for the successfully completed SaaS instance in the Symbioses Ecosystem.

But what happens if the results delivered by the Winner differ from those delivered by the Verifier? In this case, a third Producer called the Referee is brought in by SYMIO to solve the dispute. Remember that these results are already available, since at least three Producers must join the race for it to initially begin. At this point only one of the following scenarios is possible, either:

1. the Winner results are broken, i.e., the Verifier's and Referee's results match;
2. the Verifier results are broken, i.e., the Winner's and Referee's results match;
3. if all available result sets differ, another Referee is brought in to solve the dispute, and the roles of the involved parties are updated according to the points above;
4. if there are no other Producers available to solve the dispute, the execution instance is marked as broken.

Note that, due to the dark-pool nature of the protocol, the Producers have no visibility or knowledge about the role they will fulfill once they deliver their results, since this is only determined by SYMIO. Thus it is reasonable to assume that each Producer will make the effort to become the Winner of an execution instance, i.e., finishing the execution job in time and form in order to claim the biggest reward. This prevents a-priori negotiations between Producers willing to mark an execution as broken and claim the payment for it.

5.4 SYMIO: verification outcomes

When SYMIO detects a situation as described in scenario 1., the Verifier becomes the new Winner of the execution race, the Referee becomes the new Verifier, and the original Winner looses the amount of reputation points she would have been awarded in SYM tokens if correctly calculating the results. Needless to say, the original Winner does not get paid, but the new Winner (originally the Verifier) and the new Verifier (originally the Referee) proportionally share the paid amount for their provided resources, also accumulating the corresponding reputation points.

Scenario 2. creates a slightly different outcome: the Winner remains as such, the Referee becomes the new Verifier and the original Verifier loses the amount of reputation points she would have been awarded if correctly delivering the results. Similarly as before, the original Verifier does not get paid, but the Winner and the new Verifier (originally the Referee) proportionally share the paid amount of SYM tokens for their resources, also accumulating the corresponding reputation points.

In case scenario 4. occurs, the computation is marked as invalid by SYMIO, the results of which are not delivered to the Consumer, who still has to pay for the consumed resources provided by the Winner and the Verifier. In this case no reputation points are awarded to the Producers, since it is a-priori not clear whose fault it is for the results to differ. The Consumer loses reputation points for the amount of SYM tokens spent.

5.5 SYMIO: reputation tracking

As it was explained in Section 3.1, a user (Producer or Consumer) interacts with the Symbioses Ecosystem through the SYM token. The amount of SYM tokens a Symbioses user holds is continuously tracked through its blockchain interface and used as the basis to build her reputation. In this sense, the users improve their reputation level by gathering, spending and holding SYM tokens for predefined amount of time. Producers, for example, increase their reputation by gathering their rewards after providing resources for hosting SaaS instances. Consumers, on the other hand, increase their reputation by spending SYM tokens on the Ecosystem for the resources needed for the relevant SaaS instances. Note that this mechanism also allows any user to acquire Producer's reputation by simply buying SYM tokens and keeping them in the wallet associated to the Symbioses Ecosystem. Although this mechanism may initially seem as "cheating", the dynamics emerging from it will make the market for accessing SaaS in the Symbioses Ecosystem progress in

Level	**Member**	**Balanced**	**Advanced**	**Guru**
Condition	Sign up	p SYM tokens gathered in m months	$3 \cdot p$ SYM tokens gathered in m months	$18 \cdot p$ SYM tokens gathered in $2 \cdot m$ months
Validity	m months	No time limit	No time limit	No time limit
Priority	Random	Random with third highest probability	Random with second hightest probability	Random with hightest probability
Deduction	Regular	d %	$2.5 \cdot d$ %	$5 \cdot d$ %

Table 2: Proposed reputation levels for Producers.

Level	**Member**	**Balanced**	**Advanced**	**Guru**
Condition	Sign up	c SYM tokens spent in m months	$3 \cdot c$ SYM tokens spent in m months	$18 \cdot c$ SYM tokens spent in $2 \cdot m$ months
Validity	m months	No time limit	No time limit	No time limit
Assistance	r requests every m months	$3 \cdot r$ requests every m months	$6 \cdot r$ requests every m months	$12 \cdot r$ requests every m months
Fee	Regular	e % Off	$2.5 \cdot e$ % Off	$5 \cdot e$ % Off

Table 3: Proposed reputation levels for Consumers.

notable ways. Indeed, some of these behaviours could be reproduced in the simulation environment where the SYMIO protocol is tested.

Similar to Proof-of-Stake [20], the reputation level has a direct influence in the choice of the Producers for a particular execution job. In this sense, the choice of a Producer as the resources provider is done via various combinations of random selection and reputation level. For example, given two identical Producers' available on the Symbioses Ecosystem, the Producer having the higher reputation level will generally, but not consistently, be chosen by the SYMIO protocol. Again, the random component is included to increase the dynamics of the SaaS market within the Ecosystem.

Table 2 shows an initial proposal for the different reputation levels a Producer may achieve, including the amount of SYM tokens required to unlock each of them.

Similarly, Consumers reach higher reputation levels with the amount of SYM tokens spent while paying for successfully deployed SaaS instances on the Symbioses Ecosystem. The reputation levels proposed for Consumers are listed in Table 3.

The amount of SYM tokens Consumers spend in paying SaaS instances that do not complete because they run out of time do not earn them any reputation points.

In case of misbehaviour of Producers or Consumers, SYMIO will initiate the process of blacklisting the user in question and will freeze the funds during an a-priori specified amount of time. Clearly, recurrent misbehaviour is punished with frozen funds for a longer period of time, the growth of which is exponential, until they are eventually expropriated and the user's account banned.

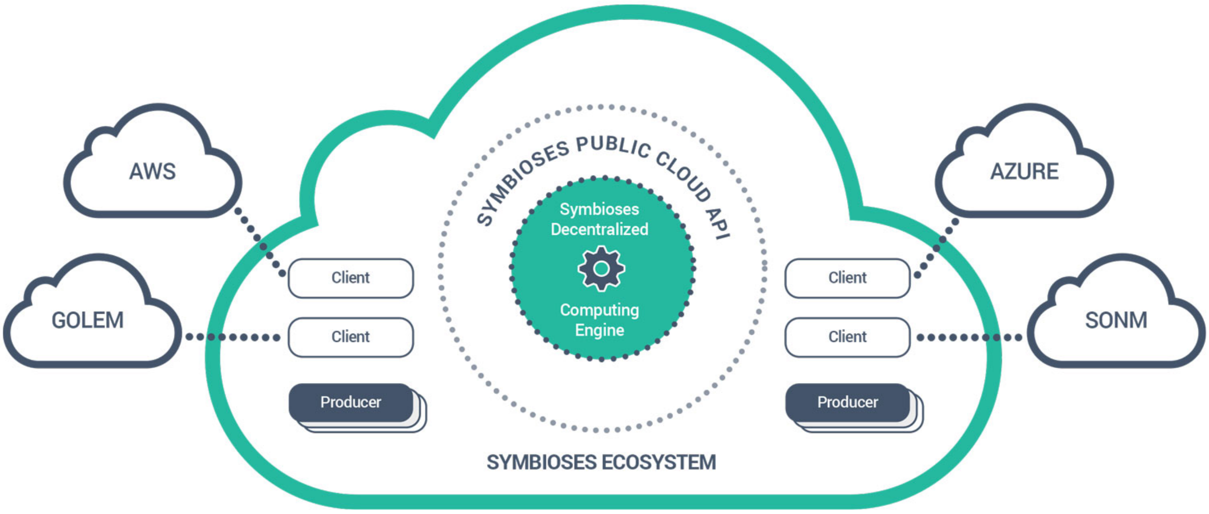

Figure 10: The Symbioses Ecosystem as a connection hub for non-interoperable networks.

6 Interoperability with External Networks

Besides all the features described in the previous Sections, Symbioses is also able to talk to external systems. This is achieved by implementing specialized Clients as part of the Symbioses Ecosystem, each of which takes care of two-way communication with a different external system, whether it is decentralized or not. Specifically, each Client implements (translates) the protocol of the relevant system, dispatching the messages to the Symbioses DCE through the Symbioses Public Cloud API. In one direction, the Clients may request additional resources in external networks when the Symbioses Ecosystem is fully occupied, i.e., no Produces are available. In the other direction, the Clients may offer Symbioses resources to these external networks, giving the Producers the possibility of offering their resources on several networks simultaneously. Furthermore, this way of communication allows to overcome the compatibility barriers, since the mentioned protocols are potentially non-interoperable between each other.

Figure 10 shows the way this feature increases the potential of the Symbioses Ecosystem by also leveraging its technology through external systems (compare this with Figure 3).

It is worth pointing out that Symbioses does aim at becoming a heterogeneous multi-chain technology. The underlying complexity of such solution is not required in the decentralized computing case, where every node in the network is independent from the rest. A multi-chain solution consisting of many parallel chains with potentially differing characteristics would create an excessive overhead, rendering the Symbioses network too slow for real-time usage. Additionally, since not all the features available in the Symbioses Ecosystem will be available in these external networks, the specific Clients will have to filter those out. For example, since as the time of this writing Golem does not support workloads other than 3D rendering using Blender, an arbitrary scientific simulation cannot be offloaded to the Golem network.

Another example of interoperability with external systems is accessing computing re-

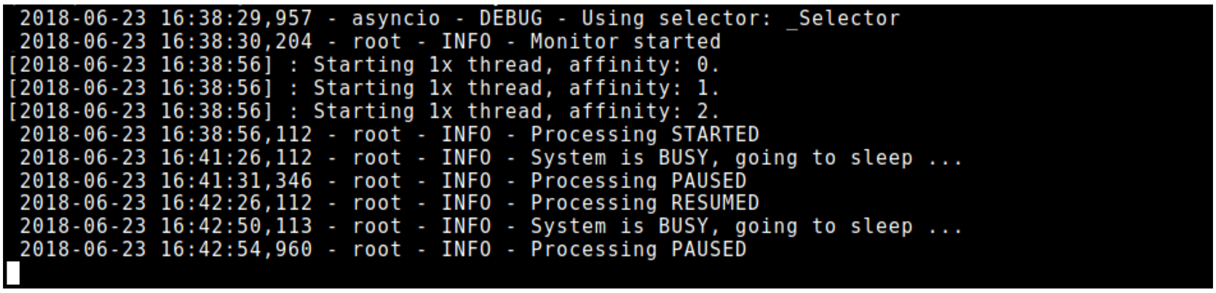

Figure 11: A traditional cloud instance sharing its computing resources with the Symbioses Ecosystem.

sources offered by centralized cloud providers, e.g., Amazon EC2, inside Symbioses. Users renting dedicated instances on traditional cloud providers can offer their idle computing resources to the Symbioses network thus partially redeeming their expenditure in such services. Listing 1 shows a code snippet to programmatically launch an EC2 instance from the compatible Symbioses Client, whereas Figure 11 shows a sample output displayed when a traditional cloud instance connects to the Symbioses network.

Listing 1: Sample code to launch an EC2 instance.

```
from boto import ec2
conn = ec2.connect_to_region("eu-central-1",
                             aws_access_key_id='<aws␣access␣key>',
                             aws_secret_access_key='<aws␣secret␣key>')
conn.run_instances('<ami-image-id>',
                   key_name='<key>',
                   instance_type='c1.xlarge',
                   security_groups=['<security␣group>'])
```

7 Economic model

In previous sections, several of the benefits decentralization brings to sharing resources and accessing readily available SaaS solutions have been discussed. However, the decentralized nature of blockchains pose interesting challenges in terms of the economic model behind them.

Other decentralized technologies like BitTorrent [32] employ the rudimentary approach of not allowing resource consumption without synchronous contribution. This approach is typically straightforward to implement but results in curtailing participation in the technology [30].

William Stanley Jevons explains the issue occurs in non-monetary economies where two entities must want a resource from each other in order for a transaction to be negotiated [17]. He calls this the coincidence of wants:

"The first difficulty in barter is to find two persons whose disposable possessions mutually suit each other's wants. There may be many people wanting, and many possessing those things wanted; but to allow of an actual act of barter there must be a double coincidence, which will rarely happen."

The rest of this Section discusses a modified version of the economic model proposed by Omega One [28] for its use in the Symbioses Ecosystem. The discussion on the negotiation protocol incorporates elements specific to the decentralized nature of SaaS solutions and their resources. This ensures the proposed approach is optimized for the Symbioses Ecosystem while providing a general framework for other decentralized protocols integrated through its architecture (see Section 4).

7.1 Model design

SYM tokens are the exchange currency in the micro economy of the Symbioses Ecosystem. However, they are not just exchanged for hosting SaaS instances, since they also provide proof of membership and reputation.

Symbioses is actively exploring exchanges and partnerships to ease the process of liquidating and/or exchanging tokens for rewards.

Figure 12 shows the involved elements in the model, a description of which follows:

- Consumers will be charged transaction fees for their usage of the Symbioses public cloud, i.e., hosting SaaS solutions.
- The fees are distributed among the Producers providing the computing resources to host these solutions.
- A small percentage of the fee is dedicated to the support and continuous development of the Symbioses public cloud, i.e., the Symbioses Ecosystem fund.
- The fees will be paid using SYM tokens, and the Producers are incentivized to hold the tokens in order to gain higher reputation levels (see Section 5.5) and thus increase

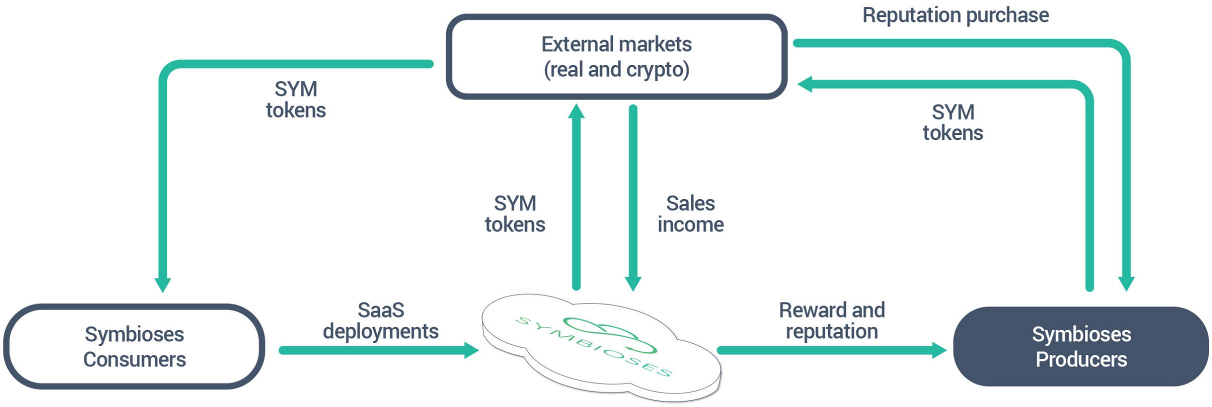

Figure 12: Elements of the economic model of the Symbioses Ecosystem.

the rewards on individual workflows. These factors should result in lowering supply on the market and reducing the velocity of the token throughout the Ecosystem, which should put upward price pressure for the SYM token.

Compared to centralized cloud providers and online (social) communities, where a nominal number of users is needed in order for the business to be profitable or the network to be regarded as relevant, Symbioses creates value with the first Producer who joins the network. Even in the case when no Consumers require SaaS services, block mining provides a baseline for the Producers' idle computing resources. On the other hand, a great amount of Producers in the presence of a handful of Consumers will drastically lower the price of the available computing resources, making Symbioses even more attractive to Consumers of traditional cloud services. This fundamental principle enables the Symbioses Ecosystem to keep growing its business with every single Producer who shares computing resources on the network.

7.1.1 Market sales of SYM tokens

Although the lack of a large central holder managing supply is one of the key features of cryptocurrencies, the inclusion of explicit reserve management commitments, mainly in the form of smart contracts, is becoming increasingly popular. However, there are different opinions about how cryptocurrency reserves should be managed. The option of implementing an automated reserve management policy has been dismissed by the Symbioses team mainly because of its lack of flexibility.

Primarily driven by protecting the service value from the volatility of cryptocurrency markets, Symbioses requires a certain level of flexibility to decide how to use reserves in order to smooth larger variations. Therefore, Symbioses will execute market sales of SYM tokens over time pursuing the goal of keeping a high service value at reasonable prices. It follows that the policies and procedures governing such sales will be established individually as regards to the ongoing circumstances. Furthermore, Symbioses is implementing vesting and lock-up periods for owners, employees and advisors in order to retain talent and align the long-term interest of the company with a successful token-economic system.

7.2 Consumers

Consumer demand for SaaS services will be derived from the computing resources required for rendering, big-data analysis, deep learning and scientific simulations. We will continually strive to expand our target market capabilities as the Ecosystem grows.

Symbioses has demonstrated the ability to reduce the cost of computing resources by up to 90% compared to the price of centralized cloud services.

At this point it should be clear that SaaS plays a central role in the Symbioses Ecosystem. While the capability to deploy both standard and decentralized applications is present in the Ecosystem, it is by providing readily available SaaS solutions in a decentralized fashion that enterprise consumers will benefit from a number of advantages the Symbioses network provides, e.g.:

Reduced time The application of choice is already deployable and configured. The Consumer has the advantage of choosing from a number of options in the SaaS catalog and have the application ready for deployment. This not only reduces the time spent in installation and configuration, but reduces the frictional issues that complicate software deployment in general.

Lower costs The SaaS license is shared by all the Consumers of a particular application, making it a differential cost compared to the traditional model. Clearly, the customer base can be greatly increased since it allows scientists, small businesses and medium businesses to use an application that otherwise would be prohibitive due to the high cost of license. Maintenance costs are reduced as well, since it is the Symbioses Ecosystem through its Producers who provide the hosting environment.

Scalability and integration SaaS solutions provided by Symbioses are scalable over multiple Producers and have integration with other centralized and decentralized offerings. Compared with the current model, Consumers do not have to register with a number of decentralized services to get the best of blend. They only need to access the Symbioses Ecosystem offering and the protocol will integrate the necessary capacity to deliver the required computing resources.

New releases The Symbioses team will regularly provide application upgrades and service expansion of relevant SaaS solutions, making them available to the Consumers. Clearly, there are no extra costs or effort from the Consumers' side associated with new application releases, the installation of which usually demands expenses for specialized services to upgrade the existing environment.

Ease of accessibility to perform proof of concepts Consumers can rely on SaaS to perform proof of concepts and test the application functionality in advance. It is also easy to have more than one instance with different versions and perform a smooth migration.

7.3 Producers

Producers within the Symbioses Ecosystem are incentivized for the resources they provide to the network. Their participation is rewarded in the form of SYM tokens, which are also used as the foundation of the reputation system. The availability of Producers' devices is modeled by means of a statistical analysis, the confidence of which grows with time and the reputation level a particular Producer achieves. Note, however, that the reputation system does not necessarily favour the most performant hardware in the Ecosystem, but it rather encapsulates different aspects influencing the delivered QoS to the Consumers.

The availability of the Producers' devices is established via the Client Application and determined by multiple factors (such as idle capacity, electricity consumption, internet connection, etc.), in parallel with Producer's selected criteria. The Producer is able to input individualized electricity cost and network-transport cost (if applicable) to view potential profitability of the individual computing devices.

Due to the dark-pool nature of the SYMIO protocol, the risk of abusing the reputation system is minimized. Similarly, it helps mitigate other kind of popular attacks in decentralized networks, such as front running or Sybil attacks. These advantages are further reinforced with the competition between Producers to quickly deliver the correct results to the Consumer.

Certainly, the Symbioses Ecosystem provides a unique opportunity for owners of spare computing power, since becoming Producers enables them to deliver these resources in the form of SaaS services to individuals, businesses and governments, i.e., the Consumers. Delivering SaaS solutions will significantly enhance the rewards generated for the Producers. In financial terminology, the return on investment (ROI) of an average GPU-enabled device can scale from 60-80% for block mining (net of average electricity expense) to as high as 400% for other specialized SaaS instances. This value add is available to Producers sharing their resources within the Symbioses network.

The rewards gathered by the Producers are directly proportional to their contributed resources. Consequently, the strong scalability [36] of the Symbioses Ecosystem grows with the number of Producers. Due to the pseudo-random component included in the Producers pre-selection process, all Producers sharing the same kind of hardware and platform will have a uniform probability of being selected. If, on the other hand, the chance of hosting a SaaS instance was proportional to the computing power of a Producer's device, slower devices would hardly ever take part of a deployment. Thus, if taking a sufficiently long period of time into account, it is expected for a Producer to get a percentage of all distributed rewards equivalent to the portion of the contributed resources. It is the role of the Symbioses Ecosystem to combine the available resources and maximize the rewards. These are transparently distributed among the Producers according to the size of their contribution to the network. In other words, the Producer gains a steady small income instead of hosting a complete SaaS instance every few months.

Symbioses is also actively looking to establish partnerships with other blockchain projects that will enable our Producers to do a swap of the gained SYM into another token. This would enable them to have instant access to a diverse array of services and physical goods should they choose to opt-in. The benefit of this approach is to further simplify the interaction of the general public with the blockchain, since the token swap is done automatically.

7.4 Secondary actions being rewarded

Producers and Consumers are also rewarded for the following actions:

- Registering to the service (welcome award).
- Referring friends to expand the Symbioses network.
- Promoting the service.
- Writing editorial content.
- Reporting bugs about the Client or the Ecosystem in general.
- Reporting security issues.

8 Security

When evaluating the security of the Symbioses Ecosystem it is important to consider the roles and interests of the different parties involved:

- **Data provider (DP)** is the party providing the input data. DP wants the data to be protected from theft and might want to control it.
- **Software provider (SP)** is the party providing the software to the delivered as SaaS. SP wants the software to be protected from theft.
- **Consumer (CU)** is the party who needs the output data that is produced by a SaaS instance after processing some input data. The CU wants to have a secure method for accessing the output data.
- **Producer (PU)** is the party providing the resources to host SaaS instances.

Some of the parties may be the same entity or person. It is also worth mentioning that although the target environment is a decentralized network, the security implications are treated as in a multi-tenant system due to the hosting of third-party SaaS and data, i.e., a Consumer's SaaS request deployed on a Producer's device.

It is important to refer to the Symbioses Ecosystem as a separate entity, since there is an inherent risk of hacking (or unauthorized access). Indeed, as the target environment is treated as a multi-tenant system we further assume a hacked infrastructure provider in order to provide additional protection levels.

8.1 Cases of trust, security and privacy

Let us now consider the following cases where the different parties interact with each other and how this interaction may influence the security model that needs to be applied over the Symbioses Ecosystem.

1. **No encryption** The data and the SaaS instance are open. In this case nothing is additionally protected with any form of encryption. As such, the workflow provides a portable, reproducible and convenient way to manage and deploy SaaS (see Figure 13).

2. **Multi-tenant security** DP, SP and CU are the same entity or they trust each other. The data and SaaS instance should be protected from other users of the Ecosystem, e.g., the Producers. There should also be a secure transmission in and out of the deployment infrastructure.

3. **DP=SP vs CU** DP and SP are the same person or they trust each other. But they do not trust CU or PU. CU also may not trust PU. So PU receives encrypted data and encrypted software. PU should be able to execute the hosted SaaS using the corresponding data, and send the encrypted output data back to the CU. But the PU shouldn't be able to decrypt anything. DP and SP also may want to limit the use of the data and the software (by a period of time or by the number of executions).

   Let us now consider an example. DP has a medical database containing patients' personal information. CU is a scientist who needs statistical information about diseases. But CU shouldn't be able to see any personal information about the patients. Clearly, DP does not trust CU. SP provides a SaaS instance for statistical processing, and guarantees the output data will not contain any of the patients' personal information. DP trusts SP, so DP may allow CU to process the data using that particular SaaS provided by SP.

4. **DP=CU vs SP** DP and CU are the same person or they trust each other. But they do not trust SP or PU. In this case it is important to safeguard that the SaaS instance cannot send data to the internet or store it anywhere but inside the encrypted archive containing the output data (see Figure 14).

   Let us now consider another example. SP provides a unique SaaS for physical modeling. A novel mathematical theory was developed especially for this software package. SP wants to prevent reverse-engineering attempts to extract the algorithms from the deployed binaries, e.g., PU. CU is an engineering company looking to initiate the physical modeling of new microcontroller, the design of which is secret and should not be shown to any third party, e.g., SP or PU.

Figure 13: A SaaS deployment without encryption.

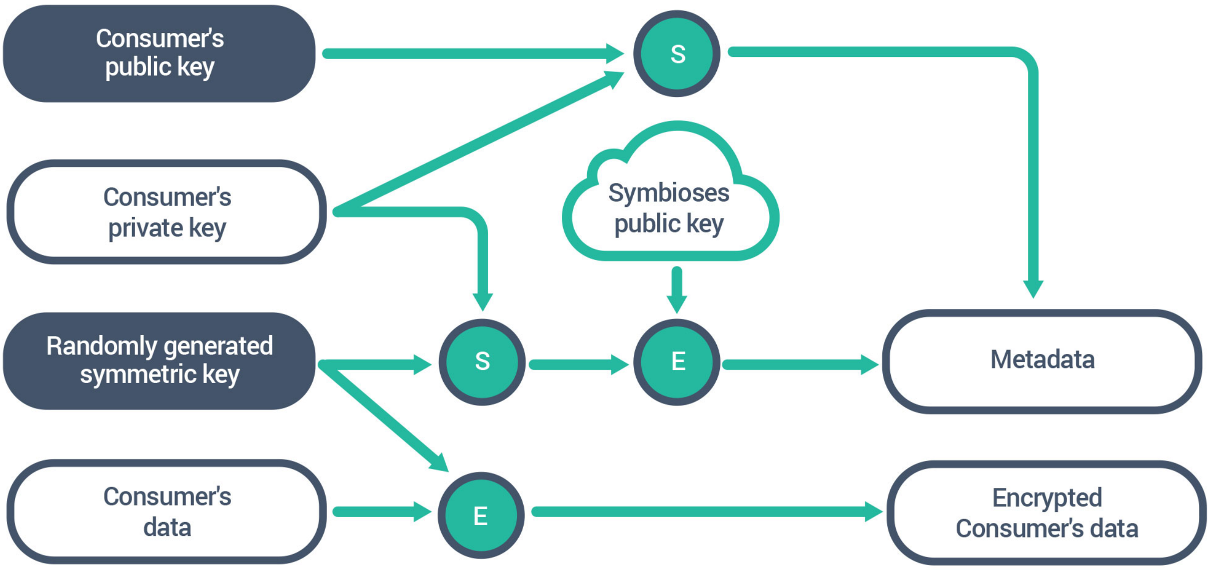

Figure 14: Consumer's workflow to prepare encrypted data to be accessed by an untrusted SaaS instance.

8.2 Workflow and support

The general workflow for securing the data consumed during the execution of a SaaS instance might be as follows:

- CU (in this example DP, SP, and CU are the same entity or they trust each other) encrypts the input data using a trusted computer, e.g., her own PC.
- CU provides the encrypted input data as part of the chosen SaaS to be hosted by one or many PUs.
- The selected PUs deploy the corresponding SaaS instance and provided input data, the files of which are mounted as part of the deployed instance.
- CU can also specify that the generated output should be encrypted.
- The output data produced by a SaaS instance is transferred back to the CU, who can adequately decrypt the data by using her secure keys.

Since the security framework should support all cases described in Section 8.1, the implementation work will be divided into several stages. The first and simplest stage will support the general workflow without providing any data or software protection, i.e., case 1. "No protection". The complexity of the other cases gradually grows together with the number of features required from the framework itself, and should follow in development.

Clearly, a solid and secure framework has to be continuously validated in practice. To this end, open challenges will be organized for the community to verify the security measures in practice (see Section 9).

8.2.1 Overview of the encryption schema

Symmetric-key algorithms use the same cryptographic key for both encryption of plain text and decryption of encrypted text [7].

Public key cryptography [11], on the other hand, uses a pair of keys: a public key which may be disseminated widely, and a private key which is known only to its owner. This arrangement accomplishes two main functions, namely encryption and authentication. Encryption is achieved since only the holder of the paired private key can decrypt the message, the content of which was originally encrypted with the public key of the same pair. The authentication aspect appears when the public key is used to verify that a holder of the paired private key is the actual sender of the message.

In the general case, the Symbioses Ecosystem has to guarantee both reading and writing of data in a secure way. Therefore symmetric encryption seems a reasonable choice. To improve the robustness of the system, the cryptographic keys are randomly generated and valid only for a certain period of time.

Referring back to cases 3. and 4. in Section 8.1, the final piece of the schema is to use public-key encryption to encrypt the symmetric key itself. Since each PU's account on the Symbioses Ecosystem holds a different public-key pair, even the same original plain text will hold a different encrypted payload.

Fully encrypted data are only transferred through the network. However, it is clear that at some point the decrypted data should be placed on the PU's device in order to be usable by a SaaS instance. After the public-key decryption step, the data is still encrypted

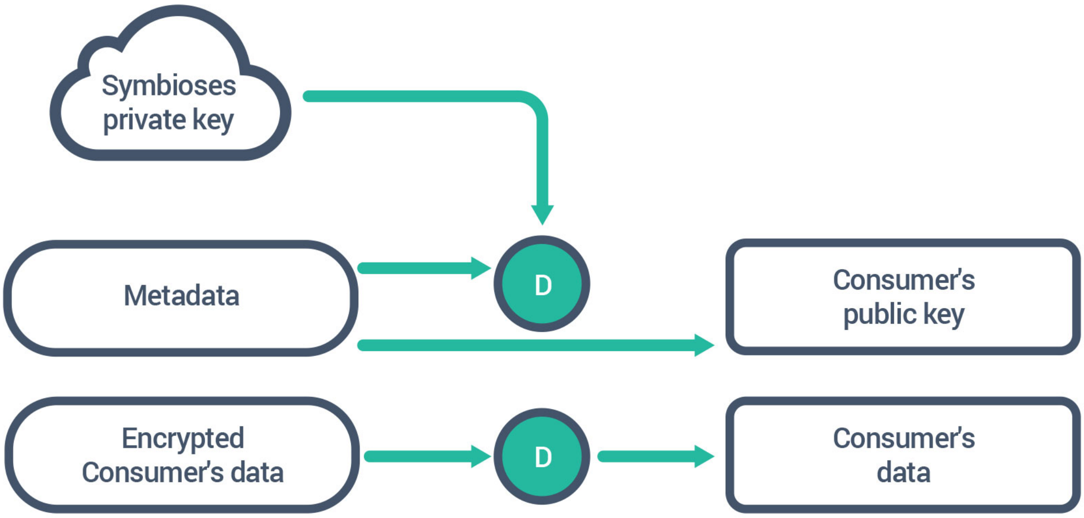

Figure 15: Accessing data that was encrypted by a Consumer.

using the symmetric-key algorithm and stored for its use. Thus, although the data has actually been decrypted, the PU cannot access it directly because it is still enclosed in the symmetric-key envelope. The SaaS instance may access the data since it is located at the same encryption level as the data itself (see Figure 15). Using a similar procedure, but with a different combination of the public-key pair, the data generated by the SaaS execution can be encrypted again before being transmitted back to the Consumer (see Figure 16).

Still, it is important to include and continuously develop risk-mitigation strategies as part of the encryption schema itself to provide a robust solution. In this sense, the validation of the framework in practice plays a key role to pursue this objective.

8.2.2 Structure of encrypted data

Consider an example encrypted data `secretData` and its internal structure described by two files, i.e.:

`secretData.dat` A compressed and encrypted archive containing the actual data.

`secretData.meta` The metadata of the actual data, including a digital signature.

The contents of the archive `secretData.dat` are encrypted using the symmetric-key algorithm, which is better performant on large data chunks. The symmetric key is also encrypted using a public-key set, the result of which is stored in `secretData.meta`. The following is a sample content of this file in JSON format:

```
{
    "name" : "secretData",
    "data_id" : "qDLGq6q4h281bGK6tqfjv7mZeUjkAUzV",
    "allowed_applications" : ["hash1", "hash2"],
    "owner_key" : "ssh-rsa AAAAB3NzaC1... user@laptop",
    "encrypted_data_key" : {
      "ssh-rsa AAAAB3NzaC1... user@laptop"       : "v9SLQh7tT...",
      "ssh-rsa AAAAB3Ngf2f... keys@symbioses.io" : "DVdyC13Qd..."
    },
```

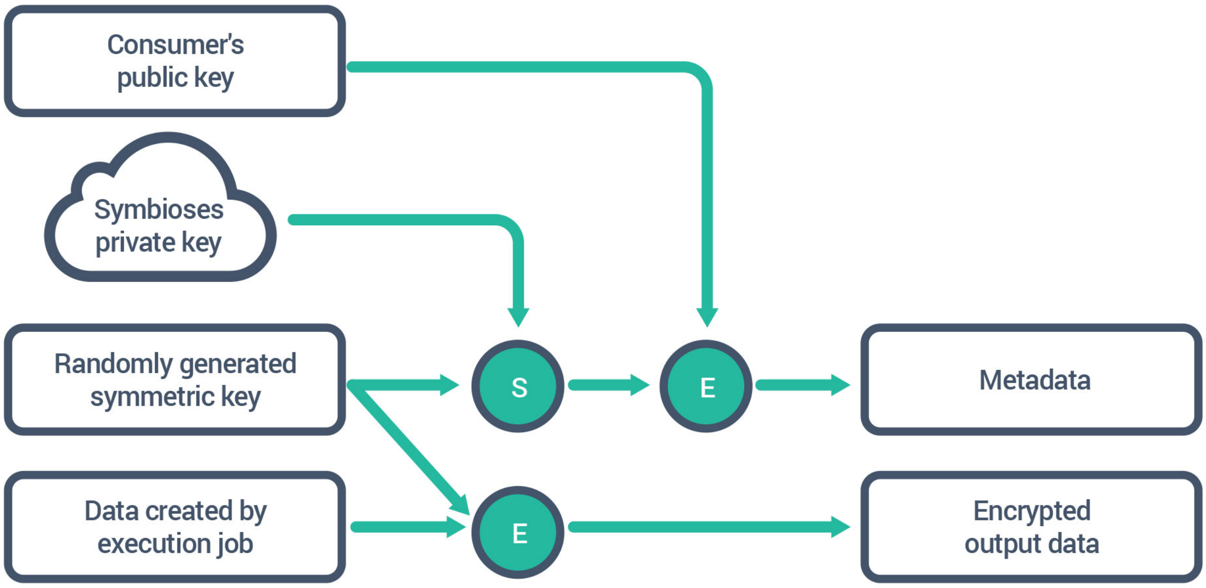

Figure 16: Encrypting data generated by a SaaS instance using the Consumer's key.

```
    "keys_to_export" : [
        "ssh-rsa AAAAB3NzaC1... user@laptop"
    ],
    "readonly" : false,
    "signature" : "qR9pj2yv9SLQhDBcY8eyenFbJDVdyC1M9P3g3QdZtLR"
}
```

The meaning of the fields shown in the listing above are described as follows:

`name` is a user-defined string to identify the encrypted data.

`data_id` is a randomly-generated, 32-byte identifier of the encrypted archive.

`allowed_applications` is a list of hashes of SaaS instances that are allowed to access the data in the encrypted archive.

`owner_key` is the s public key of the Consumer who created the encrypted data archive.

`encrypted_data_key` stores the symmetric key to decrypt the archive `secretData.dat`. Considering any *(key,value)* pair, *key* is the public key of an entity who can decrypt the data, whereas *value* is the symmetric key signed by `owner_key` and encrypted by *key*.

`keys_to_export` is a list of public keys to be used during the encryption step of the output data. Thus the value in `encrypted_data_key` is replaced by the of `keys_to_export` in the metadata file of the encrypted archive holding the SaaS-generated data.

`readonly` indicates the data is accessed in read-only mode once it is decrypted.

`signature` is the digital signature of all other field-value pairs inside the metadata file, which should be created using `owner_key`.

8.3 A note about Docker

Practically all the existing solution for distributed computing [13] [34] rely on Docker containers to isolate the execution environment on a Producer's device. Unfortunately, Docker itself is well known to have several drawbacks when used in distributed environments, the exploitation of which is sometimes trivial, and exposes the target device to multiple attacks [26]. Unfortunately, it is not fully clear from the available documentation how these projects are planning to deal with the mentioned issues. It is certain, however, that without a clear understanding about the security implications Docker has when used as the corner stone for a distributed-computing solution, there is a potential harm to either the initial objectives of the project, their reach capacity or both.

The Symbioses team is taking additional measures in order to further harden the security of the SaaS instances when deployed over Producers' devices by using a different technology that brings many advantages over a standard Docker container.

9 Development roadmap

9.1 First phase: gamers and block mining

The goals of this phase can be summarized as follows:

- Establish a robust network.
- Build out internal capacity in varying locations for testing, robustness and response of the network.
- Gather new Producers with a simple reward mechanism.
- Establish partnerships to target rewards specific to the gaming community.
- Target gamers and the general public with GPU capacity and unlimited broadband service who have no blockchain or IT knowledge.
- Diversify target platforms and hardware support.
- Let the Symbioses Ecosystem mature in practice.

The aim here is to take the technology as quickly as possible into a production environment, by testing it with a well known application. This will allow the Symbioses team to assess several of the technological challenges of the distributed SaaS architecture in practice. At the same time, it will provide an instant reward to the Producers for providing their resources to the network. This way, the growth of the number of registered Producers is one of the KPIs to measure the success of this early phase.

9.2 Second phase: diversification of the SaaS offering

During the second phase, the focus will be on the following points:

- Handle larger data transfers as part of a SaaS deployment.
- Integrate principles of data-centric applications and general-purpose computing.
- Offer a set of non-trivial applications, e.g., rendering, scientific simulations, deep learning (DNN training).
- Analyze the dynamics of the economic model in practice.

Besides increasing the supported feature set inside the Symbioses Ecosystem, the main goal of this phase in to grow of the number of registered Consumers and deployed SaaS instances. Thus the KPI of this second phase is a combination between the number registered Consumers and the rate of SaaS instances deployed per Consumer.

9.3 Third phase: minimizing latency

Regarding the third phase the objectives are as follows:

- Add the minimum-latency measure in the pricing dynamics.
- Add result validation as an additional service of the deployed SaaS instances.
- Analyze the dynamics of the bidding and cheating-prevention process in practice.
- Validate the encryption mechanisms in practice.
- Organize open hacking challenges with the security community.

The achievements of this phase could be measured as the geographical scope of the deployed SaaS instances. It is expected to see the formation of clusters of Producers' devices working together on the computing aspects of the deployed SaaS solutions. This way it will be possible to compare the impact of the latency measure in the service provided to the Consumers. Additionally, the adoption of the result validation as part of the deployed SaaS instances should also be followed. Finally, the outcomes and level of participation in an open hackaton targeted at finding security holes in the Ecosystem is a KPI of this phase.

10 Conclusion

This paper presented a decentralized SaaS ecosystem called Symbioses.

Symbioses aims at exploiting spare computing resources of off-the-shelf hardware for providing turn-key, readily available SaaS solutions. By focusing on simplicity for the general public, Symbioses offers complex systems as accessible SaaS services that underpin digital products, scientific simulations and computing services in a manner accessible to all.

Symbioses brings together the best aspects of non-interoperable centralized and decentralized networks in an intuitive and simple way. The synergy created by these technologies working together will allow them to potentiate each other in ways their original creators never thought possible. Furthermore, by allowing Producers to join or leave the network at any time, Symbioses provides a constant supply of resources without compromising the robustness of the system as a whole.

The Client connecting a Producer's device with the Symbioses Ecosystem can start generating rewards after two clicks. Its default configuration allows for a fully automated functioning, seamless integration with the availability of the device, and a complete traceability of events. Despite its simplicity, Symbioses emphasizes the absolute transparency towards the end users. By providing detailed access to all created transactions, end users can verify that the received rewards and transaction fees precisely match a particular series of events. This provides a validation mechanism for everyone to employ, giving transparency the place it deserves in an open economy.

With a unique proposition to deliver the next generation of SaaS services, Symbioses is pushing the decentralized aspect of current blockchain projects even further.

In terms of the differentiation aspects between Symbioses and other projects, it is worth mentioning their reliance on software developers to implement applications based on their technology. Symbioses fills this gap by allowing access to turn-key SaaS solutions, developed using leading technologies as mentioned above.

On the computing supply side, some of these projects are focused on users sharing their server infrastructure to provide the computing resources required to run applications. Again, Symbioses' focus on the general public allows anyone to monetize not only servers, but all kind of computing devices, starting with desktop and laptop PCs.

Regarding decentralized storage, some of these projects require a third-party service to act as an intermediary between the smart contracts and the source of data. The Symbioses Ecosystem has built-in support for centralized and decentralized storage networks thus does not require any third party services. Furthermore, because of the dark-pool nature of the SYMIO protocol, there is no risk or need to guarantee that the data come from the right source, since validation and encryption are also an integral part of it.

Needless to say, since these projects heavily rely on Ethereum smart contracts for their functioning, the sharp increase in transaction fees during the last months will have an uncertain impact on their networks. The internal protocol behind the Symbioses Ecosystem keeps practically all of its functions off-chain in order to keep the blockchain costs to their minimum, just creating the completed transactions on-chain for everyone to see.

References

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. A view of cloud computing. *Communications of the ACM 53*, 4 (2010), 50–58.

[2] Bosch, J. Architecture challenges for software ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (2010), ACM, pp. 93–95.

[3] Botta, A., De Donato, W., Persico, V., and Pescapé, A. Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems 56* (2016), 684–700.

[4] Buyya, R., Abramson, D., Giddy, J., and Stockinger, H. Economic models for resource management and scheduling in grid computing. *Concurrency and computation: practice and experience 14*, 13-15 (2002), 1507–1542.

[5] Chen, H., Chiang, R. H., and Storey, V. C. Business intelligence and analytics: From big data to big impact. *MIS quarterly 36*, 4 (2012).

[6] Data Center Knowledge. Here's a map of all the azure and aws regional offerings. `http://www.datacenterknowledge.com/archives/2016/09/21/heres-a-map-of-all-azure-and-aws-data-centers`. Accessed: June 2018.

[7] Delfs, H., and Knebl, H. Symmetric-key encryption. *Introduction to Cryptography* (2007), 11–31.

[8] Fan, J., Han, F., and Liu, H. Challenges of big data analysis. *National science review 1*, 2 (2014), 293–314.

[9] Foster, I., Zhao, Y., Raicu, I., and Lu, S. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08* (2008), IEEE, pp. 1–10.

[10] Frankel, F., and Reid, R. Big data: Distilling meaning from data. *Nature 455*, 7209 (2008), 30–30.

[11] Garfinkel, S. L. Public key cryptography. *Computer 29*, 6 (1996), 101–104.

[12] Godse, M., and Mulik, S. An approach for selecting software-as-a-service (saas) product. In *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on* (2009), IEEE, pp. 155–158.

[13] Golem. White paper. `https://golem.network`. Accessed: October 2017.

[14] Hajibaba, M., and Gorgin, S. A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing. *Journal of computing and information technology 22*, 2 (2014), 69–84.

[15] Harford, T. Big data: Are we making a big mistake? http://www.ft.com/intl/cms/s/2/21a6e7d8-b479-11e3-a09a-00144feabdc0.html#axzz2xV4TdTmn. Accessed November 2017.

[16] iEx.ec. White paper. https://iex.ec. Accessed: October 2017.

[17] Jevons, W. S. *Money and the mechanism of exchange.* D. Appleton & Co., 1876.

[18] Jin, H. Challenges of grid computing. *Advances in Web-Age Information Management* (2005), 25–31.

[19] John Walker, S. Big data: A revolution that will transform how we live, work, and think, 2014.

[20] Kiayias, A., Russell, A., David, B., and Oliynykov, R. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference* (2017), Springer, pp. 357–388.

[21] Kish, L. B. End of moore's law: thermal (noise) death of integration in micro and nano electronics. *Physics Letters A 305*, 3 (2002), 144–149.

[22] Kluger, J. Finding a second earth. *Time (Asia-Pacific edition) 183*, 1 (2014), 30–32.

[23] Komorowski, M. A history of storage costs. http://www.mkomo.com/cost-per-gigabyte. Accessed December 2017.

[24] La Repubblica. Bitcoin for paying the ski pass. http://www.repubblica.it/economia/2018/01/09/news/bitcoin_per_pagare_ski_lift_e_ferrovie_a_sankt_moritz_arrivano_le_criptosciate-186067259/. Accessed: January 2018.

[25] Lohr, S. How big data became so big. http://www.nytimes.com/2012/08/12/business/how-big-data-became-so-big-unboxed.html. Accessed November 2017.

[26] Lu, T., and Chen, J. Research of penetration testing technology in docker environment.

[27] Ma, D., and Huang, J. The pricing model of cloud computing services. In *Proceedings of the 14th Annual International Conference on Electronic Commerce* (2012), ACM, pp. 263–269.

[28] Omega One. White paper. https://omega.one/. Accessed: November 2017.

[29] Pallis, G. Cloud computing: the new frontier of internet computing. *IEEE internet computing 14*, 5 (2010), 70–73.

[30] Peng, D., Liu, W., Lin, C., Chen, Z., and Peng, X. Enhancing tit-for-tat strategy to cope with free-riding in unreliable p2p networks. In *Internet and Web Applications and Services, 2008. ICIW'08. Third International Conference on* (2008), IEEE, pp. 336–341.

[31] Poon, J., and Buterin, V. Plasma: Scalable autonomous smart contracts. *White paper* (2017).

[32] Pouwelse, J., Garbacki, P., Epema, D., and Sips, H. The bittorrent p2p file-sharing system: Measurements and analysis. In *IPTPS* (2005), vol. 5, Springer, pp. 205–216.

[33] Samuelson, W. F., and Marks, S. G. *Managerial economics.* John Wiley & Sons, 2008.

[34] SONM. White paper. `https://sonm.io`. Accessed: October 2017.

[35] Wikipedia. Distributed computing. `https://en.wikipedia.org/wiki/Distributed_computing`. Accessed: October 2017.

[36] Wikipedia. Scalability. `https://en.wikipedia.org/wiki/Scalabilty`. Accessed: November 2017.

Legal

This document is provided for information purposes. This document is not to be considered the final version of the White Paper. The information herein is subject to change until the Main Sale.

Symbioses SA is incorporated in Zug, Switzerland and a member of the Crypto Valley Association. Our legal partners in Switzerland are Ochsner & Associés, a highly reputable firm with experience in the crypto and blockchain sector.

The Symbioses token (SYM token) qualifies as a utility token as such term has been defined in the Swiss Financial Market Supervisory Authority (FINMA) Guidelines on ICOs published on February 16, 2018 ("**FINMA's ICO Guidelines**"). The SYM token purpose is to confer digital access rights to services of the Symbioses platform that can monetize computing device. The Symbioses platform has already been developed and it is functional at the time of the token issuance. The SYM token has no investment purpose whatsoever and is neither designed nor created as an investment product nor a security.

Therefore, the SYM token shall not be considered as a security under Swiss laws and regulation (including as a share, participation right, bond, structured product, collective investment schemes or derivative). The SYM tokens are not subject to the authorization or supervision by the FINMA.

The Symbioses Ecosystem works as a monetization platform for computing device. The SYM tokens are the exchange unit (and will also be used as a monetizable reward on active service payable by the consumers) exclusively within the Symbioses Ecosystem. The issuance of the SYM tokens, the main purpose of which is to provide access right to services of the Symbioses platform, should not be subject to the Swiss Anti-Money Laundering Act (AML) given that the payment function is by default an accessory function of the SYM token and given that it is exclusively used within the Symbioses Ecosystem and the Symbioses platform. As consequence, the SYM token shall not be considered as a mean of payment (that can be used on a third party's platform to acquire goods or services) under the AML or be deemed to be a "payment token/cryptocurrency" as such terms have been defined in the FINMA's ICO Guidelines.

SYM tokens will be issued on the Ethereum-based ERC20 token. This allows token holders to easily store and manage their SYM tokens using existing solutions including Ethereum Wallet.

The smart contracts controlling all aspects of the SYM token, including the multi-sig wallets that make trust-less escrow possible, will be implemented by the Symbioses technical experts and partners. These smart contracts, in turn, will follow an auditing procedure for security and correctness.

Multiple risk factors of the project include, without limitation, defects in technology, legal or regulatory exposure, market volatility & liquidity, corporate actions, or the unavailability of complete and accurate information.

Neither a solicitation, an offer, a recommendation nor a contract. This document is for informative purposes only and does not constitute, nor is it intended to be, a prospectus within the meaning of applicable Swiss law (including art. 652a or 1156 of the Swiss Code of obligation) or an offer to sell, a solicitation of an offer to buy a SYM token or a recommendation from Symbioses and its affiliates.

This White Paper is not a contract and does therefore not legally bind Symbioses or its affiliates.

Selling restrictions. Potential subscribers should note that participation in the SYM token sale(s) is subject to limitations imposed by applicable securities laws in various jurisdictions. The regulation of token and more generally blockchain is currently in evolution. Therefore, potential subscribers should be aware that some jurisdictions may introduce new regulation or adapt their current regulation and/or definitions / categories of tokens, including Switzerland. Below are examples of some limitations:

Singapore Investors from Singapore shall not be able to participate in our offering.

United States The SYM tokens will not be registered under the US Securities Act of 1933, as amended from time to time, or under any securities laws of any state or other jurisdiction of the United States. Therefore, from the United States, the SYM token may only be made available to purchase by a maximum of 99 verified "accredited investors" (as defined in Regulation D under the US Securities Act of 1933).

The list of jurisdictions above where limitations apply is not exhaustive. Other examples of selling restrictions and limitations may be outlined in the upcoming SYM Token sale's documentation.

This White Paper may not be used in any such jurisdictions or in any circumstances in which this White Paper, its content and/or the SYM tokens are not authorized for distribution and/or unlawful.