SPARKSTER

# Build and Run Decentralized Software in Plain English.

**We're building the world's fastest Decentralized Cloud for Smart Software. A specialized Blockchain designed specifically to replace Cloud Computing and execute tens of millions of transactions per second. By enabling you to build Smart *Software* in plain English, we empower you to create the future.**

# Table of Contents

# Premise

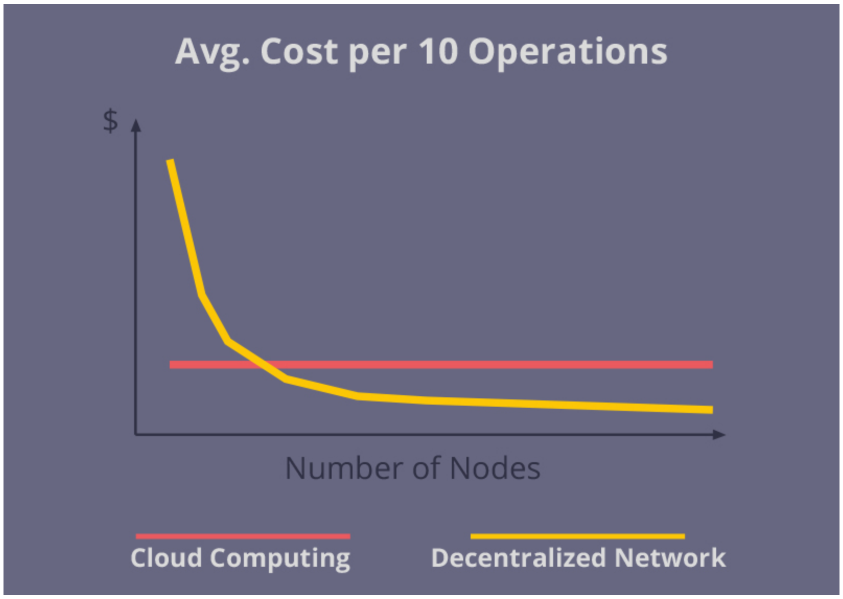
We're all Sparksters. We all see better ways of doing things. With the right tools at our finger tips, we can finally build solutions that improve efficiency, solve meaningful problems, and bring about a better world.

# Abstract

The Sparkster Decentralized Cloud

We facilitate the 'decentralized execution of Smart Software', an evolution on Smart Contracts, across a network of cell phones, notebooks, laptops or other personal devices held by our network of miners. These miners are rewarded in SPARK tokens for contributing spare capacity on their personal devices to the Sparkster Decentralized Cloud.

We envision a future in which decentralized networks can be far cheaper than traditional cloud computing from the likes of Amazon®, Microsoft®, Google® and IBM®. Given that cloud providers pay for: rack space, cooling, dedicated hardware, networking infrastructure, backup power, and support personnel; personal devices have near zero marginal cost per operation relative to cloud providers.



Hence, if each contributor is afforded the opportunity to set their own price for the use of their device, over time, natural market conditions will drive the cost of a decentralized cloud below that of traditional cloud computing. This presents an opportunity to collectively disrupt the $246 billion-dollar cloud computing industry (Darrow, 2017), and distribute a significant portion of that revenue to participants in the Sparkster Decentralized Cloud through SPARK tokens.

With traditional cloud computing, no matter how many nodes are added to the network, the average cost per operation remains constant, enabling one to scale with near linear costs. However, with a decentralized network of inexpensive nodes, as more nodes such as cell phones are added to the network, the cost per operation declines, such that the cost per operation becomes less than traditional cloud computing.

Companies wishing to run software, will pay for the execution of software on the Sparkster Decentralized Cloud in SPARK tokens. Large companies are likely to adopt a public Decentralized Cloud, like they've adopted Amazon® AWS and Microsoft® Azure, if we can satisfy the following requirements:

1) **Privacy** – all data stored and accessed is deconstructed into transaction fragments, encrypted and disseminated across a network of nodes such that it cannot be of any meaningful value when viewed on a public ledger.

2) **Security** – any devices operating on the network and interacting with client data are hardened such that any form of tampering to access data, while it is being processed, prevents the device from accessing the network.

3) **Performance** – transaction latency must be comparable with traditional centralized cloud computing, but transaction throughput must exceed **Tens of Millions** of Transactions per Second.

4) **Material Cost Savings** – Cost savings must be sufficient to justify the efforts involved in redesigning software

5) **Architecture Transparent** – Software designed in such environment must easily and automatically operate in a Decentralized environment, without requiring any knowledge or architectural design considerations for Decentralized environments by the software author.

6) **Blockchain Agnostic** - Users must have the freedom to build software that can interact with any existing blockchain of their choice – all in plain English. Today, we have existing integrations with Ethereum and IOTA, but others such as NEO, Cardano, and Stellar are on our roadmap.

7) **True Decentralized Oracles** – Integrate anything, whether it's AI, IOT or your favorite API. Most importantly, integrate anything without any code – in plain English.

The aim of this paper is to describe a purpose built decentralized blockchain design that addresses these seven concerns and presents a legitimate and practical solution, using existing but proven technologies, that collectively will give rise to a trustable, public, Decentralized Cloud facilitating the future of Cloud Computing.

## What is Sparkster?

Today, Sparkster is a platform of empowerment. We empower you to bring your ideas to life by building software without learning how to code. Use intuitive drag and drop interfaces and the ability to define business logic in plain English to build anything imaginable. Focus on WHAT you want to build, and let our platform transform your ideas into working software in seconds. Inspired by MIT Scratch, we take the position that if young children can build games with simple building blocks, then we can empower everybody to build enterprise grade applications in a similar manner.

Innovation in the 21st Century is almost exclusively dependent on software development. One need not look far for a plethora of examples: emerging companies such as Uber, Airbnb, and Instagram are exclusively software companies. The Sparkster platform aims to democratize access to innovation, and make innovation in the 21st century accessible to everybody.

The Sparkster platform is to software what Uber was to the taxi industry - a seismic disruption. Competitors such as Microsoft®, SAP®, Oracle® will have their business models fundamentally disrupted by the advent of user generated software, empowered by the Sparkster platform. We present the fastest, cheapest, easiest, most democratic and personalized way to build software, period.

Now, we're building the world's fastest Decentralized Cloud that enables people to execute this software across a network of our community's cell phones. This combined packaged is not only the cheapest, most democratic and fastest way to build software, but to run software too. A high performance decentralized cloud, as proposed by Sparkster, is necessary to mainstream blockchain adoption.

## Our Vision

Ideas are the currency of the future. We envision a future in which everybody can bring their ideas to reality. In doing so, they create financial independence for themselves, while contributing to their society in the process.

## Mission Statement

We aim to empower the dreamer within each of us to innovate. We will achieve this by democratizing access to the tools necessary for innovation in the 21st century: software development.

## Selling Ideas, not effort.

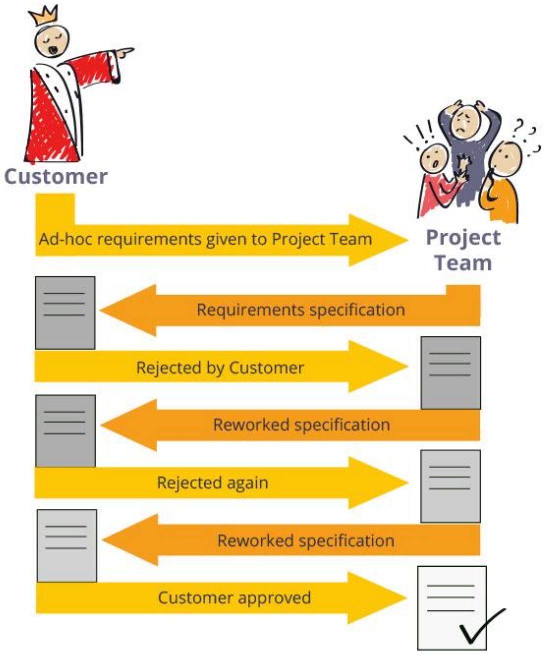
Ideas are ubiquitous, they’re not limited to certain social classes, education or family upbringing. Ideas are not the privilege of the few, but rather the prerogative of everybody. In our everyday lives, we can all see better and more efficient ways of doing things.

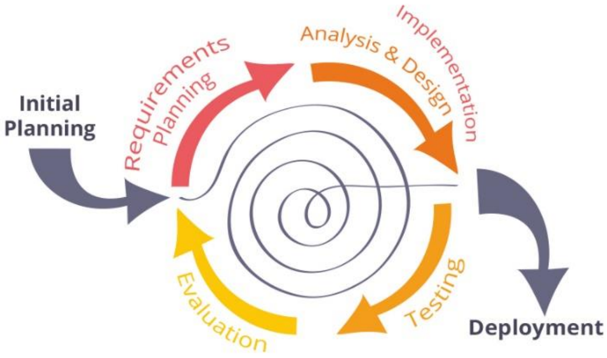
The Sparkster platform empowers you to bring these ideas to life, and unshackles you from the limitations of your job, financial disposition, or regional and cultural impediments. The Sparkster Marketplace empowers you to sell these ideas, find patrons and fans to help bring about the change you seek, and create sustainability through paying customers. The Sparkster platform presents a path to liberation and a path to financial independence.

## Problem Statement

Enterprises, startups and individuals attempting to adopt AI, IOT and Smart Contracts are entrenched with impediments to harnessing these innovative technologies. Namely, that existing enterprise technologies are architecturally incompatible, IT departments move too slowly, and qualified people familiar with these technologies are not available in sufficient supply.

### Iterative Software Development Lifecycle





The typical process in which software is currently created is rather convoluted. An expert that understands the intricacies of a business process, describe their requirements to a member of the IT team. The IT team are technicians that usually have limited understanding of the actual business. The IT team attempt to document what they understood from their meeting with the domain expert. Through an iterative process of rejection and rework, they finally and successfully document what their customer (the domain expert) wants. Then they begin a process of analyzing how they might achieve this, before they begin development.

At this point, several weeks if not months have passed. After some testing they engage the customer again. They soon discover that their understanding of the signed off document and their customer's perception of the document are quite different. Usually to their dismay, they discover that they've developed something rather different to what was desired. Hence, once again, the iterative process perpetuates as they try and close the gap in interpretation of the approved document. This process of software development, often called the Software Development Lifecycle (SDLC), is slow and painful (Berry, 2003).

Technology is developed to meet a need at a point in time, these needs change as market conditions change and businesses respond with new strategy. In order to update the software to meet these changing demands, professionals need to be consulted at great cost to make even the simplest of changes to software. Not only is there often a large cost associated with this, there is a significant time-lag in requesting changes, from the initial consultation in requesting a change, to explaining the new requirements, to development, testing and implementation of the changes. This inhibits technological advancement and instead we make do with the status-quo.

## Enterprise Software is Defunct

The reality is that almost all businesses today are running software that is absolutely incapable of interfacing with the technologies of the future. No matter who provides your business software, whether it's SAP®, Oracle®, Microsoft®, or it's homegrown, integrating Internet of Things ("**IOT") devices**, **Artificial Intelligence ("AI")**, or **Smart Contracts** into existing business processes will not be possible without other technology platforms. All this software will have to be either ripped and replaced or integrated into a new technology platform for businesses to continue to be competitive and take advantage of these revolutionary technologies (Clark, 2017).

We believe the existence of SAP® Leonardo is evidence of our premise, that business software in use today need to be replaced in order for businesses to take advantage of the technologies of the future, and to continue to be competitive. For example, IOT powered machinery in a factory cannot simply be integrated into existing business processes that replenish raw material inventory to a production line. Entirely new software architecture is necessary to consume the data being generated by these machines, and use this data to make real-time autonomous replenishment decisions. SAP® recognize this, and hence, built Leonardo - a platform for software developers.

## Impediments to Innovation

Not only are traditional ERP's architecturally incompatible with the technology of the future, the real problem is that the vast majority of software developers lack the technical knowledge to build these new technologies (Kalla, 2017). Enterprises today are struggling to find sufficient talent that can help them integrate AI and IOT into their enterprises to help bring about the benefits of these technologies materialize, let alone finding the talent to integrate Smart Contracts. This presents a significant challenge to enterprises in how they might innovate, and build new business models that harness the power of AI, IOT and Smart Contracts.

Similar to Enterprises, often, the greatest challenge that Startups face is not a lack of funding, but rather the lack of software developers available to make their ideas a reality (Aleksandrova, 2018). This too presents an impediment to innovation, as startups struggle to bring about innovative products and services that contribute value to our society.

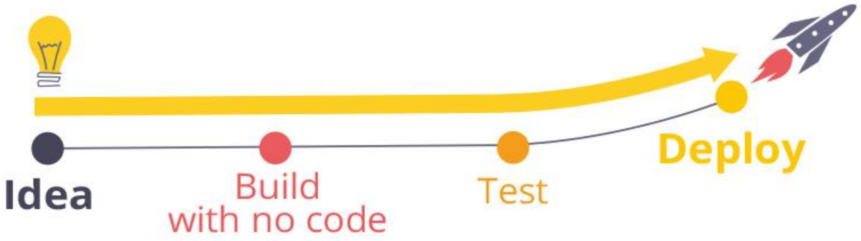
As individuals, the cryptic nature of computer programing languages prevents the majority of people from being able to materialize ideas. While much has been done to make technologies accessible via API's, these API's are still within the purview of software developers. Hence, for someone not fortunate enough to be versed in a computer programming language, manifesting their ideas most often requires significant capital, which is often difficult to accumulate. Hence, the inaccessibility of technology presents an impediment to innovation.

Developing technology has been the preserve of those that had the ability to write computer code and translate what are in essence very simple functions into cryptic language that can be understood by computers. We believe this inhibits technological development and stops those with great ideas truly realizing their potential, and building the future.

## Environmental Concerns

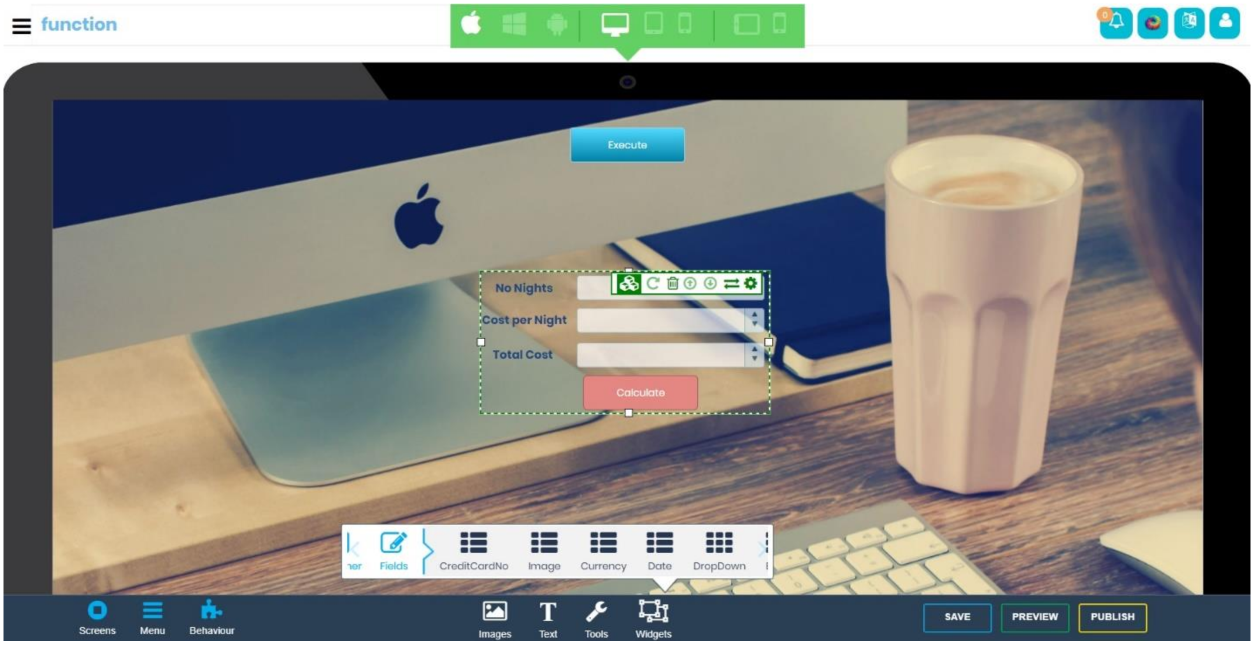
A Greenpeace report titled Make IT Green: Cloud Computing and its Contribution to Climate Change, suggested that cloud computing consumed 632 billion kilowatt hours of energy in 2007 and this expected to increase to 1,963 billion kilowatt hours by 2020. The associated CO2 emissions that are anticipated to reach 1,034 Mega Tonnes (Schmidt, 2010). Such high levels of CO2 emissions are clearly unsustainable.

# Solution: Software built for you, by you.

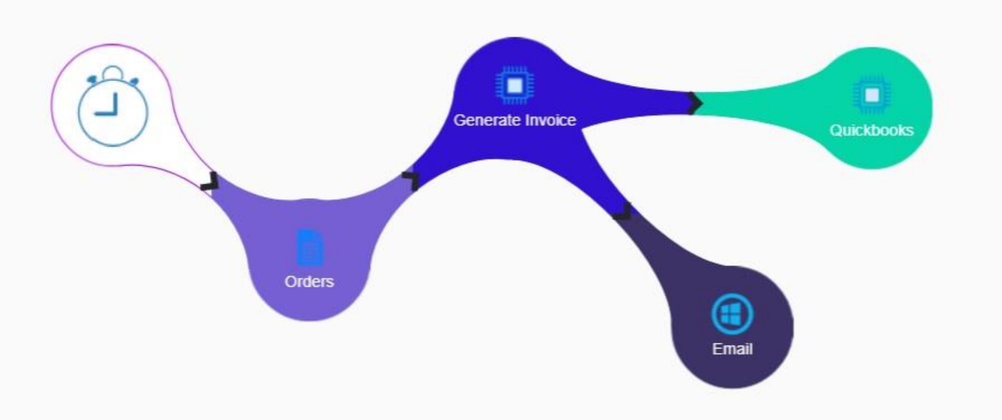


What if software could be defined in plain English? What if this software was defined on a platform architected specifically to integrate innovative technologies such as AI, IOT and Smart Contracts? What if taking advantage of these new technologies was as simple as dragging and dropping a block? That is the Sparkster platform, and it's ready to use today. We empower you to create the future, we empower you to bring your ideas to reality.

The Sparkster Platform today:



The Sparkster Platform empowers the people running the business with the tools to build the business process of the future that will ultimately keep them competitive. Sparkster makes software development accessible to the 99% who do not know how to code, and don't want to learn. Enterprises can yield the power of their human capital, ignite a new culture of innovation and inclusive participation that kindles a passionate to tinker, a passion to manifest, a culture of continual improvement among their knowledge workers, among the very people that the know the business intimately.



Respond to changing business circumstances instantly, build the future on your terms, that's the Sparkster platform. Build software designed for you, by you – at least 100x faster, cheaper and easier than any other alternative.

Building decentralized software is difficult and currently prohibitive because of high gas prices and long execution times. What if decentralized software could be built without any knowledge of decentralized architecture, and run on cheap devices such as cell phones, and IOT devices, all deployed with just the click of a button? What if "mining" with unused cell phone capacity became as simple as installing an app? What if in the process of making software execution cheaper and more democratic, we could drastically cut CO2 emissions? That's why we need the Sparkster Decentralized Cloud.

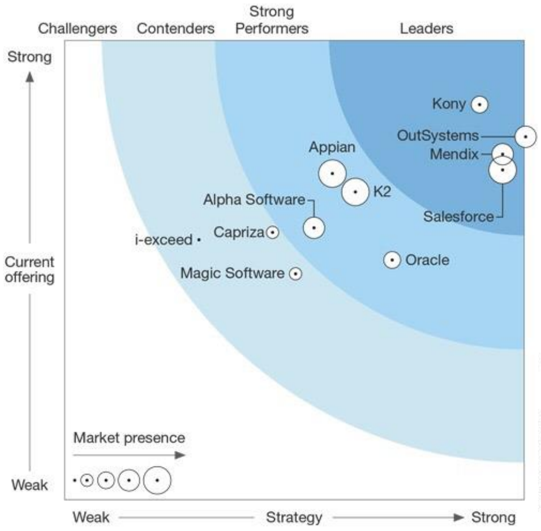
## Background

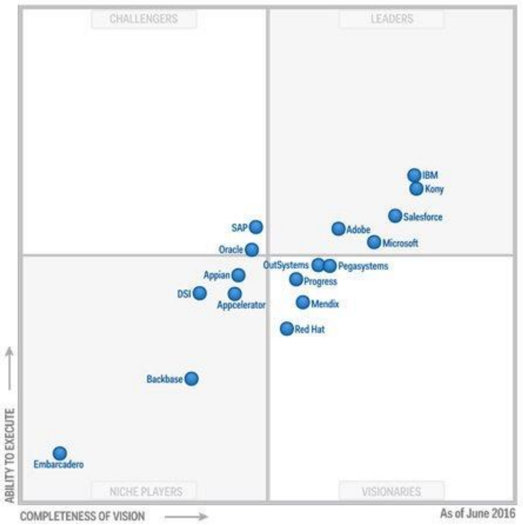
Having spent the last 14 years working with software engineers to design and build ERP software for a fledgling startup, Sparkster was born from extreme frustration and the answer to our founder exclaiming "just give me the tools, and I'll do it myself!" Hence, the idea emerged. Empowering our managerial team with the tools to define just business logic, without worrying about the technical coding, was the guiding principle with which the platform was developed. From there began a 6 year journey in R&D to find a way as to how such a platform might be built.

Today, the Sparkster platform is not simply production ready, it's enterprise ready. Designed with cloud principles, and utilizing managed and monitored Docker containers, the platform autonomously scales horizontally to support near unlimited throughput from users and devices. In addition, security is at the forefront of the design and architectural considerations of the Sparkster platform. Implementing AES 256 bit encryption and OAUTH 2.0 between all container interactions, security isn't an afterthought, it is integral to the very architecture of the Sparkster platform.

Our product has garnered interest from many enterprise company, which are some of the largest technology companies in the world. So far, we have signed official partnerships with ARM® and Libelium®.

## We're not alone





In the last few years an entire market has emerged of participants who share our concerns that software development in the traditional sense is unnecessarily complicated, burdensome, and time consuming. According to Gartner's Magic Quadrant, the low code / no code segment consists of household names such as IBM®, Microsoft®, and Salesforce®**.** We commend and support their efforts to remove inefficiency from the process of software development, but we disagree with the majority of the existing industry in one major way: we're not trying to make software development easier and faster for JUST software developers.

Rather, we want to see software development become accessible to everybody. We're focused on the 99%, where many participants in our industry are focused on the 1% who are software developers. Therefore, we believe we lead the industry in the realization of this vision. We invite you to compare our platform to those that exist in the "Magic Quadrant" and determine for yourself whose platform is easiest to use. In addition, we're far more than a platform to build nice interfaces or mobile apps. Using our revolutionary technology, you can build AI, IOT, Ethereum Smart Contracts, and IOTA Smart Transactions, all without a single line of code. We're years ahead of the competition.

# Business Overview

## Core Value Proposition

Our innovative technology results in two specific outcomes that form the core value of our platform: empowerment of team members to build software based solutions, and disruptive cost reduction.

Disruptive cost reductions are achieved by drastically reducing the amount of time spent building, and refining software and the cost of running software. Using our innovative tools, we have demonstrated that applications can be built in one hundredth the time relative to traditional application development. In addition, since knowledge of a specialized programming language is unnecessary, the cost of using skilled software developers can be drastically reduced or even eliminated. These factors combined lead to disruptive cost reductions relative to traditional software development.

Existing manual or paper based processes and onerous spreadsheets can now be replaced with custom software applications that can be built in-house, in minutes, not months. Team members are empowered to build such solutions using our innovative technology, as they're able to design such applications visually, and define all business logic in plain English. Deployments can be made with the click of a single button, never having to worry about how any of the plumbing behind the scenes. This disruptive technology emancipates middle management, and gives them the power and tools to improve business processes, whist bringing meaningful positive impact to the organization.

Our Marketplace brings both core value propositions together. Not only does it allow innovative authors to publish their high-level compositions for others to use, but our Marketplace empowers contributors to earn from their efforts by selling their creations. Earning an income from sharing one's innovation is at the heart of our mission. In addition to selling one's ideas, in the future, we aim to facilitate decentralized execution of software built in our no code platform. The participants on whose devices these functions will execute, will be compensated in SPARK tokens.

Running software defined in the Sparkster Platform on the Sparker Decentralized Network facilitates substantial reductions in the cost of running applications at scale. Traditional centralized cloud providers such as Amazon® and Microsoft® enable you to add cloud servers at a predictable cost per cloud server. But transitioning to a cost per transaction, rather than a cost per node, enables a company to run software without having to anticipate demand surges or respond to future demand while looking in the rearview mirror.

Imagine having the entire AWS cloud available to your business whenever you need it, always on, without reserving any instances. That's the Sparkster Decentralized Cloud. In addition, what if this were drastically cheaper than AWS and Azure because the fixed costs of running such as a service, like rack space, cooling, dedicated hardware, networking infrastructure, backup power, and support personnel, don't exist in a Decentralized Cloud. Companies pay for these transactions using SPARK tokens, and miners make their devices available as resources to the Sparkster Decentralized Cloud in exchange for SPARK tokens.

## Scale and Revenue Strategy

To achieve scale, Sparkster will make its Platform free to use in a personal capacity. While this will be limited to a certain number of transactions per month, in a personal capacity, this limit can be lifted by referring other users. The use of our Platform in a commercial capacity will be subject to ongoing fees in the form of user licenses, transaction fees, device fees, storage fees etc. We only get paid when you do. By engaging users and listening to feedback, we aim to make our service irreplaceable, and hence ubiquitous.  Inspired by the BYOD movement, we anticipate that our user community will solve problems that they encounter, in both their personal and professional lives, using the tools from within our platform, a platform they're familiar with.

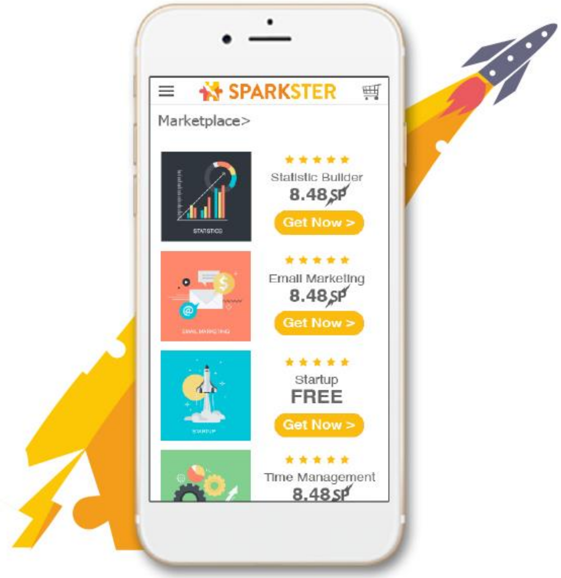
The Sparkster Platform and Decentralized Cloud are complementary products. There's significant interest today from companies large and small that want to generate new business models atop decentralized blockchain technology. The Sparkster Technology Stack presents the cheapest, easiest and fastest way of achieving this. Software can be built with Sparkster at least 100x faster and easier than traditional means. Hence, the Sparkster Platform will generate demand organically for the Sparkster Decentralized Cloud.

## Viral Growth

The Sparkster Platform will be free to use for individuals, however, there will be transaction quantity limits. Increases in personal use limits can be attained by referring a friend to our platform, much like how one gets a free ride on Uber or increase in storage space on Dropbox. Word of mouth referrals are a critical component of our growth plan.

## Network Effects

While the Sparkster platform aims to provide the fundamental constructs (e.g. IF conditions) necessary to build software applications, building applications using fundamentals is not always optimal. We see a significant opportunity for our user community to create higher level constructs (i.e. sales tax calculators) that can then be utilized by the other members of the community to build high level applications (i.e. accounting software). In effect, while we provide simple building bricks, members of our community can use these bricks to construct their own cars, houses etc.



Via a peer to peer Marketplace, these innovative users can provide or sell these higher level constructs (i.e. sales tax calculators or accounting software) to other platform users. This not only increases user engagement, it presents a model where the value that our platform provides to users increases at an exponential rate. Consumers of these higher level constructs can compose them into applications that bring about new businesses, innovative solutions to unsolved problems, and provide income generation opportunities for those willing to author and publish on our platform through our Marketplace.

One may elect to execute their software on our decentralized computing network. As we gain more participants on this network, the computing cost will decrease materially, as participants compete to earn SPARK tokens from their unused cell phone and IOT device capacity. Not only does this democratize software execution, it's present a viable and cost effective alternative to services like AWS and Azure. In the long run, given the negligible marginal cost of making ones cell phone available for decentralized computing, we believe this give rise to a disruptive software execution model relative to cloud computing. In addition to the significant cost reductions relative to cloud computing, as well as the democratizing of software execution, there are very real and impactful implications on our environment from eliminating cloud computing.

## Strategic Partnerships

Strategic Partnerships are mutually beneficial, and much like vertical markets, they present an opportunity to deliver meaningful value via our platform to an existing community of users and hence grow our user base. In addition, these partnerships provide a cross pollination opportunity, where each community benefits from interacting with one another, not just by sharing ideas, but collaborating to create solutions to existing problems.  We also see the strength of our strategic partnerships as a validation of not only our concept, but also our technology. Established organizations that would be key users of our technology have shared our vision and want to partner with us to help bring this technology to the masses.

Thus far, we have signed official strategic partnerships with Libellium, an industrial sensor and gateway distributor, as well as ARM (one of the worlds largest computer chip designers). Building deep integration with these strategic partners presents significant opportunities to expand our reach, build specific integrations that serve the needs of our partner's community, and hence grow our own customer base as well as giving us access to market leading specialists who we can leverage.

For Libelium, we've built gateway software that facilitates seamless connectivity to our cloud from a device, as well as a simple rules engine that facilitates local data processing to determine when and what data should be sent to our platform. Once data arrives in our platform, this data can be used by any other application composed on our platform to make decisions within a business process or application, orchestrate actions in the real world (i.e. turn on sprinkler system) or simply provide a meaningful graphical dashboard.

With ARM, we've developed the ability to build IOT devices, without writing a single line of code. Users have the ability to use drag-and-drop blocks to define device logic and hence develop specific IOT devices. A very simple example might be a device that detects the temperature in a greenhouse and turns on a warning LED if the temperature rises or falls outside the normal operating range, and uses a relay to open or close a window. Such a device could be programmed, controlled, and monitored without writing a single line of code. Our platform will generate the device code in C, produce a binary, and deploy to a local ARM, Arduino or ESP device completely autonomously at just a click of a button. In addition to deploying the logic itself, we also build the logic for connecting to our platform using MQTT (an IOT communication protocol), implement appropriate device security and generate a virtual representation of the device in our platform. The virtual representation facilitates bidirectional secure dataflows to our platform where that data can be used to build IOT powered applications. In addition to powerful IOT applications, dashboards can also be built in just a few clicks. For example, if you wanted to monitor the temperature of the greenhouse and the state of the window in real-time, such a dashboard could be built in minutes, without a single line of code.

## Vertical Markets

We are developing targeted solutions for specific vertical markets based on our assessment of the potential for growth and the value that such a vertical market may contribute to our community. Thus far, we have identified two vertical markets, in which we have built deep integration: Internet of Things (IOT) and Cryptocurrency Smart Contracts.

We see significant opportunities in both markets, in terms of user growth and contribution of value.

## IOT

According to market participants, the IOT industry represents a $20 Trillion opportunity by 2025. We feel this transformative technology will fundamentally disrupt existing business operations, as existing business process and technology are neither architected nor designed to take input from devices in the physical world when making decisions or executing these business processes. A practical example might be one where machines themselves can slow-down or speed-up deliveries of raw materials being picked from the warehouse and delivered to the production line, to avoid raw material congestion on the factory floor. Today, ERP systems are unable to handle unscheduled machinery maintenance in a factory in an autonomous way. Using existing technology, human intervention is needed to deal with such a scenario in many ERP in use today. As such, we've made devices a primary focus in our platform as a fully integrated yet sufficiently generic concept that can be adapted to suit a plethora of partner integrations and implementations.

## Cryptocurrency Smart Contracts

Whilst we see the vast applications and disruptive opportunities created by Smart Contracts, we perceive a fundamental problem in this burgeoning industry. We neither expect lawyers to learn to write code, nor do we expect lawyers to be entirely replaced by autonomous smart contracts. As such, today, Smart Contracts are inaccessible to the majority of potential users, and hence this has impacted commercial adoption. Our platform is uniquely positioned to address this problem by enabling Smart Contracts to be built in plain English with a focus strictly on defining the actual business logic the contract is expected to execute; with our platform handling the translation into a functioning SmartContract. As such, we've integrated the ability to generate Ethereum Smart Contracts and IOTA Flash Channel Transactions via our platform. Once authored, our platform generates the appropriate code for such a Smart Contract to execute on the respective network which means that knowledge of the implementation details of Smart Contracts is completely unnecessary.

We've built and tested an insurance Ethereum Smart Contract use case, in which a buyer purchases medical insurance, but prior to the policy being issued, the request is forwarded to an underwriter for approval. Once approval is granted, the client is emailed a copy of their policy document. The entire scenario was built without a single line of code. Note, that this scenario was simulated, and we did not work with any actual medical insurance providers or underwriters.

We've also built a Smart Contract that issues tokens whenever payment is received. This "ICO Smart Contract" was written in plain English and executed on the Ethereum network. Feel free to check out our demo video of this contract in action.

### Barriers to Entry

We aim to create barriers to entry by patenting our technology and our innovative ideas. We have engaged a leading London based technology specialist law firm, Bird and Bird, to provide an opinion of the patentability of our technology and our ideas. Based on their analysis, we have engaged Kilburn & Strode LLP to write our patent applications which are due to be filed in the first part of Q2 2018.

Being first to market with such an innovative technology presents its own opportunities too. We aim to build our community at such a rate that any new entrants will not be able to scale to the point that they represent a material threat.

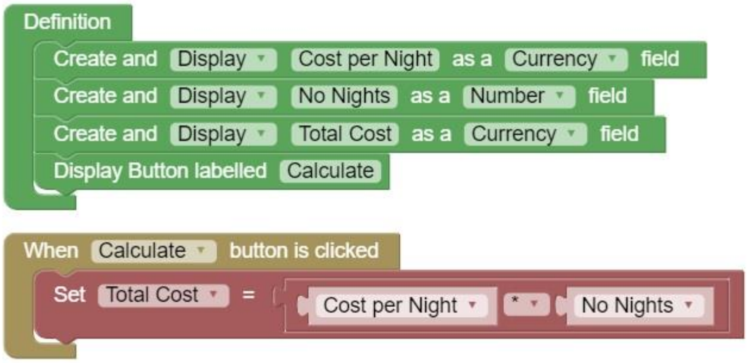
## GDPR Considerations

GDPR concerns could be addressed by limiting the Storage Nodes that can join a particular cell. Suppose a limitation were introduced that the IP address of the Storage Node must be European, this could address the EU regulatory requirements that sensitive data does not leave its jurisdiction. Our architecture lends itself uniquely to solving such a problem. With regards to the right to be forgotten, this is a particular challenge for blockchain environments given that data is immutable. There are two solutions we propose for this. Since global state is maintained architecturally by the ordered distributed hash tables a delete command would simply render historical data inaccessible to applications built atop the Sparkster Decentralized Cloud, without impacting immutability. In addition, we can re-encrypt this deleted data with a platform key that would provide an additional guarantee that this data would be inaccessible to the client and any applications built atop the Sparkster Decentralized Cloud. We shall wait and see how legislations evolves to address Blockchain environments and adjust our strategy appropriately.

# Product Overview

## Sparkster Platform

The Sparkster Platform enables you to build software without writing any code, by simply dragging and dropping blocks that are in plain English.



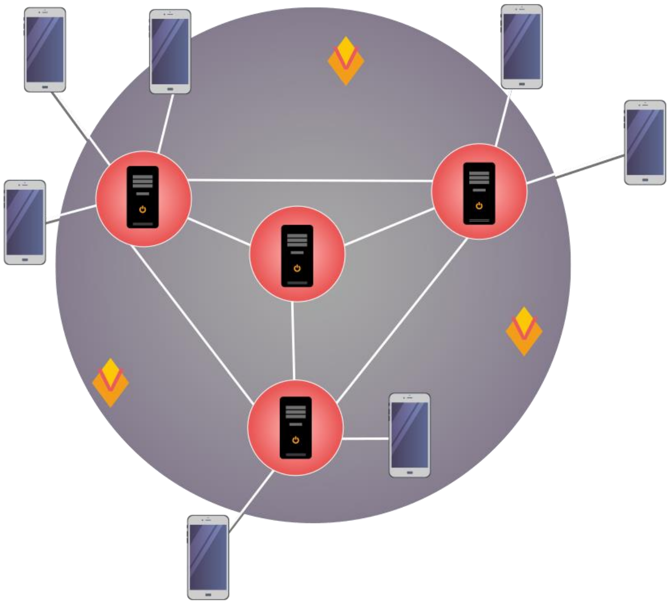
```
Definition
    Create and Display ▾ Cost per Night as a Currency ▾ field
    Create and Display ▾ No Nights as a Number ▾ field
    Create and Display ▾ Total Cost as a Currency ▾ field
    Display Button labelled Calculate

When Calculate ▾ button is clicked
    Set Total Cost ▾ = Cost per Night ▾ * ▾ No Nights ▾
```

Snap together blocks that describe **WHAT** you want to create without worrying about **how** it works. Building software becomes as easy as playing with Lego. Cut the time to make software from months to days. Easy enough that even children have built software in minutes.

Software generated with the Sparkster Platform is specifically architected for high volume, distributed applications. Horizontally scalable, implementing streaming pipelines, the platform is architected to support the veracious demands of IOT and AI environments. Security is architected at the core of this Platform, implementing AES 256 bit encryption between every interaction and OAUTH 2.0 authorization and authentication, the Sparkster Platform generates Enterprise grade software without any knowledge or concern for software design or necessity for architectural awareness.

Software generated by the Sparkster Platform is architecturally independent, meaning that software generated by the Sparkster Platform can be compatible with the architecture of the Sparkster Decentralized Cloud. The client elects where to execute their software. However, in the long run, the Sparkster Decentralized Cloud will prove far cheaper than the Sparkster Platform as an execution environment.

## Sparkster Decentralized Cloud



The Sparkster Decentralized Cloud will facilitate the execution of *Smart Software* defined using the Sparkster No Code Platform, in a decentralized fashion. A miner can install the Sparkster "mining" app on their cell phone that will provide user generated Smart Software with a local "Sparkster Runtime Environment (SRE)." The SRE would localize the architecture necessary for the execution and decentralized coordination of functions, workflows etc.

Companies and individuals wishing to have their applications run in a decentralized fashion place a bid on the Decentralized Exchange and stake SPARK tokens in accordance with the amount they're willing to pay. This is a free market, clients are free to place any bid, and miners are free to ask any amount they wish. Matching bids and asks (less a platform access fee) will be cleared and then function/workflow execution will begin.

These tokens would then be issued to the "miners" for the use of their cell phone or desktop processing power. In addition to Compute Nodes, Storage Nodes can join our network. These nodes would be responsible for decentralized data persistence (documents), and for maintaining a repository of the components designed by our community of users. In the event that data needs to be persisted in a decentralized fashion, nodes would be paid their appropriate fee in SPARK tokens (less a platform access fee) for long term data persistence. Again, clients are free to place any bid, and miners are free to ask any amount they wish. Matching bids and asks (less a platform access fee) will be cleared and then data persistence will initiate.

Miners, both Compute and Storage, stake tokens as collateral to ensure they operate honestly. Verification nodes validate transactions generated by Compute nodes and persisted by Storage nodes. If these verification nodes detect any illicit activity, they cease the stake advanced by these miners as a bounty for detecting illicit behavior. Herein lies the system of checks and balances that gives rise to a safe and secure public Decentralized Cloud.

In all instances, compute execution or data persistence, both data at rest and in transit will be fully encrypted. Neither the compute nodes nor the data nodes will be able to determine what they're executing or persisting, nor for whom. This is achievable given all execution will occur in memory on a cell phone. All data stored to storage nodes will be fully encrypted and hence will be private to the owner of the data.

# Smart Software

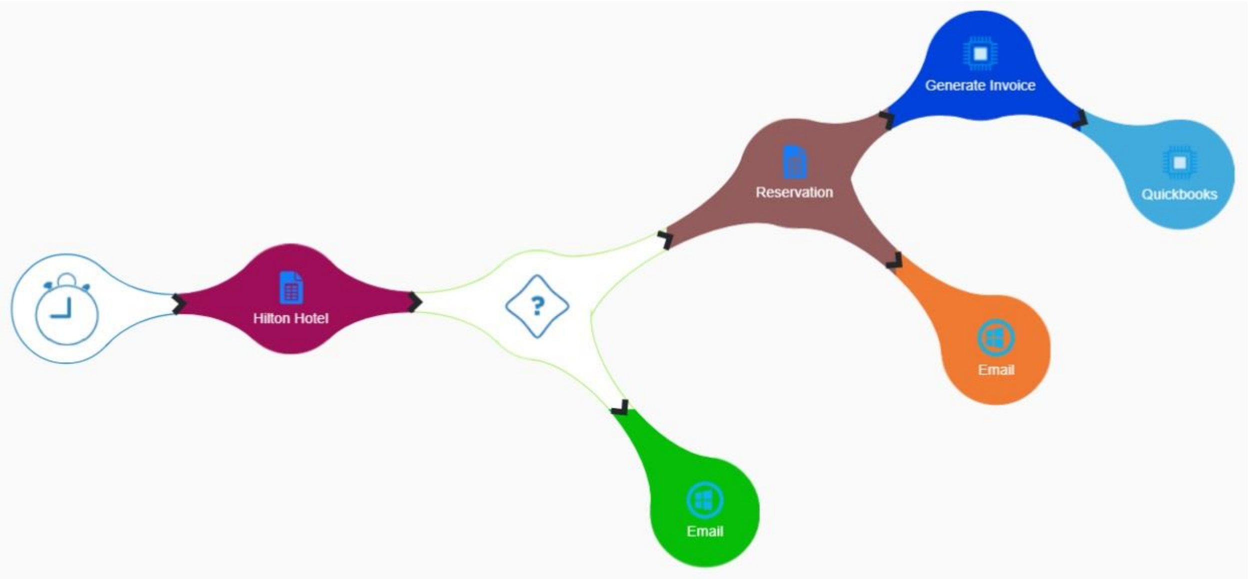
The Sparkster platform is an environment in which software can be designed in a drag and drop fashion, and business logic can be defined in plain English.

Smart Software is distinct from Smart Contracts in that they natively incorporate abilities that Smart Contracts cannot easily provide. Namely:

1. HTML responsive interfaces that are desktop and mobile compliant by default.
2. Provide REST API's to facilitate integration into existing business processes and traditional software.
3. Ability to call any external API and handle responses in a deterministic fashion through an automated Oracle processing layer.

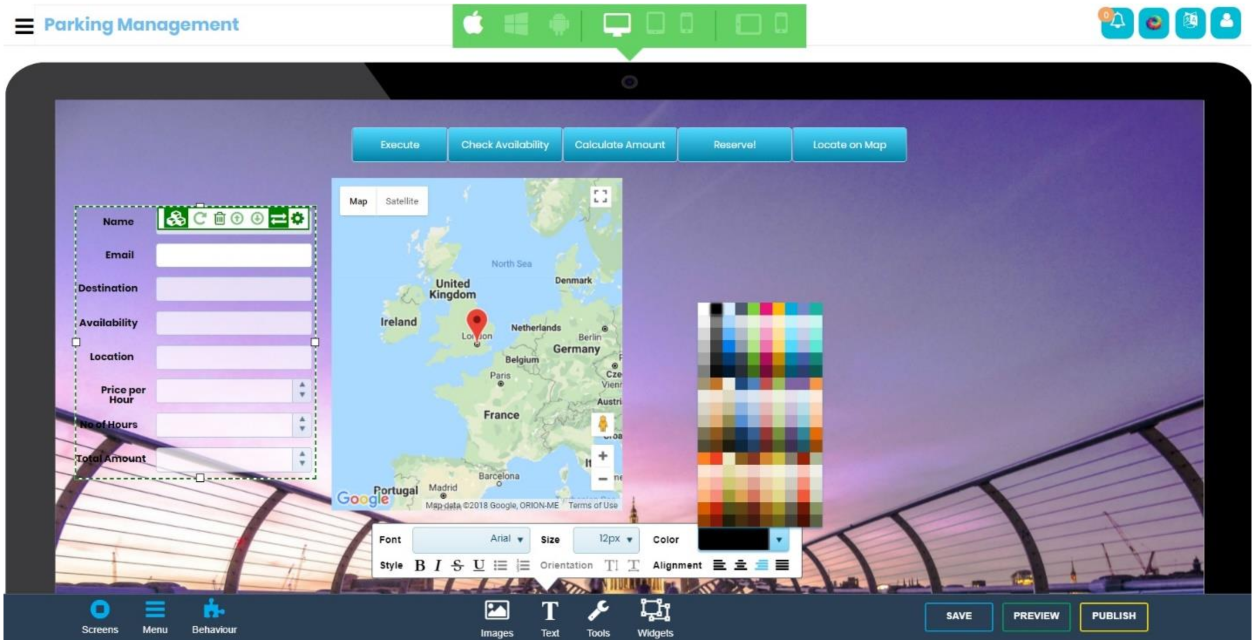
Below we will outline the various components that constitute Smart Software.

## Flows



In the Sparkster platform, Flows constitute the definition of a piece of software. These flows are an assemblage of other core components in the Sparkster platform, including Functions, Documents, Devices, Integrations, AI, etc.

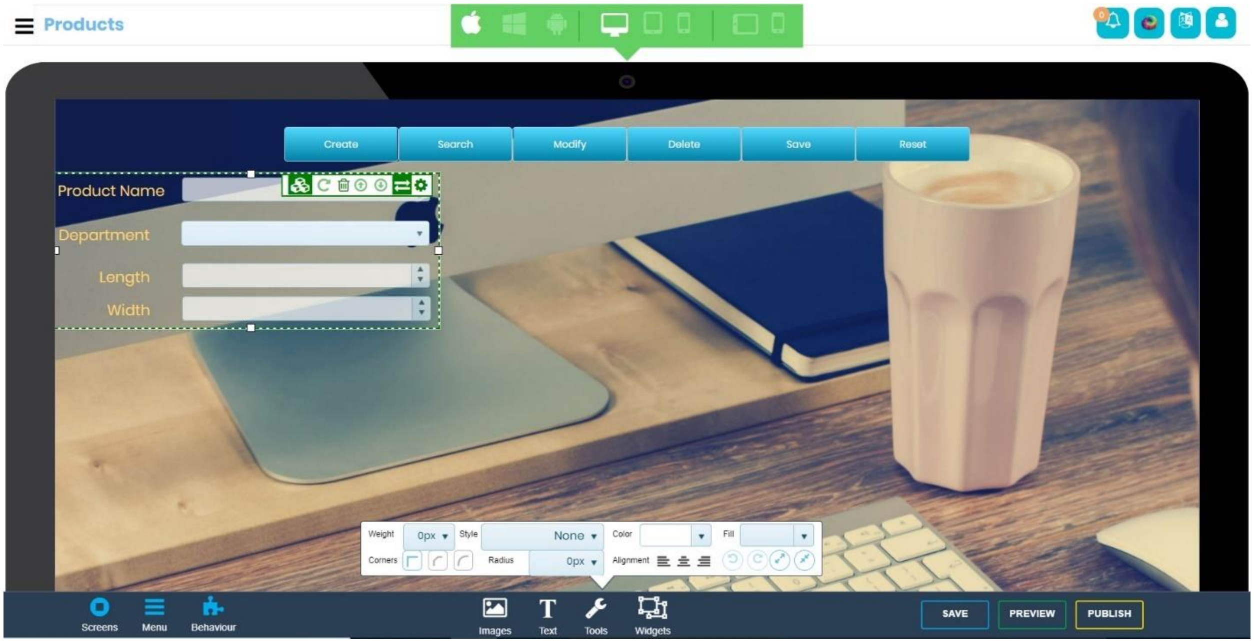
## Functions



Functions are a single building block that performs a unit of work. An easy way to think about these is that they take action. Business logic can be defined within a Function using plain English and are accompanied by a fully customizable user interface. Functions can then be plugged together (composability) to create a string of functions that results in a particular business process. An example of this is a function that calculates a particular discount for an item when certain quantity levels are reached. This can be paired with another function that calculates sales tax for your entire basket of items. This demonstrates how functions can be thought of as Lego blocks that can easily be plugged together to create meaningful applications. In addition to a user interface and business logic, functions also generate an API automatically, this becomes critical when you want to use a particular function elsewhere in your application.

Compute nodes will consist of cell phones, tablets and notebooks on the Sparkster Decentralized Cloud are responsible for executing these functions. A function consists of many operations, or actions. For example one function operation may “update a field by adding two existing numbers.” Participation in the network is as simple as installing an app on the device and setting the desired price per function operation execution. The device owner is rewarded whenever their device is selected to execute a function on behalf of a client.

## Documents

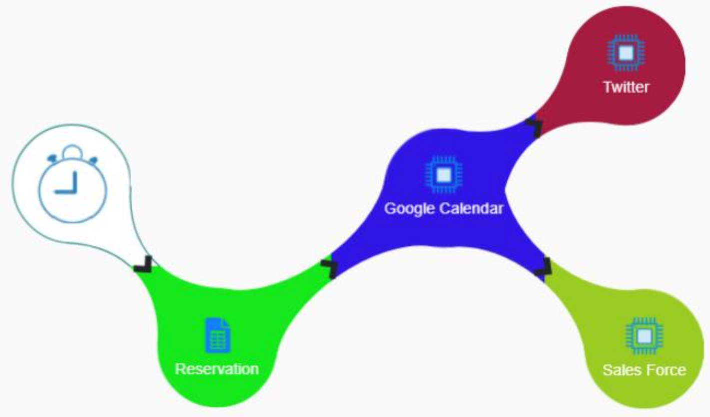


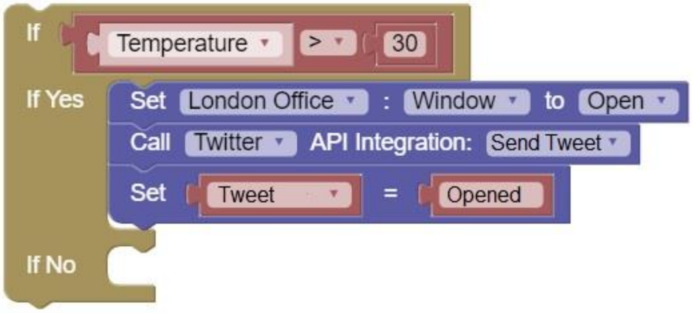
Weight 0px
Style None
Color
Fill
Corners
Radius 0px
Alignment

Documents are the basic data storage entity in the Sparkster platform. While they too provide an interface, English language business logic and an API interface. Their difference to function is that they facilitate the persistence, retrieval, updating and deleting of data from a datastore. No datastore needs to be defined, and relationships between these data storage entities (documents) can easily be established in Plain English. These documents aim to represent an entity in the real world. A good example of a document might be a patient record, or an inventory item.

Storage nodes on Sparkster Decentralized Cloud are rewarded for storing and retrieving data. Storage nodes will consist of desktops and servers operated by our community. Since data is immutable in decentralized networks, when a document is being used on the Decentralized Cloud, records can only be created, they cannot be edited or deleted. Hence, we will maintain a system for versioning records, such that updates can be adopted as new versions or new instances of a record.

## Integrations





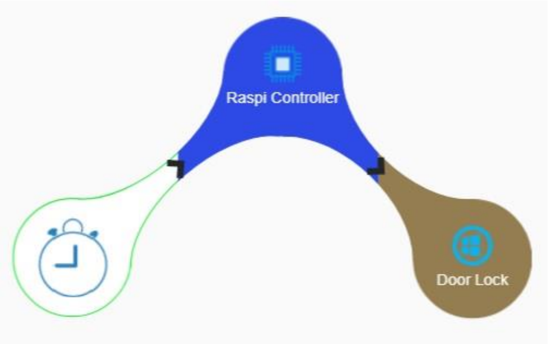
```
If      Temperature > 30
If Yes  Set London Office : Window to Open
        Call Twitter API Integration: Send Tweet
        Set Tweet = Opened
If No
```

Integrations are our interface to the outside world. They provide a simple and efficient abstraction to an API or webservice hosted and provided by an external party. These integrations can be defined without a single line of code and involve merely populating a form. However, this is likely beyond the scope of what a non-technical person would be able to do. Hence, our build once use anywhere approach is particularly valuable here. A technical user can define the integration once, then publish it to our marketplace, where a non-technical user can simply consume it and use it within their project or application. This is in line with our philosophy of not having to understand how the plumbing behind the scenes works, rather, focusing strictly on desired outcomes. A good example of an integration might be a graphical user interface that provides a shipping quote. Simply input the weight, origin and destination zip codes, and it will return a shipping price directly from the FedEx or UPS API.

In decentralized environments, such integrations are often called oracles. Oracles have externalities which make them particularly difficult to handle in decentralized networks. Imagine for example, that we're using an integration to interact with an IOT lightbulb (a device). With traditional Smart Contracts, the contract is executed by many if not all the nodes on the network. What would happen if 1,000 nodes on the network tried to turn the light off? Side stepping the flooding problem that may occur, "turn-off" messages would arrive at different times as the network attempts to arrive at consensus. Imagine, if someone manually turns the light on in the middle of this process. It would be quite annoying for the nodes to keep turning the light off during this process. Hence, today oracles are handled in a centralized manner requiring us to trust the source of data.

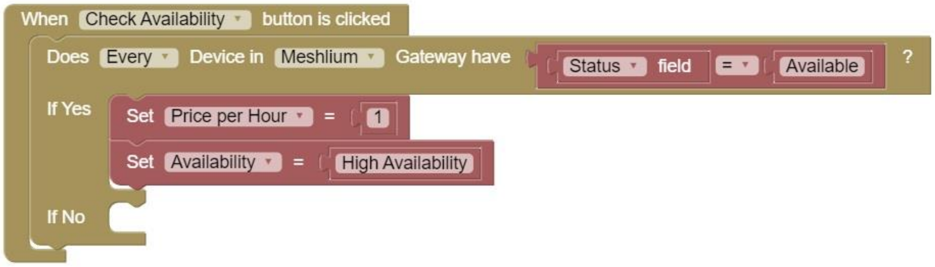
We overcome this problem in a novel way. We assign the task of turning off the light to one compute node (cell phone, tablet etc.) However, to be sure that the compute node performed the required task, the reply from the IOT device is sent to a different node. If and only if we receive a successful reply, is the first node rewarded.

## Devices



Devices are, as you might imagine, a representation of a physical device in the real world. Our virtual representation enables us to integrate physical devices into our applications. These devices consist of two core concepts: commands and fields. Fields facilitate bidirectional data transfer. As an example, imagine a greenhouse that has an IOT temperature probe, the actual temperature measurement would be sent back to the Sparkster platform via a ‘Temperature’ field. Commands are how the Sparkster platform instructs a device to take action. For example, a “turn–on” command might cause a LED light to turn on, or an “open” command might instruct a relay controlling a window would cause it to open. These two capabilities are exposed via specific device blocks that are available in all other aspects of our platform, namely: Functions, Documents, Flows etc.

## Gateways

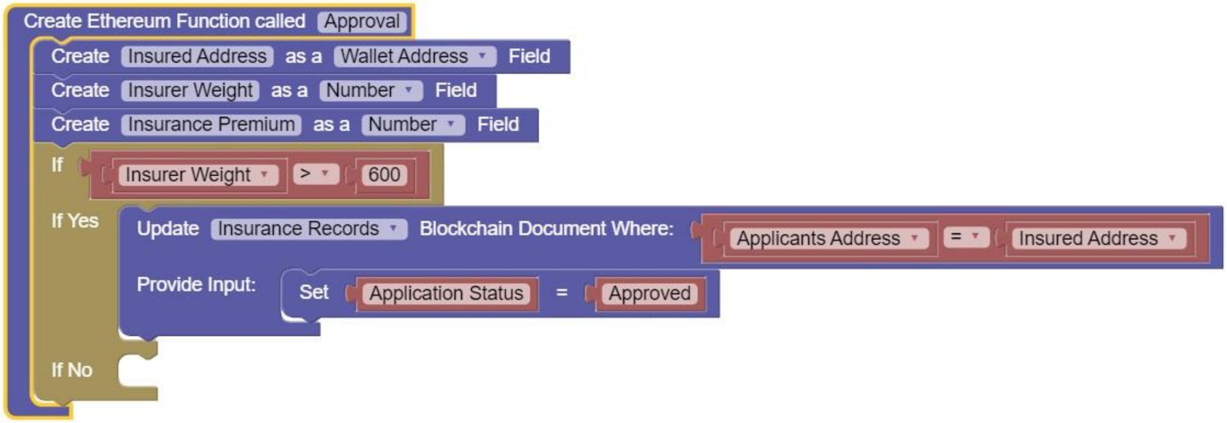


```
When Check Availability button is clicked
  Does Every Device in Meshlium Gateway have Status field = Available ?
  If Yes
    Set Price per Hour = 1
    Set Availability = High Availability
  If No
```

Gateways represent a group of devices that are all connected to the one gateway. That gateway provides the internet connectivity, so all devices can share one internet connection and reduce deployment costs. Through the gateway, our platform can interact with each device individually, or deal with all the devices as one group. For example, to talk to the August door lock over the internet you could use Apple TV as a gateway to the device. Here, we’re talking to the individual device (the door lock) but we’re not talking to it directly, but rather through Apple TV as a gateway. An example of interacting with a group of devices in aggregate via a gateway might be a car park where not every parking space device is directly connected to the internet. Rather they connect via an appropriate protocol such as Xbee to the gateway, and the gateway is connected to our platform. We have implemented this system with Libelium and we are able to provide some unique capabilities. We’re able to execute local preprocessing on the gateway before data is sent to our platform. This is particularly useful as we can significantly reduce the bandwidth consumed by only sending data to our platform over the internet if there is a change in status; i.e. when a parking space changes from being vacant to being occupied, or vice versa. Here, we don't need to worry about interacting with individual parking space devices, but rather we can just interact with the limited data that we need from the gateway.

As mentioned above, the interaction with these devices and gateways in the Sparkster Decentralized Cloud is a special case of function execution on compute nodes.

## Smart Contracts



```
Create Ethereum Function called Approval
    Create Insured Address as a Wallet Address ▾ Field
    Create Insurer Weight as a Number ▾ Field
    Create Insurance Premium as a Number ▾ Field
    If      Insurer Weight ▾  > ▾  600
    If Yes  Update Insurance Records ▾ Blockchain Document Where: Applicants Address ▾ = ▾ Insured Address ▾
            Provide Input:  Set Application Status = Approved
    If No
```

Smart Contracts enable our platform to build, execute and adjudicate transactions that have financial consequences in the real world. We currently support two versions of Smart Contracts: Ethereum Smart Contracts and IOTA Smart Transactions via IOTA Flash Channels.

Traditionally, contracts have been a way for parties to come to an agreement. These contracts define the task which need to be accomplished and by whom, as well as the terms and time period of fulfillment. Also, in the event of a dispute, the traditional contract seeks the enforcement of the terms of the agreement, at least in theory. However, in practice, resolution of disputes occurs outside of the contract itself, in a court of law, and can be a costly and time-consuming endeavor with high legal fees, and years to draw a resolution (if ever).

Smart Contracts enable the digital enforcement, verification, and facilitation of a traceable agreement between parties, without the need for 3$^{rd}$ party intermediaries. Smart Contracts are automatically executed as soon as the necessary conditions of the contract are met. What this means in practice is that funds are released once each party has performed according to the terms of the contract. The release of funds is administered by impartial entity – the contract itself.

Adjudicating transactions is one of the core strengths of Smart Contracts, in that penalties can be defined, and autonomously executed for a party failing to perform its obligations according to the terms of the Smart Contract. The Smart Contracts acts an impartial adjudicator in such a case, and executes penalty clauses, as defined by both parties, in accordance with the specific and unambiguous logic with which it was created. This reduces the necessity of legal involvement in the event that a party fails to perform relative to an agreed upon contract.

The current challenge however is that Smart Contracts require advanced programming skills to create, and still run the risk of error in logic or programming by the developer, which can lead to loss or vulnerability of assets.

With the Sparkster platform, anyone can seamlessly create Smart Contracts without writing any code. There is no need to hire a developer; you can do it yourself, with only a few clicks. Such Smart Contracts can be built in the Sparkster platform in plain English, without a single line of code, and can easily be deployed to the Ethereum network with the click of a button. We believe this is revolutionary for business and is the missing ingredient in facilitating safe and secure peer to peer trade.

Applications built using the other core components of our platform (Functions, Documents, Workflow, Devices etc.) can easily interact with Smart Contracts with appropriate English language blocks. In addition, Smart Contracts for specific use cases (i.e. an insurance Smart Contract) could be authored by one individual and made available for others to use as a Smart Contract template. This in essence would create a Marketplace of Smart Contracts, all authored in plain English and easily understandable by everybody.

From a decentralized perspective, Smart Contract would be executed client side via Meta Mask, and hence have no bearing on the Sparkster Decentralized Cloud. However, the software built using the Sparkster platform has the potential to interact with any Blockchain. We’re not limited to Ethereum and IOTA. In the future we will facilitate interaction with Neo, Cardano, Stellar, Ripple amongst others.

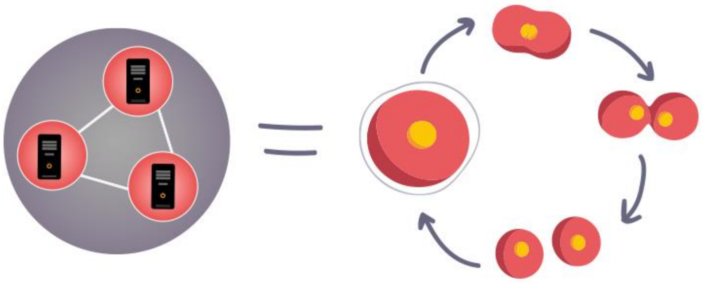
## No Bugs

Software that are autonomously generated by the Sparkster platform have an overlooked but incredibly important benefit – they’re bug free. While the business logic is in your hands, the actual code behind the scenes that makes all the magic happen is rather uniform and consistent. Hence, any application built on our platform will be entirely bug free. This can save countless hours if not weeks relative to traditional software development. Yet another reason why traditional software development environments have no hope in competing with us. Software can be built in our platform at least 100x faster than they can by traditional software developers.

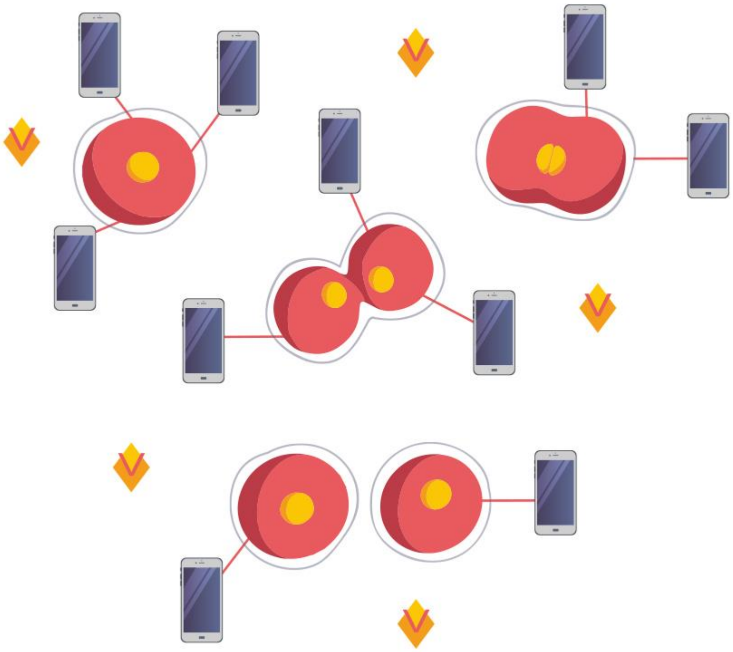
# Technical Implementation of the Sparkster Decentralized Community Network

## The Road to 10 Million+ TPS

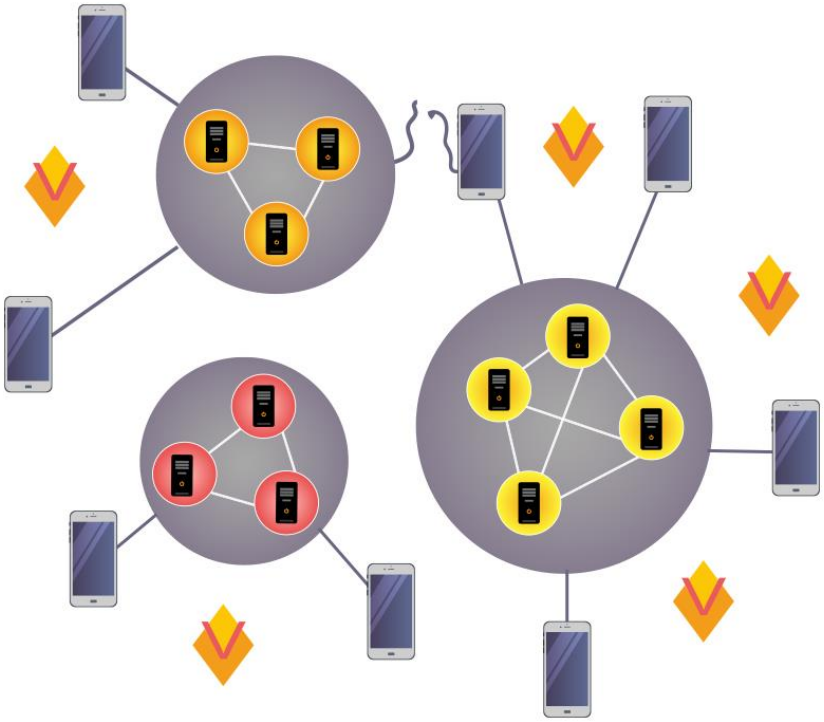
The introduction to the Sparkster Decentralized Cloud on page 20 is required reading to understand this section.



Since the Sparkster Decentralized Cloud is designed to be a specialized blockchain to be used only for the execution of Smart Software, we can eliminate some of the constraints imposed upon other blockchains. Given that clients desire to keep their company's data private, we can eliminate the necessity of maintaining Global State. This is significant, because it allows us to **shard** our Distributed Hash Tables by client groups. Most importantly, one shard **never** needs to have any awareness of any other shard – a unique benefit afforded to us as the Sparkster Decentralized Cloud is a specialized blockchain. Specifically, we're afforded clients, isolation, a key factor in building high throughput systems.



A useful analogy to explain what this means is to recall human cell division in a zygote. As the zygote grows and develops it splits and becomes an embryo – we now have two cells instead of one. The embryo continues to divide as it matures. Similarly, think of a blockchain cluster of nodes as a cell. Once there are a sufficiently large number of nodes in a particular cell, cell division can be induced. Let's take our worst case scenario of 1,000 transactions per second among a cluster of nodes. Once cell division is induced, transaction rates are doubled to 2,000 transactions per second. Similarly, cell division can continue, given sufficient nodes, and transaction rates can be increased linearly without any limiting factor. This is possible because we do not need to maintain any global state. In practice, this means that one cell can have complete isolation from all other cells, and hence, there's no performance penalty or overhead attributed to continued cell division. Isolation of node clusters is one of the core principles of building high throughput systems.



## Practical Example

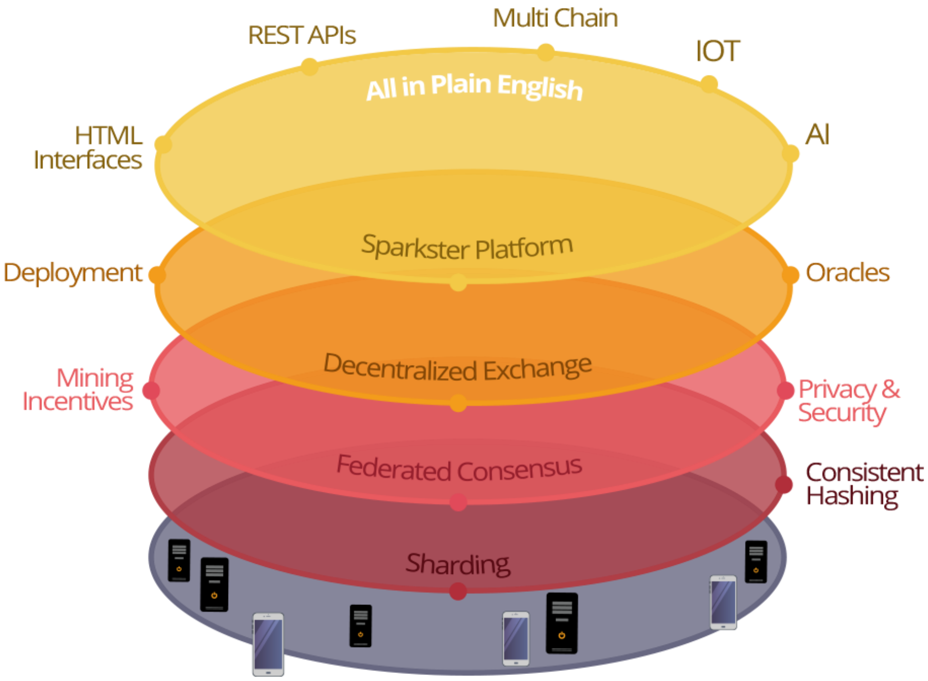
Let’s take a practical example to explain why this works. Imagine that GE® and GM® wanted to build and run software on the Sparkster Decentralized Cloud. Because GE® would never want their data accessed by anyone but employees and maybe customers of GE®, GE’s software applications would only ever need to interact with their own data. The consequences here are that that the distributed hash tables that concern GE’s data need not have any of GM’s data. The implications here are substantial. This means that the peers replicating data in the distributed hash table (DHT) for GE® (red nodes) can be completely isolated from the peers replicating data for GM® (yellow nodes), and hence GE® and GM’s data can be replicated in parallel. This is fundamentally how substantial increases in blockchain performance are achieved. Transactions can be received and processed by two different cells in parallel, and in this case doubling throughput - transactions per second. The reason that this works is because each cell is isolated from all other cells, eliminating the communication overhead involved maintain a consistent DHT between cells. We can add as many cells as necessary, without impacting performance of any one cell, but increasing throughput of the overall system with each cell appended. Note that our example only consists of two customers, but in reality, one cell would consist of thousands of customers.

In another use case, suppose one wanted to build a business or a product that will be used by millions of customers, akin to Uber® or Airbnb®. Such a company may opt to distribute customers across cells, and benefit from throughput in the millions of transactions per second. Suppose customers were distributed according to the first letter of their first name, and hence, all customers whose name begins with A would be in one cell, B in another, and so on. This is practical given that one customer's purchase history need not be seen by another customer. However, it is also incredibly valuable to company building the software as it allows them to build software that can serve millions of customer concurrently, without having to architect their software to necessarily do so.

Theoretically, there's no limits to how many cells we can create and hence, theoretically, there's no limit to the TPS we might be able to achieve, whether that's 10 million transactions per second or orders of magnitude more. The limiting factor is how many mining participants we can encourage to participate in the Sparkster Decentralized Cloud. Hence, our system of incentives becomes pivotal, and as such, we've designed a free market for participants to independently determine how much to charge for the use of their resources. Miners are free to set their own prices, and clients are free to elect how much their willing to pay.

## The Sparkster Technology Stack

The Sparkster technology stack represents a set of technologies that have been specifically designed to facilitate Decentralized Cloud Computing, where the Smart Software, an evolution on Smart Contracts, is written in plain English, by people with no prior knowledge of software development and without concern for the underlying decentralized architecture.



## Consensus

Today, there exists many different approaches to solving the Byzantine Generals Problem and ultimately arriving at consensus amongst trustless parties. By implementing Proof of Work, Bitcoin trusts whomever has the greatest hashing power (Semko, 2017), whereas Ethereum trusts whomever has the most money with their reliance on the Proof of Stake protocol (Lacina, 2015). In addition, both Bitcoin and Ethereum suffer from significant confirmation times that make them impractical for commercial applications without the use of off-chain transactions (Mitra, 2017).

The Federated Byzantine Agreement System (FBAS) proposed by the Stellar Network (Mazieres, 2016) has proven to easily perform 1,000 transactions per second with a latency of just 2-5 seconds per confirmation. Barclays Africa claimed to have achieved 10,000 transactions per second, processing over 36 million transactions an hour (How Many Transactions Per Second Can Stellar Process?, 2018). This is achieved by forming consensus amongst a slice of total participants in a network, otherwise known as a quorum slice, rather than the entire network. Since consensus is federated, meaning, party A trusts party B, and if party B approves a transaction, then party A necessarily approves the transaction, we arrive at consensus very quickly (Koller, 2017).

Hence, rather than focusing our efforts on creating a consensus algorithm that meets the throughput and latency needs of decentralized software execution network, we have chosen to implement the Stellar Consensus Protocol (SCP), a commercial implementation of FBAS.

There are some inherent limitation of the Stellar Consensus Protocol (SCP) including a lack of intrinsic incentives for good behavior, and lack of a formal methodology to determine whom to trust when forming a consensus slice (Mazières, 2016). The focus of this paper is to resolve these two limitations with SCP, to create a self-governing network with incentives that keep all actors honest, and give rise to a decentralized cloud for the execution of software.

We propose using the Stellar Consensus Protocol (SCP) for reaching consensus between parties on the network, with an additional layer to determine how tasks are allocated. This layer governs the earnings of each participant. In doing so, we aim to create a system of incentives that keeps all parties honest, minimizes the risk of an attack on the network and minimizes wasteful work being performed by participants on the network.

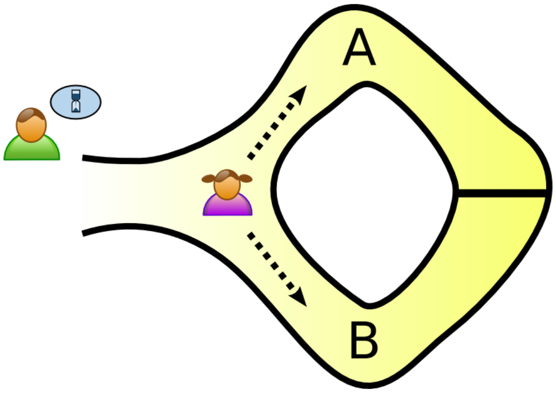
## Privacy

Privacy and security are absolutely necessary when handling transactions on behalf of corporate clients. A company may be bound by jurisdictional and legal restrictions such as the EU's General Data Protection Regulation (GDPR) (Czarnecki, 2017). However, legal restrictions are not the only reason a company would demand data privacy if they're to adopt a public blockchain. Another reason might be to protect customer or supplier data from being pilfered by competitors.

In essence, if we are to see corporations adopt public blockchain infrastructure, as they've adopted the public cloud, privacy of their data will be a major consideration. To provide such privacy, we propose using Zero Knowledge Proofs (ZKPs) and encrypting all data persisted on the Sparkster Decentralized Cloud.

Zero Knowledge Proofs limit the number of parties that are interacting with sensitive or privileged data by enabling the party executing a transaction, the prover, not to have to reveal any privileges or sensitive data to the verifying party, while still being able to prove that they have such data. In addition, ZPKs enable us to ensure any publicly observable data always remains encrypted.

The topic of Zero Knowledge Proofs has been discussed extensively by the likes of Vitalik Buterin, and many resources can be found that explain ZKPs well (Buterin, 2017). However, to provide an intuitive and oft-quoted example, consider the scenario below.



Suppose that Peggy (purple) wants to prove to Victor (green) that she knows the passphrase to a door within a cave, Peggy being private wishes to prove that she knows the passphrase but does not wish to reveal the passphrase. Peggy chooses a path to go down without Victor seeing, either A or B, and Victor yells out which path he would like her to return from. Suppose this experiment were repeated 20 times, and each time Peggy emerged successfully. The probability of Peggy being able to pull this off successfully without knowing the passphrase is about 1 in a million.

There are several examples of ZKPs in use today, one such example is ZCash which has implemented zk-SNARKs a form of Zero Knowledge cryptography.

In the example above, Peggy and Victor needed to interact multiple times, back and forth, in order to arrive at sufficient conviction that Peggy does indeed have know the passphrase. ZK-SNARKs are an evolution on these in that they are succinct and non-interactive, meaning that the proof consists of just one message sent from the prover to the verifying party. In addition, these messages are very lightweight, usually just a few hundred bytes and can be verified in milliseconds.

## Security

To ensure that a customer's sensitive data remains secure when executed on the Sparkster Decentralized Cloud, we must take steps to protect against tampering of the Sparkster mobile app. We will employ the following anti-tamper and tamper-detection approaches, some of which have been outlined below:

1. The app will compare the public key used to sign the app and confirm that it was indeed signed by Sparkster's development private key. If this comparison fails, we will know the app has been tampered with, and will be prevented from joining the Sparkster Decentralized Cloud.
2. Digital signatures and check sums will enable us to detect any possible file tampering. If tampering is detected, anything in memory will immediately be deleted by the application as a matter of protection.
3. We will ensure that any application files, libraries etc. are encrypted, to significantly limit the risk of tampering.
4. Input validation to limit the risk of injection attacks.
5. Employ third party tools to detect malicious injected payloads running in memory, thread execution hijacking and other attack techniques.
6. Ensuring all data is sent to the device using secure POST with XSRF tokens.
7. No app data will be cached. All transactions will be executed entirely in memory.
8. Securely delete data and nullify any memory after the completion of each transaction. In addition, force garbage collection after each transaction.
9. All data transmitted to and from the device will be encrypted to protect again man in the middle attacks.
10. We will ensure all communication between devices is protected with SSL/TLS.
11. Eliminate any knowledge of which customer the transaction belongs to, thereby reducing incentives to hack the application to mine data.

## Multi Chain Interoperability

Given that the Sparkster Decentralized Cloud is a blockchain specialized in cloud computing, it does not execute payments natively. Hence, we have a necessity to integrate with other blockchains and virtual currencies to facilitate payments. In essence, if you wish to create a piece of software that makes payments to other parties, the software defined on the Sparkster platform will need to interact with other networks.

Presently, the Sparkster Platform has native integration with Ethereum and IOTA. Generate Ethereum Smart Contracts in Plain English, but also interact with any Ethereum Smart Contract in plain English. The implications are that Smart Software can quickly and easily be developed that can interact with and make use of Ethereum Smart Contracts. In addition, interact with the IOTA network through IOTA Smart Transactions facilitated by IOTA Flash Channels.

The implications are substantial. Given the ability to generate Enterprise grade software with the Sparkster Platform, easily integrate these Smart Contracts and Transactions into existing business processes with ease and simplicity. Alternatively, build software that solves meaningful business problems, but pay customers in the currencies they're actually using. With Sparkster, there are no limits.

In the future, we will add support for other chains, specifically NEO, Cardano and Stellar. See the roadmap for details.

## Consistent Hashing

Without a centralized coordinator and without any global state, how do compute nodes identify which cell a client's data is stored in? That's where a smart algorithm called Consistent Hashing comes in. This algorithm allows us to hash the client's ID, and extract a bounded number. Since this is bounded, imagine the bounds to represent a point on a circle, and all cells (clusters of nodes) divide up this circle equidistantly. Now, we're easily able to assign a cell to a client and identify its cell using the Consistent Hashing algorithm. Most importantly, any compute node can determine, independently, where they might find a client's data, without interacting with any central authority or requiring any central database. Hence, we have a network that is fully decentralized.

## AI and IOT

Being that the Sparkster Platform aims to empower people to bring their ideas to reality, emerging technologies such as AI and IOT are necessary constituents of the platform. However, our aim is to make these technologies accessible to people, without having to programming, let alone the complex mathematics that underpins AI. Thus far, we have made AI based facial recognition completely codeless, and will be bringing other forms of Artificial Intelligence such as natural language processing, speech recognition amongst others, in an accessible manner to the Sparkster Platform.

Sparkster has industrial IOT partnerships today with ARM® and Libelium® and has native integration with Apple® Homekit, Amazon® Alexa, Arduino®, Raspberry Pi®. The Sparkster Platform can be thought of as an environment architected specifically for the high throughput requirements of IOT, yet making all these technologies accessible without any code. Our YouTube® channel features a video of children aged between 7 and 10 years old creating IOT devices, independently and codelessly with the Sparkster Platform. If they can do it, we can all do it.

At Mobile World Congress 2018, we demonstrated the Sparkster Platform using AI facial recognition to detect a cleaner, and opening an August® door lock, controlled by Apple® Homekit, and execute an automated payment via IOTA. The entire demo was built with no code, in plain English. You will find a video of this demo on our YouTube channel.

# System of Incentives

There exist limitations with the Stellar Consensus Protocol (SCP) that must be addressed for it to be used in a broad capacity. Namely, that SCP lacks a system of incentives to keep participants honest, and fails to provide a solution to determine whom to trust when forming consensus (Mazières, 2016).

Systems of incentive are necessary to ensure sufficient network participation. The rate of participation is directly proportional to the rate at which price per operation will decline. Hence, paying participants for their participation is necessary to delivering a Decentralized Cloud that is significantly cheaper per operation than traditional cloud computing, and critical to delivering on our vision of disrupting the $200bn cloud computing industry. However, these incentives also provide additional positive side effects. Namely, they provide us with a mechanism of encouraging participants on the network behave appropriately.

Compute nodes and Storage nodes are rewarded in SPARK tokens for their contribution to the network. Namely:

- Compute nodes contribute their mobile, tablet or notebook CPU and memory for the purpose of remotely executing software components on behalf of paying clients.
- Storage nodes contribute storage space and network bandwidth for the purpose of storing and retrieving encrypted data on behalf of paying clients.

Clients pay network participants for the execution of software and the storage of data. Given that marginal cost per operation are near negligible for Compute nodes on the Sparkster Decentralized Cloud, we quickly converge on costs for clients that are significantly cheaper than traditional Cloud computing from the likes of Amazon® and Microsoft®. This provides clients a significant incentive to adopt the Sparkster Decentralized Cloud, which in turn benefits network participants with growth in network load.

Device owners can set the “price per function operation” for the use of their device. Each statement or action will be charged this “price per function operation” set by the device owner. So, a function that has been defined with many operations will cost more to execute than one that only has only a few operations. The reason such a system is required, is to provide a disincentive for creating operations that run forever, otherwise known as infinite loops, within the functions defined by our clients. A maximum “gas limit” will be set, which is a maximum number of operations a function can legally execute. If a function reaches this limit, then the function execution will automatically terminate.  This is necessary to ensure that functions are designed correctly, i.e. small units of execution, and that infinite loops do not impact the availability of the network.  This price governs how much the device is rewarded per function whenever that node is selected to execute a function on behalf of a client or customer.

Allocation of work to Execution is governed by a Proof of Stake probability distribution wherein:

$$P_i = \frac{n_i}{N}$$

Where:

$P_i$ is the Probability of assignment of some Compute node $i$
$n_i$ are the number of tokens stakes by node $i$
$N$ is the total number of tokens stakes by all nodes

Proof of Stake is used as it provides an incentive for Compute nodes and Storage nodes to operate honestly on the network. Any misbehavior will result in this stake being seized by verification nodes that detect such misbehavior. For example, if a node claims to have executed a task, but in fact has not, their stake will be taken by the verification node. Hence, there's a strong incentive for both Compute nodes and Storage nodes to operate honestly.

Herein lies the natural incentive for verification nodes. They police the network in the hopes of capturing stakes from those who behave maliciously. While the Compute nodes and Storage nodes earn an income proportional to their stake, verification nodes earn an income proportional to the work they perform – namely verifying transactions. Here, instead of wasteful effort as in the case of Bitcoin (Hern, 2018), verification nodes on the Sparkster Decentralized Cloud perform **Useful Proof of Work**. A quorum of verification nodes is necessary in order for the detecting node to receive its reward. These quorum slice participants are chosen at random to avoid any risk of collusion. For those unwilling to stake tokens, they may still participate in the Sparkster Decentralized Cloud as verification nodes, earning from the use of their CPU and memory.

Storage nodes are rewarded for storing and retrieving data. Much like Compute nodes, the probability of being assigned a persistence request is a function of tokens staked, amongst other factors. This is necessary, as a storage node may misbehave by deleting data after it has received payment. Hence, in this case, verification nodes request data in order to validate that they do still retain that data.

Clients are able to elect how long they would like their data persisted on the Sparkster Decentralized Cloud. After this time frame has expired, verification nodes will no longer verify data persistence, and Storage nodes are free to delete this data without the risk of losing their stake. Pricing is set by Storage node owners independently on a per record per month basis.

To further encourage reliability among our Storage node participants, these participants are not paid in one lump sum for a storage agreement. Rather, they're paid on a performance basis. Namely, they're rewarded each week for storage of data for the week prior.

From time to time, storage nodes may experience hardware failures where availability is compromised. In this case, the storage node is not acting maliciously. To mitigate the risk of their stake being seized, storage node owners are able to place their node in maintenance mode for a maximum of 16 hours. After 16 hours their node is automatically added back to the network, where verification nodes will resume data requests for verification purposes.

Allocation of persistence requests to Storage nodes is governed by a probability distribution wherein:

$$P_i = \left(\frac{n_i}{N}\right) \cdot A \cdot R$$

Where:

$P_i$ is the Probability of assignment of some node $i$
$n_i$ are the number of tokens stakes by node $i$
$N$ is the total number of tokens stakes by all nodes
$A$ is the Availability Index as defined below
$R$ is the Response Time Variable as defined below

Availability is an exponentially smoothed indexed value based on the number of hours the node has gone into maintenance mode. Hence, here we introduce an incentive for storage node operators to ensure their nodes remain available. Exponential Smoothing ensures that while the past is never forgotten, recent availability performance is given greater significance. Therefore, problems of the past can be recovered from. This is derived as follows:

For some node $q$,

$$\text{Let } x_i = 1 - \left(\frac{h_i}{m_i}\right), \text{for } h, m > 0$$

Where:

$x_i$ is the number of availability index for a specific month $i$
$h_i$ is the number of maintenance hours in month $i$
$m_i$ is the total number of hours in month $i$

$$x_i \in \{x_t\}, \text{for } q \text{ such that:}$$

$$A_t = \alpha \cdot x_t + (1 - \alpha) \cdot A_{t-1}, \text{for } t > 0$$

Where:

$A_t$ is the Current Expected Availability Rate

$\alpha$ is the smoothing factor, and $0 < \alpha < 1$

Response Time is monitored by verification nodes as they're verifying transactions persisted by Storage nodes. Those nodes that fail to respond within our response threshold will have their response time score impacted. Similar to Bitcoin mining difficulty, response time thresholds will reduce automatically in a predictable way to entice participants to improve network performance over time. Those Storage nodes found to provide response times higher than the

$r_i \in \{r_n\}$, such that:

$$R = 1 - \left( \frac{\frac{\sum_{i=1}^{n} r_i}{n} - T}{T} \right)$$

Where:

$R$ is the Response Time Variable
$T$ is the Network Threshold
$\{r_n\}$ are the node's response times as determined by the verification node over n observations.

Decay of the Transaction Threshold is a function of Transactions processed at time t, such that:

$$\frac{dT}{dt} = -\gamma \cdot (T - T_n)$$

Where:

$\frac{dT}{dt}$ is the rate at which transactions are being processed
$\gamma$ is a decay constant
$T$ is the Number of Transactions
$T_n$ is the Target Transaction Quantity

$$\therefore \frac{1}{T - T_n}\, dT = -\gamma\, dt$$

$$\int \frac{1}{T - T_n}\, dT = -\gamma \int_{t=0}^{t} dt$$

$$\ln|T - T_n| = -\gamma \cdot t$$

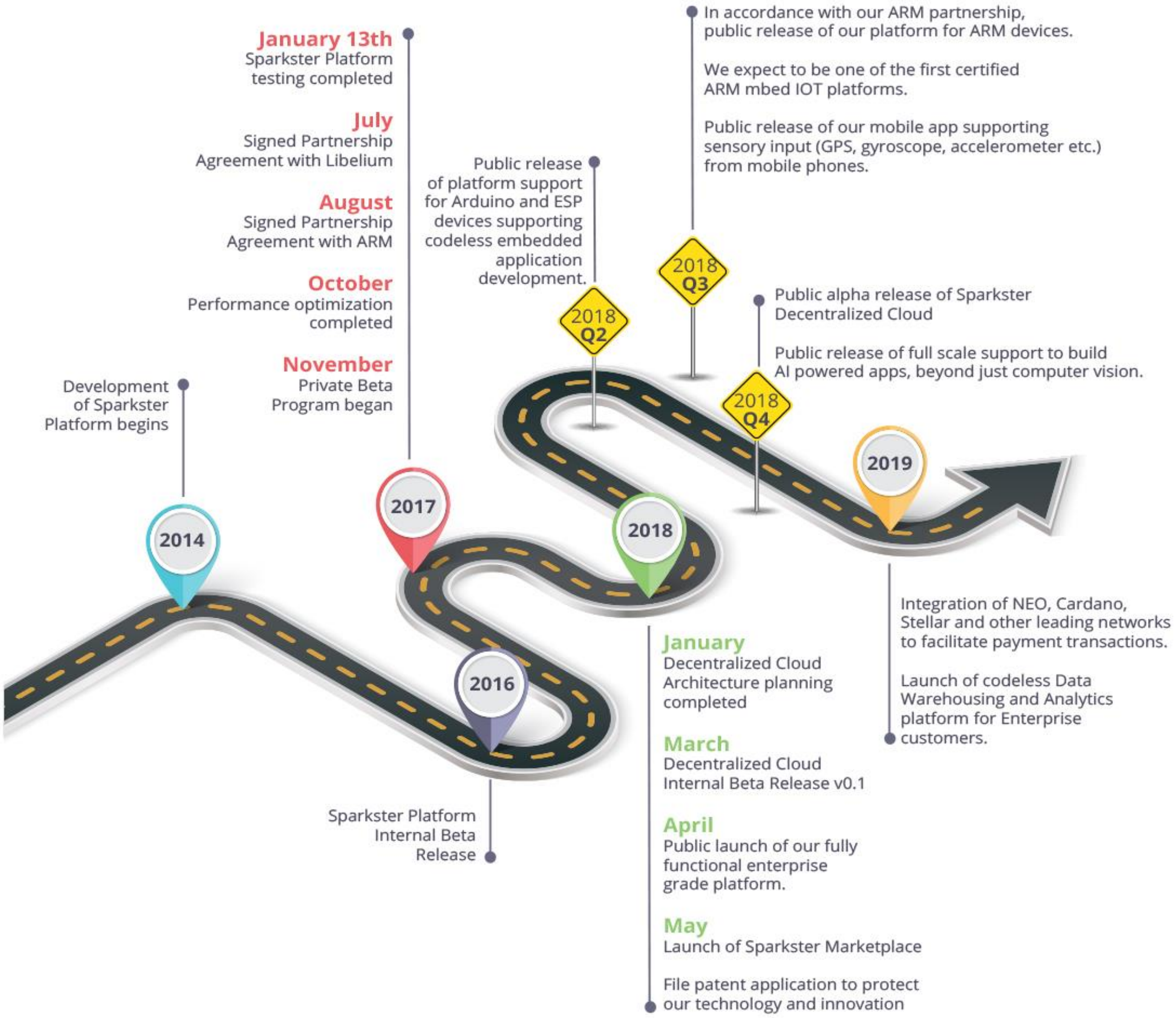
$$|T - T_n| = e^{-\gamma \cdot t}$$

$$T(t) = T_n + e^{-\gamma \cdot t}$$

## Optimizing Execution Costs for Clients

A free an open market amongst network participants will facilitate competition that ultimately will drive down execution costs as each network participating competes for execution opportunities. Clients also have the freedom to choose the amount they're willing to bid for execution, and wait for an offer that executes the order. A free market presents the fairest way to find and execute opportunities to find equilibrium in a market.

# Roadmap



2014

Development of Sparkster Platform begins

2016

Sparkster Platform Internal Beta Release

2017

**January 13th**
Sparkster Platform testing completed

**July**
Signed Partnership Agreement with Libelium

**August**
Signed Partnership Agreement with ARM

**October**
Performance optimization completed

**November**
Private Beta Program began

2018

**January**
Decentralized Cloud Architecture planning completed

**March**
Decentralized Cloud Internal Beta Release v0.1

**April**
Public launch of our fully functional enterprise grade platform.

**May**
Launch of Sparkster Marketplace

File patent application to protect our technology and innovation

2018 Q2

Public release of platform support for Arduino and ESP devices supporting codeless embedded application development.

2018 Q3

In accordance with our ARM partnership, public release of our platform for ARM devices.

We expect to be one of the first certified ARM mbed IOT platforms.

Public release of our mobile app supporting sensory input (GPS, gyroscope, accelerometer etc.) from mobile phones.

2018 Q4

Public alpha release of Sparkster Decentralized Cloud

Public release of full scale support to build AI powered apps, beyond just computer vision.

2019

Integration of NEO, Cardano, Stellar and other leading networks to facilitate payment transactions.

Launch of codeless Data Warehousing and Analytics platform for Enterprise customers.

# Token Economic Value Model

We have built a global Marketplace to bring authors together with innovators and makers via the Sparkster Platform. Authors can create components, for example an Ethereum Smart Contract for the automotive industry, and publish them to our Marketplace. Innovators and makers can discover these components from our Marketplace, and integrate them to build custom software solutions for the problem they are trying to solve.

Our tools enable software to be built 100x faster than traditional software, but our Marketplace adds an order of magnitude improvement in efficiency and productivity. The SPARK token provides a vehicle for consumers of these authored components to compensate the author for their contribution, and gives authors an incentive to contribute to our Marketplace.

The SPARK token facilitates micropayments that are not feasible using traditional fiat given exorbitant credit card transaction fees and minimum charges imposed by credit card merchant banks. Hence, the Sparkster token will be the only currency which will be accepted as compensation on the Sparkster platform, where authors will be able to publish their components and software in exchange for Sparkster tokens.

In Q4 2018 we will launch our Decentralized Cloud where these components can be executed. “Miners” will make their cell phone and IOT devices available as Decentralized Compute environment for software authored on the Sparkster No Code Platform. These miners will be rewarded in SPARK tokens, hence providing an incentive to compete and provide their unused capacity at close to the marginal cost of this capacity - near negligible.

Network effects result from our community’s participation, such that the cost of executing software on the Sparkster Cloud diminishes over time, hence, increasing client participation in the network over time. In addition, network effects result from our community contributing their creations to our Marketplace, making it faster and easier to build software on the Sparkster platform. We believe this will result in Sparkster being a market leader in the creation and development of Enterprise software.

## Token Purpose and Intent

Our tokens are intended to facilitate trade between members of our community and provide an incentive for authors and innovators to contribute to the Marketplace.  When publishing to our Marketplace, authors can opt to charge for the use of their contribution. While the author is free to choose the amount to charge, the payment will be made with Spark tokens.

Sparkster will charge the author a discretionary platform fee for any paid transactions that occur on the Sparkster marketplace, this will be deducted automatically from any remittance to the author. There are no fees charged for contributions made to the Marketplace that are offered for free.

The Sparkster Platform will be fully functional and available for use prior to any token sale.

Once the Sparkster Decentralized Cloud is released for public use, the SPARK token will be used to purchase execution on the Decentralized Cloud, and miners will be paid with these tokens. Miners will be required to stake tokens as collateral to ensure they do not cheat. Verification nodes check that these miners are honest, and if they're found to be cheating, their stakes will be captured. Hence, the SPARK token is integral to the efficient operation of the Sparkster Decentralized Cloud.

## Token Distribution Platform

Our token sale will be conducted on the Ethereum network. We will be using issuing standard ERC-20 tokens. You must NOT use wallets from an exchange such as Coinbase, Binance etc.

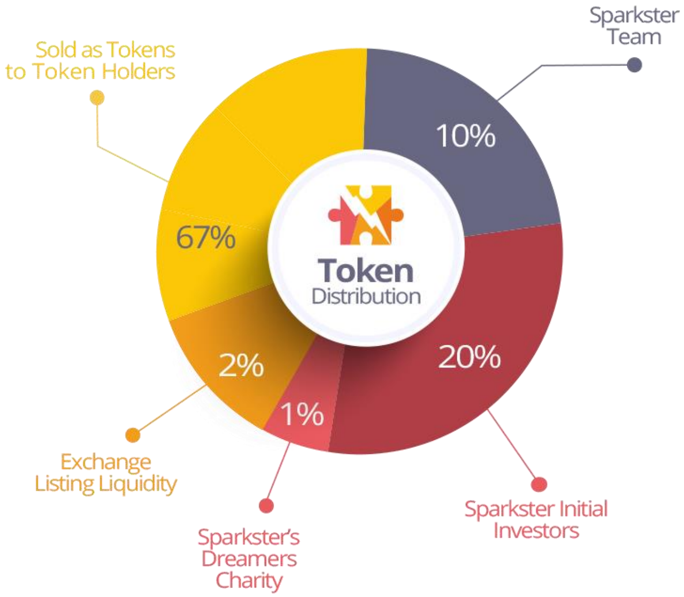
## Token Sale Procedure

In order to participate in our token sale, you will need to:

1) Have signed up to our whitelist, and received an email with confirmation of your allocation.
2) Complete our KYC and AML verification procedure.
3) Sign a SAFT and Token Sale Agreement
4) Follow instructions provided via email to complete payment.

Anybody interested in purchasing our tokens from us directly will be required to complete KYC and AML verification. US Persons, as defined by the SEC will not be permitted to purchase our tokens. Once we have verified that you qualify to purchase our tokens, we will authorize your Ethereum account request. From that point forward, you will be able to buy, sell and hold our tokens. Authorization will be revoked in the event that we determine that you are engaged in terrorist financing, or anything else that violates the guidelines provided by the SEC.

## Token Distribution Strategy



1% of total token supply will be distributed to our own charitable organization: The Dreamers Foundation. The mission of this organization is to promote education and technological innovation amongst the world's poorest people, and to create technology that promotes their growth, independence and enhances their sense of self determination.

10% of all tokens will be reserved to support our current and future team. These will be distributed as rewards, incentives, bonuses etc. to support our growth and development.

20% of all tokens will be reserved for shareholders of Sparkster as compensation for their investment over many years to build the Sparkster platform.

2% of tokens will be retained to facilitate liquidity for an exchange listing. An exchange listing is necessary to ensure that those wishing to use the Sparkster Decentralized Cloud have the ability to purchase tokens in order to do so. Any unsold tokens will be retained by Sparkster to facilitate exchange liquidity.

Team and Initial Investor tokens will be vested for a period of 2 years.

The remaining 67% of all tokens will eventually be sold via token sales in phases. In an effort to ensure that everybody is given a chance to own our tokens and not just those willing to pay exorbitant gas fees or those who show up right at the moment the sale begins, we will be maintaining a whitelist for all sales.

## Token Sale Structure

Presale – 250 million Tokens

A presale will be conducted to allow non US persons wishing to purchase more than $25,000 of tokens to do so. If you are interested in participating in our presale, please visit http://sparkster.me to join the whitelist.

Crowd sale – Remaining 38 million Tokens

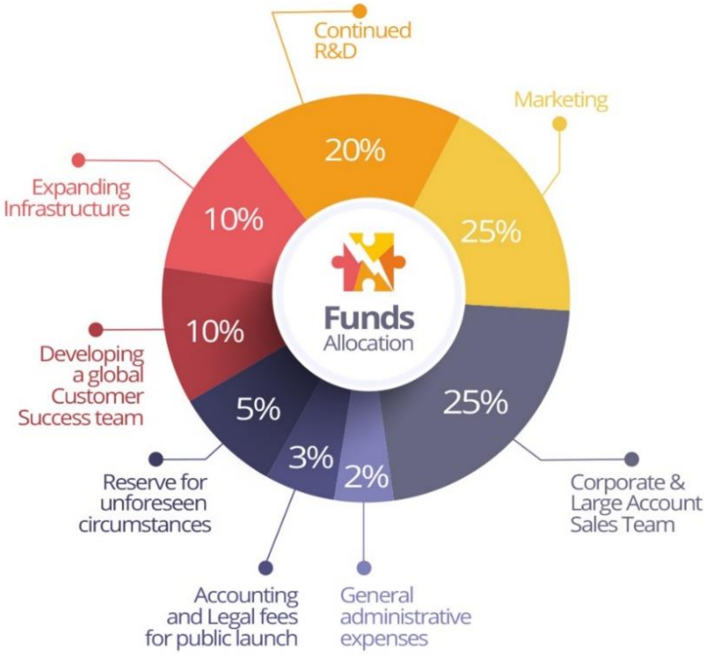
The public crowd sale will only be open to non US persons. The specifics of the crowd sale will be determined in consultation with our Telegram community.

## Currencies Accepted

You will be able to pay for your SPARK tokens in the equivalent value of Ethereum (ETH).

## Referral Rewards Program

A token reward of up to 0.5% of referred token purchases will be offered for referrals of token buyers. A personal referral URL will be offered to all those who register for our token sale. The referrer will earn their bounty once a referred user makes payment for their tokens. Any token rewards earned will not be deducted from the token allocated in each phase, but rather will offered from the remaining unallocated balance. All token holders, including those who earn tokens via our bounty program will need to be approved via our KYC and AML process.



## Allocation of Token Proceeds

The proceeds from our token sale will be allocated as follows. Following in Salesforce's footsteps, we intend to allocate a significant portion of our proceeds to sales and marketing efforts. The amounts below are just a guide for how we intend to allocate proceeds.

# Advisory Team

## Professor Gary Leavens

Chair of Computer Science, UCF
Expert in Computer Programming Languages

Gary T. Leavens is a professor and the Chair in the Department of Computer Science at the University of Central Florida, He received his Ph.D. from MIT in 1989. Before his graduate studies at MIT, he worked at Bell Laboratories in Denver Colorado as a member of the technical staff. Bell labs was once owned by AT&T but is now owned by Nokia. Professor Leavens is a recognized expert in programming languages, his research interests include programming and specification language design and semantics, program verification, and formal methods, with an emphasis on the object-oriented and aspect-oriented paradigms. He also worked on the specification for languages such as Smalltalk/Larch and C++/Larch. These languages embody insights from his work on the theory of behavioral subtyping. His best known work on language design and semantics is related to multiple dispatch languages such as MultiJava. Professor Leavens has been consulting with the Sparkster team since 2015.

## Juan Albelo

Senior Vice President, SAP

With over 20 years of sales experience in the Software Industry with companies like PeopleSoft, Oracle, SuccessFactors, SAP, SunGard, and PeopleFluent, Juan has extensive leadership, software sales, sales operations, consulting and education services experience, as well as multi-cultural abilities, business development expertise, and an extensive upper management acumen. He is passionate for success and building high-performing teams. He currently is serving as Head of HCM Sales at SAP. He has a proven track record of quarterly accomplishments, as well as several years of sales and marketing experience, for both products and services, in the highly competitive software industry.

# Leadership Team

## Sajjad Daya

CEO

A serial entrepreneur, having co-founded several companies and divisions around the world, Sajjad has been instrumental in building and leading a portfolio of global companies that generate annual revenues in excess of $50 million. In his younger years, Sajjad worked for Cisco Systems, in a technical capacity. Sajjad has diverse hands on experience building every aspect of a company from the ground up on shoestring budgets. Sparkster is the result of his unrelenting, ferocious and passionate commitment to empower the powerless.

## Amit Kumar

VP Technology

An IT Professional with 20+ years of experience in various leadership roles in the areas of Application & Database management. A seasoned veteran with experience working at Wipro & Ankh Software, he has worked in various domains - Healthcare, Manufacturing & E-Commerce. He has extensive practical knowledge in architecting Platform-as-a-Service & Software-as-a-Service development models. Amit lives and breathes our cultural mindset of "Making the Impossible Possible."

Sajjad and Amit have worked directly with one another for 11 years.

## Shabeer Kirmani

Technical Evangelist

AT&T Pfizer NBC UNIVERSAL

Shabeer is an Enthusiastic Techie, Researcher, Public Speaker, and Dreamer. Shabeer has a diverse professional background working with many Fortune 100 Firms including AT&T, Pfizer and NBCUniversal, in sectors as diverse as Finance, Telecommunications, Pharmaceuticals, Informatics, Consulting, and Government, and most recently working as a Data Scientist in the Defense & Space sector. A Certified Project Management Professional (PMP) and avid reader, Shabeer, enjoys good books, thought provoking discussions, volunteering, traveling and great food. Sajjad and Shabeer have been friends for 6 years.

## Dayanie Rajeev

VP Human Resources

citi IBM accenture

Dayanie is a true champion in our human capital management function. With a decade of experience working in organizations such as Citi Bank and Accenture, Dayanie is credited with building the team of Sparkster Warriors. These are not ordinary people, Dayanie has been instrumental in identifying individuals that have the tenacity, passion and technical prowess to overcome any technical obstacle they're presented with. Dayanie is also credited with leading large HR projects for IBM.

Sajjad and Dayanie have been working together for over 10 years.

## Sparkster Warriors

### Neeru Pallen

THE TIMES

When it comes to creative digital marketing, Neeru has a lot of strings to her bow and over a decade's worth of experience. Having worked with companies such as The Times (London), her passion for hands-on online and content marketing has helped to transform a diverse range of businesses. Working closely with a creative team to create finely-tuned, targeted content strategies that resonate with readers is Neeru's happy place. She has a gift for enhancing the performance and engagement levels of any site she touches.

### Munawar Bijani

LOCKHEED MARTIN

UCF INSTITUTE FOR SIMULATION & TRAINING

Munawar's passion for programming started at the age of 12. His curiously and desire to learn drove him to teach himself how to program. Munawar is both blind and visionary, he used his skills at programming to develop games for blind people, most notably a combat flight simulator. He formed his own company in 2010 and sold copies of his games around to world.

Munawar has worked for companies such as the Institute for Simulation and Training. The team he worked with was partnered with Lockheed Martin to develop a system for firefighters to navigate safely through smoked-filled buildings. Today, Munawar is one of our most prized developers and has been instrumental in building many of most important aspects our underlying architecture. He is an inspiration to all that meet him.

## Nagendra Obbu

wipro Infosys

Capgemini

Nagendra is a senior software engineer with a wealth of experience. He has worked for Wipro, Infosys and Capgemini. Known for his diverse range of experience, Nagendra has worked on integrative application solutions for banking projects with Discover when he worked for Capgemini.

Negendra's role includes planning, researching, developing logic, coding, testing and deploying Web projects with minimal assistance. He is a highly adaptable individual with the skills to participate in projects with varied levels of complexities and challenges.

## Siddharth Ubale

HSBC NEDCOR INVESTMENT BANK

Siddharth is a software engineer that specializes in in data analytics and data warehousing. He has 8 years of experience working for clients such as HSBC and NEDCOR bank, where he was involved in large scale big data projects. He is known for employing latest technologies and tools for data analytics and formulating methods to effectively draw business solutions from data. At Sparkster, he is involved in building an enterprise level data warehouse system that interactively renders reports for the statistical requests.

## Mithun Debnath

A software engineer with extensive experience in backend technologies. He has experience in developing large and complex applications involving migrations, integrations and SaaS applications. Analytical rigor and strategic thinking are critical aspects he would attribute to his success. He's one to dive in and own the end to end technical implementation as in the case of his work on projects for Ola and Snapdeal.

When he's not creating niche solutions, he enjoys reading science fiction and plays badminton.

## Prasad Nadendla

Prasad is a senior software engineer who is involved in the research and development of AI related technologies and solutions here at Sparkster. He has been instrumental in security, integration and IoT. He understands the business needs and applies artificial intelligence technology to solve real-world business problems. He constantly proposes and creates AI-enabled platforms that mesh multiple technologies into novel applications to drive meaningful solutions for Sparkster.

**Praveen Namoju** – Praveen plays a very crucial role in driving and leading the project teams in the analysis, design, and development of applications. He adds a perspective that spans the disciplines of software development, solution design, and project leadership. In addition to leading the team, he is great at writing architectural documentation as needed to construct and integrate systems that are not inherently compatible through extensive modification or alteration. His team monitors the platform's performance; anticipates needs and optimizations, tracks industry trends, evaluates new and alternative technologies; assists in creating reports related to present and future technology needs. His trip to Boston, USA for an extended training further enhances his ability to understand businesses and the need for technical automation across various domains.

**Swarup Kumar Roul** is a Lead software engineer with experience in solving business challenges through simple technical solutions. His specialization lies in adapting innovative and current coding techniques and practices; while still maintaining the enterprise coding standards within the organization. As a Full Stack Web Developer, he works with the business team to develop user-centric software products from idea to production.

Swarup is a passionate techie who also happens to have an indescribable affinity to Coffee. His team would describe him to be extremely friendly, sincere and ambitious in nature. A Foodie and great cook who believes that "Simplicity is the key to brilliance".

**Shaik Naimuddin** is a veteran software developer and comes with an experience rich in workflow management. In addition to working on our workflow project, he has written the automated logging and monitoring of the platform's execution. This system ensures reliable messaging which guarantees that when you click a button, things happen 100% of the time. He enjoys work that challenges him to learn something new and stretch in a different direction.

**Baggiaraj** – He specializes in designing and developing user interfaces products. He is always hungry for new challenges and innovations. He works closely with the product team in understanding and converting the design vision into prototypes and code. His ability to visualize even the most unstructured information and transform them into sleek, responsive and attractive design is most commendable. He's a sport enthusiast who loves to play cricket and badminton, when he's not working his magic with design.

**Sethuraman Babu** - Is a user interface developer who demonstrates a solid grasp of current best-practices in web and mobile UI design. He has a sharp eye for concept and execution, and has a strong ability to follow creative direction when required. Being highly organized with exceptional attention to details; and his ability to prioritize while accurately completing tasks have made him stand out as a user experience specialist.

**Udbhav Agarwal** - A software engineer specializing in open source big data technologies. He has helped firms in taking up a systematic, disciplined and quantifiable approach to building software for warehousing with sub-second latency. He works closely with product managers, and functional teams to build effective data products for clients to drive product decisions, improve customer experience, and increase overall operational efficiency. He is involved in designing and driving the implementation of scalable and reliable data pipelines.

**Monika S Krishnappa** – She is a user interface developer specializing in creating 3D interface designs. She is an Innovative conceptual designer with impeccable skills in creating interactive content and user interface design. She has extensive experience in developing and designing responsive interfaces to render a unified experience to user across multiple platforms. She can convert requirements to information architecture, user flows, and wireframes in no times, which helps her to contribute to fast paced projects, where a quick turn-around makes an impressionable difference.

**Abjeet Dhiraj** - A senior software test engineer with seven years of experience in quality assurance, design verification and testing methodologies. She has supported the holistic testing efforts for multiple projects and software releases, including integration, system, regression, performance and user-acceptance testing. Her project artifact efforts include test plans, test scenarios, test cases and test scripts. He interacts with product managers, architects and developers all through the project - from concept to implementation - and ensures that the deliverables are of the expected quality.

**Kavya Ravishanker** : Is a user interface developer. She is inclined towards strong concept development with ability to design effective graphical user interfaces for cross-platform elements that includes a focus on layout, composition, typography, website architecture and design. She handles the design, development, integration, debugging, testing and delivering of top quality, scalable, and high-performance user interface code.

**Akshay Tiwari** is a Senior UI Developer, with over 6 years of experience. Akshay's role ensures the process of making application creation easier, intuitive, efficient and accurate. Applying user-centered design processes to develop user-flows, and reusable components. He works with Web Developers, Engineers and QA in meeting the deliverables on time. He can successfully apply principles of graphic design, interactive design, user-interface design, and information design to improve the development of responsive design for all user interfaces and their elements.

**Abhijit Kumar Patel** - A software test engineer who takes part in all stages of software delivery, from requirements gathering to testing. He's a quick thinker and a suave problem solver. He works closely with the development teams to address known gaps in the SDLC process and recommends a comprehensive delivery program that lowers the ratio of issues in the deliverables and improves the overall quality in the development environment. Strong interpersonal and communication skills are his key strengths.

# References

Aleksandrova, M. (2018, March 6). *The 10 Biggest Challenges Startup Founders Face When Building a Tech Startup Team.* Retrieved from Eastern Peak: https://easternpeak.com/blog/the-10-biggest-challenges-startup-founders-face-when-building-a-tech-startup-team/

Berry, D. M. (2003, August 10). *The Inevitable Pain of Software Development: Why There Is No Silver Bullet.* Retrieved from semanticscholar.org: https://pdfs.semanticscholar.org/8c1e/65eb4b442d190a5b4a37e72b9d49197fecd5.pdf

Buterin, V. (2017, February 3). *Zk-SNARKs: Under the Hood.* Retrieved from Medium.com: https://medium.com/@VitalikButerin/zk-snarks-under-the-hood-b33151a013f6

Clark, L. (2017, June 14). *Business applications embrace IoT, blockchain to face future.* Retrieved from ComputerWeekly.com: https://www.computerweekly.com/feature/Business-applications-embrace-IoT-blockchain-to-face-future

Czarnecki, J. (2017, April 26). *Blockchains and Personal Data Protection Regulations Explained.* Retrieved from coindesk.com: https://www.coindesk.com/blockchains-personal-data-protection-regulations-explained/

Darrow, B. (2017, February 22). Retrieved from http://fortune.com/2017/02/22/cloud-growth-forecast-gartner/

Hern, A. (2018, January 17). *Bitcoin's energy usage is huge – we can't afford to ignore it.* Retrieved from The Guardian: https://www.theguardian.com/technology/2018/jan/17/bitcoin-electricity-usage-huge-climate-cryptocurrency

*How Many Transactions Per Second Can Stellar Process?* (2018, January 10). Retrieved from Lumenauts: https://www.lumenauts.com/blog/how-many-transactions-per-second-can-stellar-process

Kalla, S. (2017, July 16). *To the Moon? Blockchain's Hiring Crunch Could Last Years.* Retrieved from Coindesk: https://www.coindesk.com/moon-blockchains-big-hiring-crunch-last-years/

Koller, M. (2017, October 25). *The Stellar Consensus Protocol: Decentralization Explained.* Retrieved from itnext.io: https://itnext.io/the-stellar-consensus-protocol-decentralization-explained-338b374d0d72

Lacina, C. (2015, May 24). *The Inevitable Failure of Proof-of-Stake Blockchains and Why a New Algorithm is Needed (Op-Ed).* Retrieved from Coin Telegraph: https://cointelegraph.com/news/the-inevitable-failure-of-proof-of-stake-blockchains-and-why-a-new-algorithm-is-needed

Mazieres, D. (2016, February 25). Retrieved from Stellar.org: https://www.stellar.org/papers/stellar-consensus-protocol.pdf

Mazières, D. (2016, June 14). *The Stellar Consensus Protocol.* Retrieved from https://www.youtube.com/watch?v=vmwnhZmEZjc

Mitra, R. (2017, December 4). *Lightning Protocol & The Raiden Network: A Beginner's Guide.* Retrieved from blog.springrole.com: https://blog.springrole.com/lightning-protocol-the-raiden-network-a-beginners-guide-c9d7bc702748

Schmidt, S. (2010, April 30). *The dark side of cloud computing: soaring carbon emissions.* Retrieved from The Guardian: https://www.theguardian.com/environment/2010/apr/30/cloud-computing-carbon-emissions

Semko, R. (2017, November 12). *Decentralised cryptocurrency is dead.* Retrieved from Medium.com: https://medium.com/deviota/decentralised-cryptocurrency-is-dead-c1763dc75efb

## Disclaimer & Important Disclosure

IMPORTANT DISCLAIMER There are risks and uncertainties associated with Sparkster and their respective businesses and operations, the SPARK tokens, the Sparkster Token Sale, the Sparkster Platform and the Sparkster Decentralized Cloud (each as referred to in this Whitepaper). You can find a description of the risk related to the Token Sale under the section Legal, which should be read carefully. This Whitepaper, any part thereof and any copy thereof must not be taken or transmitted to any country where distribution or dissemination of Token Sale, Crowd sale or Initial Token Offering like the one described in this Whitepaper is prohibited or restricted. The SPARK tokens are not intended to constitute securities in any jurisdiction. Acquisition of SPARK tokens does not present an exchange of cryptocurrencies for any form of ordinary shares of the Sparkster, Ltd and a Holder of SPARK tokens is not entitled to any guaranteed form of dividend. Holders of SPARK tokens are only entitled to certain rights within the Token Sale Agreement. SPARK tokens are utility token and cannot have a performance or a particular value outside the Sparkster Platform, Sparkster Marketplace and Sparkster Decentralized Cloud. Therefore this Whitepaper cannot constitute a prospectus or offer document for investment in securities. This Whitepaper does not constitute or form part of any opinion on any advice to sell, or any solicitation of any offer by Sparkster Ltd to purchase any SPARK tokens or give any help in any investment decision. This Whitepaper is for information purposes only. The contents of this Whitepaper are not a financial promotion. Therefore, none of the contents of this Whitepaper serves as an invitation or inducement to engage in any sort of investment activity. Prospective acquirers of SPARK tokens should carefully consider and evaluate all risks and uncertainties associated with the cryptocurrencies, Sparkster, Ltd and their respective businesses and operations, the SPARK tokens and the Sparkster Token Sale. Familiarize yourself with all the information set out in this Whitepaper, Risk Notice and the Token Sale Agreement prior to any purchase of SPARK tokens. Ensure that you are aware of all of the would be risks prior to obtaining SPARK. The Risk Statement details all potential risks that you should consider. We recommend that you seek out independent financial advice before engaging in any sort of business endeavor. You are not eligible and you are not to purchase any SPARK tokens in the Sparkster Token Sale (as referred to in this Whitepaper) if you are a citizen, resident (tax or otherwise) or green card holder of the United States of America. This whitepaper is subject to change, and may be updated at any time without notice.