CROWD MACHINE

WHITEPAPER

Craig Sproule
Founder and CEO
Crowd Machine SEZC

February 1, 2018

Statement regarding Forward-Looking Statements

This Whitepaper ("**WP**") is issued by Crowd Machine SEZC ("**Crowd Machine**") and sets out the general business information for Crowd Machine. This notice applies to each recipient of this WP. This notice may be altered or updated. Crowd Machine does not have an obligation to amend, modify or update this WP (including this notice) or to otherwise notify a recipient thereof in the event that any matter stated herein, or any opinion, or forward-looking statement set forth herein, changes or subsequently becomes inaccurate.

This WP does not constitute any relations between the recipient of this WP and Crowd Machine. This WP does not constitute a prospectus or offer document of any sort, and is not intended to constitute an offer for securities or a solicitation for investment in securities in Crowd Machine or any related or associated company in any jurisdiction.

This WP is for information purposes only. The contents of this WP are not a financial promotion. None of the information or analyses presented are intended to form the basis for any investment decision, and no specific recommendations are intended. Therefore, none of the contents of this WP serve as an invitation or inducement to engage in any sort of investment activity.

This WP outlines the current and future developments of Metavine, Inc. ("**Metavine**") which will be exclusively licensed to Crowd Machine. Unless expressly specified otherwise, the products and innovations set out in this WP are currently under development and are not currently in deployment. Crowd Machine makes no warranties or representations as to the successful development or implementation of such technologies and innovation, or achievement of any other activities noted in the WP, and disclaims any warranties implied by law or otherwise.

The information in this WP is given in good faith, but no warranties, guarantees or representations are made by Crowd Machine with regard to the accuracy, completeness or suitability of the information presented. Certain statements, estimates and information contained in this WP constitute forward-looking statements or information. Such forward-looking statements or information involve known and unknown risks and uncertainties which may cause actual events or results to differ materially from the estimates or the results implied or expressed in such forward-looking statements. Neither Crowd Machine nor its advisors have independently verified any of the information, including forward-looking statements, contained in this WP. No regulatory authority has examined or approved any of the information set out in this WP. No such action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this WP does not imply that the applicable, regulatory requirements or rules have been complied with. Nor should this WP, or any part of it, form the basis of, or be relied upon in any connection with, any contract or commitment whatsoever.

Crowd Machine expressly disclaims any and all responsibility for any direct or consequential loss or damages of any kind whatsoever (whether foreseeable or not) arising directly or indirectly from: (i) reliance on any information contained in this WP, or any information which is made available in connection with any further enquiries, (ii) any error, omission or inaccuracy in any such information, (iii) any action resulting therefrom, or (iv) usage or acquisition of products. This disclaimer applies notwithstanding any negligence, default or lack of care.

This WP is not directed to, or intended for distribution to or use by, any person or entity who is a citizen or resident of or located in any state, country or other jurisdiction where such distribution, publication, availability or use would be contrary to law or regulation. The manner of distributing this WP may be restricted by law or regulation in certain states, countries or jurisdictions. Persons into whose possession this WP may come are required to inform themselves about and to observe such restrictions. By accessing this WP, a recipient hereof agrees to be bound by the foregoing limitations.

Revisions

Version 1.4: February 1, 2018

Table of Contents

01 | Abstract

Crowd Machine is revolutionizing the blockchain and decentralized apps sector. By unifying the world's surplus device processor capacity, Crowd Machine has created a highly capable decentralized Cloud. Embedded within this decentralized cloud is a revolutionary software app creation technology which enables developers and non-developers alike to create blockchain and decentralized apps at up to 45x the speed of conventional development approaches. The technology is blockchain agnostic. It allows apps to integrate with existing smart contracts and extend the functionality of those contracts to deliver contract robustness superior to that currently supported by existing technologies.

Crowd Machine is creating the **Crowd Computer**, a global decentralized app execution engine. It has been designed to execute blockchain smart contracts and decentralized apps that meet any requirement. The Crowd Computer consists of a peer-to-peer network of Crowd Virtual Machines ("**CVM**") that run on peer devices. Device owners are compensated for the use of their surplus processing power to run the CVM.

Embedded within the Crowd Computer is a comprehensive app development technology that allows anyone, from seasoned developers to novices with no coding experience, to develop blockchain and decentralized apps at rates that far exceed current approaches.

The technology employs a systems wide approach across the app lifecycle. It is fully self-contained and removes the need to understand, or pay for, any additional third-party technology required to get apps to market. It is also blockchain agnostic allowing apps to be created that leverage any blockchain including Bitcoin, Ethereum and Ripple. Furthermore, it provides a means by which sidechain apps can be created without being constrained by the functional limitations of existing blockchain networks.

Crowd Machine also allows its community to contribute source code to its **Crowd Share** repository. Each time source code is consumed and executed on the Crowd Computer the community member who contributed the code is compensated. Crowd Machine aims to usher in a new era of blockchain network adoption and decentralized apps. The combination of dramatically accelerating decentralized app creation and empowerment of a large population of code contributors, results in unparalleled speed of delivery of solutions to market.

We believe that Crowd Machine will accelerate the adoption of blockchain networks and the creation of decentralized apps leveraging those networks. It will play a key role in ushering in a new stable of market-ready apps that can disrupt the dominance of centralized infrastructure companies.

Crowd Machine has battle tested its technology with Fortune 500 companies such as General Electric, AON Hewitt, KONE and Anthem and is already generating revenue.

02 | Problem and Value Proposition

The increase in app demand is far outpacing app creation, driven in part by the exploding blockchain and IoT markets. Gartner predicts that the business value-add of blockchain will grow to slightly more than $176 billion by 2025, and will exceed $3.1 trillion by 2030 [1]*. Maverick Research also suggests that Blockchain, AI and IoT are set to undermine current centralized platform business models* [2]*. This will require the creation of decentralized apps to replace those that exist for today's centralized model.*

2.1 Skilled Labor and App Creation

A report from VisionMobile estimates that the 300,000 developers contributing to IoT apps in 2014 must grow to nearly 4.5 million developers by 2020[3]. If you add to that the requirement for blockchain and AI developers, as well as sustaining existing legacy apps across all industries, it becomes obvious that there are insufficient skilled resources to support app creation and to sustain engineering demand.

The problem is already being felt in the blockchain and cryptocurrency industry today, with a drought of experienced Solidity developers globally. With the number of Initial Coin Offering/ICO projects currently underway, this makes it challenging for ventures to keep up with the acquisition of skilled labor to deliver products and services that match the pace (and hype) set by the blockchain and cryptocurrency field.

Crowd Machine's App Studio ("**Crowd App Studio**") makes it significantly faster and cheaper to build traditional and blockchain apps for those that understand logic and rule-based processes. Furthermore, onboarding and training coders and non-coders alike can happen quickly and, in turn, lead to an explosion of new and exciting apps.

2.2 Decentralized Cloud Computing

A decentralized model is reliant upon access to compute resources for app execution. If current centralized platforms are to be displaced with decentralized models, and scale is to be achieved, an extensive network of decentralized compute resources must exist. This network must be capable of executing any type of app complexity and deliver outcomes that meet user expectations. It must be able to scale against demand, offer redundancy to ensure availability, and be highly secure. It must also allow those that provide compute power to the decentralized network to utilize their devices' spare resources and be rewarded for it.

The world needs a new methodology for creating decentralized apps at speed to meet demand. To support those apps, it needs an efficient way to access and utilize distributed compute power.

[1] Gartner - Hype Cycle for Blockchain Business, 2017

[2] Gartner - Maverick Research: How Blockchain Undermines the Value Proposition of Platform Businesses

[3] VisionMobile – IOT:Breaking Free From Internet and Things

2.3 Solving the Processing and Programming Problems

The need for software apps is outpacing their creation, while at the same time, device memory and processor capacity remains largely underutilized. Crowd Machine harnesses both demand and underutilization with its decentralized app solution. It is creating the Crowd Computer, a powerful peer-to-peer computational network that leverages global device capacity to power the execution of blockchain and decentralized apps. Device owners are paid for allowing their devices to be a part of the app execution network.

Embedded within the app execution network is the Crowd App Studio, a zero-code app development technology. This technology enables anyone to create decentralized apps at a pace up to 45x faster than traditional methods. This is democratizing decentralized app creation by allowing anyone to create apps, and share those apps via the **Crowd Share** repository, a GitHub-like repository which includes a method for developers to monetize their work.

The Crowd Machine approach provides an incentive to encourage participation by a large population of contributors (the "**Crowd Machine Community**").

Crowd Machine is also focused on ensuring that the apps that drive blockchain adoption can deliver the levels of sophistication and scalability required to bring about their widespread adoption within the enterprise environment. This allows enterprise grade apps to be developed and tightly integrated into any blockchain network where they can act as sidechain solutions and drive blockchain consumption.

The combination of a global app execution network combined with the ability to create and deploy blockchain apps faster than ever before will greatly increase the reach and pace of adoption of blockchain networks and decentralized apps.

03 | Crowd Machine Vision

Crowd Machine's vision is:

> *To deliver the world's most advanced global app execution network and accelerate the delivery of blockchain and decentralized apps.*

Crowd Machine believes in empowering everyone to create amazing apps.

3.1 The Crowd Machine Community

Crowd Machine is building a global community of the next generation of app developers and decentralized compute network participants. This addresses the growing developer shortage and the distributed compute model required to run apps. The initial research and development work, and the creation and testing of the technology has been completed by Metavine (which has perpetually licensed its intellectual property to Crowd Machine) and is delivered to the blockchain and decentralized app community.

While the ability to address the explosive demand for apps using a low-code or zero-code approach has gained traction in the cloud environment, it is only with the release of Crowd Machine that it has surfaced in the decentralized app market. The blockchain has significant potential to change the way companies do business, especially in the ecosystem area.

A recent article by Shaan Mulchandani in ITProPortal[4] identifies five ways blockchains will:

> *"…positively disrupt the inception, development, operation and consumption of products, platforms and services. The fully decentralized system lowers transaction costs, improves record-keeping and enables complete transactional traceability. These benefits make blockchains ideally suited for the hyper-connected digital era — one in which trust boundaries between devices, networks, applications and users are being constantly redefined."*

Mulchandani continues,

> *"In the ecosystem of applications, the challenges can be daunting. The platform must validate and manage consent between different parties, in-app permissions, security and compliance checks for the app or service, automated developer onboarding and payment reconciliation between all participants in the process. Additionally, there are actions relating to the generation of app ratings and authorization to, or revocation/expiration of, services for developers and users."*

If Mulchandani is correct, this implies a migration of the current centralized app market to a decentralized model. This will require the creation of decentralized apps to replace many of the apps that manage today's business. It also opens the door to a stable of new apps that will leverage blockchain in unique ways. This is already clear given the volume of new blockchain apps currently making their way to market.

Gartner predicts the value-add of blockchain to reach $176 billion by 2025. Given the trend towards the decentralization of apps, including those not directly related to blockchain, that number could prove to be conservative. Gartner also suggests that the enterprise software sector market value is expected to be $392 billion in 2017 with a CAGR of 8.5% through 2021[5].

By using proven intellectual property, Crowd Machine will automate decentralized app development and leverage blockchain technology to do so. It is positioned to be the most influential driving force in the migration of existing monolithic solutions to a decentralized paradigm.

4.1 The Competition - An Overview

Crowd Machine competitors can be categorized as follows:

- Public Cloud Services Providers; and
- Enterprise Software Providers.

Within each category, the competing products can be characterized as proprietary or open source.

[4] ITProPortal access via: http://www.itproportal.com/features/five-ways-blockchains-can-transform-products-platforms-and-services/

[5] Gartner - Forecast Alert: IT Spending, Worldwide, 2Q17 ID: G00324598

4.1.1 The Big 4 Public Cloud Service Providers

Public cloud is a capital and resource intensive business that provides infrastructure under a centralized model that takes advantage of economy of scale. To compete in the public cloud space it needs very significant investment, in the billions of dollars including a legion of technical talent, to support that infrastructure.

The top cloud computing companies provide a range of complementary products and services, giving rise to business models such as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS). Due to the resource-intensive nature of the public cloud model, it is generally understood that this is a "winner takes all" type of game with very little opportunities for new entrants. Hence, the market has consolidated to a handful of leaders.

Currently, the top 4 public clouds are:

(a) **Amazon Web Services** (AWS) - offers a complete range of IaaS and PaaS services and is currently the undisputed market leader;

(b) **Microsoft** - dominates the public cloud space with Azure along with a range of other complementary Microsoft offerings such as Office 365, the online versions of its "Dynamics" line of enterprise software and development tools;

(c) **IBM cloud** - is a latecomer to the game but with its technical and financial resources, the company is strengthening its position in this space and has reported a 50% increase in its "cloud-as-a-service" revenue. IBM's most visible cloud service is Bluemix PaaS, which is aimed primarily at enterprise development teams; and

(d) **Google** - offers a range of IaaS and PaaS services that span compute, storage, networking, big data, machine learning, developer tools and security. Its flagship offerings include Compute Engine, App Engine, Container Engine, Cloud Storage and BigQuery.

Collectively, they control about 65% of the market share. The next 10 players collectively take less than 20% of the market.

Crowd Machine provides both a XaaS platform (X is a stand-in for Infrastructure, Software, Cloud, etc.) and a software development platform (respectively, the Crowd Computer and the Crowd App Studio/Crowd Share). One key differentiator is that it provides a decentralized solution to cloud computing, which does not require huge upfront investments in concentrated infrastructure similar to AWS or Azure. In addition, the cost and revenue to participate in the network is distributed amongst many participants instead of being centralized within a company.

Compared to the public clouds, Crowd Machine is infinitely scalable (limited only by the number of connected computing devices on the internet), and is not cost prohibitive.

4.1.2 Enterprise Software

As a zero-code development platform for enterprise grade distributed app development, the Crowd App Studio draws competition from across multiple well-defined conventional market players, including integrated development tools, cloud-based enterprise app development tools, and open source decentralized app development stacks, among others.

Enterprise software covers a broad range of products that include, but are not limited to, cloud-enabled productivity tools to mission-critical vertical apps and enterprise resource planning solutions. The top 10 vendors occupy nearly half of the market. Unlike the cloud service provider market, more opportunity exists for new entrants to offer better and more effective solutions to win over customers.

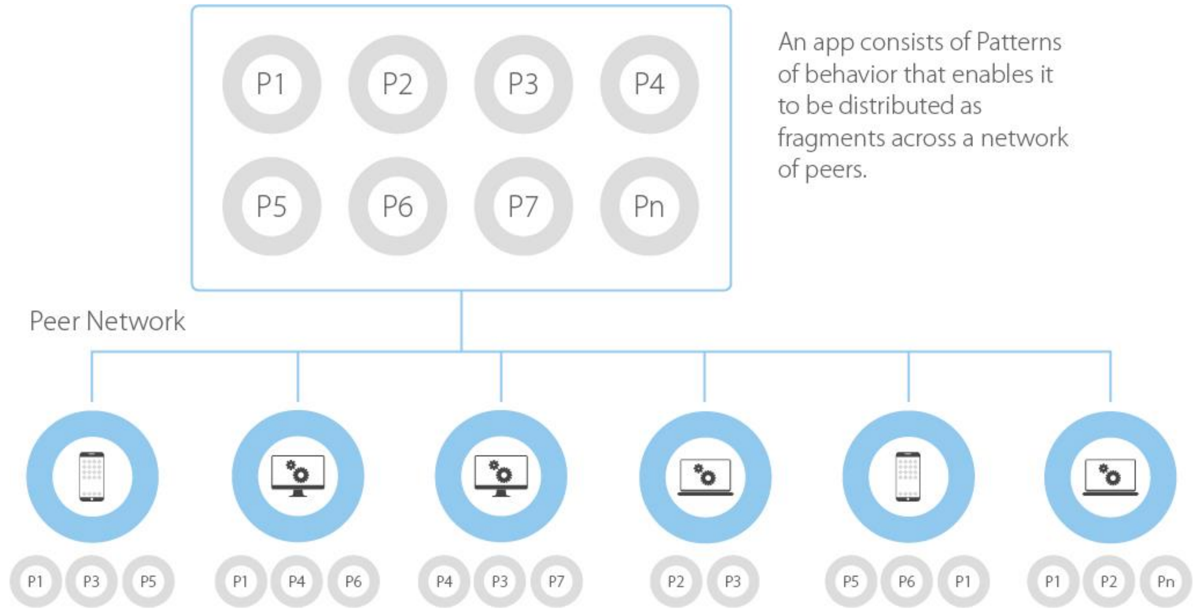
Today, tools for developing decentralized apps are mostly open source tools that present a significant learning curve, even for experienced developers. A search for the top 10 decentralized app development tools reveals mostly Ethereum and other blockchain development tools such as Mist, Truffle and the like. Microsoft recently started to offer BaaS (blockchain-as-a-service) in its Azure suite and offers a fully-fledged Ethereum IDE called Ether.Camp.

To-date, no other tool or platform offers the same combination of zero-code traditional and decentralized app development and deployment environments that Crowd Machine offers.

Crowd Machine's target market is large. From startups and small organizations to large enterprises and from 'no coders' to seasoned developers. The need is large, and so is Crowd Machine's appeal. Blockchain and IoT is exploding, and traditional markets like healthcare, banking and finance, manufacturing, mobile and consumer/retail are also increasingly focusing on software solution strategies to remain competitive. The result is an ever-increasing reliance upon apps to run the business.

Crowd Machine represents a new way of thinking about traditional and decentralized app creation and execution. It embraces a systems wide approach to deliver an intelligent decentralized cloud that contains a level of sophistication to meet any app need.

The Crowd Machine technology enables an app to be constructed and distributed as 'fragments' across a peer network. Rather than deploying an entire binary object, each fragment can be deployed and executed individually. The fragments are termed **Patterns**, and are a set of instructions that define a desired app behaviour. By combining a set of Patterns, an app is created. Patterns themselves are composed of **Activities** which represent specialized subsets of instructions.



An app consists of Patterns of behavior that enables it to be distributed as fragments across a network of peers.

Apps are created as a set of Patterns of behaviour rather than a monolithic binary executable. They consist of one or more Activities which instruct the Crowd Computer to perform the actions required of the app.

Figure 1: Apps are created as sets of Patterns rather than monolithic binary executable.

Crowd Machine's unique approach to app creation and execution allows all device types including mobile, desktop, data center and IoT devices to act in unison as a global processor. It provides the ability to leverage devices that would otherwise not have the capacity to execute a sophisticated large binary executable app. It also results in unrivalled scalability and redundancy, removes central point of authority related issues and significantly reduces the cost of app hosting.

05 | Crowd Machine Product Suite

The Crowd Machine technology is in production with numerous Fortune 500 companies including General Electric, Anthem, AON Hewitt, KONE and a number of large banking institutions.

The Crowd Machine technology consists of three core components that are combined to form a single cohesive solution. Their purpose is to deliver the most advanced blockchain and decentralized app technology available. The three components consist of:

(a) The Crowd Computer – an intelligent network of peers;

(b) The Crowd App Studio – an embedded development platform; and

(c) Crowd Share – a GitHub-like source repository.

For ease of understanding, the following sections describe each component of the Crowd Machine technology separately. However, they act in unison to deliver the speed of delivery and power of execution required to usher in the new decentralized app economy.

5.1 The Crowd Computer

The Crowd Computer network (“**Network**”) consists of peer to peer nodes of which there are two distinct types:

(a) Activity contributing nodes (“**ACNs**”); and

(b) Agoric nodes (aka super nodes).

The Network seeks to enable and optimize for a number of characteristics that consist of strong security assumptions, distribution of control, internal settlement for payment, node incentivization and trustless dispute resolution.

Computation within the Network takes place in a context that pre-determines what sorts of events can and cannot occur based on fixed, hardware-enforced rules with regards to access and changes to the state of the system.

5.1.1 Distributed Apps and Consensus

Agoric is the name of the distributed computer methodology for optimization of workload distribution and allocation of compute resources. Agoric consensus aims to provide a uniform, Byzantine-robust, secure landscape for the exchange of machine resources.

Agoric nodes form a Strong federation, with the purpose of signing blocks to advance current consensus of the Network. This has to do with balance management and settlement around the ledger, affirming who owns what resources. Additionally, Agoric nodes perform the following –

(a) Maintain and update Network topology of ACNs;

(b) Routing of generator requests to the appropriate ACN based upon the computational resource requirement; and

(c) Resolve payment channels.

Unlike monolithic apps which, during load time, often place the entirety of an app’s process into memory, Agoric deconstructs an app into parts (consisting of Patterns and Activities), generates blueprints for execution that map app components to optimal distributed machine resources, and allows apps to be synchronously and asynchronously loaded and executed by a network of distributed machines.

5.1.2 Securing the Network

Due to the substantial security guarantees that are required where trust is lacking, a Strong federation is elected to run distributed hardware responsible for specific Network tasks. This Strong federation is made up of a fixed (but addable) set of hardware distributed to selected, known, specialized, distrusting peers[6] (Agoric nodes).

Strong federations were developed as a technical solution to problems blockchain users face daily: transaction latency, commercial privacy, fungibility, and reliability - complete with Byzantine Security[7]. These are critical components to get right when striving for the guarantees required from a distributed computer.

The network operators of a Strong federation normally consist of two types of functionaries. Unlike with Sidechains (like the Liquid Network and Rootstock), where there are two types of functionaries:

(a) Blocksigners - who sign blocks of transactions defining the consensus history; and

(b) Watchmen - who are responsible for moving the assets out of network by signing transactions on another chain.

Agoric nodes facilitate cross-chain interaction via the use of the Lightning network, so there is no requirement for Watchmen.

Classically, functionaries (Blocksigners) are entities that mechanically execute defined operations if specific conditions are met, and to enhance security, certain operations are split between entities to limit the damage an attacker can cause. With Agoric, Blocksigners are given the additional responsibility of maintaining the list of possible computation resources, in addition to routing generator requests to the appropriate computational resource, and acting as dispute-resolution mechanisms under conditions where fraud/disputes occur[8].

For a Strong federation to be low latency, and eliminate the risk of reorganization from a given hostile minority, Agoric nodes replace the dynamic miner set common in blockchains such as Bitcoin with a fixed signer set (k of n, configured majority). As such, the Crowd Computer emulates the Byzantine robustness of Bitcoin where a minority of compromised signers will be unable to affect the system. In Strong federations, this k of n signing requirement requires full security of the hardware, which is distributed across multiple unknown locations and conditions. Agoric nodes utilize separate hardened devices for key storage and signing, to significantly reduce the number of avenues of attack.

[6] In addition, it is critical that functionaries (parties running secured, mechanically-secured hardware) have their economic interests aligned with the correct functioning of the federation. Incentives can be aligned through the use of escrow, functionary allocation, or external legal constructs such as insurance policies and surety bonds. Also, given the specific needs of the Crowd Machine, Functionaries could earn fees for the continued facilitation of the network.

[7] Strong federation consensus is deterministic, where each block is expected to be produced by a single party. Therefore, reorganizations cannot happen, unlike in Bitcoin or Ethereum, where they are an ordinary fact of life. In a Strong federation, Blocksigners need only to obtain consensus amongst themselves before extending history; since they are a small, well-defined set, the network heartbeat can be significantly faster than in Proof-of-Work, where settlement is probabilistic over time.

[8] Agoric nodes maintain and update the list of computation resources, route generator requests to the appropriate computational resource, and resolve disputes over proper execution.

5.1.3 Ledger of current balances

In a Strong federation, the knowledge of private keys is sufficient for the "right to spend" without the permission of any third party. Not only are the code updates open and auditable, and open to rejection in the case of coercive behavior, but the state of the system also provides a consistently reliable log that maintains immutability of state. Most importantly, the members of the federation cannot directly control any user's money inside the system other than their own. Payment between users on the Network is facilitated via public key to public key payment. For higher order contracts (outside of simple payments between parties), the Lightning network will be used.

5.1.4 Consistent and reliable network state of computational resources

Individuals contribute to the Network by electing to have their mobile devices, notebooks, desktops and servers, or other computational resources (bandwidth, storage, graphics processing, etc) onboarded to Agoric. We refer to the resources contributed by these individuals as being ACNs (i.e. Activity contributing nodes). This section seeks to describe how an ACN is onboarded so that they are 'available' for use by apps across the Network. It also outlines the methodology the Network utilizes for keeping a globally-consistent perspective on the state of all ACNs concurrently.

Connection to, and use of, the Network requires that there be a universal understanding of all known current resources across the Network. An onboarding ACN connects to the known functionary set, and from that point on (with periodic re-authentication and benchmarking), connects directly to the lowest-latency functionary. All communications are end-to-end encrypted, signed to guarantee security and authenticated-peering, and require minute amounts of value to be posted.

View from an arbitrary ACN:

An ACN connects to the Network via the known public Internet addresses associated with the Agoric federation. Communication is authenticated, and the ACN commits to a specific entropy value (identifier). The ACN is assigned a benchmarking task to test varied attributes[9] associated with their machine (this can include user-mediated value input in order to set firm/hard limits on bandwidth, etc). After the benchmark runs, the ACN is awarded a score, and a specific functionary node with which to communicate under most common conditions. The ACN will periodically have to prove it continues to have the characteristics evaluated in the benchmarking.

The Network can then allocate work to the ACN under conditions where an app's requirement (whether they be latency, cost, or otherwise) is met by the ACN.

View from the Network:

An ACN connects to the Network via the known public internet addresses associated with the Agoric federation. The Network queries the node via the standard benchmark (with additional, specific benchmarks possible under conditions where the node meets, exceeds, or fails certain thresholds). Then, either based on a fixed standard, or relative to all currently connected ACNs, the Network assigns a score for the ACN depending on task and type of computation/behaviour executed.

[9] The benchmarking function is critical for scoring the fundamental capacity and effectiveness of a node. Test needs to include: (a) Up-time over a random sampling period; (b) Last-n-accepted results from compute - Latency to arbitrary Network nodes; (c) Bandwidth, spontaneous and consistent; (d) Processing power; (e) Ram; (f) Storage capacity.

This score can be by specific task, or as a general metric. The Network will periodically have ACNs prove that they continue to have the characteristics evaluated in their respective benchmarking, with variance captured when possible to allow for more fine-grained market mechanisms. The Network allocates work to arbitrary ACNs under conditions where an app's requirements (whether they be latency, cost, or otherwise) are met by the ACN.

5.1.5 Generator and Blueprints overview

An Agoric generator is connected to other generators via payment channels for determining optimal routes for app task, execution pathway blueprints. A blueprint guides an app through the resources most uniformly optimized for its load and execution needs. Each generator will return one or more possible blueprint resolutions for an app's computational resource, load and sustained execution needs. Using one of, or a series of, payment channel relationships with associated Script Structures (refer to MAST at 5.1.6 below for more information), a generator can natively communicate with its source app from an Agoric node. The app then enters into Payment Channel relationships with usury bonds associated with the particular ACN/s selected for blueprint/s based on the app's requirements.

Regardless of our best efforts, it is likely that there will be sybil and adversarial conditions in play whereby individuals attempt to mimic or bypass the ACN benchmarking methodology, and result in less-than-optimal behavior with regards to specific app blueprints. Unfortunately, there is no known way to avoid these adversarial conditions - the best we can strive for and work towards is an ever-growing absolute value and proportion of the Network as honest nodes.

A node's behavior can be tracked, and their relative reliability can be measured (and assigned) based on their consistent (or inconsistent) adherence to tasks (app failures etc incrementally lower a node, app success incrementally increases a node's score). At certain thresholds (defined by the federation, or by app type/specification) certain nodes are disqualified from participating in the blueprinting process, and at even further thresholds, entirely partitioned off the Network.

5.1.6 Generating Blueprints and Resource Routing Requests

Agoric nodes view apps as their respective functional components which consist of Patterns, Activities and natural language expressions. Agoric node generators construct blueprints utilizing a Merkylized Abstract Syntax Tree ("**MAST**") of the Patterns and Activities of the app, only revealing portions of the app necessary for execution and redemption. Unused patterns and activities of app execution are pruned off, enhancing privacy and decreasing resource requirements.

The generators generate blueprints that utilize the benchmarking provided by ACNs to form pathways from source app patterns and map to optimal compute resources. Each pathway for blueprint generation can prioritize for a variety of requirements while optimizing for capable resource availability. To ensure robustness of the solution, redundant secondary and tertiary ACNs accompany all primary ACNs in the blueprint. If a primary ACN becomes unavailable, Agoric nodes fails over to subsequent nodes.

Blueprint failover ACNs exist on separate Network routes to obviate the risk of route failure or compromise impacting the expected compute outcome. If the secondary or tertiary ACNs are also unavailable, Agoric nodes will dynamically calculate a new blueprint using currently available ACNs. This generation process can occur until the Network has resulted in the desired outcome.

Benchmarking combined with market pricing dynamics will be the driving factors in blueprint generation for optimal app execution. The use of market mechanisms can yield orderly systems beyond the ability of any individual to plan, implement, or understand. Agoric nodes allow economic conditions to be independently maintained by node owners across geographies. This creates market driven Network competition that ultimately drives pricing outcomes. To remediate centralization bias to the extent possible, Agoric nodes dispatch asynchronous tasks farther afield by removing latency from the design for those tasks. This approach results in sharing load across the Network and decentralizing the execution of apps.

5.1.7 Blueprint optimization for geo-local differences

Because Agoric nodes have a complete view of the computational topology, they can allocate resources optimally from a multi-linear perspective. Apps compete for the minimum cost ACNs, meeting their relative needs amongst themselves, with the dependencies of a given app representing the minimum values that allow for an ACN to compete for executability. In generating a blueprint, Agoric takes into consideration the origination point of the request when weighing the competitive eligibility of the potential executing ACNs. Depending upon the user preference for cost versus performance, differing blueprints are potentially generated for differing geo-locals.

5.1.8 ACNs and Payment Channels

ACN Network activity is managed via Payment channels. The channel features are designed to take advantage of HTLC and MAST, along with secure communications, authentication (eg. PGP), and enable for n-of-m parties to exchange pre-publication encrypted meta-data with sufficient consensus for the transaction. Support for API extensions for other protocols as well as HD-key generation for addresses and Lightening is built into the Crowd Computer.

5.1.9 Proof of execution

Agoric will make use of usury bonds to ensure that execution attempts are made in good faith. Step-functioned Script Ratcheting provides tighter control over the distribution of an app’s resource allocation. This in turn ultimately secures fund distribution upon ‘completion of execution’ and minimizes adversarial conditions caused by capitalized attackers.

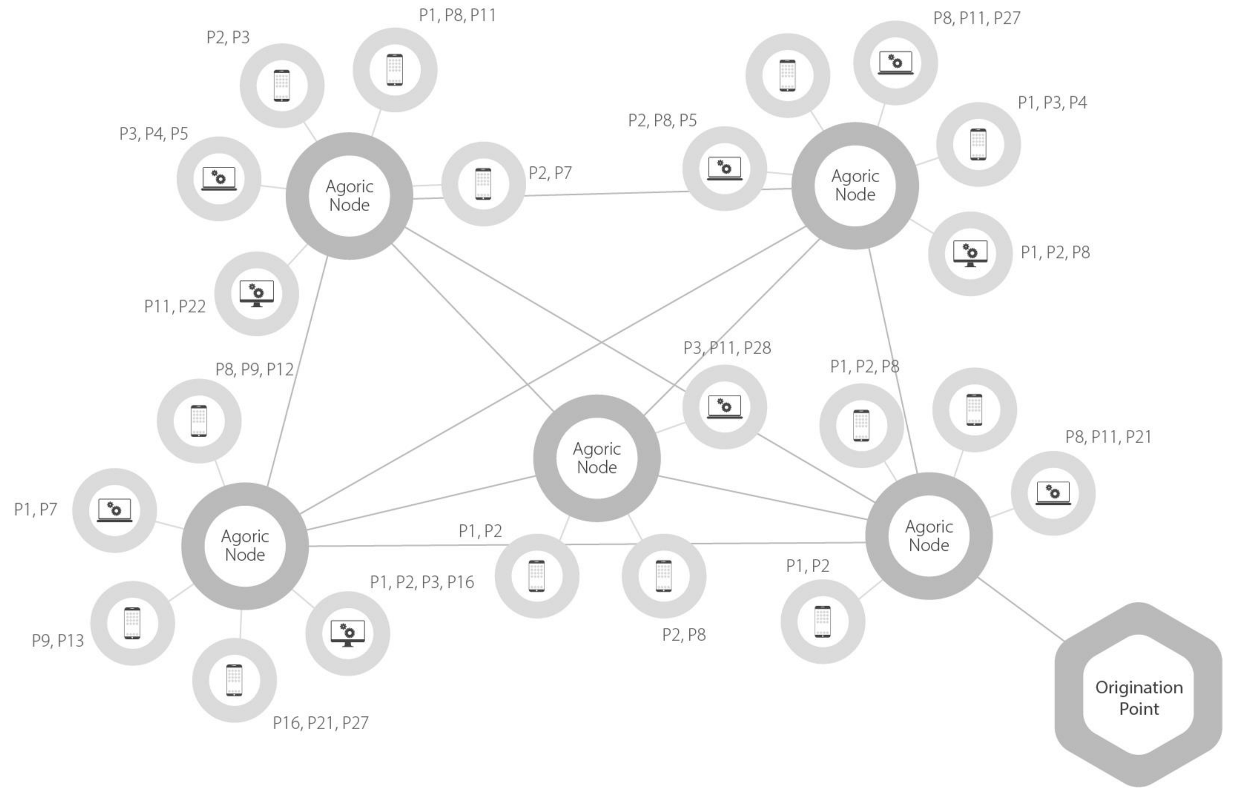


Figure 2: Crowd Computer network illustrating Agoric nodes and ACNs.

5.1.10 The Crowd Virtual Machine

Underpinning the Crowd Computer is a network of CVMs (i.e. Crowd Virtual Machines) that are responsible for executing the Patterns that define an app. Simply, a Pattern is a set of Activities that instruct the CVM in its function to produce the desired app outcome. Given that apps exist as individual Patterns, an app can be executed across many CVMs acting in unison to deliver the defined functionality. By utilizing the currently untapped depth of existing computational power on billions of devices, Crowd Machine creates a new paradigm in distributed computing allowing massive app scalability, no single point of authority and at reduced costs in comparison to current centralized models.

5.1.11 How it Works

As mentioned at 5.1.10, each Activity in a Pattern is as a set of instructions that directs a CVM's function. In executing Activities, the app requirements are met.

As an app request is made, it is distributed to a CVM as directed by an Agoric blueprint for the app. Once a request is accepted, the CVM initializes its state, loads its Activity data and executes the developer defined Activity instructions. The result is the execution of the desired app requirement.

The CVM utilizes numerous techniques to maximize Activity throughput performance. Activity definitions are held in memory to be available for immediate execution upon request thereby removing the need to load data from storage.

It also evaluates each Activity to determine its eligibility for asynchronous processing. When an Activity is judged eligible, the Crowd Computer calculates a blueprint that reduces the latency weighting for ACN eligibility for execution of that task. The approach assists in ensuring apps scale to meet load demand and provides improved user experience.

5.1.12 Developer Extensions to the Crowd Virtual Machine

The CVM is open to extension using the Crowd App Studio. The Crowd App Studio allows functionality to be defined and exposed as RESTful APIs. The APIs become available as extensions of the CVM, offering extended functionality to apps deployed on the Crowd Computer. Furthermore, extensions can also act purely as RESTful relays to expose third party or external functionality on the Network. The approach provides infinite extensibility to the functionality available to the CVM and the apps that run on it.

5.1.13 Node Rewards and Performance Versus Price App Tuning

Of utmost importance is the proper identification of rewards for ACN and Agoric nodes. This incentivizes the Crowd Computer participants to cover the costs utilizing their devices to provide compute power to the Network. Such cost will depend on where the nodes are located (for example, Internet connectivity may be more expensive in Africa than in the USA). As it is impossible to determine the relevant conditions in all environment types, the node participants will have the ability to determine their cost in their sovereign currency, which shall be converted to a tokenized form. App developers and customers will then be able to adjust whether they want performance as the basis of running their apps or cost efficiency or in some instances both.

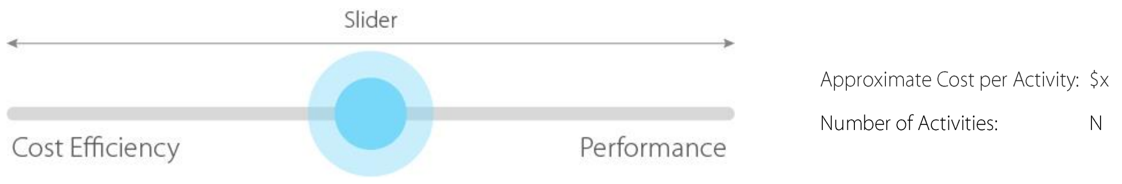


Figure 3: Determination of Performance.

When an app is served to an originator of a request, the Agoric nodes will distribute the Activity computation to the ACNs based on the cost preference of the app developer. Once the work of the ACN is complete, it will pass back the resultant computation to the first Agoric node that relayed the work and, at that time, the compute cost for the ACN will be calculated and logged. The Agoric node involved in relaying a request will also be rewarded for its work. Note that an Agoric node may connect to another Agoric node, but the ultimate ACN node will revert back the information to the first Agoric node. In this instance:

Agoric node 1 Reward > Agoric node 2 Reward

At the end of a predetermined block time, the node rewards will be distributed pursuant to the record of the work performed by same. Together with this, Crowd Machine shall obtain a percentage of the reward for the utilization of the Network. Therefore, Crowd Machine's revenue shall be tied to the utilization of Activity and Patterns built in the Crowd App Studio and run on the Crowd Computer.

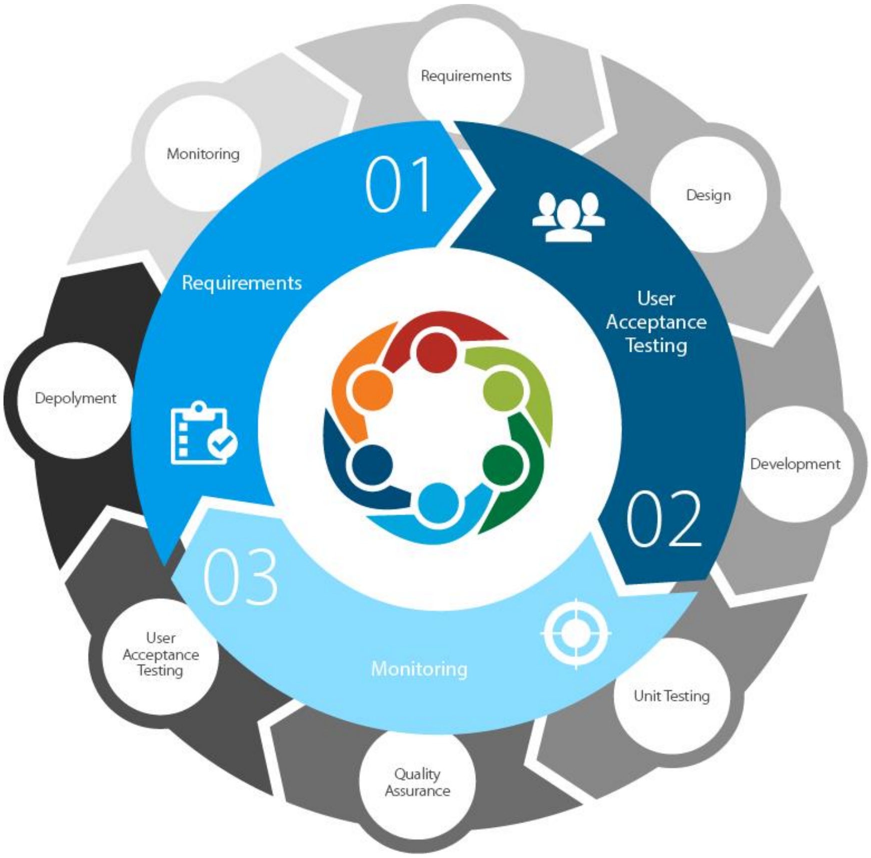
5.1.14 Open Source

The Crowd Virtual Machine will be made available as an open source project in Q4 2018.

5.2 The Crowd App Studio

Crowd Machine has reinvented the nature of app delivery and execution. It accelerates the creation and management of decentralized apps by automating the app lifecycle process. With Crowd Machine, non-technical users can create sophisticated traditional, blockchain and decentralized apps without writing code.

It allows app features and functions to be drawn as logic diagrams, making it faster and easier to create apps than current development approaches.



Crowd Machine automates many aspects of the app lifecycle enabling apps to be created at speeds of up to 45x faster than today's approaches.

Figure 4: Crowd Machine Automation Vs 8-Stage Traditional Model

The approach allows a much larger community of app developers to participate, and its accelerated app delivery dramatically impacts the pace of blockchain adoption.

5.2.1 How it Works

All apps exhibit behaviors where each behavior represents some form of functionality.

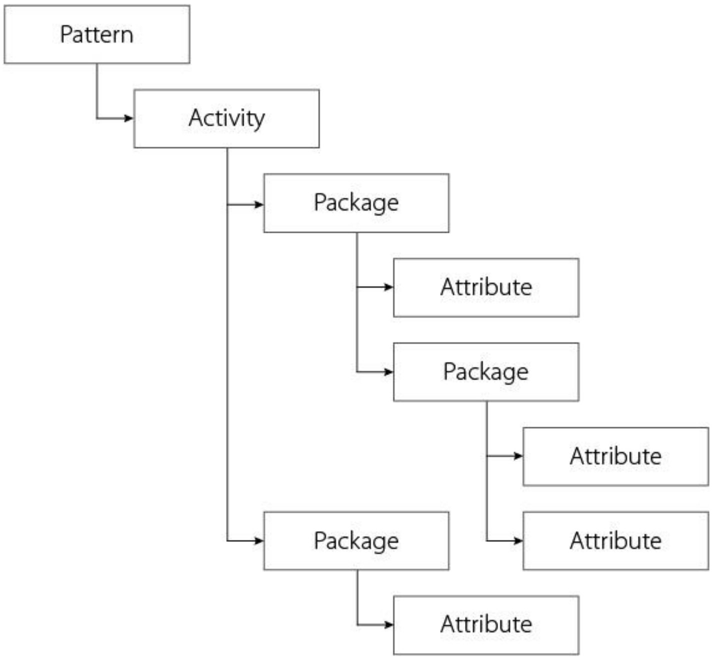
Crowd Machine allows developers to model behaviors as a Pattern, rather than writing code.

Patterns are a diagrammatic presentation of logic that represents an app requirement. They are composed of one or more abstract types known as an Activity.

An Activity is a logical block that acts on one or more sets of data. The data containers applied to an Activity are known as **Packages** and they contain artefacts of data termed an **Attribute**. The approach allows a 'class' like representation to be modelled, against which executable logic can be applied. Furthermore, Packages can be sourced from external systems and represented virtually within the Activity.

This allows composite construction of remote system data and functionality.

By combining Patterns, an app is formed. The unique approach removes the requirement to understand any form of computer language or the traditional intricacies of app development. This speeds the pace of app delivery by removing the development and unit testing steps of the app lifecycle, allowing a developer to go directly from requirements to user acceptance testing.



Each pattern contains at least one Activity, which in turn contains one or more Packages. Packages are intelligent containers that hold Attributes of data. The approach provides a method of modelling relationships. An Activity can contain multiple top-level Packages, which act independently of one another. Where a Package is hierarchically related to another Package, a relationship is formed that causes automated linking of data and functionality.

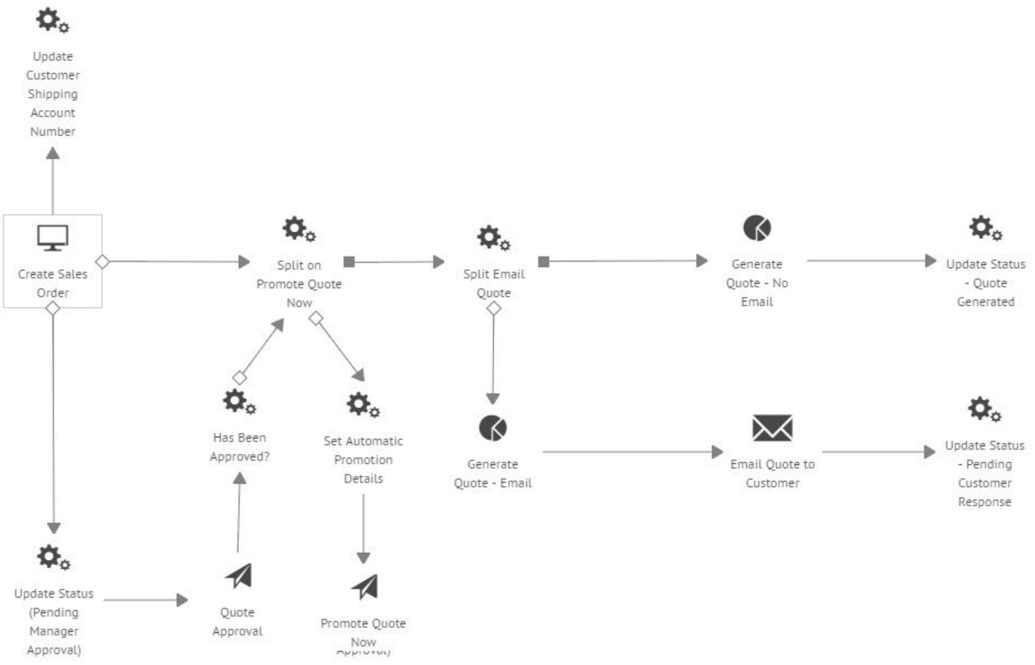
Figure 5: Patterns model app behaviors.

Activities can be joined together using **Links**. Links allow unlimited 'if-then-else' type logic to be applied to the movement between Activities and define the pathway through which runtime execution will occur. They also provide a means by which exception conditions can be modelled. The approach enables visualization of conditional branching within a Pattern.

Each Activity type contains a set of preformed capabilities native to that type. Activity types consist of:

- User Interface
- Background task
- Integration
- Scheduling
- Messaging
- Reporting
- Pattern Launch

Activities act to reduce the amount of effort required to create functionality in an app. They are templates that allow a developer to define Activity specific properties that drive their final behavior. By defining the properties, an Activity will function in the Pattern as instructed.


Update Customer Shipping Account Number
Create Sales Order
Split on Promote Quote Now
Split Email Quote
Generate Quote - No Email
Update Status - Quote Generated
Has Been Approved?
Set Automatic Promotion Details
Generate Quote - Email
Email Quote to Customer
Update Status - Pending Customer Response
Update Status (Pending Manager Approval)
Quote Approval
Promote Quote Now

Each Activity in a Pattern acts as a template for a specific behaviour. Developers configure an Activity to perform the desired task instead of writing code.

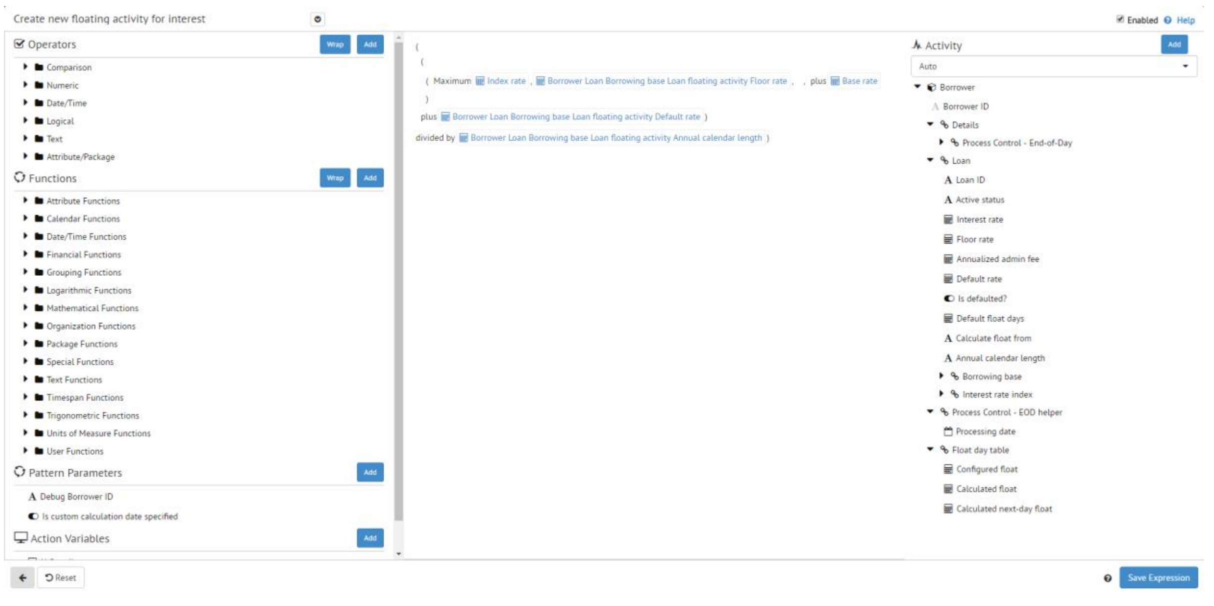
Figure 6: Logic is modelled using a diagrammatic form.

Activities enable both structural definition and properties to be defined. The Activity Definition panel is the area in which the Activity is told which Packages it will utilize and act upon. The Activity Properties panel is the area in which behavioral properties are applied. Properties vary based upon Activity type and instruct the Activity how to act upon its Packages.

5.2.2 Natural Language Expressions

All Activities enable logic to be defined against them. The logic takes the form of a natural language expression where the expression utilizes the developer-defined properties to act upon the **Activity Definition** – the Packages and Attributes associated with the Activity.

Crowd Machine's natural expression logic alleviates the need to understand a computer language and dramatically accelerates the implementation of an app. It is comprehensive and unlimited in its capacity to define any form of logic required. It is driven by an expert-system style approach whereby the developer responds to questions posed based upon responses to previous questions and Package relationship context. All expressions begin with a root node question set and branch conditionally from that point. Complex expressions are also supported and include access to a vast library of internal and third party methods that may be applied. The expert system driving the construction of an expression restricts the developer from creating code bugs. It enforces strict type checking and removes the ability to create problematic conditions such as infinite loops.



```
(
 (
  ( Maximum Index rate , Borrower Loan Borrowing base Loan floating activity Floor rate ,  , plus Base rate
  )
 plus Borrower Loan Borrowing base Loan floating activity Default rate )
divided by Borrower Loan Borrowing base Loan floating activity Annual calendar length )
```

Crowd Machine contains many pre-built expressions that accelerate the development process. It also allows for third party expressions to be called giving the developer access to any external services they require.

Figure 7: Crowd Machine pre-built expressions.

Expression dependency analysis is undertaken on behalf of the developer. This allows the developer to focus on the logic and not on the sequence in which it must be applied. Orphaned expressions are also identified automatically and excluded from the execution process.

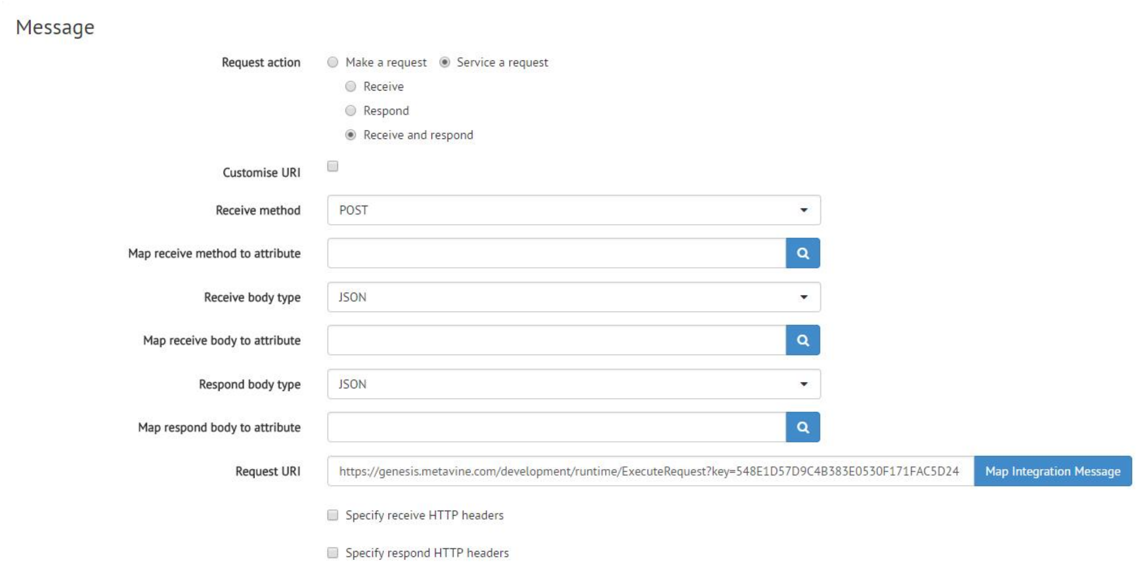
5.2.3 UI Design

The Crowd App Studio contains a forms design environment for the creation of any 'look and feel' required. The forms designer allows complete customization of experience across multiple form factors. It natively supports responsive design against specific device resolutions.

Class definitions and style sheets can be graphically defined and applied to form controls. Natural expression logic can also be used to customize user experience at runtime. There is a large library of native form control types that can be drawn upon during the design experience to ensure that developers can accommodate a variety of design outcomes.

5.2.4 Integration

The Crowd App Studio provides a number of approaches for integrating with external solutions including integration with Third Party APIs, Databases and authentication services. It also provides extensive RESTful API support for both request and response operations allowing the consumption and publishing of RESTful APIs. All forms of authentication types are supported including OAuth2. Templates are provided to automate the authentication process with many of the more popular vendors such as Google, Facebook, Twitter and others.



Message

Request action: ○ Make a request ● Service a request
○ Receive
○ Respond
● Receive and respond

Customise URI: □

Receive method: POST

Map receive method to attribute:

Receive body type: JSON

Map receive body to attribute:

Respond body type: JSON

Map respond body to attribute:

Request URI: https://genesis.metavine.com/development/runtime/ExecuteRequest?key=548E1D57D9C4B383E0530F171FAC5D24

□ Specify receive HTTP headers

□ Specify respond HTTP headers

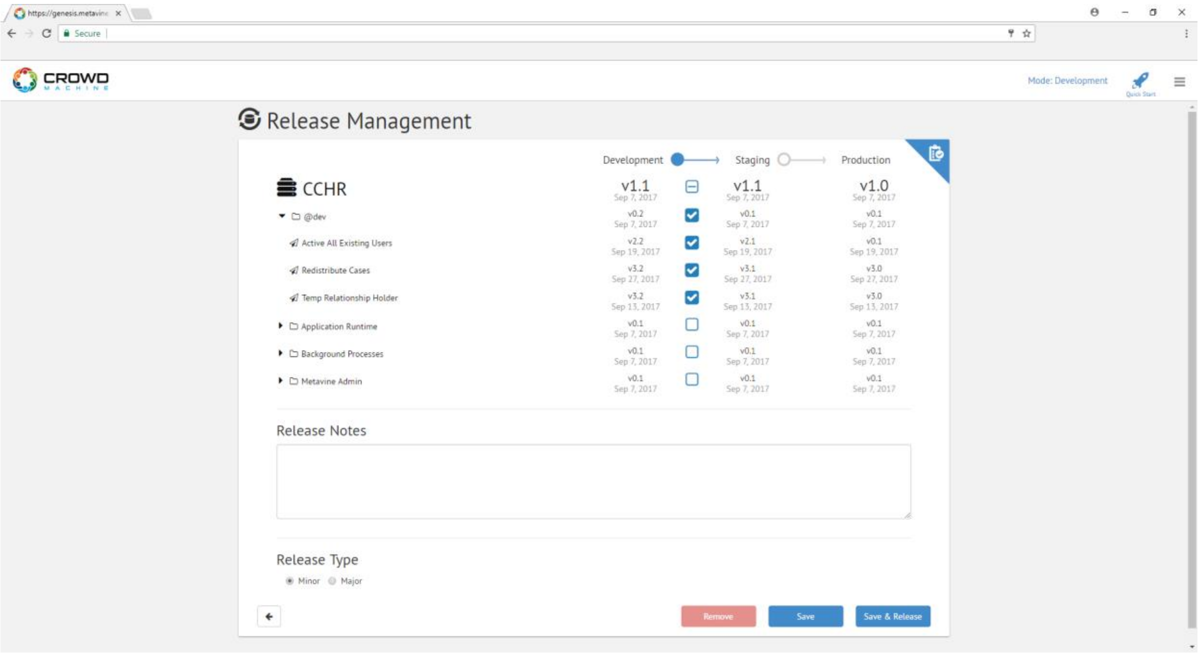
Crowd Machine supports many integration types including RESTful APIs for both request and response types. Integrating into external RPC interfaces is dramatically simplified and further accelerates the pace at which an app is created.

Figure 8: Integration Types supported by Crowd Machine.

The RESTful API integration capability of the Crowd App Studio enables blockchain Network integration. Further details on blockchain integration support are provided below.

5.2.5 Release Management

The Crowd App Studio contains the ability to perform app releases to the Crowd Computer. Patterns are versioned and moved from Development to Staging and then Production as required. Dependency analysis is conducted whenever releases are conducted. When inconsistencies are detected the developer is presented with a list of potential release issues and the release process is halted until the issues are resolved. Replication of the app Patterns across the Crowd Computer is undertaken automatically on the developer's behalf. There is no requirement to utilize deployment containers such as Docker.



Release Management

Development → Staging → Production

CCHR	v1.1 Sep 7, 2017	⊟	v1.1 Sep 7, 2017	v1.0 Sep 7, 2017
@dev	v0.2 Sep 7, 2017	☑	v0.1 Sep 7, 2017	v0.1 Sep 7, 2017
Active All Existing Users	v2.2 Sep 19, 2017	☑	v2.1 Sep 19, 2017	v0.1 Sep 19, 2017
Redistribute Cases	v3.2 Sep 27, 2017	☑	v3.1 Sep 27, 2017	v3.0 Sep 27, 2017
Temp Relationship Holder	v3.2 Sep 13, 2017	☑	v3.1 Sep 13, 2017	v3.0 Sep 13, 2017
Application Runtime	v0.1 Sep 7, 2017	□	v0.1 Sep 7, 2017	v0.1 Sep 7, 2017
Background Processes	v0.1 Sep 7, 2017	□	v0.1 Sep 7, 2017	v0.1 Sep 7, 2017
Metavine Admin	v0.1 Sep 7, 2017	□	v0.1 Sep 7, 2017	v0.1 Sep 7, 2017

Release Notes

Release Type

● Minor ○ Major

A release management console with inbuilt versioning is provided within the Crowd App Studio. Releases can be made from Development through Staging to Production modes. Comprehensive impact analysis is undertaken at each release stage to ensure that releases do not result in error states.

Figure 9: Release management console.

5.2.6 Security

Crowd Machine remains compliant with a number of standard protocols to ensure that user, device and service provider authentication requirements are challenged according to security standards. Each tier of the Crowd Machine product employs independent security mechanisms, including multiple levels of firewalls and standard industry intrusion detection and prevention mechanisms.

At its core, every user on the Network is assigned a profile associated with a private key that drives both the authentication and authorization mechanisms. For a user to obtain a profile, it must be allocated to them either by an app developer or via an app running on the Crowd Computer. A verification process is used to ensure the details of the profile are valid.

Authentication is achieved through a custom security model which is defined by app developers. The authentication is implemented using a standard username and password model, or via integration with an external SAML2 provider, which is uniquely linked to a given profile. The model includes an administrator configurable password policy, which can be strengthened as needed (e.g. increase the frequency of password changes). Once a given profile has been authenticated, the profile is authorized to access a limited set of resources on the Network (e.g. Patterns), based on the custom security model.

Authorization is achieved using OAUTH2 which governs the issuing of tokens for a given profile and then subsequently used to approve requests to access specific resources. The Network and CVM (Crowd Virtual Machine) have been designed to verify access tokens at various boundaries in the Network and on the CVM.

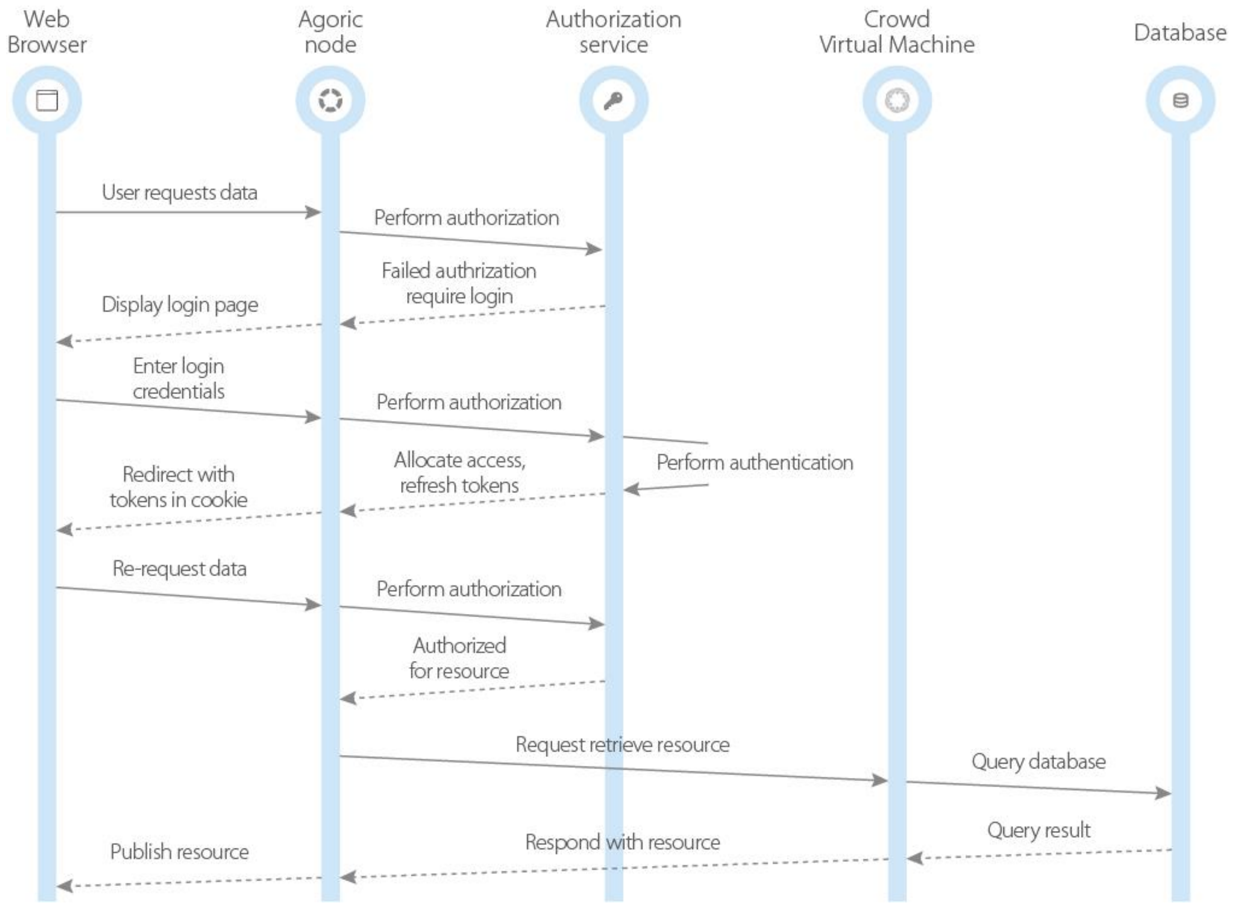
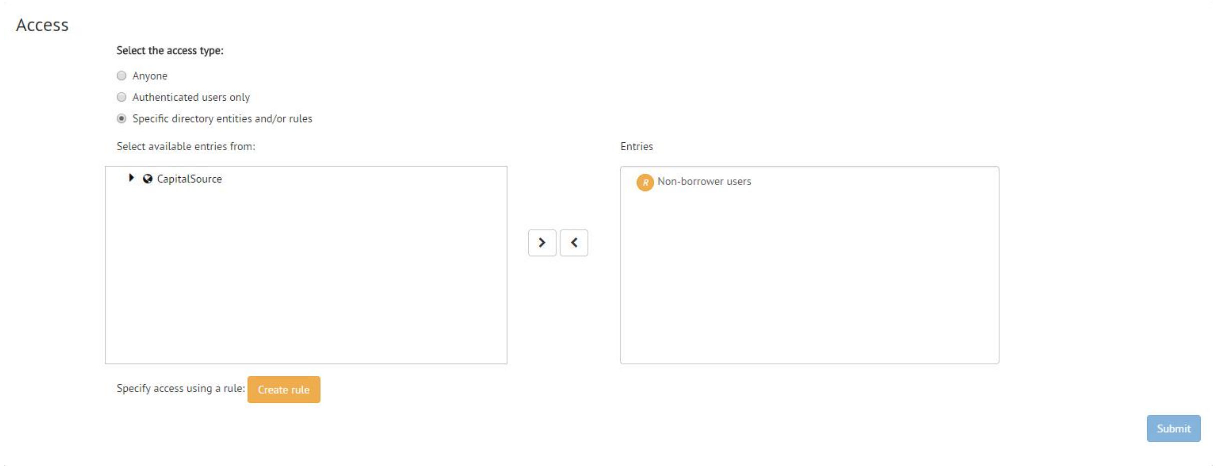


Figure 10: Example authentication/authorization flow for executing a Pattern.

As an example of the above process – let us assume a user wishes to execute a given app on the Network, so they launch a web browser with the relevant URI to that app. The Agoric node determines that the user has not been previously authorized, and redirects the user to a login page. The user provides login credentials, and the Agoric node authenticates the user via the Authorization service. If authentication is successful, it authorizes them to launch the initially requested Pattern and any Patterns for which they have been granted access. The web browser redirects to the initially requested Pattern URI and executes the first activity in that Pattern.

Additional security measures can be applied at a logical level within an app. The developer can optionally extend the security layer to restrict access using natural language expressions. Permissions can also be applied to groups and directory based objects natively or via expressions.



Access

Select the access type:

○ Anyone
○ Authenticated users only
● Specific directory entities and/or rules

Select available entries from:

CapitalSource

Entries

Non-borrower users

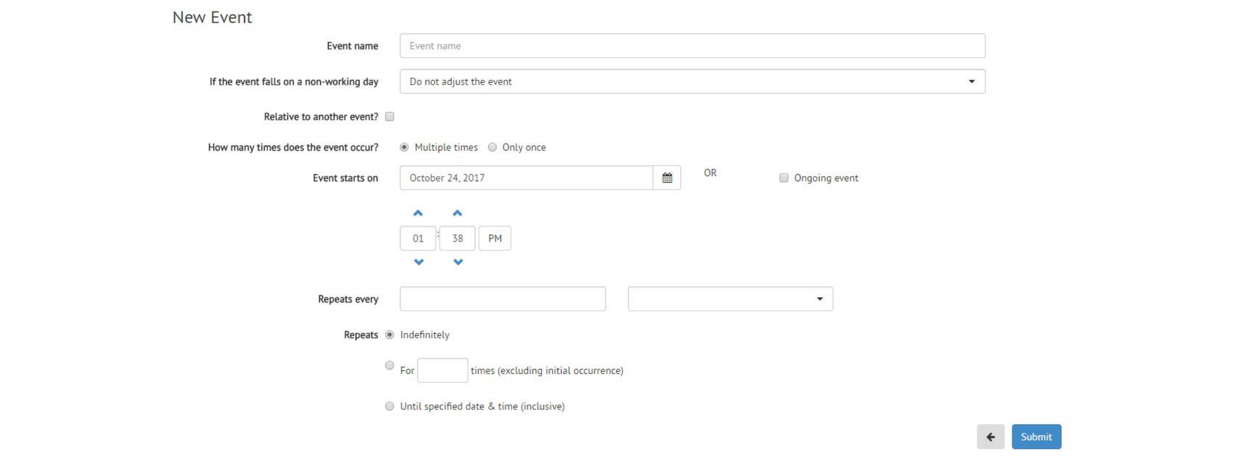
Specify access using a rule: Create rule

Developers have extensive access control to their apps. Natural language expressions can be formed to control permissions or behaviors within an app. Crowd Machine applies industry standard cryptographic protocols to all communications and data.

Figure 11: Extensive access control to apps for developers.

5.2.7 Events and Event Scheduling

Data and time events can be captured by Crowd Machine and used to initiate Patterns. After defining an event to be monitored in the Crowd App Studio, it is linked to an app Pattern causing automatic Pattern execution upon occurrence of that event. Patterns can also be scheduled to execute at specific times as well as being subject to natural language expression invocation. Additionally, events are used to disrupt or alter Pattern behavior as required.



New Event

Event name: Event name

If the event falls on a non-working day: Do not adjust the event

Relative to another event? □

How many times does the event occur? ● Multiple times ○ Only once

Event starts on: October 24, 2017 OR □ Ongoing event

01 : 38 PM

Repeats every

Repeats ● Indefinitely

○ For times (excluding initial occurrence)

○ Until specified date & time (inclusive)

Developers can schedule events to invoke a Pattern. They can also create data event monitors where a change of data state can be linked to a Pattern causing the Pattern to fire when the data state is altered.

Figure 12: Ability to schedule events to invoice a Pattern.

5.2.8 Blockchain Oracle Creation and Auto-Expression of Smart Contract Methods

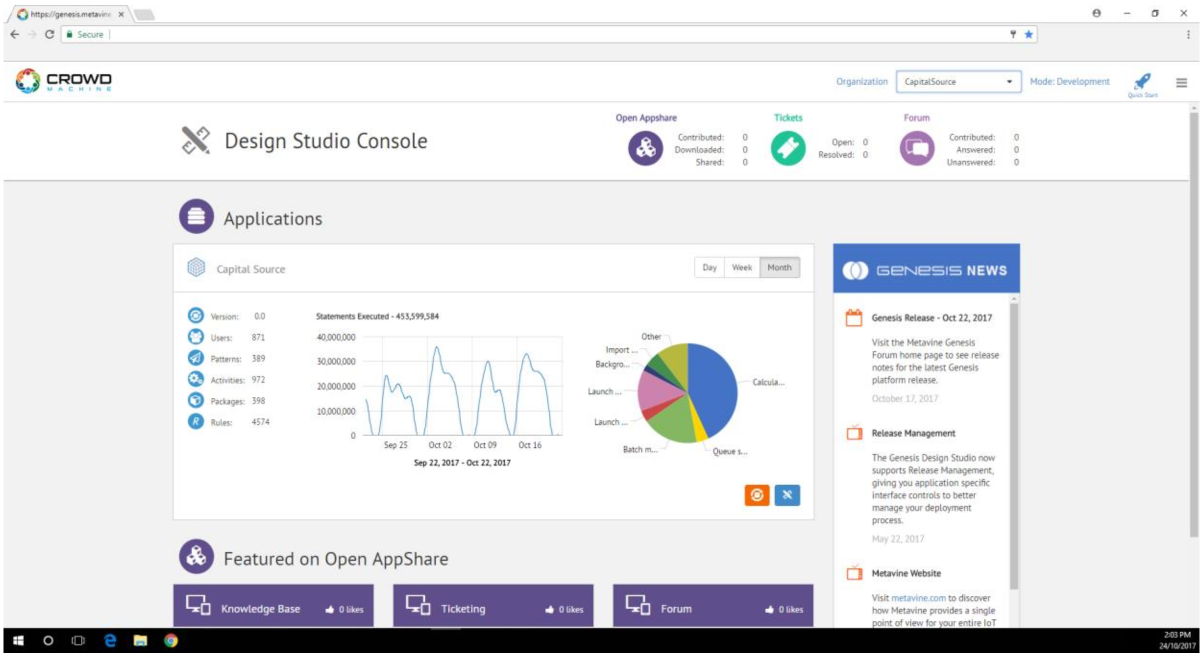
Events can be used to induce blockchain Oracle apps. By creating an event that monitors the state of an arbitrary data source, any change in that state can result in the invocation of a Pattern, which executes its defined function. The Pattern in turn may call a Smart Contract method to perform a desired task.

The Crowd App Studio allows the developer to import an Ethereum smart contract Application Binary Interface ("**ABI**"). The process of importing the ABI generates method definitions for use within Crowd Machine natural language expressions. The methods can be called from expressions to enable interaction with the smart contract as required.

By providing the ability to monitor the state of data and make calls to smart contract methods, Crowd Machine brings unlimited extensibility to Ethereum smart contracts. Apps can be created to extend the power of the smart contract where the contract itself is limited in its functional capability. It is an off-chain method of providing capabilities such as workflow, third party system integration, process management or any other requirement to the smart contract.

5.2.9 Crowd App Console

The Crowd App Console enables a developer to monitor the use of a deployed app on the Crowd Computer. It provides granular insight into app performance down to a natural language expression level. Effectively, it reports the number of blocks of logic executed within an app. It uses this information to calculate the cost of running the app on the Network.



Organization: CapitalSource
Mode: Development

Contributed: 0
Downloaded: 0
Shared: 0

Open: 0
Resolved: 0

Contributed: 0
Answered: 0
Unanswered: 0

Version: 0.0
Users: 871
Patterns: 389
Activities: 972
Packages: 398
Rules: 4574

Genesis Release - Oct 22, 2017

Visit the Metavine Genesis Forum home page to see release notes for the latest Genesis platform release.

October 17, 2017

Release Management

The Genesis Design Studio now supports Release Management, giving you application specific interface controls to better manage your deployment process.

May 22, 2017

Metavine Website

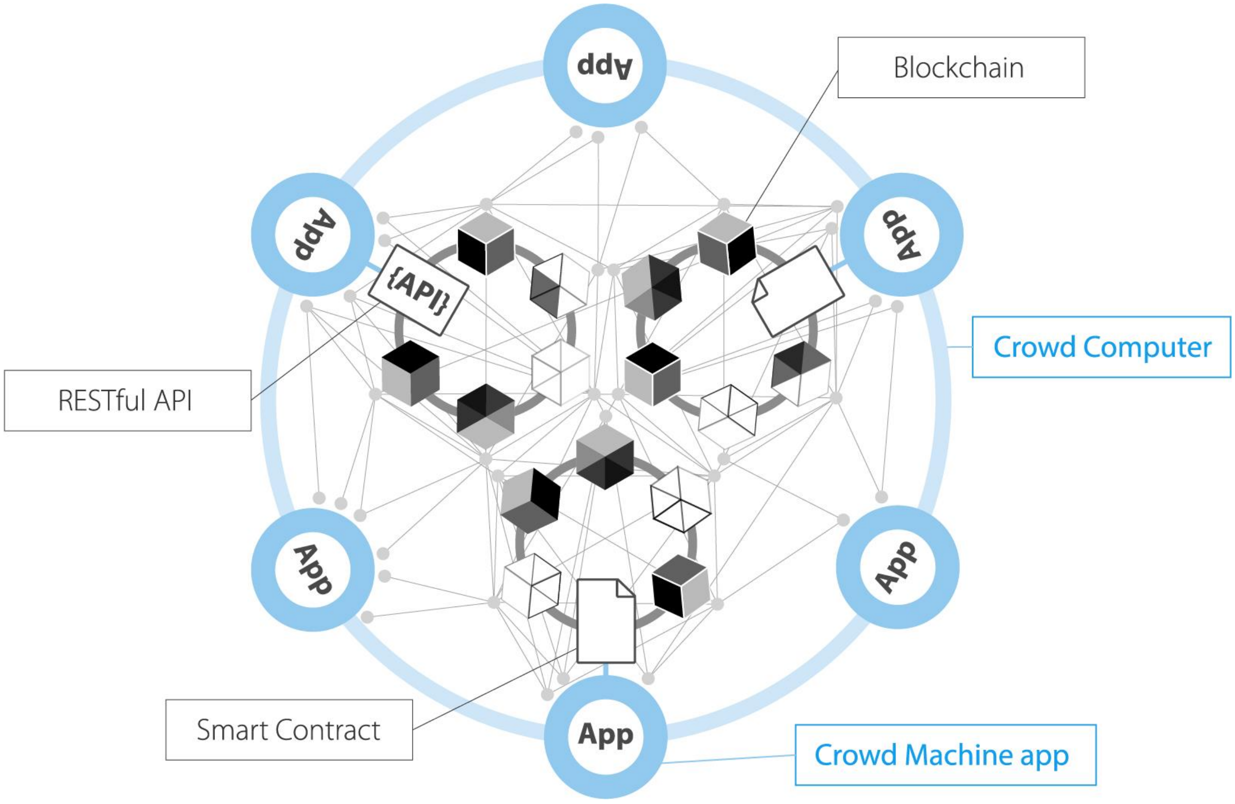
Visit metavine.com to discover how Metavine provides a single point of view for your entire IoT

The Crowd App Console provides insight into app utilization on the Crowd Computer – a network of peer to peer devices. Apps are distributed as Patterns across devices on the Network where they are executed upon demand.

Figure 13: Crowd App Console.

5.2.10 Agnostic Blockchain Support

Crowd Machine has an open approach to the utilization of blockchain networks. By creating an app using Crowd Machine, developers have the freedom to quickly migrate their app from one blockchain to another if required. Crowd Machine allows off-chain apps to be built that offer far greater power and flexibility than a smart contract alone.



Crowd Machine is blockchain agnostic. It allows off-chain apps to be created and run on the Network that integrate with existing blockchains and smart contracts. The Crowd App Studio allows decentralized apps to be created that dramatically extend the flexibility and power of blockchain deployed smart contracts.

Figure 14: Crowd Machine is blockchain agnostic.

The following sections outline Crowd Machine support for some of the more popular blockchain networks.

1. *Ethereum*

Crowd Machine will shortly be releasing functionality to facilitate the rapid creation of Smart Contracts without writing code. Smart Contracts are being implemented as a Pattern behavior with contract specific requirements defined using Crowd Machine natural expressions. Crowd Machine natural expressions are then converted to Solidity for compilation. At time of compilation the ABI will be automatically imported, and expressions generated and made available as Crowd Machine natural language expression calls. Crowd Machine will also be hosting a test network for deployment and test of Smart Contracts.

The Crowd Machine approach to Smart Contracts applies an additional level of security by removing the risk of developer coding error in the creation of the contract. Additionally, Crowd Machine will offer the developer the opportunity to have their contract audited prior to its release to the Ethereum network.

Crowd Machine provides extensive integration capabilities including support for RESTful APIs. Access to the Ethereum JSON-RPC methods is a simple process from within the Crowd App Console[10].

2. *Ripple*

Crowd Machine supports the Rippled JSON-RPC APIs for integration into the Ripple XRP Ledger[11].

3. *Other Blockchains*

The integration capabilities within Crowd Machine are extensive. The technology supports integration into numerous databases, RESTful APIs, Web Services and SOAP services. RESTful APIs can also be created to respond to external requests. Basic and OAuth2 authentication is also natively supported for both response and request types. The integration capabilities allow the product to support all external blockchain based services that support the integrations types listed above.

Crowd Machine's blockchain agnostic approach allows an app to interact with any number of blockchains required. A developer may elect to have their app function with blockchains that provide specific capabilities such as authentication, payment transactions or title transfer as examples. There is no limit to the type or number of blockchains that a Crowd Machine app can leverage.

5.3 Crowd Share

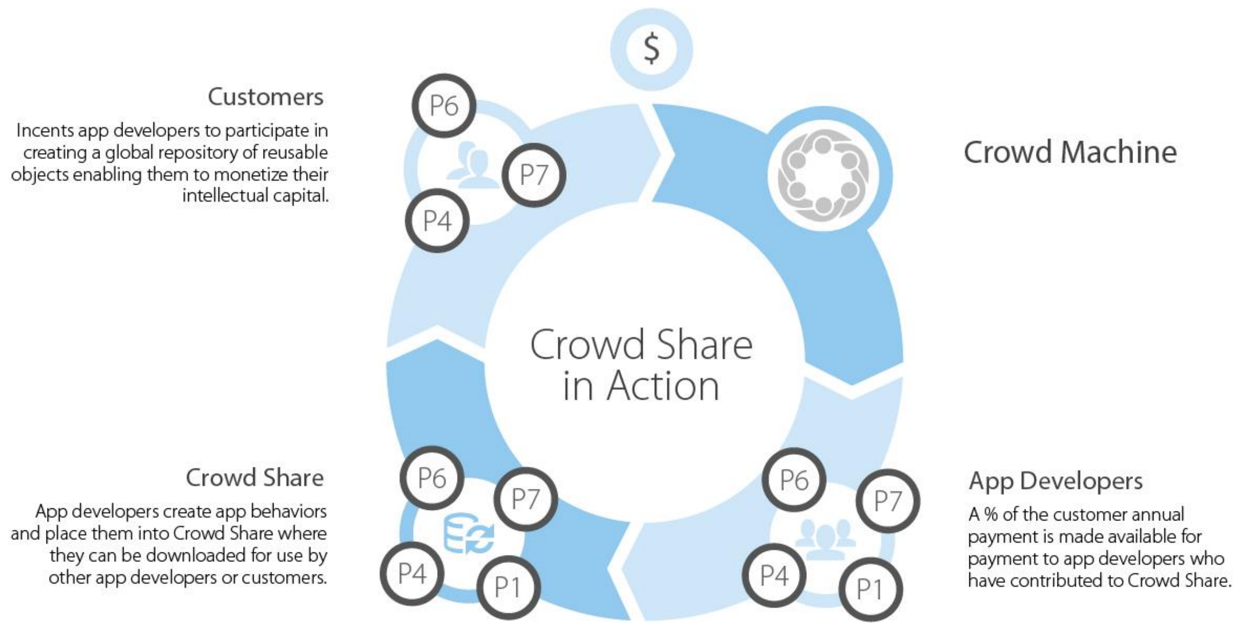
Crowd Share is a GitHub-like source repository enabling the commercialization of source code by the developer community. It allows developers to monetize their intellectual capital in an open marketplace. Crowd Share is a growing repository of ready to use, plug and play Patterns, allowing apps to be assembled rather than created from the ground up. Crowd Share further accelerates app time to market.

5.3.1 The Crowd Share Ecosystem

The Crowd Share ecosystem allows the developer community to share their apps and Patterns, and dramatically accelerates delivery of apps. It is an app marketplace where Crowd Machine content can be downloaded and consumed free of charge by Crowd Machine customers. Crowd Machine sets aside a percentage of the revenue generated by its customers to compensate the developer community for the lifetime of the customer.

[10] For further information on the Ethereum JSON-RPC APIs refer to: https://github.com/ethereum/wiki/wiki/JSON-RPC

[11] For further information on Rippled refer to https://ripple.com/build/rippled-apis/



Customers
Incents app developers to participate in creating a global repository of reusable objects enabling them to monetize their intellectual capital.

Crowd Machine

Crowd Share
App developers create app behaviors and place them into Crowd Share where they can be downloaded for use by other app developers or customers.

App Developers
A % of the customer annual payment is made available for payment to app developers who have contributed to Crowd Share.

Figure 15: Crowd Share process functions illustration.

The Crowd Share compensation model is based on the consumption of Patterns. For example, assume that a Crowd Machine customer is paying $180,000 per annum for their Crowd Machine use, the customer is using 10 Patterns contributed by a developer, and those 10 Patterns represent 10% of the customer's consumption. The compensation to that developer would be as follows –

Total of customer contract	$180,000 per annum
Crowd Share pool	15%
Crowd Share value	$27,000
Developer’s compensation	$2,700 per annum

If 50 customers were using those same Patterns at the same consumption rate, then the developer would be earning $135,000 per annum. The Crowd Share model also takes into consideration the complexity of the Pattern, and allows for derivative works where compensation is distributed across all parties who have participated in those works. Each upload and download of a behavior from Crowd Share is written to a blockchain which provides a mechanism to trace derivative works and identifies the payment pathway. The technology removes any dispute as to content ownership and rights.

5.3.2 The Crowd Share Federation

Crowd Machine will be extending its Crowd Share program to develop a federation of vendors who can participate in the program. Vendors will gain access to the **Vendor Market Portal** where community members and customers building apps can search for and consume vendor products. It will provide an opportunity for vendors to monetize their offerings to a large community of potential customers. The Vendor Market Portal will be provided for free to vendors wishing to participate.

The Vendor Market Portal will also offer automated billing services to federation participants. Where a vendor's product is consumed, the customer will be billed in Crowd Machine Compute Tokens ("**CMCTs**") which will be converted into FIAT currency for payment to the vendor.

The Crowd Share program has been established to promote Crowd Machine's adoption by both Crowd App Studio and Network participants. It is a crowd app development program that establishes a market where developers' intellectual capital can be rewarded, which creates an ever-growing repository of apps and Patterns for customers to choose from. The Vendor Market Portal will further expand the offerings available to the community and customers using Crowd Machine which in turn will further accelerate adoption of the technology.

06 | The Crowd Machine Business Model

The Crowd Machine business model is relatively simple to understand and execute.

It consists of the following:

1. Crowd Machine will make access to the Crowd App Studio and Crowd Share available to a global community of developers and customers for free;

2. Crowd Machine will allow device owners to download and run its CVM as a member of a peer to peer network (Crowd Computer);

3. Crowd Machine will market its technology to startups and enterprises and promote the use of the Crowd Share repository through the Crowd Machine Community and customers;

4. Crowd Machine only monetizes on the compute resources required to execute an app on the Network. All app development and vendor related services are provided for free. As apps are executed, the following is relevant:

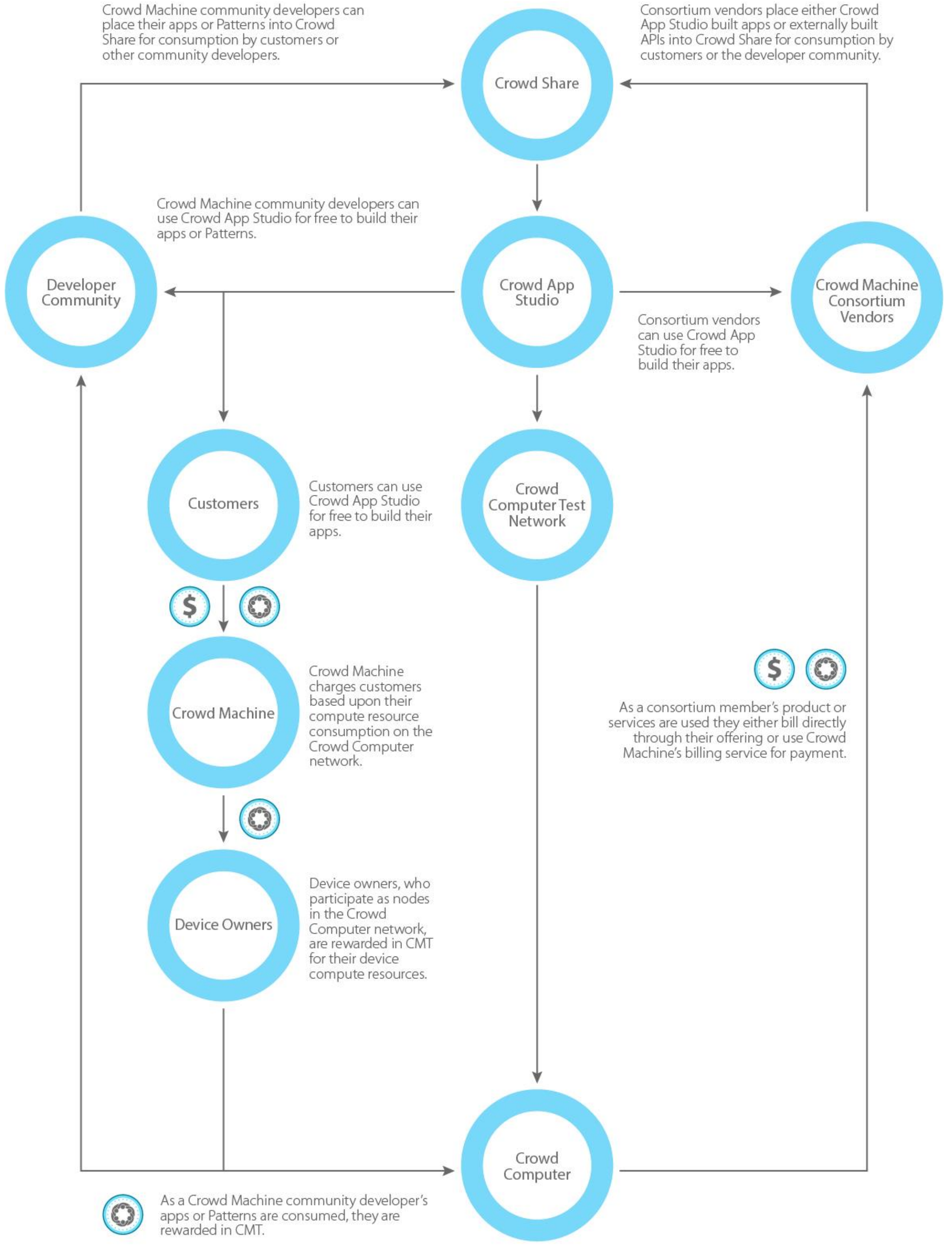
   A. Crowd Machine will be rewarded for use of its Network proportionally to the amount of compute resource required to execute the app.

   B. When a developer's Crowd Share content, utilized by a customer in production, is executed it will result in a reward being given to that developer for that consumption proportionate to the percentage of compute resources it takes to execute that content; and

   C. Crowd Computer Agoric nodes and Activity contributing nodes will be rewarded based on the work they do in routing Activity requests and outputting the relevant computation back to the originator of the request. That is, as app Patterns are executed on the CVM, the device owner will be compensated for that execution proportionate to the percentage of compute resources it takes to execute that content.

Broad adoption of the technology will be accelerated by the creation of a large library of Crowd Share content. To accelerate that process, Crowd Machine will invest in establishing a global community of Crowd Share contributors. This process will consist of the creation of training materials and establishing an accreditation process to accelerate the understanding and adoption of the technology.

06 | The Crowd Machine Business Model



Crowd Machine community developers can place their apps or Patterns into Crowd Share for consumption by customers or other community developers.

Consortium vendors place either Crowd App Studio built apps or externally built APIs into Crowd Share for consumption by customers or the developer community.

Crowd Machine community developers can use Crowd App Studio for free to build their apps or Patterns.

Consortium vendors can use Crowd App Studio for free to build their apps.

Customers can use Crowd App Studio for free to build their apps.

Crowd Machine charges customers based upon their compute resource consumption on the Crowd Computer network.

As a consortium member's product or services are used they either bill directly through their offering or use Crowd Machine's billing service for payment.

Device owners, who participate as nodes in the Crowd Computer network, are rewarded in CMT for their device compute resources.

As a Crowd Machine community developer's apps or Patterns are consumed, they are rewarded in CMT.

Figure 16: Crowd Machine Business Model.

06 | The Crowd Machine Business Model

The Crowd App Studio and Crowd Share are free for community developers, vendors and customers to use. Community developers who place Patterns or apps into Crowd Share that are consumed by customers, are rewarded in CMCT each time that content is executed on the Crowd Computer. Device owners participating on the Crowd Computer are rewarded in CMCT each time their device's compute resources are utilized to run app content. Vendors are paid in CMCT or FIAT currency when their content is consumed either directly via their product offering or through the Crowd Machine billing service. Crowd Machine customers pay for use of the Network only.

As previously highlighted, Crowd Machine Community developers will be incentivized to participate by being rewarded for the use of their contributions by Crowd Machine customers. Additionally, the vendor federation program will make available a suite of app extensions that will expand the offering of apps available to the market and further accelerate adoption of the Crowd Machine product.

07 | RoadMap

Crowd Machine has already demonstrated its prowess with the early adoption of its product by Fortune 500 companies (via Metavine). The first version of the Crowd App Studio and Crowd Virtual Machine were released to market in 2017.

The following timeline is provided as a guide only and is subject to change:

Jan 2018	• Release Crowd App Studio, Crowd Share and Crowd Virtual Machine to public beta • Commence developer training programs and release of the Crowd Academy certification program
Apr 2018	• Begin public CMCT token offer • The opening date for the offer will be midnight UTC April 1, 2018
Apr 2018	• Close the Crowd App Studio, Crowd Share and Crowd Virtual Machine public beta • Make the technology available for production use
May 2018	• Release Crowd Computer public beta 1
Sep 2018	• End Crowd Computer public beta 1 • Beta feedback will determine whether a second beta will be necessary
Oct 2018	• Release of the Crowd Virtual Machine to open source
May 2019	• End public CMCT token offer • The closing date of the offer will be 20:00 UTC May 22, 2019

07 | RoadMap

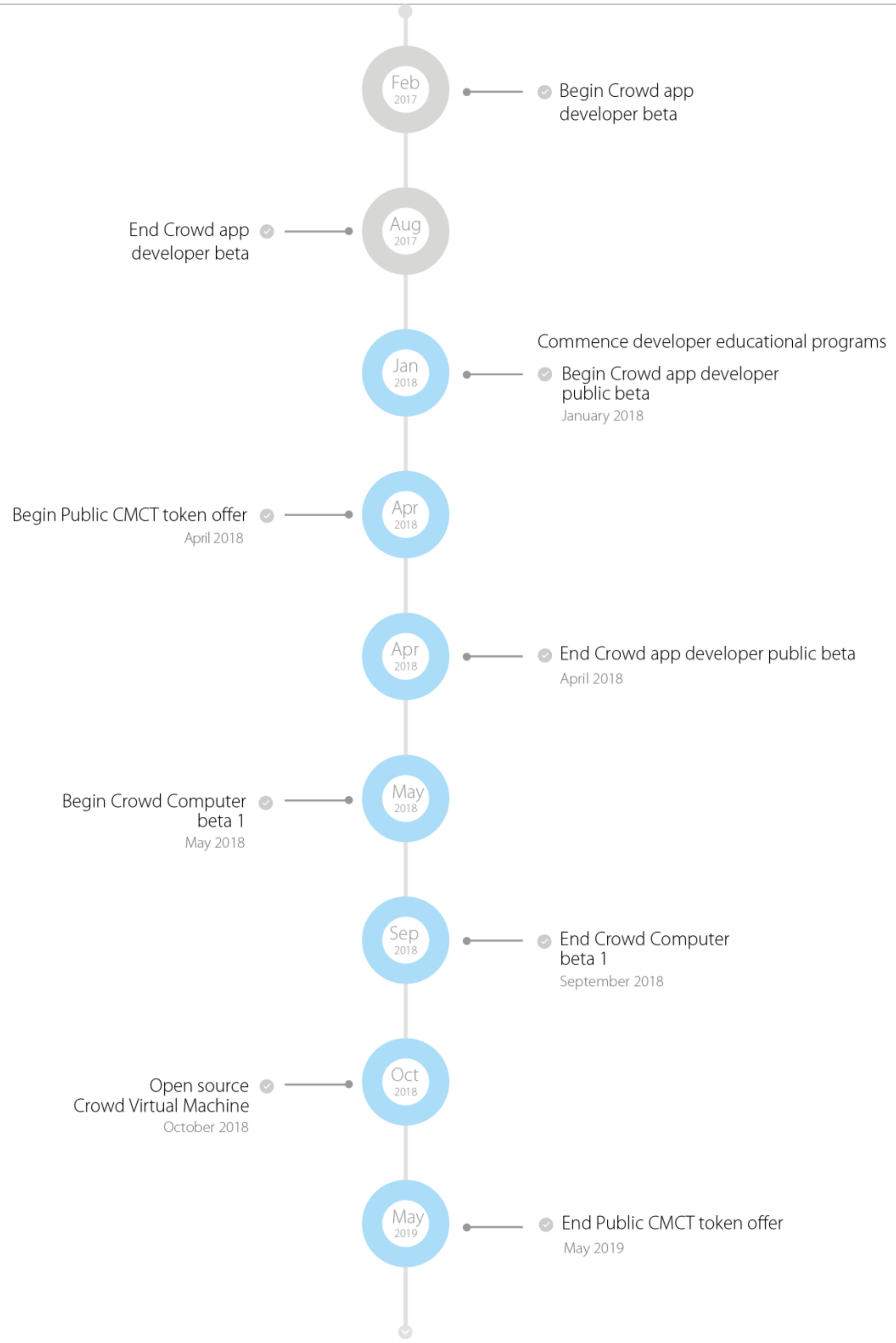

Feb 2017
Begin Crowd app developer beta
Aug 2017
End Crowd app developer beta
Jan 2018
Commence developer educational programs
Begin Crowd app developer public beta
January 2018
Apr 2018
Begin Public CMCT token offer
April 2018
Apr 2018
End Crowd app developer public beta
April 2018
May 2018
Begin Crowd Computer beta 1
May 2018
Sep 2018
End Crowd Computer beta 1
September 2018
Oct 2018
Open source Crowd Virtual Machine
October 2018
May 2019
End Public CMCT token offer
May 2019

Figure 17: Crowd Machine RoadMap.

(takes into consideration previous R&D effort with a product already deployed to market)

08 | Conclusion

Crowd Machine has developed a breakthrough technology that will accelerate the adoption of blockchain and decentralized apps by enabling unparalleled rapid creation, deployment and execution of apps. The technology will begin the process of displacing the existing centralized app and infrastructure markets. Crowd Machine is focused on developing a global community of developers and a network of device owners who will function to execute the decentralized apps of tomorrow. Developers and device owners will gain access to Crowd Machine, and will be paid for their contributions.

The Crowd Computer is the only distributed network technology that enables both the creation and management of decentralized apps. Furthermore, it is a feature rich, powerful solution that can create apps that meet complex, mission critical needs. It provides the ability for startups through to enterprises to create and deploy blockchain solutions without being subjected to the complexities of existing blockchain networks and paves the way for increased blockchain adoption.

Crowd Machine is already in market with many Fortune 500 companies who are making significant investments in the technology. It enables them to get apps to market at unparalleled speeds while reducing the dependency upon highly skilled and scarce resources. Furthermore, using the Crowd Computer, infrastructure costs will diminish which will have the effect of further accelerating the adoption of decentralized solutions in the enterprise.

Crowd Machine's Crowd Share and incentive programs will cause the proliferation of a new stable of apps that will set about replacing the monolithic solutions that exist today. Developers will be able to join the community and be paid for their source content contributions. Developers will also be able to draw from an ever-increasing repository of content to build out the apps of tomorrow. Crowd Machine is also creating a federation of vendors who will have the ability to market their products to the Crowd Machine developer community and Crowd Machine customers. These programs will further accelerate the movement of apps from a centralized to a decentralized model.

A | Annexure A - Glossary

ABI means application binary interface.

Activity means a logical block of functionality that acts on one or more sets of data.

Activity Definition means the Packages and Attributes associated with the Activity.

Crowd App Studio means the zero-code app creation environment which includes the console environment through which an app developer can monitor the use of their apps deployed on the Crowd Computer.

Crowd Computer means a decentralized cloud compute network made up of compute devices.

Crowd Machine refers to Crowd Machine SEZC, a Cayman Islands based company, and the products it offers.

Crowd Machine Community means a global community of developers creating apps using the Crowd App Studio, and decentralized compute network participants who contribute to the Crowd Machine.

CMCT means Crowd Machine token.

Crowd Share means a GitHub-like environment where app developers can publish Patterns for use by others. Every time a Crowd Share Pattern is used by another app developer, the originator of the Pattern gets rewarded.

CVM means the Crowd Virtual Machine.

Links are the method by which Activities are joined together in the Crowd App Studio. They also provide a means by which exception conditions can be modelled.

MAST means Merkylized Abstract Syntax Tree.

Network means the Crowd Machine network.

Package means an intelligent container that holds attributes of data.

Pattern means the mapping of specific behaviors, composed of a set of Activities.

Vendor Market Portal means the environment through which community members and customers building apps can search for and consume vendor products.

B | Annexure B – Market Introspection

B.1 Is the market big enough?

Crowd Machine is a “cross-over” offering that can simultaneously address multiple markets. Two markets of primary concern are the “Public Cloud Service” market and the “Enterprise Software” market. The worldwide public cloud services market is currently valued at $246B in 2017 and the Enterprise Software Sector Market is valued at $392B in 2017. Together, they represent a combined market size of about $638B.

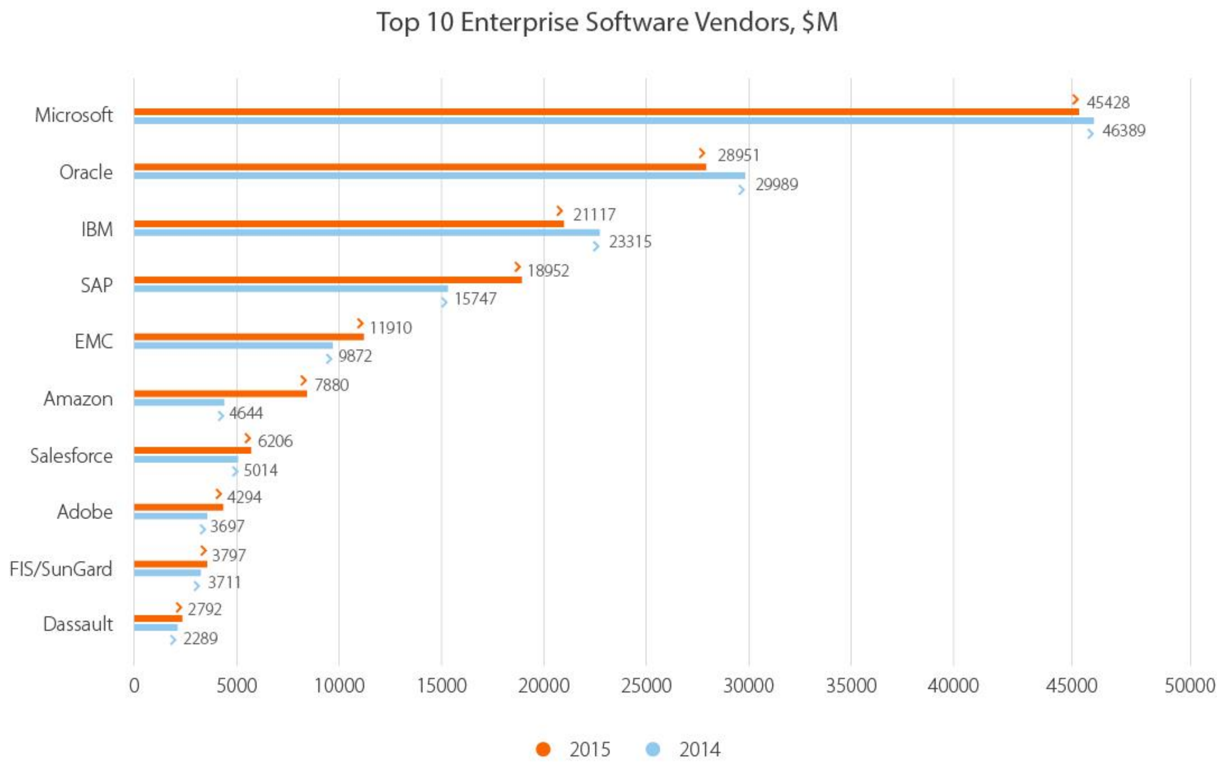


Figure 18: Top 10 Enterprise Software Vendors. *(Source: Apps Run the World, April 2016)*

B.2 Does the market have enough headroom to allow the business to grow?

Since the financial meltdown of 2008, there has been a grassroots movement for people to learn how to code and becoming an app developer has turned into something of a trend. The U.S. government predicts that the software development workforce will grow 22% over the next ten years, and many in Silicon Valley would argue that such an estimate is quite conservative. However, current development workflow still represents a significant learning hurdle for newcomers. Therefore, as more novice developers join the developer ranks, Crowd Machine’s no-code development platform is very likely to capture a substantial share of the new comers as well as the less productive programmers who are looking for a performance boost. Because Crowd Machine’s current market share is still small, there is still large headroom for Crowd Machine to grow.

B.3 Market Size Data

2017 Total Market Size for Cloud Services $246B[12]:

Cloud Business Process Services (BPaaS)	=	43,772	(17%)
Cloud Application Infrastructure Services (PaaS)	=	8,851	(3.5%)
Cloud Application Services (SaaS)	=	46,331	(18.7%)
Cloud Management and Security Services	=	8,768	(3.5%)
Cloud System Infrastructure Services (IaaS)	=	34,603	(14%)
Cloud Advertising	=	104,516	(42.3%)

Market ($MM)	2016	2017	2018	2019	2020
Cloud Business Process Services (BPaaS)	40,812	43,772	47,556	51,652	56,176
Cloud Application Infrastructure Services (PaaS)	7,169	8,851	10,616	12,580	14,798
Cloud Application Services (SaaS)	38,567	46,331	55,143	64,870	75,734
Cloud Management and Security Services	7,150	8,768	10,427	12,159	14,004
Cloud System Infrastructure Services (IaaS)	25,290	34,603	45,559	57,897	71,552
Cloud Advertising	90,257	104,516	118,520	133,566	151,091
Total Market	**209,244**	**246,841**	**287,820**	**332,723**	**383,355**

[12] "Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017" available at http://www.gartner.com/newsroom/id/3616417

C | Annexure C – Business Modelling Considerations

By using proven Intellectual Property, Crowd Machine automates decentralized app development and leverages blockchain technology to do so. It is positioned to be the most influential driving force in the migration of existing monolithic solutions to a decentralized paradigm.

Moreover, the app market is growing in leaps and bounds. According to TechCrunch:

> *"… the apps market is expected to reach $6.3 tln by 2021. By the end of this year, there will have been a total of 268 bln apps downloaded, with revenue exceeding $77 bln. This is a staggeringly huge market, and with Google and Apple taking over a 30 percent cut of the profits, it's a market ripe for disruption. While many startups are trying to find ways to profit from this enormous market, to date they have been hampered by the Blockchains they build upon. Both Bitcoin and Ethereum are capable of about seven transactions per second, which is clearly not enough capacity to support a transformation of the apps market."*

As these statistics indicate, Crowd Machine stands poised to be a monumental disruptive force in both the app development and cloud markets. With Crowd Share and Crowd Computer, the realization of scalable decentralized apps is a reality and not just a vision.

C.1 Reality of the Cloud

The 3 largest cloud providers can provide this service because of massive infrastructure. What is called the cloud is essentially siloed, centralized data storage and processing facilities - a "cloud in a bottle" so to speak. This limits their deployment and the feasibility of reaching markets that would not provide a return on investment.

Let us look at the maps of AWS, Azure, and Google Cloud, respectively:

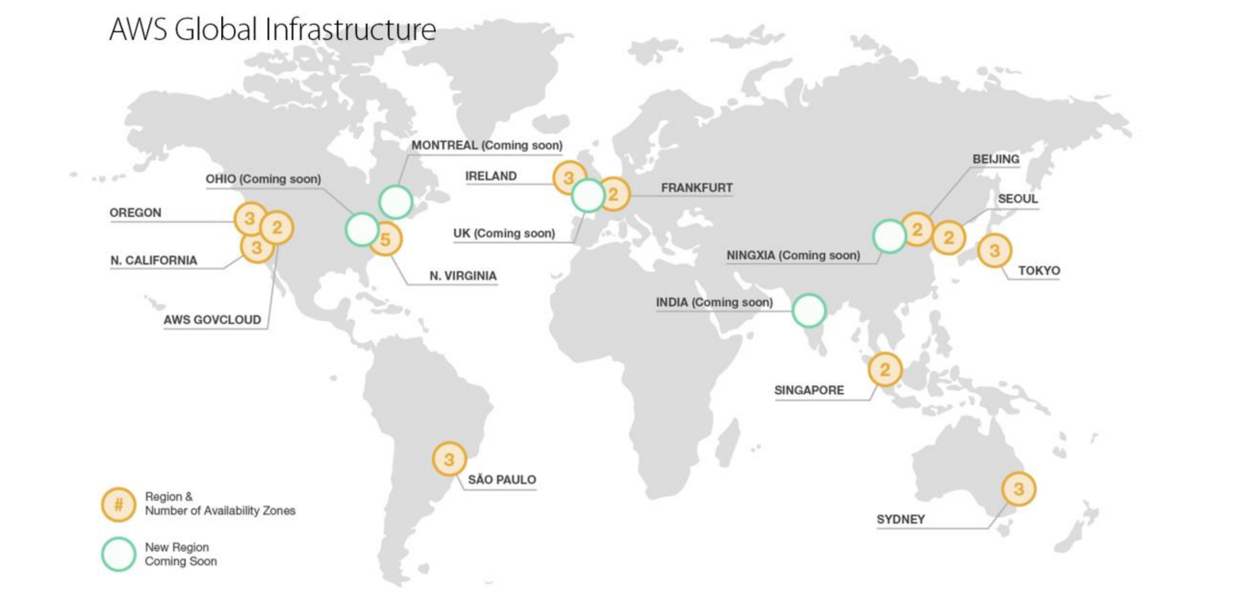


Figure 19: AWS Global Infrastructure. *(Source: Amazon AWS, 2017)*

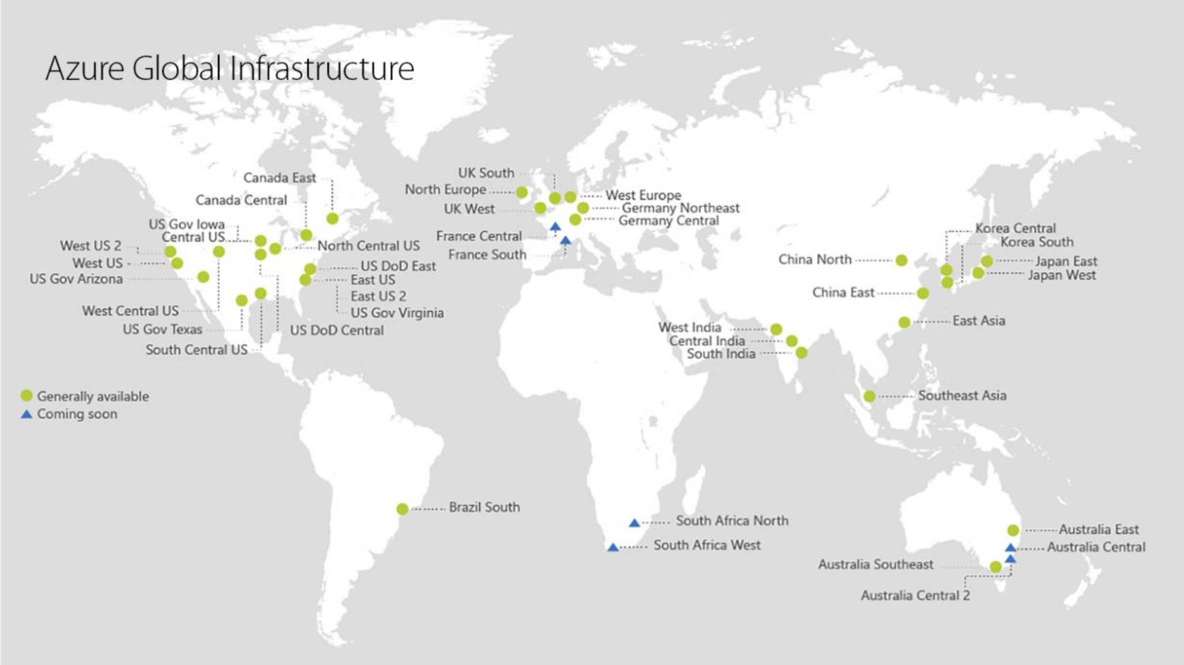


Figure 20: Azure Global Infrastructure. *(Source: Microsoft Azure, 2017)*

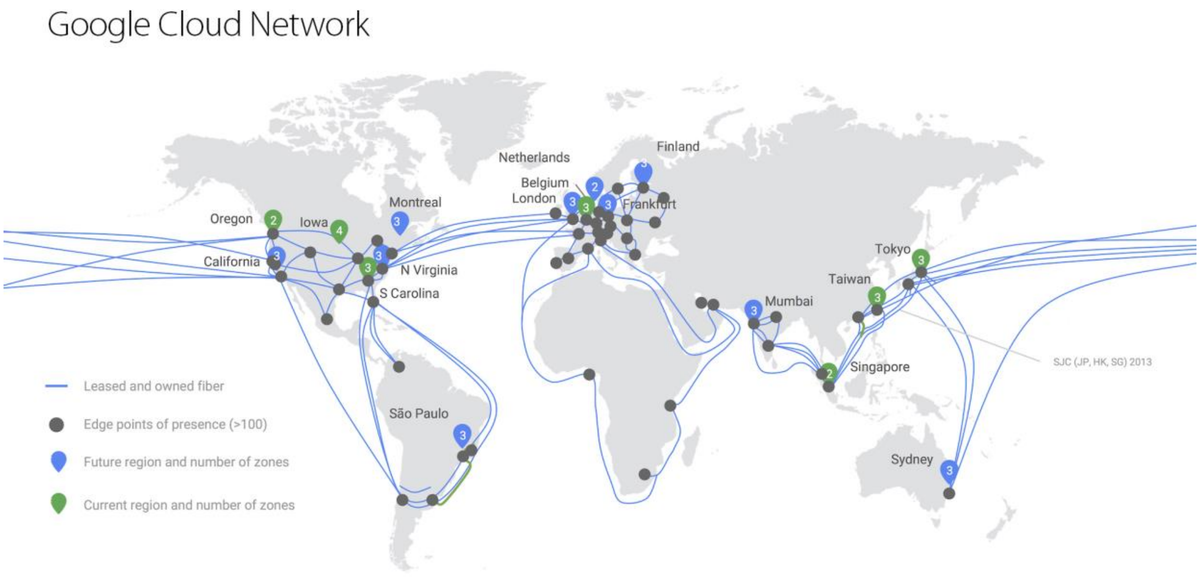


Figure 21: Google Cloud Network. *(Source: Google Cloud, 2017)*

C.2 A New Proposition

The Crowd Computer eliminates the need to have massive infrastructure development be deploying its protocols on everyday devices and coordinating them to complete processing. The Crowd Computer would be distributed, which is massively more efficient and secure. The map below is of the Crowd Computer - anywhere the grid can reach is a potential processing node, where nodes are paid for the processing and transfer of data. It promotes the enhanced integrity of the internet by incenting its proliferation.

Figure 22: Crowd Computer infrastructure.

The figure above shows a rendering of the Crowd Computer infrastructure.

C.3 Business Model and Financial Projections

Metavine (a Crowd Machine related company) currently falls into a subset of the app software market catering to enterprise clientele. The average annual subscription to the Metavine platform is $180,000 per annum. Per the Gartner report, the enterprise software sector market has an expected compound annual growth rate of 8.5%.

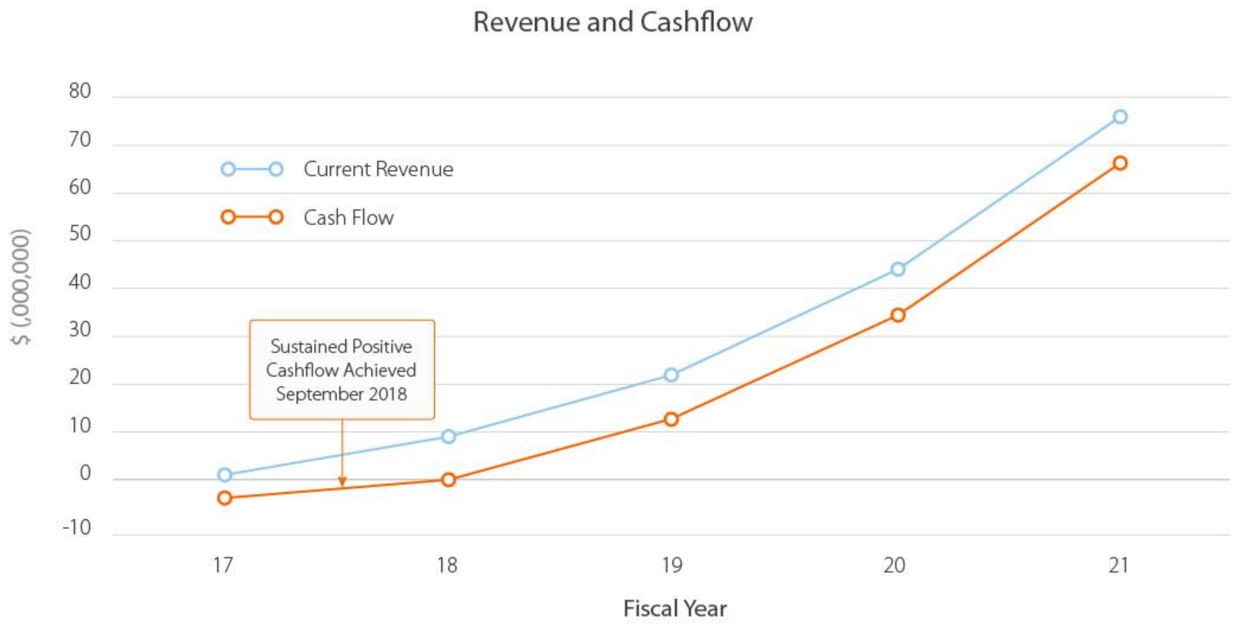


Figure 23: Revenue and Cashflow. *(Source: Derived from Crowd Machine 5Y Financial Projection Data)*

C.4 Customer Acquisition Cost

Crowd Machine estimates robust growth of new accounts in each of its 6 tiers of service. Under the current business model, sales and marketing is focused on acquiring new customers in 6 categories. The categories are defined from a threshold of Crowd Computer compute usage where each category represents a band of compute usage. They have been created to ease the budgeting process of customers and is based on market experience from existing sales. Where a customer's usage exceeds a band, they are migrated to the next band above. Customers also have the option to elect not to pay for a consumption band but pay on a raw compute usage basis only.

Starter:	$60k/year
Department:	$120k/year
Enterprise:	$240k/year
Enterprise Agreements:	$500k-$1MM/year
Channel Sales:	$120k/year
Professional Sales:	$180k/year

Note: fees will be offset in favor of larger volumes using a reward percentage to Crowd Machine.

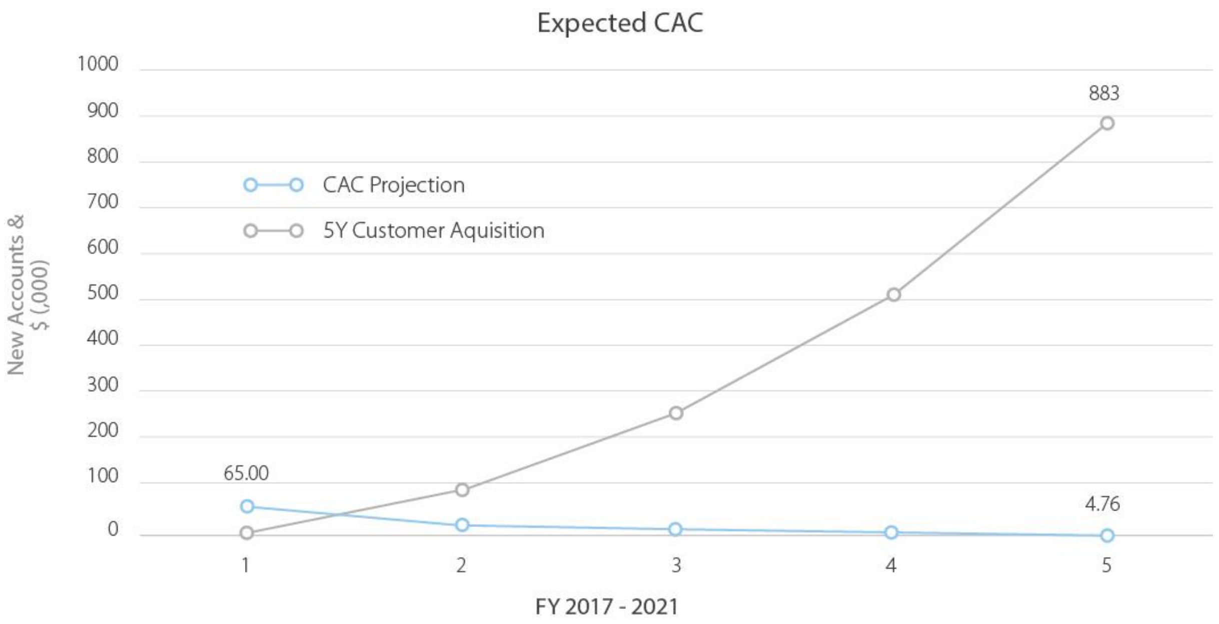


Figure 24: Expected Customer Acquisition Cost.
(Source: Derived from Crowd Machine 5Y Financial Projection Data)

C.5 Hosting Costs

Crowd Machine utilizes Amazon AWS to host its customers' apps. Under the current business model, this cost will vary as revenues increase. As seen below, the creation of Crowd Machine hosting becomes a revenue item.

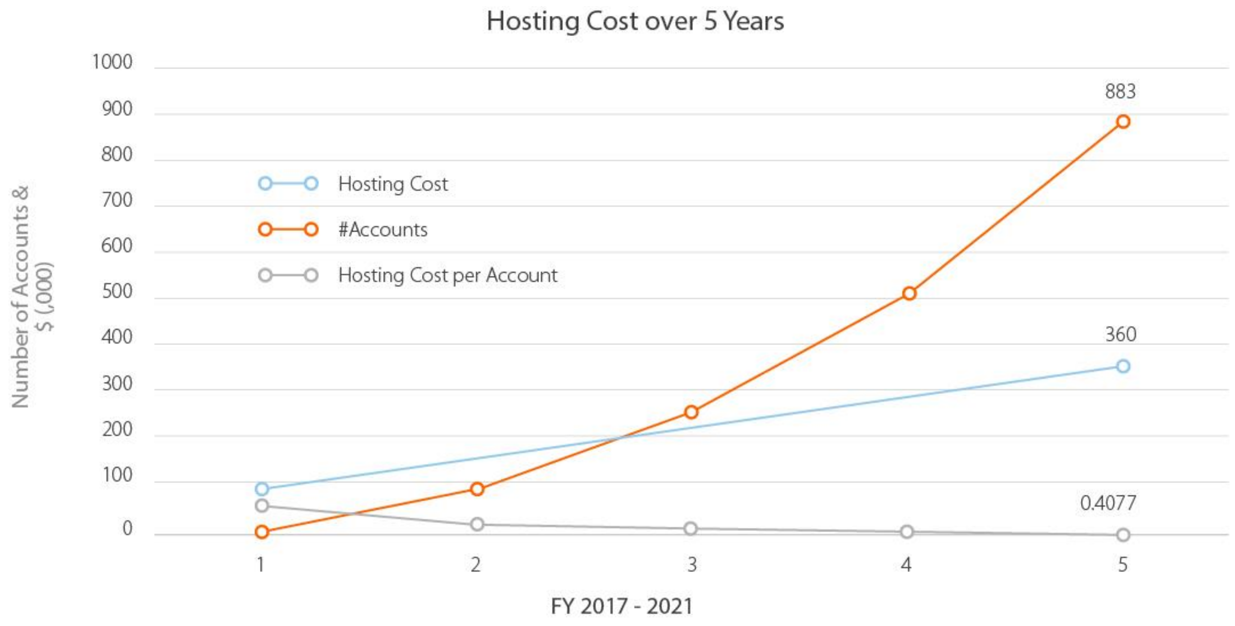


Figure 25: Hosting Costs over 5 Years. *(Source: Derived from Crowd Machine 5Y Financial Projection Data)*