A Vision for a Dignified World Computer Digital Economy

The World Computer Organization
www.WorldComputer.org

July 2020

Table of Contents

1.0 In Pursuit of a New Decentralized Digital Ecosystem for All

The Bitcoin whitepaper and the idea of Blockchain that informed it, introduced an entirely new way in which people can transact with one another without requiring a trusted 3rd party authority or a central server. The promise of this technology is the eventual disruption of virtually all traditional "middle-man" industries. Allowing two parties to agree securely on a transaction without having to trust an intermediary is a paradigm shifting idea. *Blockchain* and other *Distributed Ledger Technologies* (DLTs) promise to allow this, and many well-funded technology startups are racing to be the DLT of choice for any number of industries.

The problem with these initiatives, however, is that they struggle with how to *control* and thus *profit* from a platform that is designed specifically to be open. The technology is specifically designed to operate with no central trusted party, but their very platforms are facilitated by otherwise standard profit seeking SaaS-based businesses. This creates a natural impedance mismatch between the openness they are trying to sell, and the goal of selling it at all. The situation is foundationally untenable if a truly open platform with no trusted 3rd party is in fact the real goal of the system. Hence, these ventures typically degrade from lofty open intentions to *permissioned* systems, controlled by some form of central human authority.

UnoSys (for more information see the World Computer Project accompanying document "***UnoSys – An Overview of the Operating System Design for the World Computer***", referred to as the *UnoSys paper* for brevity going forward in this document), as the open, *permissionless*, P2P operating system of the World Computer[1] must be different. It cannot be controlled by a single company, or even a consortium of companies. Rather, UnoSys, and the World Computer Project as a whole, must be run and controlled by the *commons* – the very members of the network contributing their spare compute, bandwidth and storage resources to it. And yet, there are many aspects of UnoSys that must be decided upon – everything from who chooses a node's identity, to which nodes run in which roles, to who determines a node has misbehaved and needs to be quarantined, to name just three. This creates the obvious problem of who is to be trusted to make these decisions?

In the absence of not being able to trust anyone, the UnoSys distributed operating system is designed to make these critical decisions for itself. The goal is to arrive at a distributed P2P system that *autonomously* manages its own execution within the context of a preprogrammed deterministic set of rules, without the need to trust humans. Therefore, the World Computer Project seeks to replace *trusted 3rd parties* with a single *trusted 1st party* – in the form of the UnoSys operating system itself.

The key engineering and governance challenge is how to create and manage a massively scalable P2P system that achieves *global* awareness from otherwise distributed parts. That is, how to create a decentralized, autonomous *trusted whole* that is greater than the sum of its *untrusted peer parts*, such that it is capable of making comprehensive self-governing decisions about the system in its entirety. It turns out that in building such a system an entirely new digital ecosystem can emerge with a provably

[1] Note: Throughout this document the terms *World Computer Project* and *UnoSys* are used more or less interchangeably since that former is an abstract concept made concrete by the latter.

trustworthy social contract that puts human dignity above all else, directly addressing the inequality that permeates our existing digital economy.

The following document provides a glimpse into how a trusted, decentralized, autonomous World Computer forged from a community of untrusted distributed peers, will give rise to a new self-sustainable ecosystem that represents a significant new opportunity for anyone and everyone to profit in a next generation, inclusive digital economy.

1.1 World Computer Governance

At the core of the World Computer Project is the desire to jettison any notion of a "network" from the vocabulary of its application developers and end users, and expose only a simplified single world-computing model to all users of the system. This means the abolishment of the concept of *servers*, and there single point of control. As an Internet-scale P2P system, UnoSys lives everywhere and nowhere simultaneous – everywhere because it runs across all computers provided by the commons, and nowhere because no single entity controls it. In this way of thinking The World Computer is best describe not as *cloud* computing, but as *atmosphere* computing. However, in this setting, traditional norms around centralized governance break down. This section discusses how UnoSys will be managed and maintained by the commons in a way that is provably fair, trustworthy and yet fully decentralized.

1.1.1 Accountability

Bitcoin is an anonymous P2P cryptocurrency network allowing users to securely transact with one another, without either party, or the public using the system, knowing who the parties are. This is a deliberate aspect of the design of the network, as privacy of the participants from regulating governing bodies is a major motivation for its operational behavior. This is simultaneously a powerful characteristic of the Bitcoin network, as well as a major problem since it elicits a number of unintended (yet predictably inevitable) devious uses of the network from community members that hide their malicious behavior behind the veil of privacy the network affords.

In contrast, the UnoSys network defines a strong sense of identity for its peers, its users, and the applications that run on it. Moreover, those peers, users, and applications are held to high degree of accountability for how they use the network. Indeed, it is through accountability that a meaningful reputation is established in UnoSys.

As discussed in the *UnoSys paper*, each UnoSys.Node.exe running in the UnoSys network is uniquely generated. At the core of that uniqueness is the *processor instance's* identity. This node identity is how UnoSys accounts for a peer's reputation – its measure of behavior. Furthermore, a node cannot join the UnoSys network without first being associated with the identity of the *account* that is installing it (i.e., an individual user or an entity representing an entire corporation or business). That is, as with traditional SaaS based software models, a person or organization will have to establish an *account* with UnoSys before being able to install the UnoSys.Node.exe on one of his computers. In this way UnoSys tracks which accounts are accountable for which unique nodes.

Importantly, the UnoSys P2P operating system is *permissionless* but it is not *anonymous*. Permissionless means anyone can create an account and install a UnoSys.Node.exe. In this way, UnoSys is inclusive of anyone wishing to join the network, but they cannot do so anonymously.

An *account* will be required to be created for a person to install the UnoSys operating system on one of his computers. A minimum of information will be required to uniquely identify each account (i.e., a username and email address). The act of registration with UnoSys will require that the person installing the node prove they own and control the email address they provide, by replying to an automated email verification message sent by UnoSys, as is the typical pattern used by most SaaS registration processes. However, what is different about UnoSys is that the database of registered accounts will not be accessible by any single human or any group of humans, including, importantly, the core development team of UnoSys. Rather, the UnoSys account database (and many others as it turns out) will only ever be accessible and managed by UnoSys itself, under *autonomous* control.

1.1.2 Autonomy

In the UnoSys permissionless, distributed P2P UnoSys operating system, there is no equivalent of a traditional “administrator” or “super user” account you would find in other operating systems, precisely because no one person or entity (or group of people) can be trusted to act in that role for the entire system[2]. This creates an interesting challenge since, from a security perspective the operating system nonetheless needs to run code under a well-defined security context commensurate with the sensitivities of the code in question. It is relatively easy to imagine how applications that run on top of the operating system will run in the security context of an end-user account. However, perhaps harder to imagine is how UnoSys runs code that would typically be constrained to the traditional administrator of the system.

The design of UnoSys fully acknowledges and embraces the fact that to be truly independent of any trusted 3rd party influence, it has to be able to execute significant portions of its own code fully autonomously. Similar to the way *smart contracts* are designed to run autonomously on the Ethereum Blockchain network, UnoSys will necessarily have portions of its code that once started, execute entirely unattended. The code will be self-executing and self-enforcing, ensuring that its intent is always fully carried out. Significantly, once running, this code cannot be influenced in any way, including stopped, by humans. In the UnoSys paper we have already seen how the operating system provides a fortress to run tamperproof code securely and reliably on any given UnoSys node. However, it remains to be seen how *autonomous UnoSys code* will run. Specifically, from a security context perspective, what identity will this code run under?

To address this challenge, the process of transitioning the operating system from under development to the “Go Live” version[3] will automatically generate a special ***UnoSysAdmin*** account that no human knows

[2] This is true of the Go-Live version of UnoSys only and does not apply to the Development version. See the accompanying World Computer Project document “***A Roadmap to Socially Engineering the World Computer***” for more details.

[3] See the accompanying World Computer Project document “***A Roadmap to Socially Engineering the World Computer***” for more details.

(or can find out) the credentials for. The credentials of this account will be protected with all of the built-in data security protections UnoSys affords, as discussed in the *UnoSys paper*. This special security account is provided elevated permissions by default and is used to run autonomous processes that only the operating system itself can execute. There will be no way for any human, including any member of the core develop team, to take over or otherwise assume the elevated privileges of this account. In a very real sense, the operating system itself will be its own administrator of the system, able to do things that no other user has permissions to do.

One of the first and most important responsibilities of the UnoSysAdmin account is to have exclusive responsibility over the UnoSys *Account* and *Peer* database tables, in which the identities of all people who have installed peers, and the identities of those peers themselves, reside. Importantly, only the UnoSysAdmin account will have access to and maintain this information. For example, it will be provably impossible for any human being to generate a list of nodes comprising the UnoSys operating system, or to determine a list of accounts that control those nodes. These lists will be available to the UnoSysAdmin account and therefore the software that is permissioned to run under that account, but to no one else. This is markedly different from a typical SaaS scenario where the owners of the business know their customers. In contrast, the accounts and peer identities that make up the fabric of the UnoSys platform will never be available, in whole or in part, to any human being and will instead be 100% maintained and secured by the UnoSys operating system itself. In this way, UnoSys will become its own trusted authority – trusting no one but itself for the most sensitive operations.

Essentially, for the first time, a piece of software, and not a person or business, will have stewardship over the identities of the accounts and nodes contributing to a massive P2P system. This has far reaching implications with respect to how the operating system is designed and functions, and for the new innovation that it will enable. Importantly, the operating system will have to determine for itself which accounts and peers are behaving in its overall best interest and which are not, and then act accordingly. Therefore, an immediate question arises regarding how exactly UnoSys quantifies the behavior of its peers?

1.1.3 Reputation

An end-user will only install and run the UnoSys code if he trusts it to perform its intended function of supplying reliable, secure, operating system services. In a massively distributed P2P system, the user must first and foremost trust the operating system software running on his computer. The *Fortress* security architecture discussed in the UnoSys paper provides a trustworthy environment to execute tamperproof code. The user must also trust the entire platform the software is a part of. In particular, he must trust that UnoSys has fully vetted as trustworthy the neighboring peers automatically chosen for him, and that the operating system acts accordingly if peers are determined to be untrustworthy. UnoSys uses a built-in *reputation* subsystem as the measure of that trust.

In the real world, a person's reputation is important. It has tangible value because it directly correlates to the level of trust one is afforded by others. Reputation often takes a lifetime to build and can be destroyed in only a moment. In this way, it is truly precious and most people will go to great lengths to protect their reputation.

UnoSys uses reputation extensively to determine the value of its peers. Peers that have earned a high degree of reputation are rewarded with more responsibilities. And the peers with more responsibilities have the greatest opportunity to participate in the UnoSys economy and create real-world revenues for their owners. Conversely, peers found not to be operating in the best interest of the Unosys network as a whole, are disciplined for their negative behavior by the operating system. Depending on the severity of the negative behavior, consequences range from reduced reputation scores and/or financial fines, to network participation quarantining, or removal.

The reputation of a peer in the UnoSys operating system network is based on two independently evaluated components – *fair-play* and *capabilities*. The first component – *fair-play* – attempts to measure a peer in a resource-independent way. In contrast, the second *capabilities* component takes the resources of the computer into account such as how much compute, bandwidth and storage capacity it exhibits.

When the UnoSys.Node.exe software is first installed on a computer, the peer joins the network in a probationary context. During this probationary period, the peer is limited to only acting as a client for UnoSys requests from applications. Specifically, it is not permitted to provide server-role capabilities such as storing end-user data persistently. For a period of up to several months, UnoSys will monitor the peer in order to establish both its fair-play and capabilities reputation scores. Eventually, autonomous agents within UnoSys will make a deterministic determination based entirely on reputation data collected during this probationary period, whether to authorize the peer for more responsibility within the network.

The fair-play measure is equivalent to a typical business' *equal employment opportunity* policy in that UnoSys attempts to first assess the peer to see if it is a "cultural" fit for the UnoSys network.   The assessment does not consider the specific competences of the peer, but instead attempts only to determine how likely it is to "get a long" with the other peers in the network. For example, given two computers – one a student's laptop and the other a supercomputer – UnoSys makes a fair-play reputation evaluation of how "nicely" the computers participated in the network during their probationary period based entirely on non-discriminatory baseline criteria it requires that all peers of the network exhibit:

- Good peer reachability (i.e., up time)
- Good IP address reliability
- Good response times

If the peers earn a suitably high fair-play reputation score, they are effectively accepted into the network as a "node in reserve".   The UnoSys operating system will then use data collected about their individual capabilities during the probationary period to subsequently determine the best way to utilize them within the network[4]. Similar to how businesses employ many people with broad skillsets, the

[4] At this point the peers have been classified for a specific set of roles and may or may not be immediately used by UnoSys. It may be the case that the network has all of the resources it needs at the moment in which case these

breadth of functionality the UnoSys operating system offers means there are roles for peers of every capability level – from student laptop to supercomputer. Significantly, however, it is the earned fair-play reputation of each peer that determines its right to participate in the UnoSys network at all.

The reputation of an individual registered user *account* is a different matter and is monitored separately by UnoSys from that of the individual peers. The reputation of an account is largely defined by the cumulative reputation assessed for all of the peers associated with that account. It is one matter for the UnoSys operating system to autonomously discipline a misbehaving peer by automatically removing it from the network. However, if enough such peers associated with a single registered account are disciplined, UnoSys may decide that, for the safety of the entire network, the account itself needs to be removed from the network, along with any other associated peers that may otherwise still be deemed to be behaving properly.

UnoSys arrives at disciplinary actions impartially and autonomously, based on coded and vetted business rules that no person or group can unilaterally change. These rules are transparent to the community, and every account holder must understand and agree to them before installing her first UnoSys.Node.exe peer instance.

Similar to the way parents are responsible for the behavior of their minor children, UnoSys sees to it that account holders are accountable for the collective behavior of all of the peers associated with them. This encourages vigilance on the account holders' part that appropriate peers are chosen and maintained at all times, while providing a suitably harsh deterrent (e.g., in the worst case, lifetime banishment form the UnoSys network) for behavior that is found to be negligent or malicious. UnoSys bases its decisions fairly on transparent and irrefutable telemetry data collected regularly from every peer in the network. Therefore, all autonomous reputation-based decisions the operating system makes can be suitably justified.

Reputation continues to accumulate for peers and accounts for the lifetime of their participation in the UnoSys network. Appropriately, every peer and every account must also constantly justify that they deserve to remain in the UnoSys network by being accountable for their behavior at all times. As long as they play fair according to the established rules, the UnoSys operating system will allow them to participate. And because of the heterogeneity of the computers that will participate in the network, there will be diverse roles that peers at every capability level can fill. As such, UnoSys will be an inclusive network where every well behaved computer, no matter how small, will be able to participate in the UnoSys network in some way.

UnoSys is designed to provide a comprehensive autonomous reputation system governing all peers and their account owners. However, it would be naïve to think that Artificial Intelligence (AI) technology has progressed to the point where an entirely autonomous distributed operating system can be developed that is capable of self-executing, self-policing and self-sustaining itself. Indeed, perceived general AI capabilities today are significantly overstated and the truth is that human beings will play a central role

peers will remain in a "reserve" status until such time as they are called into action by the operating system on an as needed basis.

in bringing the UnoSys vision to fruition and maintaining it. Therefore, the question remains – how will humans influence and inform the otherwise autonomous operation and maintenance of the UnoSys operating system?

1.1.4 Human Oversight

UnoSys is not an open source project. This fact will initially disturb a great many people because of the traditional association between open source and trust. Intuitively, many people do not trust mission critical software they cannot download, read and build for themselves. Of course, only a very small fraction of people actually read the source code to large open source projects and even fewer actually confirm the source code they are reading is really the software they are running, by downloading and recompiling the code themselves. But it's the perception that counts – that it is more safe and/or secure to run software that *someone else* has download, read and built the source code for. Lack of transparency breeds distrust.

Still, UnoSys cannot be an open source project because the unique security context of a permissionless, P2P, distributed operating system project cannot allow its community members to download and re-compile the source code to the operating system. To allow this would inevitably open the door to tampered versions of the software entering the network, which is strictly not permitted by the UnoSys fortress security model the project is based on, as described in detail in the *UnoSys paper*.

However, UnoSys will be an *open review* project. That is, anybody will be able to review the entire source code to the operating system whenever they wish using a special tool installed with every UnoSys node. All UnoSys operating system source code will be published and stored in an internal database that, like every other UnoSys database, is stored securely and distributed across its peer members. This database will only be accessible via the source code review tool mentioned above which will execute through delegation under the elevated UnoSysAdmin operating system account mentioned earlier with read-only permissions.

In this way, the source code to the operating system will become part of and managed by the operating system itself. The review tool will permit anyone with interest to view/search/hyperlink around the *current* source code, or any of its previous versions. However, importantly, no official project *build system* will be provided for UnoSys. This will make it impossible for anyone to recreate official UnoSys.Kernel.dll binary code that can be loaded into specific peer UnoSys.Node.exe execution instances, as described earlier[5].

Furthermore, a UnoSys processor *manufacturing* tool will also be provided to make it possible to generate new UnoSys.Processor installation packages on demand *directly from* the current source code

[5] The UnoSys.Processor installation package manufacturing process not only generates a unique version of each node, but also produces the cryptography keys to protect that node and the data it serves. Because they are dynamically generated as needed, these keys are not part of the UnoSys source code proper, and therefore will not be available for review.

database[6]. This will go a long way to giving confidence to the vast majority of end-users that they are running the actual *current* UnoSys software. For example, they will be able to go to a friend or colleague that they know and trust and who is running and endorsing the UnoSys platform, and ask them to generate a unique new UnoSys installation package for them.

It is expected that end-users and critics will eventually accept the decision to open review the UnoSys project source code rather than open source it, as the best choice for the community as a whole, once they fully understand and appreciate the unique security context a permissionless P2P distributed operating system like UnoSys faces. UnoSys will provide full transparency to its code base for peer review without compromising the significant technological investment it has made to ensure the operating system code runs at all times in a secure, tamperproof way.

1.1.5 Versioning

The UnoSys operating system is a living breathing entity which starts with its source code. It is under constant active development. Moreover, it is expected to continue to undergo changes as bugs are discovered that need fixing, and new features are incorporated. The obvious governance question becomes, who authorizes these changes?

As with most large public (traditionally open source) software projects, there is expected to be a core set of UnoSys developers who will contribute to the project on a regular if not fulltime (i.e., it is their job) basis. The World Computer Project accompany document "***A Roadmap to Socially Engineering the World Computer***" describes the three broad phases the project is expected to transition through:

- Incubation
- Growth
- Maturation

All source code governance during the *Incubation* phase is expected to be decided directly by the core development team. This is necessary to fulfill the initial stated vision of version 1.0 of the software with the least amount of bureaucracy. However, entering the *Growth* phase will be signaled by a formal go-live cutover process, after which the UnoSys software *change manage process* will be autonomously controlled by the operating system itself, based on community oversight.

In the Growth phase, any vetted member of the community (i.e., a user with enough amassed reputation) can make *suggestions* for bug fixes and/or enhancements to any aspect of the operating system. The community will be given an opportunity to vote on the usefulness and priority of these suggestions. The core development team will make the changes to the source code that have been voted for in majority and will then submit them for *community review*. The community will review the

[6] While the source code will come directly from the UnoSys database, it will undergo an automated obfuscation process designed to create a difficult to read, unique low-level code layout in memory without otherwise altering its function. This is possible because the C# language in which the UnoSys operating system code is written supports the rearrangement of classes within a namespace and fields/methods/properties/events within classes, in any order. Hence, each and every UnoSys.Kernel.dll assembly created for each and every execution instance of UnoSys.Node.exe on each and every node, will have a uniquely and radically different layout in memory.

software changes and vote on whether to accept them as made. Accepted changes will then be pushed first to a development environment for regression testing and then, assuming it passes these tests, onto a staging environment for further testing, and then, if accepted, finally to production.   Importantly, at all times, advancement through this formal change management process will be controlled *autonomously* by the UnoSys operating system itself.

Special secure voting tools will be built and incorporated directly into the operating system itself, ensuring that the various source code change management process milestones cannot be bypassed without a majority community vote. That is, the operating system itself will police the automated change management process and only allow the publication of changes to the software when a majority of the voting community agrees. In this manner, there will always only be one (and only one) current version of the software running (i.e., it will be impossible for the software *forks* that plague the management process of other open software projects to occur), and that version will have been majority agreed upon by the community as a whole.

In the event of emergencies (i.e., committed changes that cause critical software breakages or introduce unintended security holes, etc.), it will be possible to roll back software to previous versions, again via a formal UnoSys controlled change management process based on community majority vote. Thus, there need not be fears anytime soon for the possibility of a *singularity* occurrence whereby UnoSys acquires a science-fiction Artificial General Intelligence (AGI) and develops a motivated conscience with its own goals to take over entire control of the system from the governing human oversight of the community[7]. The operation of the UnoSys operating system software will be driven and controlled by the community at all times, but always in a transparent and provably democratic manner where every change is confirmed via majority community consensus by a secure/tamperproof voting platform that only UnoSys itself controls.

Significantly, it is a design goal of UnoSys that once version 1.0 of the operating system reaches the go-live state, security measures built directly into it will ensure that the core development team no longer has the ability to make unilateral changes to any aspect of the UnoSys software. Instead, they will be required to follow the same change management process that any other member of the community must, and submit their change requests for community review and subsequent majority vote approval.

Community vote notifications will be suitably distributed directly by the operating system to the member accounts that have reached a required level of reputation and that have expressed interest in being involved in such votes.   A suitable time period, commiserate with the urgency of the change request, will be allotted in which the votes/comments will be collected from the community directly by the operating system and counted, after which the operating system itself will decide the consensus outcome (e.g., to publish the change or roll it back) within its built-in secure/tamperproof/fault tolerant distributed consensus voting system.   In this manner, governance around the operation of the UnoSys operating system is a unique partnership between the human community that proposes, makes, and

[7] It is envisioned, however, that eventually, a deep machine learning AI platform will be built directly into UnoSys as a general purpose subsystem, integrating seamlessly with its already built-in streaming file and database I/O subsystems in order to make large scale, distributed, big-data AI application development ubiquitous and easy.

tests the changes, and the UnoSys operating system software that autonomously oversees the change management process and deployment of the change in a provably fair and egalitarian way.

By methodically combining human oversight with autonomous software execution, UnoSys will achieve the vision of a planetary scale *Computer* open for all to use, controlled by no one single person or entity, and yet able to grow and mature with the needs of the community it serves.

1.2 Towards Digital Economy Realignment

Markets form around trade. One party gives something of value to another party, in exchange for something of value from that party. The two parties negotiate a transaction and execute it, and historically this happened in person, across a table at the local public bazaar.

Over the Internet, however, the digital economy is often convoluted and difficult to understand. There is an entire digital sub-economy engaging in business not for the purposes of trading digital or physical assets they have of value with other parties, but rather to own, control and profit off of the equivalent of the public bazaar itself. These so-called *multi-sided marketplace* business models offer "free" tables at the 24/7 global digital bazaars they provide so that vendors from all over the world can setup shop and trade with each other, while they (the purveyors of the marketplace) extract a significant percentage of the transaction value for their trouble.

In more convoluted scenarios, the vendors are not charged to use the marketplace at all. Instead, other 3rd parties are charged to have access to the crowds that congregate within the marketplace. This model, which is premised on restricting access to large audiences, is the model of digital advertising and is the dominant revenue model of the web generating tens of billions of dollars every year.

It has become clear that two decades of these models have created a lopsided, closed digital economy where the largest, well-funded marketplace providers extract enormously disproportionate percentages of the transacted value. These market platforms have been called the perfect business model. They have near zero cost of goods sold and the users bring all of the content. They are the quintessential "middle-man" businesses, providing no tradable goods of their own, but rather being paid to merely facilitate the trade of other people's goods through their platforms. Moreover, the vast majority of *free* service providers control and then sell *access* to the massive audiences they garner to 3rd parties who are themselves only interested in selling to the audiences.

The concerning issue is that as these marketplaces have become bigger and bigger through organic growth and mergers and acquisitions, they quash their competition and dominate in monopolistic ways. The result is too few businesses control the majority of what happens digitally on today's Internet. Hence, an overwhelming disproportionate percent of the digital economy goes to only a handful of the largest marketplace providers. There are growing numbers of people who believe this situation is not healthy for the greater commons and is sorely in need of realignment.

UnoSys will disrupt the status quo of the current digital economy by making the bazaar public once more, facilitating trade between interested parties without charging a transaction fee, or prostituting them to 3rd party advertisers. As a result, more transactions will take place, reducing the friction of doing business on the Internet, while keeping more value in the pockets of the trading parties. Moreover, by abolishing transaction fees, smaller transactions (i.e., micro payments) become practical for the first time at scale, allowing for long promised aspects of the digital economy such as the Internet-of-Things (IoT) to finally start flourishing.

UnoSys will provide the public, free marketplaces, and will amass great crowds that will inevitably attract billions in advertising dollars. But instead of keeping advertising revenue for itself, UnoSys will pay it out directly to the users consuming the advertising, if, as, and when those users choose to. Such an approach returns all of the value the "middle-men" otherwise extract from the market to the very commons responsible for both the platform and the commerce that goes on within it.

This section describes how UnoSys as a, permissionless, Internet-scale P2P operating system, run by the commons will usher in an entirely new, self-sustainable digital economy, fair for all to participate and prosper in.

1.2.1 Incentives

To this point in the document, there has been no discussion about what would motivate a typical person to join and participate in the UnoSys P2P distributed network. It turns out that the World Computer that UnoSys will bring about will attract people of every age and walk of life to take part and contribute. UnoSys will create a transformative digital *sharing economy* that will provide great value to everyone in the network in an inclusive way, by offering *every* user multiple concrete incentives to participate.

The tragedy of the commons is a well know phenomenon where individual users acting independently according to their own self-interest behave contrary to the common good of all users by depleting or spoiling a shared-resource through their collective action. In P2P systems, this manifests itself in the so-called "free riders" problem where people use a system's shared resources without contributing resources to it themselves. The goal of any successful P2P system is to have all peers contributing to the system maximally. Only then will the network effect benefits of the system be fully realized.

To create the distributed operating system that will power the next generation of truly innovative Internet applications, UnoSys needs the community to share their computer's spare CPU, bandwidth and storage resources as an active member of the network. UnoSys incentivizes this active participation by providing contributors the opportunity to receive an ongoing payment for this participation. Users that share their spare computer resources with the community are called *providers.* Providers are paid real monetary value directly by UnoSys whenever they serve up requested end-user data.

However, peers cannot become providers immediately after joining the UnoSys network. As previously mentioned, UnoSys requires a vetting period during which the peers must build their reputation as reliable nodes. During this time, no resources of the peer are contributed to or used by the overall network, and the peer is given consumer (i.e.; client) only capabilities. That is, they can run (consume) the applications running on UnoSys available to them. However, as we will see, consumers also have real opportunities to earn payments as well just from using the system. Therefore, UnoSys will grow rapidly because *any* user has the opportunity to financially benefit by being a member of the community.

1.2.1.1 Joining the UnoSys Network

A *user* begins his relationship with UnoSys by obtaining a *unique*, automatically generated, UnoSys.Node.exe software installation package from somewhere[8] and executes it on one of his computers. The first thing the installation process will do is require the user to register an account with UnoSys or if they have already done so, login to that existing account[9]. Once the account's identity has been determined, the UnoSys.Node.exe installation process can complete with the installed peer being associated with the account. At that moment, the peer will begin operating with *consumer* only responsibilities during a probationary period designed to evaluate the peer to see if it would be well suited for more responsibilities within the UnoSys network.

As previously noted, UnoSys carefully collects telemetry data about the node over a protracted period of time in order to compute its *fair-play* and *capabilities* reputation scores. After the probationary period, if the peer has acquired the minimum required fair-play reputation, an offer will be made by UnoSys to the account holder of the peer to participate in the network with *provider* responsibilities, and be eligible to earn provider payments from the network. If the account holder accepts provider responsibilities, then he will automatically be paid when pay-for-use end-user content is written to and later accessed from his peer.[10]

1.2.1.2 UnoSys Resource Accounting

Similar to big tech *Cloud* platforms, the UnoSys operating system requires the use of three main types of resources from the computers in its network:

- Compute
- Bandwidth
- Storage

However, in contrast to those Cloud platforms, UnoSys takes a very different approach to billing for the use of those resources. Of the above three resources, the only one that UnoSys will charge a consumer to consume, and compensate a provider to provide, is *storage*.

UnoSys views an Internet connected computer as a commodity in this day and age. The vast majority of the people in the developed world have a least one Internet connected computing device capable of

[8] For more information on how the World Computer will bootstrap itself and how a user would find out-of-band UnoSys.Node.exe installation packages, see The World Computer Project accompany document "***A Roadmap to Socially Engineering the World Computer***".

[9] The UnoSys.Node.exe installation process will offer the user the register/login choice directly. That is, this is not available from a centralized website since UnoSys is fully decentralized.

[10] The act of accepting provider responsibilities informs UnoSys that the node is ready to start storing and serving end user data. However, UnoSys may not have need of the node right at the moment. Instead the node will be kept in reserve to be later dynamically incorporated into storage clusters on an as needed basis. As a result, depending on the needs of the network, the node may *not* begin receiving payments immediately.

running UnoSys. And since *compute* and *bandwidth* are the bare minimum required to achieve an *Internet connected* device, UnoSys deems having those capabilities as the price of admission to participate in the UnoSys network, as either a consumer or provider. However, it views storage differently.

The world generates data at an amazing exponential rate. Over 90% of the data that exists in the world today was generated in the last two years alone. We are generating data at the rate of 2.5 quintillion bytes each day according to Forbes magazine, and the Internet of Things (IoT), with its immense data output expectations, hasn't even gotten started yet. Because we need to store most of this data, storage will be in endless demand. As such, it appears to be an excellent *commodity* to base a digital economy on.

What ultimately differentiates a UnoSys consumer from a provider is that a provider is committing to having end-user data stored locally on its peer computer and, in return, UnoSys will compensate the account holder whenever that data is served up upon request. In this way, when we refer to a UnoSys *provider*, we are speaking specifically of a peer *storage provider*, since UnoSys does not compensate for compute and bandwidth used on its peers. It is worth reminding the reader that as a P2P distributed operating system, it is still possible for UnoSys consumer peers (i.e., non-providers) to run applications that read and write data. This is possible since UnoSys supports fully distributed file and/or database subsystems whose data is transparently spread across provider peers on the rest of the network. Hence, its single compute model supports *any* application on *any* peer independent of whether or not that peer is participating in local storage (i.e., is a provider).

At first blush, excluding compute and bandwidth resource usage from the provider compensation model may seem concerning since history is full of previous P2P software systems/applications that perform excessive computations pinning CPUs, or saturating Internet connections due to massive file uploads or downloads. However, these scenarios will not occur under normal operation on a UnoSys peer because of the way it is designed.

First, UnoSys is not a parallel computing platform. There are no large number crunching compute-bound services in the UnoSys operation system proper[11]. Instead, the vast majority of UnoSys services that run on a peer are of a simple fast throughput, low latency I/O request/response nature (the main reason for pair-wise peer communications within the UnoSys network) using virtually no CPU. Second, large files are never stored on UnoSys peers. Instead, small (i.e., 64K) blocks are stored uniformly across all peers so requests for end-user block data are small and fast, and are load-balanced across all provider nodes in the system. Finally, all pair-wise communications between UnoSys peers is *batched*, allowing many diverse message types to be shipped back and forth between nodes as a single unit, keeping the number of such exchanges to a minimum.

[11] It is still possible that an application *developer* develops a UnoSys *application* that does CPU-bound activities on the peer computers it is running on. However, the choice to run those applications is entirely with the peer owner. The main take away is that the UnoSys operating system *itself* does not perform long running CPU-bound tasks on peers in the network.

Hence the metrics around *compute* and *bandwidth* resource usage would typically be too small to be of practical concern to UnoSys or its peers and therefore it has been deemed these resources are not worth the effort to account for. Instead, from a resource utilization perspective, UnoSys focuses solely on *storage* – both consuming it and providing it.

However, UnoSys accounts for storage in a subtly different way than typical Cloud platforms do. Cloud platforms typically charge for storage via a set rate per gigabyte per month. Different storage types may have different rates (e.g., hot vs. cold storage, etc.), but the general pattern is that a consumer pays for each gigabyte of storage *each* month[12]. UnoSys takes a different approach.

UnoSys views end-user data as *content*. The platform seeks to attract the best content – whether that is a music file, or a video file, or document, or a database – in order to then attract the end-users wanting to consume that content. There are three stakeholders to consider when thinking about content in the UnoSys ecosystem:

1. The party that *creates* the content
2. The party that consumes the content
3. The party that *stores* the content

Each of these stakeholders has a different interest in the content storage economy of UnoSys, and the operating system accounts for them accordingly.

1.2.1.2.1 Content Creation Accounting

When content is created, the *creator* must be charged to store it in the UnoSys network. However, the way UnoSys does this is unique in the industry in several ways.

First, unlike the prevailing cloud platform model, UnoSys does not charge a creator a per-gigabyte per-month fee. Instead, the creator is charged a per-gigabyte *publishing* fee the *first time* content is written to the network[13] [14]. After this initial publishing fee, the creator does not pay an additional monthly charge to store the content in the UnoSys network for the next 11 months. However, upon the annual anniversary of the content's creation date within the UnoSys network, the creator is charged a sliding depreciating *continued storage* fee (also per gigabyte) for the content[15]. Importantly, all proceeds from the *publishing* and *continued storage* fees collected from content creators go directly to the storage providers in the UnoSys network.

[12] Note that there are also usually charges to access the data once stored, but those are usually billed as part of ingress and/or egress bandwidth charges from a billing perspective.

[13] The amount UnoSys will charge for storage per gigabyte has not yet been set, but the intent is that it less than the prevailing Cloud Platform market value for equivalent storage.

[14] Content is charged at the UnoSys streaming block I/O level and is billed at the end of the month the block was first written to the platform. The publishing fee will be equivalent to a single month's storage fee charged in the prevailing Cloud platforms.

[15] The amount of sliding depreciated discount has not been determined yet, but it is expected to be significant (e.g., 30+%)

With this model, UnoSys content creators will enjoy file and database storage costs significantly cheaper than prevailing Cloud platforms initially, and these costs will only get cheaper the longer the content is stored. Specifically, the sliding depreciating *continued storage* formula ensures that the cost for a creator to store a specific piece of content tends toward zero over time.

Configurable High-Availability

Secondly, upon publishing of the content the creator can decide what the high availability *replication factor* for the content should be. That is, the content creator must decide how many replicas of the content should be made. The default storage replication factor for UnoSys is three[16], which means that every block of content is replicated and maintained on three separate nodes by default, ensuring a reasonable default level of high availability. By default this allows the UnoSys network to tolerate any two of the replicas being offline and still be able to provide access to the content.

Replica costs factor into the *publishing* and *continued storage* fees, in that their rates are multiplied by the replication factor to determine the final cost. Hence the more redundant a creator wishes to make his content, the more it will cost to store. However, even with relatively high replication factors, it is anticipated that the total annual content storage fee for a creator will be considerably less on UnoSys than the standard market value of Cloud platform storage, over the course of any given year. This makes sense since UnoSys has no expensive data centers to maintain because the community provider account holders supply all of the storage.

Note that after the initial *publishing* charge for creating a block, updating it (i.e., overwriting or replacing a specific I/O block) is a free operation for creators and consuming applications[17]. Deleting the block is also a free operation for the creator.

Zero Copy

A third major benefit to creators storing content within UnoSys is that there is no unnecessary duplication of the data. Creators, providers, and consumers all share the same single-computer abstraction that UnoSys represents, which necessarily has only a single massive virtual drive to store all the world's data. Therefore, when a creator publishes a file or database, consumers have direct access to the blocks of this content right where they are stored, subject to having suitable access control permissions. In contrast to the web where it is typical to upload/download a file to/from another computer (i.e.; server) for example, there is no notion of file uploads/downloads in UnoSys because there is no network and therefore no *other* computer (server). There is only ever one logical copy of a

[16] The minimum allowed high availability replication factor is 2.

[17] However, applications that update data excessively may be charged fines, if the UnoSys system deems the frequency with which any particular block is updated as abuse of this free update operation policy (i.e.; in the form of attempted denial of service attacks), which may result in disciplinary action for the applications in the form of negative reputation, or worse.

file or database[18] and consumers access it using parallel *streaming* I/O directly from the providers that store its blocks.

This means content creators needn't worry about piracy since consumers never get an actual (i.e., file) *copy* of their content. Rather, the content is *streamed* in place from where it resides. Therefore it is not possible to have unintended copies of content strewn across the network. It also means that a creator can revoke access to the content from any single user, or group of users, or altogether, simply because there is only one logical copy of the content to manage. As a result, creators have complete control over their content at all times and are better able to manage it no matter how many people are consuming it.

Creators Don't Pay For Content Consumption

However, perhaps the most significant way that the UnoSys storage model differs from that of the industry Cloud platforms is that a Cloud user creates a storage account and is charged for *both* storing the data *and* when it is accessed in Cloud platforms. With UnoSys, the content creator only pays to store the data using the model mentioned above. *Accessing* the data is typically paid for by someone else. Moreover, significantly, UnoSys achieves this separation while allowing the creator to retain control over the decision of what value to place on his content.

By separating who pays for content creation from who pays for content consumption, UnoSys will usher in a bold new digital economy that is inclusive, fair and self-sustainable for all. This economy is better aligned with the supply and demand dynamics of a traditional market as will be born out below.

1.2.1.2.2 Content Consumption Accounting

UnoSys *may* also charge a *content access fee* whenever content is consumed (i.e., read) from the network. However, this charge is levied against the *consumer* who is accessing the content, not the creator (i.e.; owner) of the content, typical of traditional Cloud platform storage models[19]. Moreover, UnoSys supports and promotes self-publication by placing the decision as to whether or not there is a charge to access the content, entirely under the control of the content creator.

Controlling Content Value

Whenever a creator publishes content to the UnoSys network (i.e., creates a file in the file system or creates a table in the database system), a *digital rights management* (DRM) *policy* must be associated with it. The DRM policy allows the creator to control *how* the content can be accessed, including what if anything is to be charged for that access. If the creator does not explicitly assign a DRM policy, then UnoSys will automatically assign a default DRM policy[20] for accessing any piece of content published to

[18] The file or database is stored redundantly using multiple replicas for high-availability. However, UnoSys knows only of a single *logical* version of it. Any *copy* a creator makes of the file is ostensibly considered a uniquely separate *version* of the file.

[19] For some applications, the creator of the content may also be the consumer of the content. However, this is typically not the case for common media applications like online music and movie services.

[20] More detail on the UnoSys DRM subsystem is provided later in this document.

the network. This default policy states that the content is *unrestricted* and *free of charge*. Unrestricted means any consumer can access it as many times as she likes. Free of charge means there is no charge to read the data[21]. This generally equates to the familiar notion of *public domain* meaning freely available[22].

Importantly, content creators are free to set any desired monetary value on their content that consumers must pay to access it. This renders content creators' reliance on classic *publishers* obsolete, because the creators can now be in complete control of their own self-publication within the UnoSys ecosystem, allowing them to retain a significantly greater percentage of the income their content generates.

However, the creator doesn't receive all of the consumption revenue UnoSys collects from consumers on its behalf. UnoSys not only provides content creators with *self-publication* capabilities but also *self-distribution* capabilities. The UnoSys operating system effectively provides direct-to-consumer (i.e.; peer-to-peer) delivery of all content it stores, made possible only by the storage providers serving up the content. Hence, UnoSys redirects a fixed percentage of *premium* content consumption fees set by the content creator and collected from the content consumers and pays it directly to the storage providers responsible for its delivery[23].

With this approach, UnoSys automatically, reliably and securely collects fees that consumers pay to access premium content, and distributes them proportionally to the creator of the content and the storage providers responsible for serving it up on demand, all the while ensuring that the creators remain in control of setting the value of the content. The approach ensures the value proposition brought by all stakeholders – creators, consumers and providers – is equitably represented, fostering a fair and self-sustaining digital economy that benefits everyone.

1.2.1.2.3 Storage Provider Accounting

Storage providers earn money in exchange for storing and/or retrieving content within the UnoSys network. As mentioned above, the creator pays the provider whenever a block of content is published for the first time on one of the peers associated with the provider's account. The creator also pays (on a sliding depreciating basis) on each one year anniversary of the contents creation, assuming it is still being stored (i.e., the creator hasn't deleted it). In addition, if the creator has created a DRM policy that charges consumers to access the content (i.e., the content is premium), the provider will earn a percentage of those fees in equitable exchange for delivering the content on-demand each time the content is accessed.

[21] Note that the two concepts are mutually exclusive in the UnoSys network, as it is entirely possible to store content that has a charge to access, or to store data that does not cost to access but that is not available to everyone.

[22] However, the additional aspect of *access control security* discussed later in this document controls *who* has access to the content. The default access control security for any new published content is *private* to the owner. Therefore, by default new content on the UnoSys network is not public domain until the owner changes its access control security to allow everyone to access it.

[23] At this time the fixed profit sharing percentage paid to the storage providers when premium content is served has not been determined but it is expected to be substantial – i.e.; in the order of 10-30%.

Supporting the Public Domain

In the case where the content creator has elected to place the content in the public domain and not charge consumers to access it, the storage provider is *not* paid anything to store or provide access to the content. UnoSys enables storage providers to *profit-share* in the delivery of a creator's premium content. However, conversely, the operating system expects providers to donate a percentage of their storage services to delivering public domain content as well. That is, UnoSys expects each storage provider to actively support one of the fundamental ambitions of UnoSys, to create a planetary scale computer freely available to everyone, with the information it stores accessible to all according to the intentions of its creators[24].

UnoSys respects and supports the right for a creator to set the value of his digital creations, or those he owns legal distribution rights over. UnoSys also fully appreciates the need to provide open access to works of the public domain and to provide a place where creators of new content can offer their digital works on a world stage in an open, free, and secure way. Furthermore, in order to operate correctly the UnoSys operating system itself generates, requires storage for, and subsequently accesses significant amounts of its *own* data. This data is also stored and accessed on provider peers but is not compensated for. Only end-user *premium* content marked with a pay-for-access DRM policy will generate payments to providers.

UnoSys understands that a provider's commitment of storage to the network is not a philanthropic one,[25] and successful network effect dynamics require the operating system to properly incentivize storage providers. Therefore, the operating system sets a known limit on the amount of unpaid content that will be stored on each storage provider peer[26]. This will ensure that there is plenty of opportunity for providers to earn revenue from serving premium content.

Winning the Lottery

Importantly, because the file and database subsystems of UnoSys are based on streaming *block I/O*, the operating system can guarantee *fairness* in its storage provider remunerations model at the block level. Providers are paid when premium content blocks are stored and subsequently accessed, and since blocks are arbitrarily yet uniformly distributed across all provider nodes in the network, a peer has an equal chance of storing any particular block.

Note that premium storage payments, (payments earned only for the *storage* of the block, not its *access/delivery*) are constant and stable for each provider since every premium block is paid for at the time it is initially written (i.e.; created) and then again, on a sliding scale, on each annual anniversary of its creation date, as explained previously. However, from a consumption perspective, not all blocks will

[24] *Creator* means the person or entity that owns the copyright and/or distribution rights of the content.

[25] However, UnoSys will offer providers the opportunity to donate their storage revenues to charitable organizations if they so choose, and will automatically make the donation transfers on behalf of the provider safely and securely.

[26] The final percentage of unpaid content to be stored on a provider's hard disk has not been decided, but will likely be between 30-50%.

be created equal. There will be specific blocks (even within the same logical file or database table) being more *popular* than others simply because they are accessed more frequently. However, significantly, any storage provider peer regardless of the amount of storage it commits to the network has a statistically proportional chance of storing a relatively popular (i.e., lucrative) block.

Consider a Hollywood motion picture company choosing to distribute their latest block buster production directly to the masses via UnoSys. This will be an easy decision since they simply have to publish it to UnoSys to securely distribute it to the entire world who can watch it on demand, without concern over digital pirating. The movie will have a DRM policy associated with it that charges users to consume it. UnoSys will reliably and securely administer the DRM policy, ensuing that consumers are charged and the motion picture company and storage providers are paid, with complete neutral and transparent accountability.

Consumers begin watching the movie in the free-to-use movie player software that ships with the UnoSys operation system[27]. The movie is a huge digital file (logically), stored across the UnoSys P2P computer as a series of many small blocks. It turns out that blocks at the beginning of the movie (representing say the opening scene of the movie) will likely be more valuable than those near the end of the movie (the scrolling credits) if only because many people won't watch the movie file through to the very end, choosing instead to stop playing it when the credits begin to roll. Therefore, because UnoSys file I/O is stream based, there is a good chance that the blocks making up the tail end of the movie will not get accessed and therefore delivered as often as those at the beginning of the movie, and therefore will generate proportionally less revenue.

A similar scenario will happen with databases. Each record of every table in a database is stored as its own unique block. If the same Hollywood motion picture company publishes a pay-to-access database of movie stars, for example, the peers storing the blocks associated with the records containing the career information for Robert De Niro or Meryl Streep are likely to earn more revenue than those peers storing the record blocks for less known or accomplished actors, simply because there is likely to be a greater number of searches and therefore record retrievals for the more famous actors.

Therefore, throughout the UnoSys storage network there will be millions of such *popular blocks* that will generate relatively higher incomes for the storage providers lucky enough to store them. However, UnoSys ensures all storage peers, no matter their committed storage capacities, have a proportional opportunity to win the lottery and acquire and therefore profit from these desirable blocks.

1.2.1.2.4 Fairness

UnoSys ensures fairness and therefore an even distribution of storage revenue (relative to the amount of storage provided) across all storage provider peers because the blocks are randomly yet uniformly

[27] Software itself is just another form of digital content that is *consumed* by executing it. As we will see later creative DRM policies attached to software can be used to generate revenues for their developers. In this scenario, however, because the movie player software is part of the UnoSys operating system it has a public domain DRM policy associated with it and so does not cost anything for the end-user to consume (i.e., run). However, the movie itself is content that the end-user must pay to consume.

distributed across the peers so that there can be no way to predict which blocks will reside on which peers. Furthermore, since block read requests are randomly yet uniformly distributed across block *replica* storage provider nodes according to the creator specified replication factor, any replica holding a copy of a particular block has an equal chance among its replica peers of receiving the access request to deliver it, and therefore any access revenue that may be associated with it. This means that as long as a peer maintains a high *fair-play* reputation, keeping itself available and responsive over the Internet for long periods of time, it will earn its statistically proportional share of the storage revenue regardless of its actual compute capabilities.

This approach sharply contrasts other P2P systems such as Blockchain DLTs that rely on Proof-of-Work or Proof-of-Stake schemes to unfairly decide who gets to participate in the network – advantaging those with powerful computer rigs, and/or lots of money respectively. Instead, UnoSys embodies open and fair access for the World Computer it enables by providing a level playing field where both the average person with a home computer and a large Fortune 100 company with a massive number of large servers can earn their relative share of the UnoSys digital economy.

1.2.2 Cryptocurrency and Distributed Ledger Technology

Throughout the above conversation there have been several references to UnoSys making payments of value on behalf of *creators* and *consumers* to *providers* and *creators* respectively. However, no mention has been made regarding how these payments are physically carried out. In order to achieve digital commerce across the UnoSys ecosystem, and to ensure fair, secure and trusted financial transactions within it, there is a requirement for the operating system itself to implement a digital cryptocurrency and accompanying distributed ledger technology (DTL) infrastructure akin to Bitcoin and Blockchain. However, unlike today's primary use of Bitcoin as a speculative asset to be largely traded for profit, the only design goal of the Unosys cryptocurrency *token* is utilitarian – to enable the accounting for and trading of value within the UnoSys economy.

Blockchain technology was introduced to the world in 2008 by the otherwise anonymous Satoshi Nakamoto in his landmark whitepaper "Bitcoin: A Peer-to-Peer Electronic Cash System", sparking a global movement to remove hereto required 3rd party trust from transactional systems. The P2P algorithm to accomplish this is complex, and the Bitcoin network it created is slow, difficult to scale and, perhaps most importantly, ultimately unfair in that it is effectively closed today to participation from all but the most powerful computers. Nonetheless, this remarkable paper launched an entire Blockchain-based industry centered on the pursuit of large-scale, secure, peer-to-peer distributed ledger technology.

UnoSys is able to offer DLT technology built on top of its foundational *fortress* security infrastructure. However, UnoSys does it without the need for Blockchain. In UnoSys there is no need to have the entire contents of a DTL completely copied to every node in its network. That is how Blockchain works because it has to, but the design is not and never has been a requirement to secure, immutable, distributed data storage. Moreover the *full replication* approach of Blockchain is not tenable long term as the number of transactions and the number of members in the network grows.

The core driving motivation for DLT is that no party within the network is trusted and no 3rd party trust is allowed or assumed in the execution of a transaction between two or more parties. The Blockchain algorithm achieves these characteristics through a complex Hashcash algorithm centered on cryptographic computational proof-of-work puzzles. However, we have already seen how UnoSys provides built-in autonomous database technology that similarly trusts no single member of its community and assumes no 3rd party trust, and is therefore not controlled or influenced by any single person or entity.

Therefore, creating a distributed ledger in UnoSys is achieved by simply creating an otherwise regular built-in, massively-scalable UnoSys database that runs under the autonomous UnoSysAdmin identity that no human or group has access to. Said another way, *every* UnoSys database is effectively (from the perspective of security) a distributed ledger as long as *only* the autonomous UnoSysAdmin account can access it.

As has been previously discussed in the *UnoSys paper*, UnoSys databases are at all times provably secure from both a privacy and tamperproof perspective because (and despite) being stored uniformly across the P2P network of peers. They are also fully redundant to a configurable replication factor, and therefore ensure high availability and fault tolerance to a configurable number of faulty nodes. Furthermore, because UnoSys limits access to the database to the built-in UnoSysAdmin for which no human knows (or can ever find out) the credentials, distributed ledger operating system services such as a network wide payments subsystem become almost trivial to build into UnoSys.

Hence, UnoSys is able to implement a complete cryptocurrency distributed ledger payment system directly within the operating system. Essentially, every registered account, every peer and every application is automatically allocated the equivalent of a unique cryptocurrency wallet account to manage its monetary value balance that only it has access to. In reality, however, UnoSys goes well beyond a mere wallet by allocating to every account, every peer and every application in the network a complete set of accounting books – akin to a general ledger and its sub ledgers in generally accepted accounting principles (GAAP) – to keep track of all transactions involving each entity.

So rather than having everyone's transactions on a single public ledger for the entire community to see as Blockchain does, the UnoSys operating system, acting as an autonomous 1st person trusted broker between all services and applications running on it, will broker consensus-confirmed transactions between parties across multiple ledgers. The operating system will ensure the transaction *atomically* updates the general ledgers of each entity involved (i.e., debiting one and crediting the other automatically), as well as its own internal transaction journal (i.e., audit table). Thus, on behalf of any user, peer or application, the UnoSys operating system can initiate a transfer of value transaction to be stored securely and immutably in its distributed ledger database infrastructure. The operating system itself, working in a distributed fashion, will come to consensus[28] on the transaction (i.e., ensure that

[28] UnoSys uses proven Paxos-based distributed consensus algorithms which rely on redundant communications between a quorum of peers to agree on transaction orders and values. While several peers are used to perform this consensus algorithm ensuring fault tolerance in the face of a configurable number of coordinating malicious

both the sender and receiver exists and the sender has the funds to make the payment), facilitate the transfer of value between the two parties by committing the transaction atomically to each of their general ledgers, and simultaneously writing a record of the transaction to its own autonomously controlled audit ledger so that it can irrevocably prove when the transaction transpired.

By implementing its own built-in accounting system on top of a distributed, P2P digital currency and distributed ledger payments platform, transaction support becomes a first-class service of the UnoSys operating system. The operating system itself, running as the UnoSysAdmin identity, can ensure transfers of value are atomically made between parties in a provable, secure, party-wise private, efficient, and immutable manner – ensuring that the double-spend problem is an impossibility.

1.2.2.1 Greasing the UnoSys Economy

Significantly, since UnoSys is brokering the transactions itself in the 1st person, and the operating system runs on the commons, there is no expectation of a requirement for a transaction fee. Removing the friction caused by transaction fees will drive adoption of the payment system as a vehicle for commerce within the UnoSys ecosystem, because it will, by definition, return a significant amount of the value of the transaction back to the contributors who make it possible. Moreover, for the first time, a global payment system will be possible that will support micropayments at Internet-scale – tiny payments of fractions of cents that will be required to realize an economy for the Internet of Things (IoT).

Account balances will be tracked with the previously mentioned custom UnoSys cryptocurrency *tokens*. These tokens will be the means by which value is transferred between entities within the UnoSys ecosystem. UnoSys will charge creators and consumers in tokens and pay providers in tokens. Furthermore, developers will be able to easily build ecommerce transaction capabilities into the applications they build for the UnoSys platform, in order that the applications can receive and/or send payments in tokens.

Account holders can choose to spend accrued token balances by consuming services for themselves within the UnoSys ecosystem. Alternatively, tokens can be “cashed out” by transferring them out of the system altogether into a local real-world currency[29], allowing account holders to earn real-world incomes from their activities within the UnoSys ecosystem. Initially the monetary real-world unit value of a UnoSys token will be pegged to the average unit price to store a gigabyte of data for one month, across all major Cloud platforms[30].

agents, the number of peers required to achieve consensus is orders of magnitude less than that required by Blockchain-like algorithms which, as a consequence, are correspondingly orders of magnitude slower.

[29] Transfer fees will apply to cover the costs charged by real world payments systems such as PayPal. This will naturally encourage the spending of accrued tokens *within* the UnoSys economy wherever possible, so that transaction fees can be avoided all together.

[30] As UnoSys eventually achieves critical mass, and because of the natural declining pricing of storage as a whole, it is expected that Cloud platforms storage pricing will tend toward zero.   When this happens, UnoSys will likely adopt a change in model that allows individual peer storage providers the opportunity to set their own storage pricing, creating a competitive market for storage among peer owners. When this happens, the calculation of the unit price of token will transition to be based on this new UnoSys storage market’s average cost of storage instead.

Once UnoSys reaches critical mass and account holders begin accruing larger token balances, it is expected that more and more existing real-world businesses will want to establish themselves within the UnoSys ecosystem in order to vie for the intrinsic value the tokens hold. Consider for example that it will be much easier for any ten year-old UnoSys account holder to generate a balance with real monetary value inside the UnoSys ecosystem, than he could on the open Internet. No credit cards are necessary to do this in UnoSys. All he needs to earn his portion of the UnoSys digital economy is to share his laptop resources with the community by running the UnoSys.Node.exe.

This will represent a significant new market for existing businesses because, without a credit card (typically required to secure a PayPal account or other online payment system), the ten year old will have difficulty cashing out those tokens for real money. He will then naturally seek instead to spend them within the UnoSys ecosystem itself. Initially, digital goods such as games and online media services will likely be the target of those funds. However, as online retailers begin to realize the size of the opportunity, they will establish themselves within the UnoSys ecosystem in order to enable ten year olds to exchange their accumulated tokens for physical goods shipped directly to their home. Moms and Dads everywhere will applaud the initiative that little Johnny shows in the creative financing of his new baseball glove, by participating in the growing Uber and Airbnb inspired "sharing economy" – using the spare resources of his laptop to earn him extra pocket money, in a safe, legal, moral and family friendly way.

1.2.3 Self-Sovereign Identity and Privacy

The Internet has been a boon to mankind in countless ways. But there are many aspects not to like about it as well. One such aspect is the loss of privacy a person experiences with the increased use of Internet services. At its core, this occurs because each Internet service a user expresses interest in requires her to exchange some level of personal information (if only a seemingly unimportant email address, or IP address, or search query) with the vendor for the *right* to use the service. Every time this happens, she loses a little bit more control over her identity and therefore her privacy. Furthermore, in the rise of the "attention economy", this translates into a tangible loss of real monetary value – value that only the service vendors and their business relationships reap.

The cumulative effect is countless and sometimes egregious violations of personal privacy – in the form of spam email, web browser cookie tracking, annoying on-line advertisement, and leaked personal information through security breaches, to name just four – committed not just by the service vendors, but by a chain of 3rd parties to whom the vendors have licensed or sold the personal information to. In a cruel irony, everyone in this value chain profits from the user's personal information except the rightful owner. As the old saying goes, "*if the website is free, then you are the product*".

This all occurs because these online vendors have, through years of convention, claimed license over a user's personal information in the fine print of their website terms of use and privacy policies, and are emboldened to use the profiles they meticulously build on each of their users in any way they like, under the guise of "personalizing your experience" and misleading regulations such as "know your customer" (KYC). By amassing reams of this data, geo-locating it, categorizing it and selling it in bulk, an entire personal data industry thrives for the largest properties on the web at the lost cost of opportunity of the lowly end-user.

UnoSys is in a unique position to offer a solution to this broken aspect of digital life on today's Internet. The operating system design draws inspiration from the Ten Principles of Self-Sovereign Identity towards achieving the first viable planetary-scale identity layer for the Internet. To achieve self-sovereign identity on a global scale requires every person, organization and thing connected to the Internet to have its own digital identity that they own and control, and which cannot be taken away from them[31]. UnoSys, as a fully autonomous, fully trusted, self-policing, commons provided network, owned by nobody and everybody, becomes a natural platform on which to build this much needed capability.

As with Blockchain, there are several other project initiatives pursuing this noble goal of providing self-sovereign identity. However, most are doing so from a traditional SaaS model approach. There is a deep paradoxical irony in trying to create such a capability from a single destination on the Internet – i.e., controlled by a single trusted 3rd party. Appointing a 3rd party not-for-profit company or organization as the gate keeper of this initiative, no matter how well intentioned, does not obviate the need to create a provable, *decentralized* trust authority.

Instead, UnoSys looks to once again exploit its provable fortress security foundation and community maintained and controlled P2P network to create the *autonomous* trust required for self-sovereign digital identity open to everyone.

1.2.3.1 Strong Identity

UnoSys, acting as its own self-sovereign public key certificate authority, allocates all stakeholder entities within its ecosystem a *strong* (i.e., cryptographic-backed) *identity*:

- users (representing a person or an organization entity)
- peers
- applications

These entities form a hierarchy within the operating system where a *user* is at the top, having zero or more *peers* and/or zero or more *applications* associated with it[32]. UnoSys itself generates and manages the cryptographic material required to define and maintain strong identities, including the regular rotating of keys. That is, UnoSys obviates the need to use a trusted 3rd party for the public key infrastructure, since it is able to use its own fortress security foundation, as well as its secure distributed database sub system, to achieve the same end.

All users, peers and applications are created by a person via a special registration process. In the case of users, a minimum of a traditional email address and username is required to be provided. In the case of peers or applications, only a descriptor (e.g., device name, or application name), unique within the context of the controlling user, is required. However, because peers and applications are associated

[31] Assuming there is no abuse to be disciplined.
[32] IoT *Connected things* are considered to have the role of *applications* in this context.

with a single user, only previously registered users can create them. Note that the special internal UnoSysAdmin account mentioned previously is just another user in the system, except that it is automatically created the moment UnoSys goes live, by the operating system itself, thus cryptographically ensuring no human knows or can find out its credentials.

Each *entity* is afforded a complete set of accounting ledgers, as well as the equivalent of a bank account (i.e., a cryptocurrency wallet), as mentioned previously, in order to record token currency transactions against it. This is because *users*, *peers* and *applications* can each earn revenue and/or generate expenses, and must be able to properly account for those transactions uniquely in isolation.

In addition, when peers or applications record transactions, those transactions are also atomically *rolled up* into the account records of the user that they are associated with, as if the user is a parent company and the peers and applications are its subsidiaries. This occurs because when a peer or application earns revenue or is charged an expense, the transaction must be reported for at the peer or application level, but then also ultimately journaled against the user associated with it. In this way, a user can see a consolidated view of all totaled transactions affecting him, and can drill down into the sub ledgers representing its peers and applications to see the details behind those transaction totals.

1.2.3.2 Claims-Based Identity

Each identity is supported by one or more *claims*. A claim is a statement that one entity makes about itself or another entity. For example, a user can make a claim about itself that his *name* is John Smith. The UnoSys operating system (i.e., UnoSysAdmin) can make a claim that the user John Smith owns (i.e., created) the file FamilyPhoto2018.jpg in the UnoSys file system. The operating system can also make a claim that a peer is associated with (i.e., owned by) its user. A user can make a claim that an application is public domain (i.e., free to execute).

Claims should not be confused with access control permissions on a UnoSys operating system object like a file or database. Claims are not what the entity can and cannot do. They are what the entity is or is not. It is up to the UnoSys operating system to map the *is*/*is not* claims to the *may*/*may not* permissions of the entity receiving the claims.

1.2.3.3 Access Control

UnoSys is a fully *user-secured* operating system in that every resource such as a file or a database table is controllable by access control permissions. Initially, the creator of the resource sets these permissions to control which other users or applications can access them. Access control rules map entity identities to traditional resource permissions such as read, write, delete, update, etc. To ensure baseline security is always in-place, *default security policies* (i.e., sets of permission rules that protect resources in common scenarios) are automatically applied to a resource when it is created, whenever no explicit security policy is specified. For example, if a user creates a file on the UnoSys file system, unless they specify otherwise, that file will be hidden from the rest of the community and therefore exclusively accessible by that user only.

1.2.3.3.1 User Groups

To make assigning security policy to large numbers of users easier, UnoSys employs the common notion of *user groups.* A user group is simply a collection of user identities. The creator of the user group (usually the UnoSys operating system itself) can then set the desired access control permissions on the user group as a whole once, and they will apply to all users in that group.

For example, whenever an application developer publishes an application (e.g., MyApp) to UnoSys, the operating system automatically provisions a set of user groups in association with the application – for example MyAppCustomers. Later, whenever a user indicates she would like to purchases a license to the MyApp application in the UnoSys ecosystem, the operating system will automatically include that user in the MyAppCustomers user group, if and only if the user first consents. In this way, UnoSys automatically manages the construction of the MyApp's customer list on behalf of the application developer.

Furthermore, just as the user has a choice as to whether or not to be known as a customer of MyApp, the application developer also has choices. If the user declines to be added to the MyAppCustomers user group for MyApp, it is up to the application developer whether to still allow the user access to the application. These and other similar *engagement rules* are defined by the application developer at the time the application is published using the *digital rights management* (DRM) subsystem built directly into the operating system. They allow the developer to define, in clear terms, the conditions under which his premium content (i.e., his application) can be engaged. The DRM subsystem is discussed later in this document.

1.2.3.4 Privacy by Default

Importantly, UnoSys provides strong protections for identities by design and by default. Inspired by the General Data Protection Regulation (GDPR) initiative, the operating system integrates its strong notion of self-sovereign identity with built-in features to strongly protect personal data, privacy and end-user rights. For example, in the above scenario, if the user does in fact consent to being added to the MyAppCustomers user group of the MyApp application, he later always has the *right to erasure* – to remove himself from the group at a later date. In fact, any user can easily ask the operating system at any time for a complete list of all user groups she belongs to. If she so chooses, she can remove herself from any or all such groups without the permission of the applications those groups are associated with. If she were to remove herself from all user groups, she would effectively "go off the grid" becoming anonymous to all but the UnoSys operating system itself, which always maintains awareness of all users in the systems.

1.2.3.4.1 Permissioned Messaging

However, even when a user is a member of an application's customer user group, UnoSys affords them strong protections because the application developer will only be able to obtain the user's online username by default. Specifically, an application developer does not have enough information (e.g., email address) to contact the user directly because UnoSys protects this contact information very carefully. Instead, application developers must ask the UnoSys operating system itself to send a communication to the user and, again, only if the user has consented to such communications.

UnoSys implements a robust messaging subsystem on top of its built-in, secure, and massively scalable file and database subsystems that allows entities to communicate with each other in real-time through messages, voice and video. *Users*, *peers* and *applications* can all be made to communicate with other users. Even the operating system itself uses the messaging system to communicate with its users from time to time. However, like so much of UnoSys, the built-in messaging system is radically designed to remove the middle-man and, as a result, it turns the table on the prevailing Internet economy in far-reaching ways.

1.2.3.5 Profiting from the Attention Economy

In order to solve the rampant problem of *spam messages* which invariable occurs in any freely usable messaging system, Unosys requires message *senders* to assign a monetary *attention value* to the messages they send. Users who receive *unsolicited messages* (i.e., messages from senders that are not in the users' contact list) are then empowered to set messaging rules defining what their attention is tangibly worth. For example, if a message comes from an unknown source, a recipient can choose to only receive the message if its *attention value* is greater than X tokens. This rule effectively states that the user's attention is worth at least X tokens and if the sender wants that attention they must pay at least X tokens to get it.

1.2.3.5.1 Spam Killer

In order to send such a message, the sender must have a valid *user reference.* User references are anonymous and only the UnoSys operating system itself can map a user reference to an actual user. Therefore, a user must ask the UnoSys operating system to message a user reference on its behalf. Hence, given the scenario above where an application developer has a list of users (i.e., user references) automatically built up by UnoSys in his application's MyAppCustomers user group, he would be able to quickly create a *messaging campaign* in order to communicate with his entire customer base. He would use the appropriate UnoSys built-in tool to compose a single message to be targeted to every member of the group, customized similar to a mail merge in MS-Word through the automatic substitution of *macros* like {Username}. The messaging campaign is then assigned a monetary *attention value* of X tokens (note that X can be zero), and a duration by the sender, after which the campaign is passed on to UnoSys for execution.

The UnoSys operating system will then iterate through the list of user references and filter out all those end-users that have indicated they do *not* want to receive unsolicited messages. Of the Y users that remain, the operating system then automatically counts the Z number of users whose unsolicited message *attention worth* threshold value is less than or equal to the *attention value* the sender marked the message with. UnoSys then ensures the sender can pay for the messaging campaign by automatically and atomically debiting his account by Z x X tokens and storing the tokens in a messaging campaign escrow account for the duration of the campaign. The operating system then delivers the message to each of the Z recipients.

At this point the receiving user will see a summary (like a subject line) of the message and the X attention value the sender attributed to the message and can decided whether she would like to read it in order to be paid X tokens. If so, she opens the message and reads it[33] and the UnoSys operating system automatically pays X tokens from the message campaign's escrow account to the recipient. After the message campaign's time duration has elapsed, UnoSys will delete all unviewed recipient messages and the remaining balance of the message campaign's escrow account is credited back to the sender. In addition the UnoSys operating system will prepare a campaign statistics report for the sender detailing qualitatively and quantitatively such information as how many users accepted and rejected the message, average amount of time it took users to accept or reject the message, and average amount of time it took users to view the message and answer any the *proof-of-view* question that may have been supplied.

1.2.3.5.2 Rethinking Digital Advertising

As another example of how UnoSys will rewrite classic attention economy axioms, the platform prohibits the equivalent of website advertisements or advertisements embedded into content such as streaming video, streaming music or live broadcasts. If creators are found (i.e., it is reported by users of the applications) to be embedding advertisements in any manner within the premium content they develop for UnoSys, they will be disciplined heavily.

This rule is not to punish advertisers. Indeed, advertisers are welcome within the UnoSys ecosystem as an important source of revenue. Instead, the purpose of this rule is to protect consumers by removing the middle-man from extracting value from a transaction – in this case an attention economy transaction. Therefore, UnoSys will provide advertisers (careful) access to its users using the same built-in messaging subsystem and infrastructure mentioned above.

If an advertiser wants to get a message in-front of every consenting user on the UnoSys platform, they will similarly create a messaging campaign, specifying their advertising budget including the attention value they deem their message is worth and the maximum number of users they wish to reach. UnoSys will then randomly select the specified number of users who have consented to receiving pay-to-receive advertisements, and place the message in their internal message subsystem in-boxes. As before, those users that choose to look at the advertisement will immediately receive the promised attention value in tokens.

At first blush, this radical new approach to paying to send messages sounds infeasible, until you consider the original system it is replacing. Consider that spam, for example, only occurs because the current cost to send an email is effectively zero. If instead a cost for that email is set, and a user consents to reading the message in return for that cost, the message is no longer spam but rather a financial transaction based on trade – the receivers attention for the price of the email. Requiring senders of unsolicited message to pay for each message they send would effectively end the problem (from the receiver's perspective) of spam overnight. However, who in their right mind would pay to send an email?

[33] UnoSys may require proof that the user read the message by asking a simple context based question about the message, before completing the payment.

The answer is every advertiser out there, if they could be sure it was delivered directly to a potential new customer who will give it their undivided attention. Consider that every advertisement is by definition an unsolicited message of some sort. Also factor in that most advertisers have a budget for online advertisement. Therefore, advertisers are already paying for unsolicited messages today. The only difference between the current model of online advertising and the model that UnoSys will usher in is who that payment goes to.

In today's web, the websites that attract the largest audiences are the only ones that reap the benefit of significant advertisement revenue, because the advertisers pay the websites every time a user of the website views an ad. This is neither effective nor fair. First, the vast majority of advertising is not of interest to the user, but advertises still pay for ads delivered to that user. Second, the website that hosted the ad receives the advertising revenue rather than the user who is being distracted by it. This isn't equitable since it's the user that is providing the *attention* but the website that reaps all of the rewards. In effect, the website extracts 100% of the value paid by the advertisers for using its billboard on the side of the Internet super-highway that all users are forced to look at as they drive by, whether they want to or not. Instead, UnoSys seeks to realign the transaction and cut out the middle-man by providing all of the billboards itself, and paying the *users* to look at them if (and only if) they want to.

Through its innovative messaging platform, the operating system will enable those same advertisers to pay out those same advertising budgets, but in a much more efficient away, by ensuring that only the users interested in the ads are paid directly to view them. Users who consent to receiving ads will have the opportunity to add as much or as little personal information about themselves to their secure and private self-sovereign identity profile managed by UnoSys. UnoSys can then offer advertisers rich campaign profile categories to better target their messaging to those that will be most interested in it – similar to the advertising platforms of Google and Facebook. The end-user is therefore rewarded with additional attention revenue since the more targeted an advertisement is the more an advertiser is willing to pay for it.

UnoSys protects the privacy of the users because their personal information remains 100% under user control at all times and is never revealed to the advertiser or any other 3rd party[34]. All messaging is done strictly by UnoSys during the execution of the messaging campaign, on behalf of the advertiser. Furthermore, as always, the user is in complete control of whether to receive unsolicited messages or not. For the users that have a rigid "no solicitation" policy, a simple check of a box will ensure they are never again annoyed by a digital advertiser. However, for those looking to earn a share of the estimated $300+ billion global online advertising budget next year, accepting solicitations from advertisers in return for a payment will be an interesting new revenue model for any UnoSys user.

In a very real sense, UnoSys allows users to reap the rewards for their undivided attention that has historically only gone to the big web destinations that they visit. This becomes a major new and viable revenue stream for *any* user of the UnoSys system, while simultaneously increasing the return on

[34] Notice there is no attempt to qualify this statement by adding the words "without prior consent". That is, under no circumstance does UnoSys reveal personal information to any third party. Period.

investment for advertisers wanting to reach those users because they only pay for a targeted user's full consideration.

UnoSys will succeed in realigning the digital economy around the only two parties that matter – the buyer and the seller. By removing the unnecessary intermediaries of the existing digital advertising model, a fairer economy will prosper, featuring happier users and happy advertisers, who in turn fuel the growth and network effect for the entire system.

1.2.4 Digital Rights Management

The digital rights management (DRM) policy capabilities built-in to UnoSys will transform the digital economy because it gives content creators complete control over their creative works. DRM policy is woven from inception into the very fabric of the I/O systems of UnoSys. DRM policies are effectively additional attributes of any file or database table stored by the operating system, and can be thought of as an extension of the user-centric access control security system.

Hence, just as every file or table created in UnoSys has default user access control security it also has a default DRM policy. Whereas access control security controls *who* (i.e., which users or user groups) can access a file or database or application, DRM policy controls *how* that access takes place. DRM policy not only allows a creator to create premium content by assigning an access value to it, but it also allows for the specification of when those premiums are charged and, optionally, to whom.

Consider an author who has written an eBook on emerging new digital economies. The book is twelve chapters in length. It becomes an easy decision to self-publish and self-distribute the book on the UnoSys planetary computer for maximum reach and profit, because the built-in DRM policy subsystem supports creative pricing and distribution models. For example, the author can define a DRM policy for her book that allows the first three chapters to be consumed for free, after which a cost of $6 is to be charged to the consumer to unlock the remaining nine chapters. The budding music artist or independent film maker, or games software developer has similar flexibility in easily defining *a try-before-you-buy* pricing model for their self-published and self-distributed digital works.

By seamlessly taking care of all accounting for the exchange-of-value in the consumption of content, in a provably secure and reliable way, the digital *maker* economy will quickly flourish on the world UnoSys computer. However, the DRM subsystem in UnoSys will offer even more for content creators than *self-publication* and *self-distribution* capabilities. It will also enable innovative new *self-promotion* capabilities as well.

1.2.4.1 Built-In Affiliates

There are millions of talented digital content creators out there that all share a common issue – how to get the word out about their creations. The DRM subsystem of UnoSys will provide help here as well by enabling content creators to establish profit-sharing affiliate marketing programs associated directly with the content they create. Therefore, our eBook author from above can decide to create a DRM policy containing an explicit content affiliate marketing rule that agrees to automatically pay a 25%

commission to any user that formally refers her book to another user who in turn ultimately goes on to read past the 3rd chapter.

In this way, users who consume and enjoy digital content on the UnoSys network can generate revenue from their recommendations. Any user can generate and send a *content recommendation reference* (similar in concept to a web *universal resource locator* URL) for any content they have consumed[35] to her family, friends or colleagues through the UnoSys messaging subsystem previously discussed. If that recommendation is acted upon, the premium content's designated *recommendation commission* will be debited from the recommendation-derived profit the creator received, and in turn credited to the recommending user's account, automatically by UnoSys. In this way the content creator rewards organic word-of-mouth recommendations of happy consumers to everyone's benefit.

The people who are naturally inclined to review and endorse all manner of digital media content, especially those who already have influence over an audience, will suddenly find themselves with a new and potentially lucrative revenue stream. The recommendations themselves will be evaluated by those who acted upon them and recommenders will thus develop a reputation score for how well received their suggestions are over time. Subject matter experts will especially benefit as they will be sought after for their opinion. The bigger their following and higher their reputation scores, the more income they will potentially make.

As more people are charmed by a specific piece of premium content and tell two friends about it, who in turn, tell two friends about it, and so on, and so on, the content grows exponentially in popularity. As a result, talented digital makers who may have otherwise gone unnoticed will quickly garner their share of the attention economy through the wisdom of the crowd. Moreover, those who took the time to write a glowing review upon which others acted will benefit right along with them. Hence, premium content that goes viral earns revenue for not just the creator, but for all those that helped spread the word.

UnoSys automatically and immediately handles the accounting and payment of affiliate commissions securely behind the scenes, making even sophisticated premium content DRM policies feel intuitive from both the creator and recommender point-of-view. Through the innovative way UnoSys integrates its DRM subsystem into its file and database subsystems, a new equitable digital economy will emerge, better aligned with the exchange of value that *all* participants bring to it.

1.2.5 An Application Renaissance

The distributed design and simplified programming model of the UnoSys operating system will enable the creation of the next-generation of Internet scale applications. Those applications will themselves be *content* created by creators – i.e., *developers*. Moreover, similar to other types of content – i.e., files and databases – UnoSys allows for creative DRM policies to be attached to applications to enable innovative new distribution and revenue models. Therefore, applications can be written for fun and published to the UnoSys operating system where anyone can execute them at no charge. Alternatively, applications can be written and published to UnoSys with a profit motive, and UnoSys will dutifully

[35] UnoSys will require that a consumer must themselves have previously consumed (i.e., paid for) premium content before they can in turn receive affiliate revenue for recommending it.

facilitate the exchange-of-value between the user and the developer when those applications are consumed (i.e., run) by the user.

1.2.5.1 Application Identity

Files and databases are passive in that they can't behave improperly on their own. However, applications are active, and therefore require their own formal security context within the UnoSys operating system. In essence, each application has a unique *identity* within UnoSys through which the operating system associates the application with a specific user account – the *developer* – that authored it. It is for this reason that applications, like users and peers, have reputations and accounting ledgers associated with them.

To develop an application for UnoSys, an account holder must pay a small annual subscription fee[36] to UnoSys to obtain a unique *developer certificate* which is associated cryptographically with every application the developer publishes to UnoSys. This developer certificate allows UnoSys to establish reputation metrics for each application (such as its popularity, or the number of times it has crashed, etc.) and associate them back to the developer who wrote it. It is also the way UnoSys keeps track of profits earned by the applications as well as storage resources consumed by the application. UnoSys will then reconcile these values with the developer automatically each month.

If a community member encounters an application that they suspect is acting maliciously or otherwise misbehaving, they can report the application to the UnoSys operating system. An investigation process will then be launched to determine if the complaints are warranted[37]. In this way, application developers are held responsible for ensuring their applications behave in a manner that is in keeping with the broader interests of the community.

1.2.5.2 Enhanced Application Security

Another unique aspect of UnoSys applications is that, by default, they only have access to content that the developer either owns himself, or that is public domain. That is, other premium content owners must explicitly authorize an application to consume their content, since the operating system will not allow this automatically. This additional UnoSys security requirement is fundamentally different from other operating systems which determine access rights to files and databases based solely on the *user identity* running the application performing the access.

Consider a developer who creates a fantastic new movie player application. The application obviously needs movies to play. If the developer published his own movie content he could play those in the application because the operating system will allow an application to consume content if they share the same owner. Also, the operating system would allow any movie content published to the *public domain* by other contributors to be played in the movie player application because, by definition, it is available to be consumed by anyone.

[36] The application developer annual subscription fee amount to be charged has not yet been determined, but it is expected to be nominal – in the order of tens of dollars per year. Moreover, the developer certificate obtained with this fee can be used to write an unlimited number of applications.

[37] This will likely involve a voluntary human task force of community members.

However, if a major Hollywood motion picture company uploaded a pay-to-view movie, it would have to explicitly specify the list of movie player applications authorized to play the movie[38]. In this way, and in keeping with the design principals of UnoSys, the control over how a piece of content is consumed rests entirely with the content creator. When using 3rd party applications for distribution, professional content creators have precise control over how their product is ultimately presented to end-users, since presumably they will vet applications they allow access to their content to ensure they meet their standards. This protection is built into the DRM subsystem as a first class concept, thus enabling content creators to establish exclusive agreements of consumption with specific applications/developers. This also permits content creators to easily revoke agreements they have previously granted at a later date.

1.2.5.3 Protecting Intellectual Property

Another powerful advantage to this approach is that it effectively eliminates unauthorized access to content such as piracy, copyright infringement, and virus infection. Consider a typical movie file on a modern Linux, Mac or Windows computer – say an .MPEG4 or a .WMV or a .MOV file. Once the file itself is accessible (i.e., a user has permissions to open, read and/or write to it), then it is accessible by *any* piece of software *that* user can run against the file. For example, it would be possible to open the software with an off-the-shelf file editor (or custom application) and (given read/write permissions) change the movie file's contents in any way imaginable – such as removing a watermark or embedding a virus. The core issue here is that on these popular operating systems the file is passive. Outside of having a set of user-oriented control access permissions required to access it, the file does not care which *application* does the accessing.

In UnoSys, this is not possible by default. Any file has both user control access permissions and DRM policy associated with it. Both sets of rules have to be satisfied before the operating system will allow the file to be accessed by an application. That is, the *user* running the application has to have the access control permissions to access the file, and the *application* itself has to have the DRM permissions to access the file. Furthermore, since an application is itself ostensibly a file, the only piece of software that can access it is the UnoSys operating system itself. Therefore, it would be impossible to open the application file itself (e.g., MyApp.exe) with the equivalent of low-level file I/O APIs to read/write it from another piece of software. This makes it impossible for unauthorized software to modify the MyApp.exe file in any way, thereby completely eliminating viruses and other forms of binary executable manipulations.

In addition to this superior security UnoSys affords applications, developers will enjoy all of the other DRM policy benefits discussed previously, such as the ability to establish creative try-before-you-buy and affiliate marketing rules that UnoSys will automatically enforce whenever a premium application is consumed (i.e., run) by a user. And just as with other types of content, a developer will not pay for an application's storage if the application is placed in the public domain. Conversely, if the application is premium (i.e., has a cost to execute), then every time a user executes it, the operating system will

[38] The movie owner can elect to enable all applications access to the content if she so chooses.

proportionally compensate the storage providers who delivered the blocks of the application file to the operating system to launch, out of the revenue the application garnered[39].

As has been noted previously, no special skills are necessary to build UnoSys applications. As well, application development significantly simplifies when all notion of a network is removed from the developer taxonomy and the program model reduces to a local single computer abstraction. Lastly, any modern development language can be used to write apps for UnoSys as long as it has a basic HTTP/REST stack. These three facts, in combination with the support UnoSys has built-in to help distribute, protect, promote and reward developers for their creativity in an equitable and open way, will drive significant interest in application development[40] for fun and profit on the UnoSys world computer.

1.2.6 A New Breed of Curation

Content is king on the web today. In fact, there is so much content that it becomes necessary to have it indexed, categorized and organized in order to find what we are looking for. The blogosphere does an admirable job of structuring reams of information so that people can better understand it. There is a blog written by subject matter experts on every topic imaginable. Bloggers exert tremendous effort to coalesce, analyze, interpret, simplify and/or provide commentary on every possible topic of interest. And some bloggers earn themselves a large and dedicated following that can drive lucrative streams of revenue – for example advertisement revenue from programs such as Google's AdSense.[41] But from above, UnoSys disrupts traditional advertising revenue models that popular bloggers rely on, so what is the future of blogging and content curation in general on UnoSys? The answer is it transitions to a premium-content service that users pay to consume.

At first blush, it is easy to think this will never work because people are too used to getting their blog content for free on the web. However, these same users are also not used to being paid to view an advertisement, or read an email, or make a recommendation on the Internet. Hence, existing revenue models will transform into new ones so that everyone – not just bloggers – with have more disposable income in the UnoSys economy to spend – and they *will* want to spend it. Therefore, the very best bloggers will begin charging for their content, and interested users will have the additional UnoSys

[39] In the context of a running a premium application, the developer must declare via DRM policy *who* is actually consuming the application. There are two choices – the application/developer or the user. In the former case, the storage *access* fees paid to providers to access the application file are taken out of the proceeds earned by the developer. In the latter case, those fees are passed onto the user who consents to pay them ahead of running the application.

[40] It is worth pointing out that there are no web services in UnoSys, since there is no network and therefore no web. There is just the single UnoSys computer. Therefore, every little service you are used to using on the web (say a weather forecast web service) will instead be re-implemented in UnoSys as an *application*. Moreover, these applications will be significantly easier and cheaper to build, deploy and maintain on UnoSys then the original services they are replacing, since they will only be comprised of whatever U.I. they require along with simple code to access a UnoSys hosted database to provide the necessary data. Complex 3-tier architectures with load-balanced web service, highly-available middle tiers and clustered data servers, will be a thing of the past on UnoSys.

[41] However, only the most popular professional blogs are self-sustaining – i.e., the revenue they earn covers the cost of content creation – through advertising revenue.

tokens to pay for it, and in turn will earn their own revenue for recommending the best premium blog content.

However, there is an entirely novel and untapped curation opportunity, on a truly massive scale, that has yet to be discussed. It will create an extraordinary new wealth source for both the creators of the content and curators who make sense of it for others. And only a world-computing platform like UnoSys can make it a reality.

1.2.6.1 The Next Big Asset Class

Today's Internet curators work in a document centric world. They make sense of information in document form. They may research dozens of digital or analog resources on a particular topic and then consolidate and condense these inputs into a concise blog post or educational article so that their readers can digest the essential takeaways in a timely fashion. However, that input comes virtually 100% from documents – specifically webpages.

Yet, there is another, grossly underserved source of input that is much more voluminous than that of all of the webpages on the web today. This input source is in chronic need of curation into useful, actionable information. And consumers will pay handsomely for it. This input source is of course the raw structured data of every imaginable type, housed in all of the world's databases.

Consider for a moment the UnoSys value proposition for a typical business managing lots of day-to-day operational data. These businesses spend a great deal of money to store this data and keep it accessible and the P2P distributivity of UnoSys can offer this storage and access, securely and reliably, at a much cheaper cost then prevailing Cloud platform providers, as has already been discussed. This is a compelling value proposition in its own right, but only UnoSys can offer substantially more for any database owner that chooses to store its data in UnoSys. Specifically, UnoSys can offer the ability to transform any data set into a potentially substantial new and untapped revenue stream simply by publishing it.

When you stop to think about it, there is a consumer for almost any data set in the world. There is a student somewhere that needs access to a database containing CEO salaries in order to complete her PhD thesis. A startup exists somewhere that has invented a new kind of toilet and needs to know the amount of toilet paper sold last year in North Dakota, to help determine the size of their initial target market. There is a big pharma company somewhere interested in wound registry data to help with the research for a new kind of disinfectant. And there is someone trying to determine how many hours of direct sunlight a year their southern exposed garage roof will receive in their Seattle subdivision, in order to figure out if it is worth investing in solar panels. One or more data sets exist somewhere in the world to answer each of these questions, but it is traditionally very difficult for the average person to i) locate them, ii) gain access to them, and iii) make sense of the information within them.

Google does an amazing job of indexing the massive amount of data available on the web today so that anyone can search it. The company, through the magic of technology, takes a complex labyrinth of web pages and creates order from them for the masses. However, the input data to those indexes takes the form of published web page documents. Furthermore, these indexed web pages are by definition publically accessible and therefore, in general, do not represent trade secrets of their owners.

In contrast, there is no equivalent of Google search for data that resides in the world's databases. There are several reasons for this:

1. The databases reside in disparate places, typically within private networks, and are therefore not accessible
2. The databases often contain raw data which may be deemed sensitive to those that collect it
3. The databases are cost prohibitive to expose publically
4. The databases are increasingly becoming very large, with billion row tables no longer uncommon

For these reasons and more, no single company could ever create a search engine for the world's databases. However, UnoSys believes a world-computer can.

1.2.6.1.1 The Network Effect of Data

Observe that data sets exhibit a network effect. That is, the more data that is brought together in one place, the more valuable it will become. This is analogous to search on the web. Search becomes more valuable when the content of webpages can be cross-referenced to determine link juice, and keyword relevance scores, etc., that go directly to determining the relative priority to rank and display search results. Likewise, if the data from multiple databases from disparate sources could somehow be cross-referenced, its utility and therefore value would increase exponentially.

For example, it would be very valuable to a company trying to sell boats to have access to a database of people who recently purchased lakeside cottages. However, if that list can be cross-referenced to a database of recent attendees at a local boat show convention, as well as a database of people who have a boating license in a geographical area, the resultant list just grew exponentially in value as a sales lead report.

The above three sets of data (i.e., cottage real estate purchases, boat show attendees, boating licenses), would typically be maintained by different businesses or government agencies, and in many situations privacy rules would forbid them from being shared. However, storing all of those databases in UnoSys creates a truly unique opportunity for an entirely new revenue stream for their owners. Significantly, UnoSys can achieve this *without* undermining the security of the data itself or the personal privacy of people referenced in that data[42].

By leveraging the UnoSys subsystems already introduced in this paper, a vision emerges for an innovative, built-in *knowledge curation service* that will unearth the deep intrinsic monetary value within virtually any data set, and surface it for the owner. Moreover, in keeping with the deep sense of

[42] UnoSys deals with geographic data rules by supporting geo-storage clustering. If required, a user account (i.e., a business) can specify premium storage be used for a specific application, made up of storage providers that are all within a specific geographical area. Hence the data is still distributed P2P across many distinct peers at the block level, but all peers exist in the desired geographically region.

fairness by means of equitable chain-of-value exchange that permeates the design of UnoSys, this data curation service will also create, as by-product, a significant revenue stream for the people whose personal data created the network effect of the data in the first place.

However, before this can be enabled securely and confidentially, a major paradigm shift is required by the database owners with respect to data ownership, and it begins the moment an owner agrees to expose their databases to UnoSys curators.

1.2.6.2 Who Owns Me?

When a database owner is interested in exposing his data set to the UnoSys knowledge curation service, he must first import (i.e., publish) it into the database subsystem of UnoSys. During this import process the database undergoes a *scrubbing* procedure where sensitive personal data that can be used to identify a person such as names, addresses, email addresses, phone numbers, social security numbers, etc., are extracted from the database. UnoSys uses this extracted personal information itself to check for a match against all known UnoSys registered users.

If, by chance, a match is found, then all of the corresponding personal data in the database record is removed and replaced in mass by a single anonymous identifier that only UnoSys itself can link back to the matching self-sovereign user identity. In this way, where the database record used to contain personal information about a person, it now contains only a direct *anonymous user reference* link back to a UnoSys user. This is secure and confidential since only the UnoSys autonomous account has access to the identities and the anonymized identifiers that reference them (i.e.; only UnoSys can map an anonymous user reference to an actual user). Moreover, it amplifies the network effect of the data since it allows it to be linked in powerful new ways with other disparate databases (themselves similarly scrubbed) to extract previously unavailable value.

In the much more likely event (at least before UnoSys reaches critical mass) that no such matching registered UnoSys user representing the equivalent personal information in the database record can be found, UnoSys creates a new self-sovereign user identity *placeholder* to contain the personal information in the database and then proceeds to replace it in the database record with a link back to this anonymous placeholder identity as above. This placeholder identity essentially represents a user that even UnoSys doesn't know about.

The net result is that *all* personal information in *every* database published to UnoSys is replaced with a reference to a privately and confidentially managed UnoSys self-sovereign identity, the majority of which are placeholders – i.e., not yet (at least initially) UnoSys registered users.

Over time, as people register new user accounts for themselves on UnoSys, the operating system checks the list of *placeholder* self-sovereign identities that were materialized from importing databases to see if it has ever seen a record referencing that user. If a match is found, the operating system asks the registering user to confirm the personal information in the profile does in fact pertain to them and, if so, promotes the placeholder self-sovereign identity account to being the official UnoSys identity account for the user. If no match is found, a brand new self-sovereign identity account is associated with the user.

Thus, over time, a profile of every user is built up in UnoSys based on information provided voluntarily by the user on their UnoSys profile page, as well as from information automatically derived by importing databases the user is referenced in. This will give terrific insight to the user about which business, government agencies, academic institutions, health care systems, etc., records information about him. This will also give a registered UnoSys user unprecedented control over their identity as defined by their profile. For example, a user would be able to remove herself from any database she did not want to be associated with on UnoSys. Finally, it will generate intrinsic value around a user's self-sovereign identity.

1.2.6.2.1 Self-Worth

Importantly, the same personal information of a person can/will reside in a great many databases and, over time, as these databases are imported into Unosys and scrubbed of personal information, the above process provides a means for the operating system to automatically cross-reference individual person records in these disparate databases in order to create semantic links between the databases forming a *digital footprint graph* about the user. These links are similar to the links between web pages, in that they surface implicit relationships between the records of otherwise disparate databases owned by unrelated owners. Critically, these discovered relationships are *never* disclosed to the database owners, but are instead held anonymously and confidentially by UnoSys and made available to the user only.

In noteworthy irony, UnoSys then does to the database owners what the database owners themselves have done to their customers for decades. That is, UnoSys will automatically perform the same tracking and profiling of people's personal data that big tech data companies perform routinely and that many people deem an invasion of privacy. And UnoSys will do this for the very same fundamental reason these big tech companies do it, to extract the *network effect value* inherent in the uncovered relationships within the data linked by a person's identity.

However, there are two big differences between UnoSys and the big tech data companies:

- UnoSys will do this user profiling and cross-referencing on a truly massive scale – i.e., across *every* database published to UnoSys that contains references to people
- the monetary value derived from the profiling and cross-referencing of a user's personal data will be shared with the user that ultimately made it possible

UnoSys takes the stance that it's the person and not the database owner that should own and control his own personal identity data. Therefore, essentially, UnoSys *requires* that the owners of any database that contains personal data about an individual – i.e., data that can be used to identify a person – must be willing to give up control over it before they can expose it to the built-in knowledge curation service. That is, the database owners must be willing to lose their understanding of who a customer (i.e.; person) is in order to participate in the new curated information economy and earn potentially significant new revenues from their data.

1.2.6.2.2 Realignment

This is a profound new idea that will completely upend contemporary notions of data ownership, control and privacy. It requires database owners to agree to give up control over data they think they own and control today. It forces the recognition that the personal data the owners record about people is only of limited use and value when considered in the isolation of their own data sets. It compels the realization that only by allowing UnoSys to anonymize and then link their data to other owners of disparate databases via anonymous self-sovereign links, will their data attain its maximal value potential which they can then monetize. And finally, it appeals to their sense of justice and fairmindedness in accepting that the individual person whose data they captured in their database is largely responsible for that optimal value potential, and that it is only equitable that they share proportionally in its monetization.

A similar leap of faith is required on the part of the users. It requires the user to understand that their personal identity has great intrinsic value and it becomes more valuable for them only when a graph of their digital footprint, as captured across all databases they reside in, is built. It compels them to accept that they will never have more control over how their identity is used, including how to profit from it while retaining the level of privacy they desire, than when that graph is maximal. And finally, it asks them to acknowledge that the value of their digital footprint graph is enhanced through the *context* the databases they reside in provides, and that it is only equitable that the database owners share proportionally in that value's monetization.

This symbiotic accord significantly and simultaneously raises the value of data for database owners and the value of identity for the users. Moreover, it ensures that control over how a user's identity is used remains entirely with the user at all times. UnoSys has been designed with a database subsystem that will support the real-time storage and retrieval of huge data sets in a secure, reliable, uniformly distributed way across a network of P2P nodes, cheaper than the prevailing Cloud storage platforms. Moreover, in so doing, UnoSys will enable the discovery of significant latent value in virtually any database asset, especially those that record the activities of people. As references to users are discovered in more databases, the automatic cross-referencing that UnoSys performs will amplify the value of their identities exponentially. This value will be proportionately shared between the database owners who provide the context for the relationships, and the people whose identities are stored in those databases.

1.2.6.3 Unearthing Buried Treasure in UnoSys – An Example

Businesses, government agencies, academic institutions, health care providers, etc., can all expose their databases securely and anonymously in the UnoSys operating system, to anyone (i.e., *data curators*) with an interest in mining this data for its immense dormant value. However, the database owners must first agree to send their databases through a scrubbing procedure to anonymize it using built-in tooling supplied by the UnoSys operating system. These anonymized databases then become the raw input for a new class of curator who creates an information report definition (i.e., a search query) and then asks UnoSys to execute it on their behalf.

Only UnoSys itself, running anonymously, can turn these report definitions into meaningful, actionable results. UnoSys can provably guarantee that the databases in the query are linked correctly and that any results found are known to be actual UnoSys users[43]. Both the database owners, who give the data context with respect to the report, and the users who give the report its intrinsic targeted value, will then share in the profits that the curator ultimately generates. Importantly, UnoSys executes the report query autonomously against the databases and its internal self-sovereign identity database with guaranteed GRDP like data protections in place at all times.

As an example, imagine such a report commissioned by a boating company. The boating company advertises that it will pay the equivalent of $100 per entry for a report listing users that have, in the last year, and in a given designated geographically area:

- bought a lakeside cottage,
- attended a boating or outdoor living convention, and
- have a valid boating license

The curator's job is then to determine the databases within UnoSys that can supply the pertinent data and construct the appropriate report definition containing a spanning query across those databases. After some simple database catalog keyword searches[44], the curator locates three databases within UnoSys, each published by a separate owner, that look promising:

- a national MLS real estate database
- an events company with convention type and attendee information
- a government boat license database

The curator quickly creates a simple report definition that joins the data across all three of these databases using the anonymous self-sovereign user identifiers that each hold, and then asks UnoSys to execute it to generate the resultant report.

The resultant report produces a list of 27 entries – each itself little more than an anonymous *user reference* discussed previously with no personal data associated with any given person[45]. Only UnoSys itself can resolve these user references back to an actual self-sovereign identity managed autonomously by the operating system. Therefore, at no time can the curator or his boating company client learn details of any person listed in the report.[46] Finally, the curator then asks UnoSys to broker the final

[43] Entries in databases that do not reference actual UnoSys users would not be included in the report.

[44] Database owners will add metadata descriptions and keywords to describe the kind of data that is in their databases when they publish it to UnoSys

[45] The execution of the report automatically filters out any users that have chosen not to accept unsolicited messages, so that the boating company is not paying for a message that would not be delivered.

[46] Furthermore, the user references are keyed to the *report instance*, which means that the results are only good for this one report result instance. The report definition itself is content owned by the curator. However, the results of running the report have been purchased one-time by the boating company. The curator could in theory run the report for a different company to generate a new report for them, but she would not be able to sell the other company the results from the first report.

exchanged of information-for-value with the boating company and the operating system delivers the report and automatically collects the equivalent of $2,700 from the boating company.

The $2,700 must then be divided equitably among the report distribution chain of value, spanning the various roles involved as follows:

- Built-in UnoSys ***reporting service*** **application** - 0 tokens since the tool is provided free by the operating system and therefore there is no cost to the curator to consume this application
- The **real estate company** as ***creator*** of the real estate database would get X x 27 x 0.75[47] tokens for the valuable context the database lends to the data, because X was the value the company associated with accessing its anonymized records in the database via its premium content DRM policy
- The **storage providers** that stored the real estate company records would get X x 27 x 0.25[48] for their part in delivering the data
- The **events company** as ***creator*** of the boat show and outdoor living databases would get Y x 27 x 0.75[49] tokens for the valuable context the database lend to the data because Y was the value the company associated with the records in the database via its premium content DRM policy
- The **storage providers** that stored the events company records would get Y x 27 x 0.25[50] for their part in delivering the data
- The **government agency** as ***creator*** of the boating license database would get 0 tokens because the government agency does not set a DRM charge to access its database
- The **storage providers** that stored the government agency database records would get 0 tokens because the content it served up was not premium
- Finally the **curator** as **creator** of the report would receive the equivalent of **$2,700 – ((X x 27) + (Y x 27))** tokens for her efforts.

Therefore, the proceeds are divided between the curator and the database owners, according to the DRM rates associated with the database, because the curator's report is not possible without the data from the database owners. The database owners must in turn profit share with the storage providers whenever its premium content is served up.

Now that the boating company has an actionable sales lead report, it would then use the UnoSys built-in messaging subsystem discussed previously to create a messaging campaign advertising a sales promotion for their brand new 18 foot bow rider, offering a 20% discount and throwing in a $1,500 water ski package. The company gives the message the equivalent of a $20 *attention value* and asks UnoSys to send an unsolicited message containing a 30 second video commercial advertisement to all the users in the report. UnoSys then confidentially executes the messaging campaign, and delivers the 27 messages – automatically debiting the equivalent of 27 x $20 worth of tokens from the boat company's account and placing it in escrow for the duration of the campaign.

[47] Assuming a fixed (i.e.; UnoSys-wide) profit sharing data access rate of 25%
[48] Assuming a fixed (i.e.; UnoSys-wide) profit sharing data access rate of 25%
[49] Assuming a fixed (i.e.; UnoSys-wide) profit sharing data access rate of 25%
[50] Assuming a fixed (i.e.; UnoSys-wide) profit sharing data access rate of 25%

As the users receive the message and choose to open and view it, and then answer a *proof-of-view* question at the end of the message (e.g., what color is the strip on the boat in the video?) they will automatically be paid the equivalent of $20 worth of tokens from the escrow account UnoSys is managing. Finally, after an expiration time period the boating company set in the messaging campaign, any unopened user messages related to the campaign are automatically deleted by UnoSys and the unspent tokens transferred back from the escrow account to the boating account, to complete the campaign execution.

In the end, assuming all users viewed the message, it cost the boating company $2,700 + (27 * $20) = $3,240 to get the undivided attention of 27 excellent sales leads for 30 seconds each, which equates to $120 per lead. This would represent a significant savings over any similar attempt to obtain the same quality of lead from other advertising channels. Moreover, 100% of the entire $3,240 advertising budget was dispensed into the network creating value for the curator, the database owners, the storage providers and the targeted users who together were *all* responsible for the value that was generated. UnoSys managed the entire interaction between parties, securely and privately, with no loss of self-sovereign identity and respecting all user-specified consent/privacy rules, with full financial transaction accountability across all parties, all without any additional fees.

1.3 A New Self-Sustainable Digital Economy

For any economy to form, there needs to exist a set of *scarce resources* together with processes like *production*, *trade*, and *consumption* that arise from people's interactions. Any resource that has a non-zero cost to consume is scarce, but what really matters is relative scarcity. The World Computer, through the capabilities of its underlying UnoSys operating system, provides a rich economy with multiple scarce resources such as *reputation*, *storage*, *content*, and *attention*. Each of these resources is relatively scarce and effort must be spent to produce them. Production occurs in innovative ways when new users contribute peers, earn reputation, and supply storage, when creators of every persuasion invent new and valuable digital content, and when users apply the vast influence of their attention. Trade of these resources occurs organically when diverse members of the community exchange the resources they have in abundance, for those they desire. Resource consumption then follows naturally as content (and the storage it requires) and attentions are devoured in the never ending desire to digitally entertain and inform us. This in turn drives demand for more resource production, driving the engine of a self-sustaining digital economy.

To grease the economic wheels, seamless trading within an economy is necessary. UnoSys enables transactions to flow securely, privately, with full accountability between participants in a provably honest and fair way, and most importantly, at no additional cost. It essentially provides the public marketplace and infrastructure for trade to occur organically and without friction.

UnoSys introduces a digital cryptocurrency whose value is based on the cost of the prevailing Cloud platform storage market, which is set independently outside the platform. Cryptocurrency expansion is tied to the growth of the network and is backed by the amount of storage in it. Together these factors mean the price of the cryptocurrency will remain stable as its supply expands steadily and fairly. Thus, the cryptocurrency will predictably retain its store of value over time allowing it to take on a utilitarian use (as opposed to being just another speculative cryptocurrency with significant volatility in valuation from day to day) within the economy as a means of trade.

Uniquely, the UnoSys economy ensures fairness through inclusiveness. The world UnoSys computer is permissionless and therefore open for anyone to join. In turn, everyone claims their rightful self-sovereign digital identity which they control 100%. Moreover, UnoSys uniquely creates tangible monetary value for a user's identity, with complete confidentiality, protecting its privacy autonomously with the full conviction of its state-of-the-art fortress security infrastructure.

The World Computer incentivizes the growth of the economy through innovative yet simple to understand new revenue models that *anyone* can benefit from including:

- Earning revenue for providing storage
- Earning revenue for your attention
- Earning revenue for your recommendations
- Earning revenue for creating new original premium content that others consume
- Earning revenue curating information out of other people's content

UnoSys provides the provably trustworthy infrastructure required to facilitate honest trade and equitable transfer of value among its members. This requires the removal of any and all forms of middle-man, ensuring that the entire value of the transaction can be fairly and seamlessly distributed to all parties involved in the chain-of-value exchange, with full accountability. UnoSys achieves this acting autonomously, with no profit motivation of its own and at all times impartially represents and protects the interests of all stakeholders to the transaction – the storage providers, the content creators, the user identities, and the consumers.

In this way, The World Computer Project will usher in a realigned digital economy at planetary scale that is run, maintained and controlled by its collective members. It is designed from the outset to be inclusive and fair to all, by rebalancing traditionally lopsided digital revenue models proportionally among the members who are directly responsible for them. The result will be a dignified new self-sustainable digital economy rich in value proposition for members of every background, interest and capability, where all participants in the chain-of-value exchange benefit equitably.