

**ZillionGrid:**  
**“Decentralized Hybrid Blockchain Infrastructure”**

*by Conrad Roberts aka Mr.Zillion*

*conrad(at)zilliongrid(dot)com*

Author Note:

*Creating a High Capacity Blockchain Database Infrastructure.*

*ZillionGrid Version 1 – Phase 1*

Table of Contents

Table of Contents ..... 2

Abstract ..... 4

The ZillionGrid: “Decentralized Hybrid Blockchain Infrastructure” ..... 5

ZillionGrid Component 1: The Decentralized Grid ..... 9

    ZillionGrid Hybrid Topology ..... 10

        Servers and Server Classes ..... 10

        Server Hubs ..... 11

        Hub Clusters ..... 11

    Grid Management ..... 12

        Hub Manager Service ..... 12

        Hub Director Controller ..... 12

        Server Agent Package ..... 13

    Grid Blockchain ..... 13

        Wallets and CPU Miner ..... 13

        GPU Mining Hardware ..... 13

        Zillion Server Nodes aka MasterNodes ..... 14

        IoT Devices ..... 14

    Grid Access and File Management ..... 14

        Zillion Login: Smartphone Secured Access ..... 14

        Zillion Seal: Asset Validation and Authentication ..... 15

        Zillion Sharding: File Management ..... 15

        Zillion Containers: Websites, pages and profiles on the Grid ..... 15

        Zillion ZLN Protocol: Communications on the Grid ..... 16

ZillionGrid Component 2: The Universal API ..... 17

    REST API ..... 17

        Scripting Languages ..... 18

        Programming Languages ..... 18

        Platforms and Access ..... 18

    Crypto Payment Gateway ..... 18

Crypto Payments .....	18
Micro Transactions.....	19
Recurring Payments and Subscriptions .....	19
Any Crypto Blockchain, Present and Future .....	19
Intellectual Property and Digital Assets Library.....	19
Digital Fingerprints .....	19
Syndication .....	20
Certificate of Authenticity, Manufacturing, Origin.....	20
Copyrights, Trademarks, Patents .....	20
ZillionGrid Component 3: The Software Factory.....	21
Zillion Browser: Navigating the Grid .....	21
Zillion Mail: Encrypted Communications Including Attachments.....	22
Zillion Wall and Zillion People: Social Interactions with Real People .....	22
Zillion Suite: Office Suite on the Grid.....	23

### Abstract

In the world of crypto, the concept of blockchains and payments go hand in hand. Smart contracts are all the rage, and many initiatives are searching for real world applications. The world at large is puzzled by this new paradigm: what can we do with it and how do we use this in everyday life? Blockchain programming requires top caliber developers and big budgets; many of the millions of “traditional” programmers are scrambling to catch up.

As it turns out, the blockchain is good as an “immutable ledger” keeper and cryptographic payment system but lacks in data storage, querying abilities, and most of all, performance. The blockchain is a flat-file-based ledger, thus, architecturally speaking, the slowest possible architecture. A little research confirms these shortcomings, hence the absence of real-world applications: it is not practical to build applications using the limited capacity of today’s blockchains. Crypto cloud storage, word processing, excel spreadsheets, secure messaging, and communications are barely a blip on the radar when talking about blockchain integration. Real application infrastructures still require fast databases and massive storage needs for media distribution on any of the current platforms: web, desktop and mobile.

This paper presents a solution in the form of a total integrated network: media, database and blockchain servers working together and providing the first ever “Decentralized Hybrid Blockchain Infrastructure,” and bringing blockchain integrated applications to mass markets.

## The ZillionGrid:

### “Decentralized Hybrid Blockchain Infrastructure”

The original Bitcoin blockchain was made with one purpose in mind: **decentralized and peer to peer payment system**. With that solution came an immutable ledger in the form of a blockchain with new blocks being created at preset intervals. Each block is linked together in a chain structure and cryptographically secured, which makes double spending a problem of the past. This paper will not explain these concepts, just acknowledge the current architecture and its shortcomings when applied to an application infrastructure.

When we look deeper under the hood, we see **a series of flat files** containing all the transactions and addresses stored in a local download folder. All these files make up the chain and contain everything that has ever occurred on this chain. Technically speaking this is a breakthrough accomplishment, but from a programmer’s point of view it is the most inefficient architecture for accommodating data storage and querying functionalities. There is **no space to store any application data** without creating bloated chains, something that eventually becomes unmanageable.

If we are to extend this innovation to applications, some changes must be made to accommodate these requests. How can this be achieved? One way is bigger blocks. These will hold more transactions and eventually we will be able to store more data on them, but can a blockchain with a file-based architecture really store all our word documents, images, and video files? Most likely not, so we must look for a more efficient architecture.

Databases come to mind, since they have been proven reliable for many decades and eventually the blockchain inner workings will migrate to a more efficient management system. There are a few initiatives already out there but almost all of them have the same challenges:

migrating the blockchain's unique characteristics to a database system will take time and a lot of effort.

Enter the ZillionGrid. What is the ZillionGrid, why was it created and how will it be built? First and foremost, it is **a complete application infrastructure** including hardware, network security, storage, and everything else needed to accommodate applications, websites, and social interactions with people around the world.

After months of research we broke down our infrastructure architecture into **three main components**:

Component 1: Network part with Server Classes A, B and C clustered in hubs

Component 2: Universal API for programmers to access those resources

Component 3: Software Factory with turn key application suites for the end user

**Class A** servers are defined as Media and Storage servers, high performing servers with large storage capacity.

**Class B** servers are defined and configured as Database Servers optimized for processing database transactions on any database in existence.

**Class C** is defined as Blockchain Servers: any currently existing or future blockchain can be integrated into this grid.

We also have Class D and E for later use, being User Nodes and AI Operations nodes (for Deep Learning, Dynamic Benchmarking, and Grid Request Routing, also known as K-Nearest).

These will be defined and introduced in Phase 3 of the ZillionGrid building plan.

Class A, B, and C Servers are organized in Hubs, reporting their status to the Hub Manager via pulse metric software agents installed on each server.

Hubs are organized in Clusters and each Hub Manager reports to the Hub Director, thus creating a real time status and availability.

The Universal API and Software Factory calls are (eventually) routed via AI and Machine Learning algorithms directing and routing all requests to the nearest and most available server. All metrics are taken into account: CPU load, RAM and HD usage, and Internet Upload and Download speed. Slower servers are ranked lower and will handle less requests. These rankings will be communicated to node operators, so they have the option to intervene when needed.

The decentralized aspect of this infrastructure poses a few challenges: since anyone can operate a node, how do we handle quality control of newly added nodes? Our solution comes in the form of a “ZillionGrid Node Launcher”, a tool with a built-in benchmark to automatically classify the hardware it is analyzing. That benchmark number will eventually assign the A, B, or C classification. This classification will be assigned via unique identifiers to a server profile stored on the blockchain. Once the node becomes operational it will receive shared block rewards based on the class and provide recurring revenue to the node operator. This will avoid having to rely on advertisement for income, and because these block rewards are a form of smart contracts, no middlemen are needed, and payments are done in real time and automatically. The only way to discontinue payments is when the node goes offline, which is a decision made solely by the node operator himself.

We wanted to start building this right away, while acknowledging a potentially long development timeline before we could reach our ultimate goal: a blockchain database architecture capable of storing unlimited data with unlimited transaction speeds and real time querying capabilities, while still preserving the blockchain qualities.

The Zillion Team is working in that direction and since this will not be built overnight, we propose to break it down in three phases.

First Phase will be a grid structure with a software resource manager that might look partially centralized but is not actually centralized. This will allow us to build a grid with three server classes and create the Universal API to allow any “traditional” programmer to access those resources using any programming language in existence.

Second Phase will be a peer to peer implementation of encryption, sharding, and all the data storage aspects that are needed to make a high performing application network.

Third Phase will be the goal of a database blockchain with the same qualities of a file based blockchain system.

Technology changes constantly and by the time this paper gets (re)published, parts of it might already be outdated. For this reason, we will elaborate on the First Phase approach and publish updates as our work continues and moves into the other two phases, whenever that will be.



### **ZillionGrid Component 1: The Decentralized Grid**

The emergence of the World Wide Web in the early 1990s came with high hopes and many expectations. It changed humanity in numerous ways, but today only a handful of companies have centralized it for their own purpose, at the expense of individual users' privacy and security. Your data is now the highest priced commodity in the hands of a few profit chasers, completely unregulated and without any compensation. Impersonation, identity theft, spam, and phishing, are just a few ways that scammers and abusers target everyday users. Centralization has proven to be a tempting power broker where the unscrupulous rule and dictate what we can or cannot do, what we can or cannot buy, and what we can or cannot browse.

To get the user back in control of their own data, a new infrastructure is needed. This paper describes one of the possible solutions to those problems. The ZillionGrid allows for anyone to contribute resources and generate recurring revenue. This is not dependent on advertisers and the immense intrusions that come with it, all at the expense of the end user. Some sites make it almost impossible to get to the content, cluttered by ads and broken by all the browser versions and countless scripting libraries that make the viewers' experience a nightmare.

There are multiple ways one can create income on the grid. Mining crypto currencies is one way, while running a Zillion Server Node (or Grid Node) is another way. As a grid user, anyone can sell items, such as digital assets, or create income using video streaming or podcasting. Micro transactions, buy and sell transactions, subscriptions, pay per view: everything is built in and natively supported by this infrastructure. Natively means there is no need for third parties, credit card processors, banks, or other intermediaries to manage your transactions. You decide what happens to your data and your digital assets on the grid.

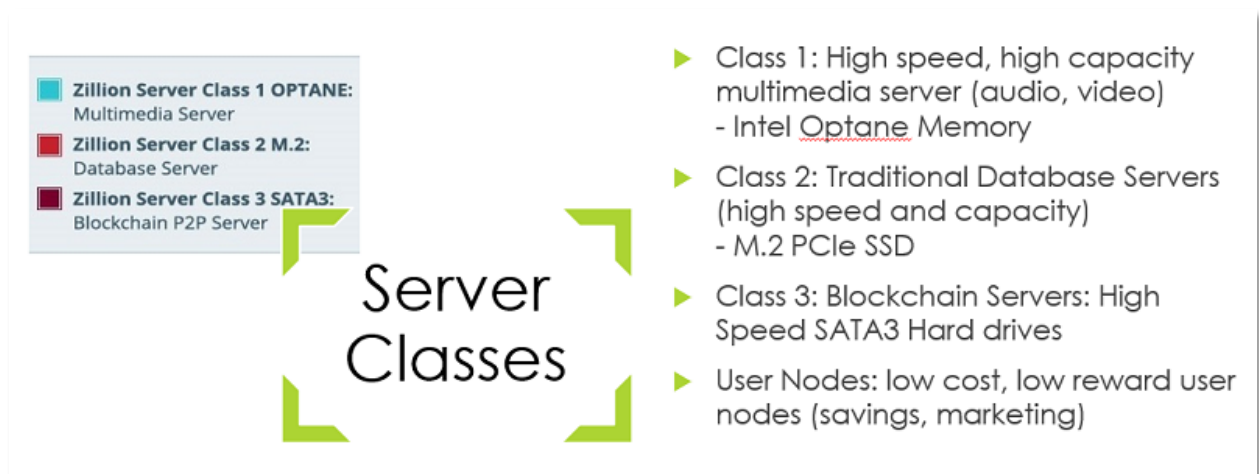
## ZillionGrid Hybrid Topology

To accommodate our First Phase objectives and build the grid, we defined three server classes to handle all data types. Since anyone can add hardware to the grid, we created a software installation package including benchmarks so we can automatically assign each server with a classification. Servers can be stood up or taken down without affecting the data stored on the grid. To accomplish this, we created redundancy and duplication services, data queues, and time synchronizers, all of which will make way for P2P database technology in our Second Phase.

### Servers and Server Classes.

To achieve speed and low latencies we suggest optimal performing components, including fast CPU's, fast RAM, and fast hard drives. We are aware of the added cost, but eventually, the crypto rewards should provide a decent ROI to make that investment worthwhile. In addition to shared block rewards, node operators will receive transaction fees. The more capacity the server has, the more transactions it can perform, and the more revenue it can generate.

*Image 1: Server Classes A, B and C*



*Image 1. Zillion Server Nodes Class A, B and C including User Nodes*

### *Server Class A*

Media Servers: These servers will do the heavy lifting regarding storage, audio and video, images, and big files. RAID systems will be needed with RAM drives for fast response times.

### *Server Class B*

Database Servers: Similar to Class A servers regarding speed and capacity but the configurations will be optimized for database operations.

### *Server Class C*

Blockchain Servers: These will act as full nodes and accommodate all blockchain requests. Since the grid is “technology agnostic”, we can host any format, database, or blockchain.

### *User Nodes*

For later.

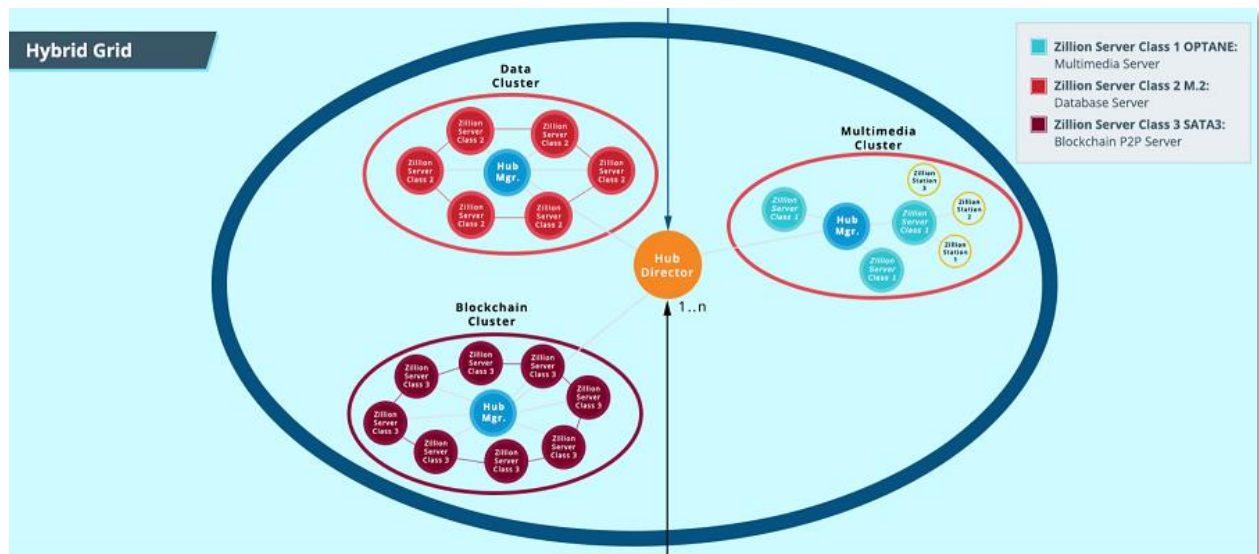
### **Server Hubs**

Servers are organized in hubs by class. New servers can join existing hubs and synchronize data; once synchronized they can serve API and/or app requests.

### **Hub Clusters**

Hubs are organized in Clusters. Hub Managers report to Hub Director, and the Hub Director connects to API or Software Factory.

*Image 2: ZillionGrid Server Hubs and Clusters*



*Image 2. ZillionGrid Server Topology including Hubs and Cluster organization*

## Grid Management

To manage all resources, we created the following structure:

### Hub Manager Service

The Hub Manager takes all the metrics from the Server Agents and keeps track of all the connected resources. It is a software service that runs in the background and manages the data request queue, assigns timestamping, takes care of server synchronization, and configures new servers added to the Hub.

### Hub Director Controller

The Hub Director Controller receives info from all Hub Managers and creates a health diagram of all Hub Clusters. The Hub Director visualizes the total grid health and capacity and routes all incoming calls to and from the grid. Eventually this will run AI and Machine Learning algorithms.

### **Server Agent Package**

Software installed on each server node sends metrics to the Hub Manager regarding server resources and load. It keeps track of and reports RAM and HD capacity, CPU usage, and internet upload and download speeds.

### **Grid Blockchain**

Many of the grid functions are blockchain secured, hashes and digital fingerprints are immutably stored, encryption and financial transactions are provided by “ZillionCoin” thus implementing the “ZillionFLUX” blockchain algorithm. These have their own whitepapers and are managed by the open source community. The main reason for its integration is that for the grid to function, blockchain architecture has to differ from what is currently out there since data storage, data querying, and other functions needed by the grid are not built into other blockchains to the extent that is needed. The ZillionFLUX blockchain accommodates those special needs. Since the grid is “technology agnostic,” any current and future coin can be integrated to the extent that their blockchain architecture allows for the needed functionalities. If nothing else, they can be used for integrating payments and micro transactions.

### **Wallets and CPU Miner**

The ZillionCoin wallet is available for free and source code is provided by the open source community. The wallet accommodates crypto transactions and has a CPU miner built in.

### **GPU Mining Hardware**

Once the CPU mining phase is over, we will increase capacity by switching to GPU mining. We will provide hardware configuration guidelines and we might provide turn key hardware in the future.

### **Zillion Server Nodes aka MasterNodes**

Our design calls for Zillion Server Node integration and shared block rewards with miners to provide ROI to server operators. This takes away the need to depend on ad revenue, and since shared block rewards are a form of smart contracts, no middlemen are needed, and payments are executed automatically and in real time.

### **IoT Devices**

The Internet of Things (IoT) is a huge market with unlimited growth potential for early application providers. We planned our own version of an IoT network linked to blockchain transactions and we created our own line of IoT devices (including Iridium Satellite trackers). More about that in Second and Third Phase implementations of the ZillionGrid.

### **Grid Access and File Management**

Providing secure access without making it too complicated is a challenge. We chose a dual access system: smartphone secured access via QR codes and email/password systems with SMS OTP (One Time Password). Neither of them is foolproof but they insert additional layers of complexity not found in current internet access systems. Everything starts with a unique profile of which the hash of “digital fingerprint” is stored on the blockchain and used to validate users throughout the grid.

#### **Zillion Login: Smartphone Secured Access**

The first step for acquiring grid access is to create a profile using your smartphone, email, and secure password. Once this information is submitted, a digital fingerprint will be created and stored on the blockchain. This digital fingerprint is then later used to validate your profile when used within the grid.

### **Zillion Seal: Asset Validation and Authentication**

The Zillion Seal technology is used throughout the grid to validate and authenticate digital assets using the digital fingerprint and, when confirmed, seal the process with a verified approval status of the asset. This will prevent fraud, impersonation, virus insertion, and other malware protection.

### **Zillion Sharding: File Management**

The term “sharding” is mostly used within database context and we will do the same. To maintain our terminology throughout our Three Phases, we started implementing sharding in the First Phase as it pertains to storing files and, more specifically, big files. Every file stored on the grid is encrypted and broken into smaller pieces and stored randomly on available servers. We keep the meta data and when a user wants to reconstruct and decrypt the file, we know how to do it. It is to some extent like IPFS (Interplanetary File System), but we use it in addition to IPFS. It provides extra security at a performance penalty.

### **Zillion Containers: Websites, pages and profiles on the Grid**

Websites, HTTP, and Internet are all synonymous and are proven methods to establish web presences around the world. Much of its security is debatable, but we are all familiar with the shortcomings, annoyances, and security intrusions. Never mind the spyware, malware, viruses, phishing, and a host of other less desirable phenomena that make the web a very unpleasant place to hang out, shop, or communicate. The ZillionGrid will try and establish an alternative to those problems while keeping compatibility with existing systems.

“Zillion Containers” are a first-generation ZillionGrid websites where we use IPFS as the base to create websites (pages, profiles) on the grid. There are a few shortcomings that we need to resolve, and we will present that solution in the form of “packaged” files secured by a

“container” hash to verify authenticity, hence originality, as envisioned by its creators (minus the viruses, malware, spyware, etc.).

### **Zillion ZLN Protocol: Communications on the Grid**

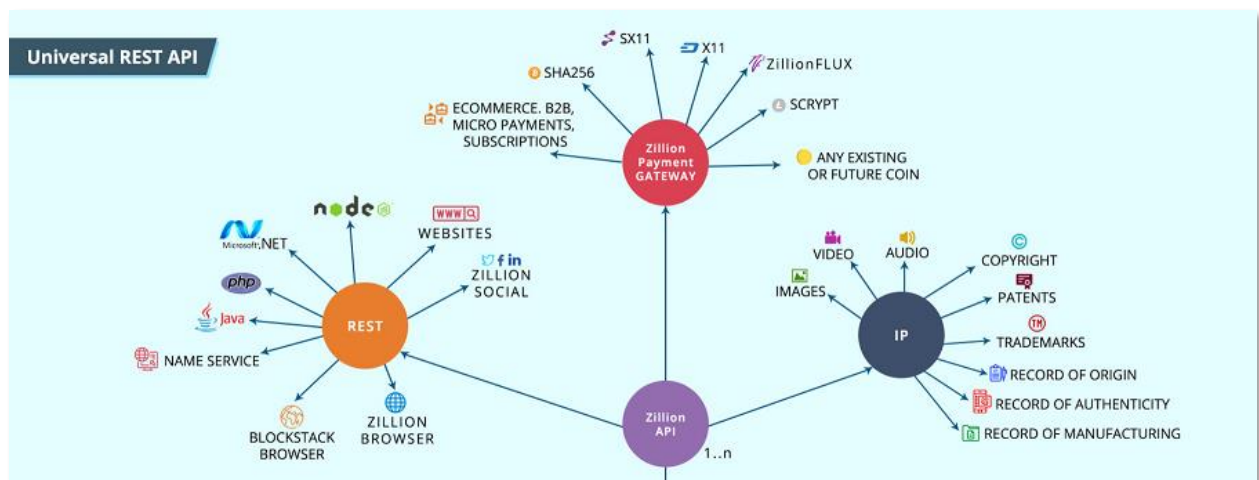
The task of managing routing, data transfers, addressing destinations, profiles, hashtags, containers, and any similar tasks falls on the shoulders of our newly created Zillion Protocol (ZLN Protocol). That requires a paper by itself and will be presented later. Right now, the ZLN:// prefix is used in our Zillion Browser for grid communications and is also available as browser plugins for Chrome and Firefox. With those plugins, certain grid functionalities will be available to standard browsers.



## ZillionGrid Component 2: The Universal API

Building the grid infrastructure is one challenge, while creating easy access is another. We propose an easy solution by programming an API on top of that grid. All grid resources are now open to developers and instead of multiple connections to multiple resources, we have everything in one place, ready to be integrated in your applications, be it desktop, web, or mobile.

*Image 3: ZillionGrid Universal API*



*Image 3: ZillionGrid Universal API with REST, Payment Gateway and Intellectual Property Digital Asset Library*

### REST API

Blockchain programming is not easy, it is rather low-level programming and yet? requires top talent, which is expensive and hard to find. Our API allows easy blockchain access via Universal API and any programming language in existence. Suddenly, millions of “traditional” programmers now have access without having to learn the specifics of blockchain programming. REST is universally known and easy to integrate once access is established using your Zillion Login profile.

### **Scripting Languages**

Many websites and applications use scripting languages, be it JavaScript, PHP, ASP, Python, Ruby, or anything else: as long as you can make REST connections, you are able to access any application platform using any programming language.

### **Programming Languages**

For desktop or mobile programming, REST is easy to integrate in Java, C, or any .NET language including C# and Visual Basic. All REST calls are returned via JSON result sets and integrating these in your application is easy.

### **Platforms and Access**

Same easy access for any desktop, web, or mobile platform including NodeJS, Angular, React, and any other library in existence including Name Services and Zillion Browser, including Chrome plugins.

### **Crypto Payment Gateway**

Our payment gateway includes any traditional payment mechanism, except we converted it to crypto using any coin in existence. If you want to setup payments in Bitcoin, Bitcoin Cash, Dash, or Litecoin, you can do so. You can setup one-time payments, recurring payments, subscriptions, per minute, per view, or any combination thereof.

### **Crypto Payments**

If you have something to sell, you can integrate payments with your favorite coin, or multiple coins. This goes for digital assets as well. Images, reports, and newsletters can all be integrated with our crypto payment system.

### **Micro Transactions**

Micro transactions are small amounts that traditional credit cards will not process. For example, integrated payments per minute of listening or viewing content, or each time an image is used, or a book is read. You can integrate payments with your podcast or streaming station.

### **Recurring Payments and Subscriptions**

Recurring payments are automatically scheduled at your defined interval. It can cover memberships or content channels, online stores or digital catalogs.

### **Any Crypto Blockchain, Present and Future**

We can integrate almost any coin in existence as well as future coins; nothing is off limits. Tokens are a separate class of payment models and we plan on integrating those also.

### **Intellectual Property and Digital Assets Library**

Many apps will process or access digital assets and Intellectual Property. For this purpose, we created a mechanism to register, track, and monetize these assets in a digital library. Once your assets are registered, you can then incorporate them in your applications and track their usage throughout the grid. If compensation is required, our payment gateway can take care of those transactions for you.

### **Digital Fingerprints**

Any digital asset created on the grid can register a digital fingerprint and claim ownership with proven identity. These assets can be text files (reports), media files in the form of audio or

video files, images, or anything else that can be made available for digital distribution. This fingerprint can be used for tracking, billing, subscriptions, or anything else provided on the grid.

### **Syndication**

Once the grid has a substantial number of users, syndication can be a vehicle to greater exposure. This will most likely be used for subsequent versions of the grid.

### **Certificate of Authenticity, Manufacturing, Origin**

The blockchain provides an excellent method for certifying and providing records of Authenticity, Manufacturing, Origin, and many more variations on this theme. If implemented well, this can track goods from raw materials to finished goods, distribution, sales, and support.

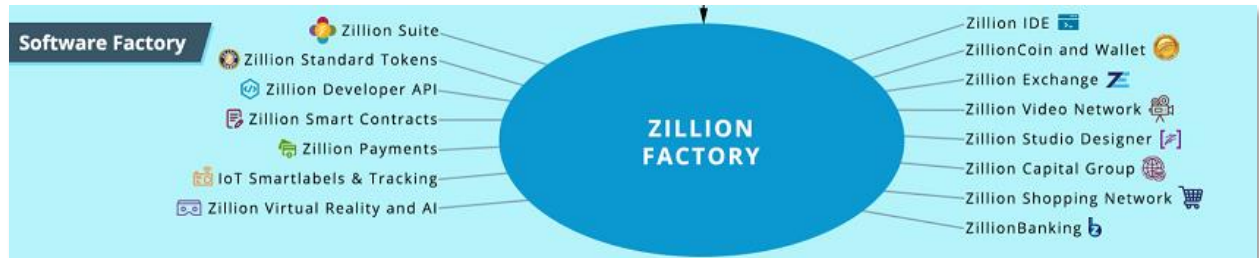
### **Copyrights, Trademarks, Patents**

This requires integration with existing systems and will most likely be implemented in subsequent versions of the grid.

### ZillionGrid Component 3: The Software Factory

Turn key software solutions for enterprises and home users.

*Image 4: ZillionGrid Software Factory*



*Image 4: ZillionGrid Software Factory: turn key software solutions*

### Zillion Browser: Navigating the Grid

Decentralized websites are in a very early stage, so there are many challenges to make them work in a practical fashion. Our first generation is built on the IPFS structure with some limitations. We created our own browser based on the JavaFX rendering engine, in addition to a Chromium version (Electron with NodeJS) for those who want to be part of the Google Universe. By design, we turned off all tracking and you control your cache and cookies. We prefer to disconnect from anything Google, but currently that comes at a penalty since Google cornered and owns the browser market. Our JavaFX implementation will get better over time; this is a start and comes at a price of limited functionalities and page rendering. Full compatibility of traditional internet browsing will be maintained. We do not want a closed system, but rather an enhanced platform for those who prefer to own their data and not be tracked with every click. This is not 100% foolproof yet but is a first step in that direction.

Each browser installation comes with a local mini-database where you can store your own activities and browsing data. This mini database can then be made available to advertisers, but only at your request. You can offer your data on the Ad Market and get paid in real time via

smart contracts. Your local data repository can also be encrypted, again at a performance penalty. Everything is under your control, and you decide what happens to your private data.

The Zillion Browser has an applet system built in and can connect directly to your local wallet. Right now, wallets are still full nodes and require synchronization, but future versions will be lighter, like SPV (Simple Payment Verification).

### **Zillion Mail: Encrypted Communications Including Attachments**

Communications are crucial in any enterprise and this encrypted messaging system allows for just that: secure communications including audio and video messages and file attachments of any type. Using this system, there will be no more snooping or intrusive mail analysis for serving ads, and no more massive invasion of your privacy. Your communications are encrypted and sharded, including any attachments, such as audio and video files. It works like a traditional email system except it is more secure and your messages are secured by our Zillion Seal technology. Each message has its original digital fingerprint and after reassembly and encryption, it is verified and authenticated against the original fingerprint stored on the blockchain. If anything changed in your message, you will know about it. Some of the advantages are spam limitations, a lack of impersonators (the sender is verified against their profile), and no more viruses or malware in your inbox.

### **Zillion Wall and Zillion People: Social Interactions with Real People**

This is our version and implementation of social networking. Since everyone is validated using their smart phone, we anticipate only real people on our network. All messages and postings are validated for their origins and content. Our aim is to make it much harder for abuse and false information to proliferate on this network. It is still possible, but much harder to circumvent the system in big numbers. This way we have quality over quantity and as with all our applications,

you own your data, you store it locally, and you make it available on the Ad Market for monetary gain, but only if you decide to do so.

### **Zillion Suite: Office Suite on the Grid**

Office applications are still the staple of every business. Word processing and spreadsheets are used every day in millions of businesses around the world. Our Zillion Suite provides full compatibility with existing suites on the market. You can store your files locally, in the cloud, or on the Zillion Grid. Grid storage comes with the option of encrypting and securing your files for later retrieval and validation using the Zillion Seal technology. If you choose to charge for your files, you can do so without much effort. Just define your parameters when storing to the grid, present it on your profile and people can download, pay, and enjoy our work. This will all be seamlessly integrated on one platform with many crypto coin options ad payment.

All these applications will be available on desktop first, then on tablet and mobile platforms. We cover any market, from enterprise to smaller companies to private individuals.

*For All Mankind.*