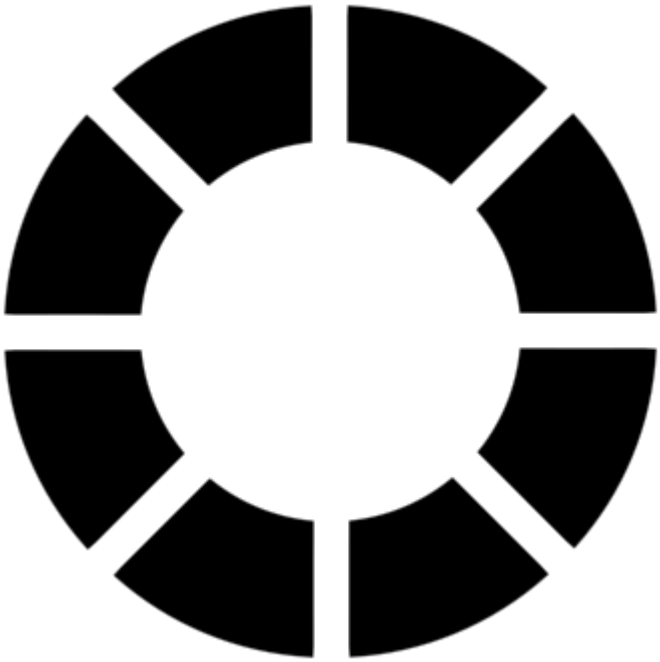


ULTRANET

Technical White Paper

v 0.1 Alpha Stage

ultranet.org
axiomion.com
twitter.com/ultranetorg

Abstract: Ultranet introduces an open decentralized infrastructure for a new generation of Internet applications which combines advantages of both web and local platforms with protection and security provided by the blockchain technology. From the local platforms, this technology takes the high performance of native code execution, rich UI capabilities, and local resources access. From the Web platform, this technology inherits its platform-independent network nature, safety, and ease of use. The blockchain technology, in turn, is used for protecting the whole system from malicious software and other threats. Decentralized application and smart-contract technologies, like Ethereum, provide us with a global decentralized 'back-end' while Ultranet technology complements it with the global decentralized 'front-end' and together they begin a new era of the Internet.

The technology consists of five major components Free Distribution Network (FDN), Anti-malware Protection Network (APN), Distributed Application Platform (DAP), Unified Operating Superstructure (UOS) and Unified User Interaction Environment (UUIE). These components together allow developers to create cross-platform applications which are run as easily as the opening of a web page but with full power of local system without any issues of today's cross-platform development. A lot of various web applications can utilize the power of true desktop-like UI to provide with a rich and fast user interface experience unreachable with classic HTML-based technologies. The Ultranet applications run under Unified User Interaction Environment are not limited by classic windows GUI but also capable of rendering any 3D visual interfaces that is important to support existing and future VR devices.

This platform does not stick to a particular OS, and once it is accepted worldwide, it will make many existing operating systems unnecessary. This would constitute a quantum leap. Many years ago platforms like Windows and MacOS brought us multitasking, and graphic UI and nothing new has happened since then. Until now. An overlooked shortcoming of existing OSs involves privacy; all major proprietary OS's spy on their users. Those days are over. Started as a cross-platform technology Ultranet may finally require only one, free, secure, open OS, like Linux, thus ending the reign of big corporations in this sector of the economy.

Contents

The Problems of the Web

Every site in the web is using HTML to generate its pages. HTML is a static text markup of page elements. It was developed as a language that allows creating rich-text documents. Rich-text means a text document that can also contain various media elements like images, animations, links, tables and other. JavaScript language is used to bring some dynamics to static nature of HTML pages. And finally, there is CSS that makes web design and markup a bit easier.

These are three major technologies that all websites use and these technologies are perfect for a sites that represent itself mainly as linked rich-text documents. An additional benefit of text-based web technology is the capability to be intrinsically indexable by search engines. As the Web grew, something more flexible and faster than HTML became a critical requirement for many websites. If we take a close look at these websites, then we realize that they are more applications than sites. And their requirements differ from the ones for websites. They need a complex and responsive GUI, they require to work as much performance as possible, they don't need to be search engine indexable.

Here are some examples of these web applications:

- Social networks and other sites that provide their content for authenticated users only
- Corporate, public and other portals
- Video, photo and other media hosting
- Site admin and personal sections
- Email clients
- Games
- Development tools, task managers, source controls
- Exchanges, online-banking, accounting, POS
- Services control panels like web hosting panels
- Online office tools
- Maps
- Streaming services
- Various online tools like converters, editors other tools
- Many, many others

Several technologies provide solutions. Most known is the Flash. While the Flash allows creating things that are impossible under pure HTML and JavaScript, this is proprietary product and still something alien to the web. The Internet still needs some technology that will bring power and flexibility needed for web applications. Moreover, security requirements hinder the progress in this area. Main web technologies do not allow running native code and do not give access to critical parts of a host OS by design. These restrictions prevent almost all malicious impact of a website code and make a web surfing much safer than installing software locally.

The Problems of Local Platforms

Under Local Platforms we understand locally installed desktop or mobile operating systems like Windows, Linux, MacOS, iOS, Android and similar. Local Applications, in turn, are software that installed and run under Local Platforms. Software vendors want their products to run on as many platforms as possible. However, cross-platform software development is difficult. Platform differences and the look and feel of different GUIs frequently cause a dilemma:

EITHER

A quick and cheap development using some cross-platform framework with bad user experience

OR

Long-term and expensive mostly separate development for each platform with the best user experience

On the other side, ARE there any significant differences between local operating systems available on the market? All systems come with a very similar set of functions: multitasking, 2D window GUI, security, networking and some other. So we can see that cross-platform development difficulties caused by the platforms differences are not inevitable. If we would introduce a standard for cross-platform API and unified OS-independent GUI standard, then we would be free from OS specifics and differences.

Another problem with current platforms is that their GUIs do not fit properly in existing and future 3D display devices and it would be nearly impossible to effectively adapt legacy 2D window user interfaces to the 3D environment provided by 3D gears. Also, classic GUI has some problems with utilizing screen space of wide and ultra-wide display monitors - maximizing makes windows too big, and it’s also hard to find some good solution as it very likely will require changing applications’ code.

Existing solutions

There are several technologies exist that address major issues of local and web platforms. Below is the comparison table of its pros and cons.

	Industry Standard	Open-source	Cross-platform support	Local System Access API	UI Integration (mostly)	Performance
HTML5	Yes	Yes	Yes	No	Inside a browser	Low
Flash (Adobe)	No	No	Major platforms	No	Inside a browser	Moderate
ActiveX (Microsoft)	No	No	No	Yes	Inside a browser	High
Silverlight (Microsoft)	No	No	Minimal	Partial	Inside a browser	Moderate
AIR (Adobe)	No	No	Major platforms	Partial	Local OS GUI	Moderate

	Maximum Acceptable Level of UI complexity	Malware Resistance	VR Support (3D UI Environment)	Deprecated
HTML5	Low	High	No	No
Flash (Adobe)	Moderate	High	No	Yes, for browsers
ActiveX (Microsoft)	Moderate	Low	No	Yes
Silverlight (Microsoft)	Moderate	High	No	Yes
AIR (Adobe)	Moderate	High	No	No

Requirements

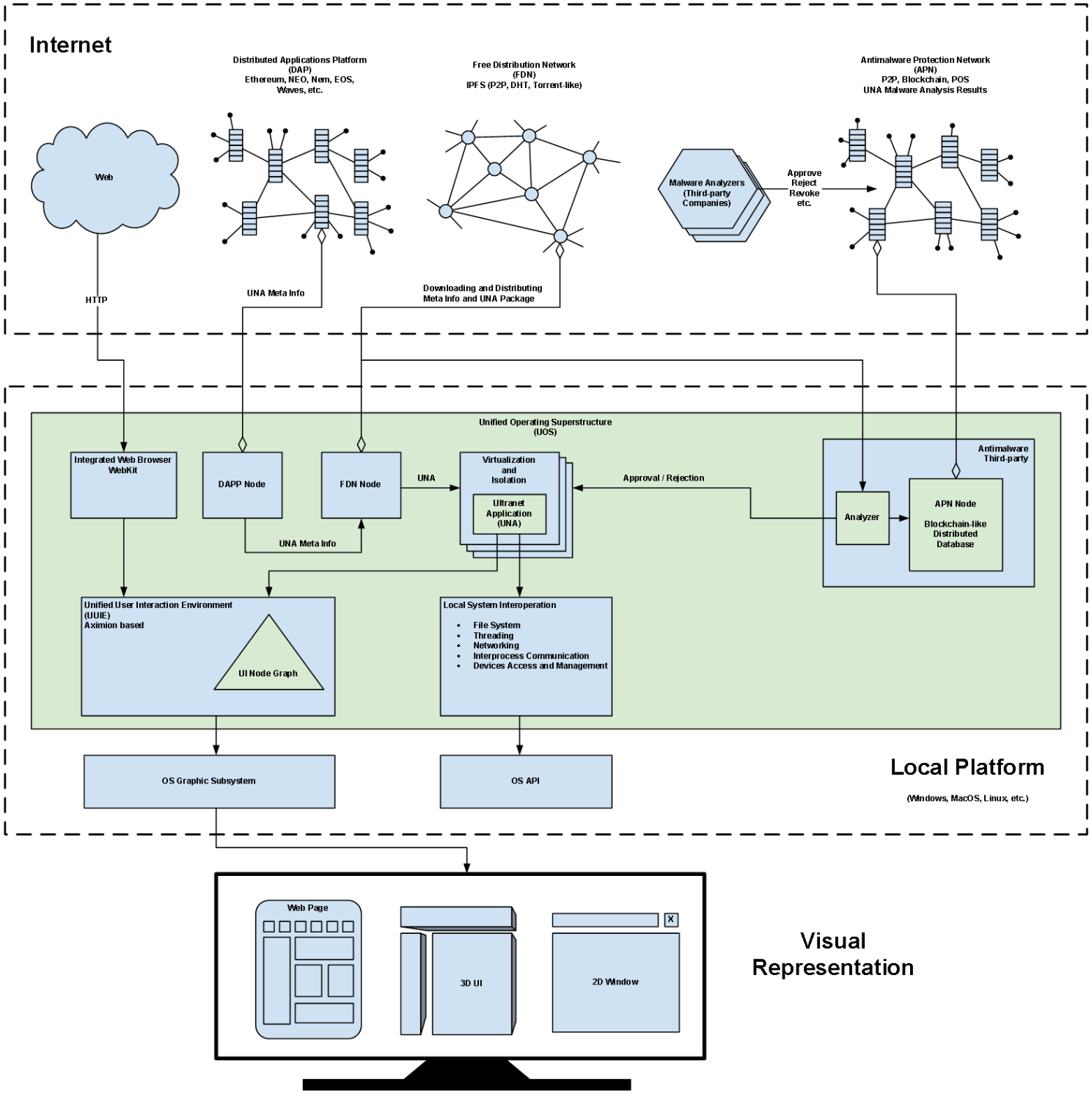
The following is a summary of all requirements for a new technology that will solve the problems described above.

- Make it possible to replace application-like HTML websites with a new technology that brings a full power of local applications.
- Give a platform that allows building OS-independent applications which are accessible like web pages but appear and function like conventional local applications.
- Provide free distribution without any censorship but with a mechanism for malware-spreading prevention.
- Minimize the impact of malicious code on a local system, other applications, and user data.
- Make local OS choice unimportant for users. Users will not need to choose between Windows, MacOS or Linux anymore.
- The new platform should not require installation and uninstallation. Applications should be callable directly through the Internet using well-known URL navigation.
- Provide a unified UI concept with the same look and feel across all existing and future platforms and the ability to display any UI graphics. It should provide a new environment that is designed to support regular and ultra-wide(and ultra-big) monitors, 3D display devices, VR gears, and ideally, mobile devices. It should also provide integrated support of a web browsing

The Solution

We propose the technology that meets all these requirements - Ultranet, the next step in the Internet evolution

(Click to enlarge)



The following are the major components of Ultranet technology:

FDN - Free Distribution Network provides free and decentralized distribution of Ultranet Applications and implemented on top of IPFS.

APN - Antimalware Protection Network consists of several components that minimize malicious code impact on the whole Ultranet ecosystem. It is a distributed blockchain database of approval, rejection and other records and also a real-time signaling network for the rapid broadcasting of recently identified threats.

UOS - Unified Operation Superstructure is a cross-platform local system layer that provides safe execution environment and exposes standard API for Ultranet Applications.

UNA - Ultranet Applications is a new class of platform-independent Internet applications that are distributed via FDN, run under UOS and must comply with the UNA Open Standard which specifications cover API, building, packaging and deployment.

DAP – third-party blockchain smart contract platform which is used to host UNA Meta Info management program. The primary purpose of this program is to get different Meta Info from the FDN using a single global unique address of UNA.

UUIE - Unified User Interaction Environment evolved from the Aximion project. Aximion is a new unified graphical user interface technology that inherits important aspects of both Local and Web platforms and is capable of rendering any graphics with a full performance of local hardware. It provides a single environment for a classic Windows UI, web UI, and for future 3D user interfaces. All implementations must strictly comply with a special technical standard to guarantee the same look-and-feel across all platforms.

Additional benefits that come with Ultranet:

- It's completely free to use for all users. Only publishers have to pay a small price and only for increasing a level of trust to their products
- Utilizing various DAPs as application server-side (back-end) and Ultranet as application client-side (front-end) gives rise to an ecosystem that is invulnerable for Distributed DOS attacks. Moreover, Ultranet proposes a special protocol which is capable to actively shutdown any botnet that participates in such kind of attacks.
- These is potential to create applications that work on various incompatible Linux distributions, which would give a second breath to open-source platforms
- It's not possible to ban particular application or its distribution as it does not depend on specific URL or IP address. Even publishers (without special measures) have no power to prevent users from using older version if users are not satisfied with the latest ones.

Free Distribution Network

FDN provided participation in a distribution of applications and implemented using Distributed Hash Table (DHT) technology on top of IPFS or as a separate network. Anyone can publish applications, and anyone can download and seed the applications. No censorship is possible. It can distribute not only stand-alone applications but also component dependencies. Downloading of these components and the main application is automatic.

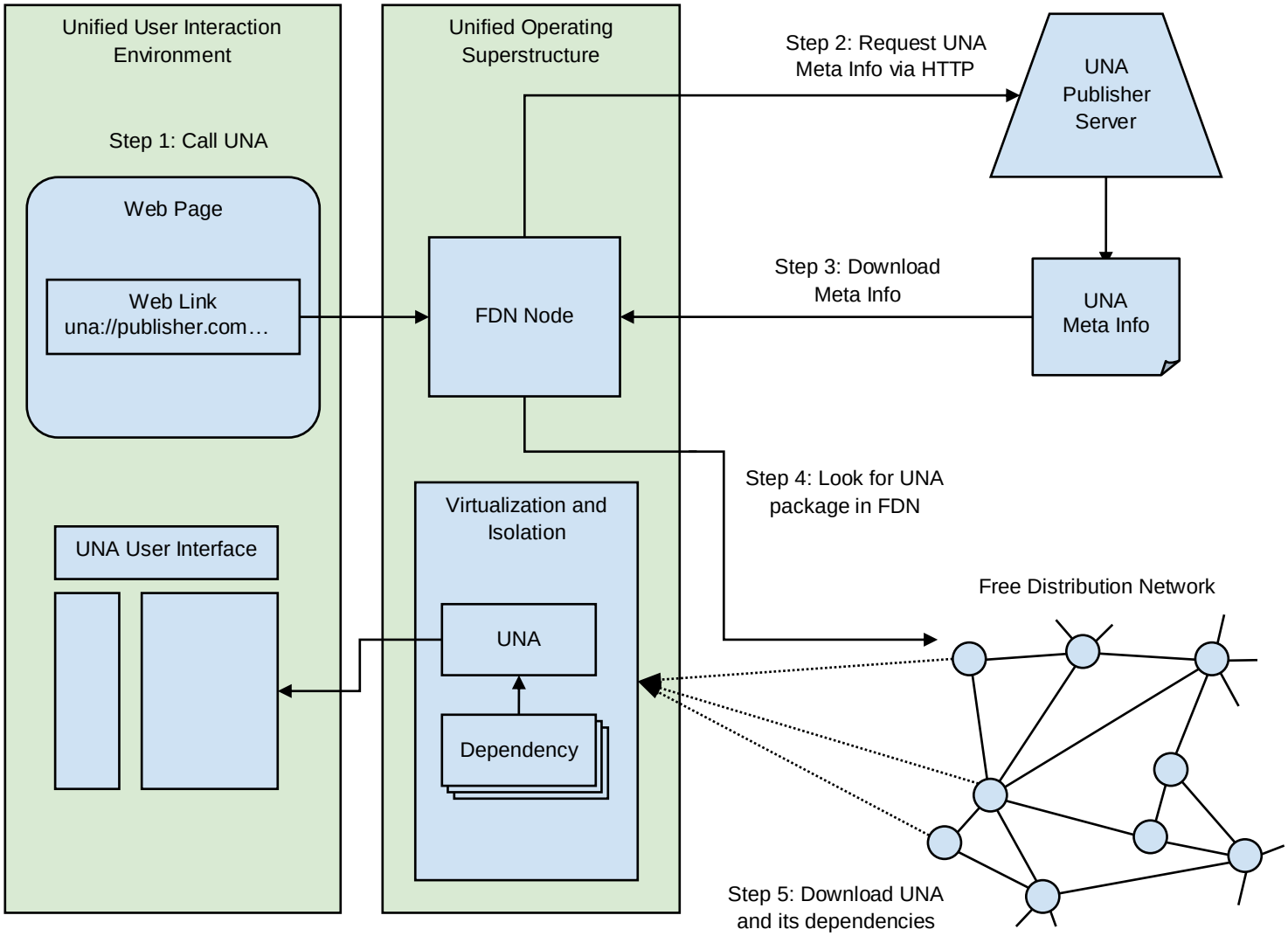
To gain additional flexibility, UNA Meta Info is used as a mediator to provide the following useful features:

- Basic description of UNA to retrieve (title, company, version)
- Application files and resources list (hashes)
- Dependencies (Meta Info hashes) required running UNA
- Visual stub, this may be an image or even a very simple 3D Model which is used to show in the UI Environment during a download of UNA components
- Publisher Digital Signature

There are two ways to publish UNA Meta Info. The first is by using a publisher web server. The second is by using distributed applications running under an existing DAPP network (Ethereum by default) – this is fully decentralized approach with its many benefits.

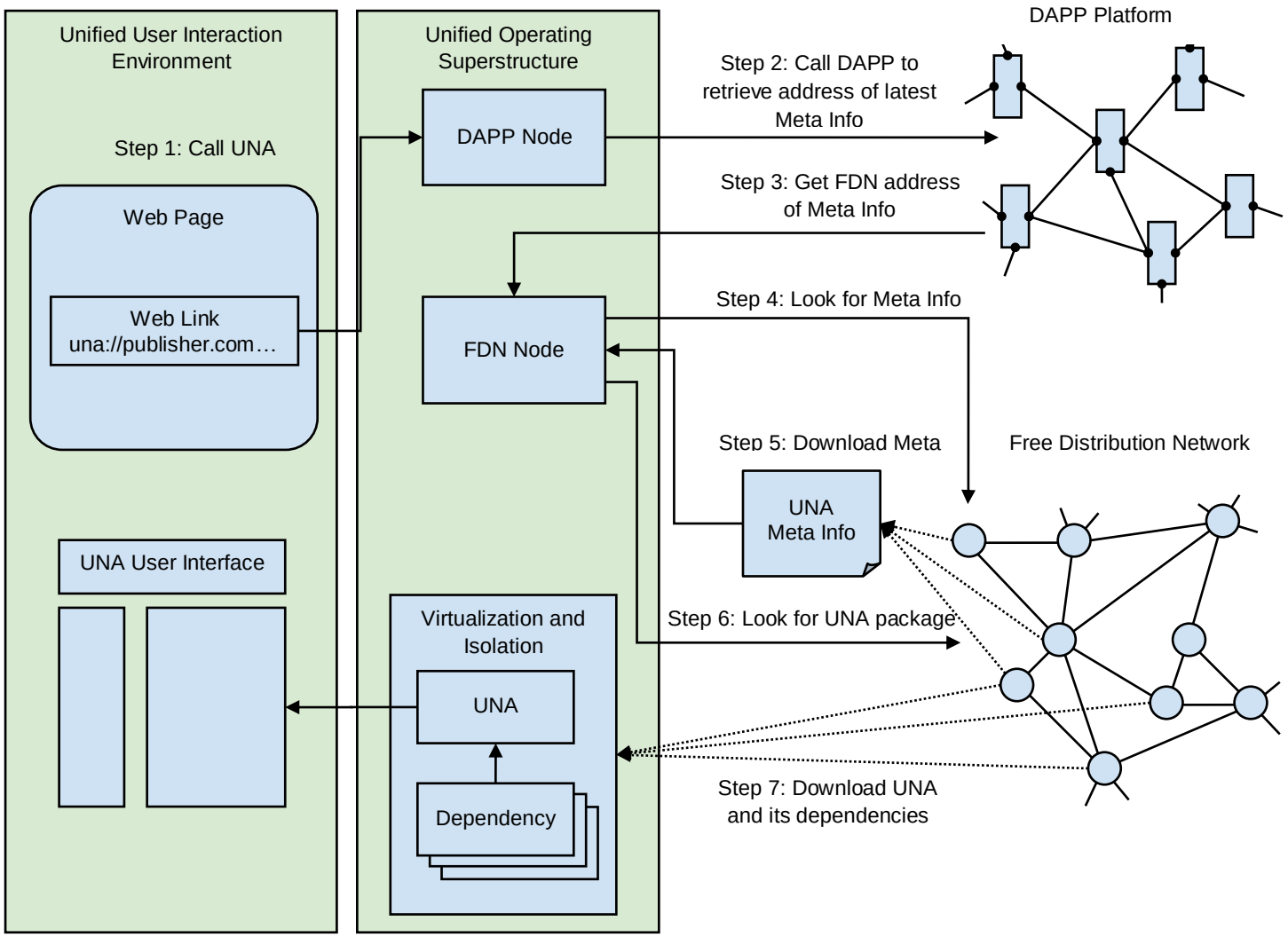
In the first case, when the user requests to run UNA the system performs the following steps:

1. The user clicks on the link.
2. If the link protocol identifier is “una“, then the system calls the UNA URL handler.
3. The UNA URL handler does an HTTP GET Request using a specified URL path to a UNA publishing server.
4. In response to this request, the server returns UNA Meta Info in a standard format.
5. The system reads Meta Info and shows a visual stub in the UI environment with download progress notification.
   Simultaneously, the system reads the application and its dependencies hashes and sends requests to the Free Distribution Network looking for peers from which to download UNA and shared components.
   Simultaneously, the system checks all hashes for anti-malware approvals in the APN database (explained in the next section)
6. Once the application download is complete, its integrity and high approval level are either confirmed or manually overridden by the user it then runs in default configuration in a virtualized and isolated execution environment.

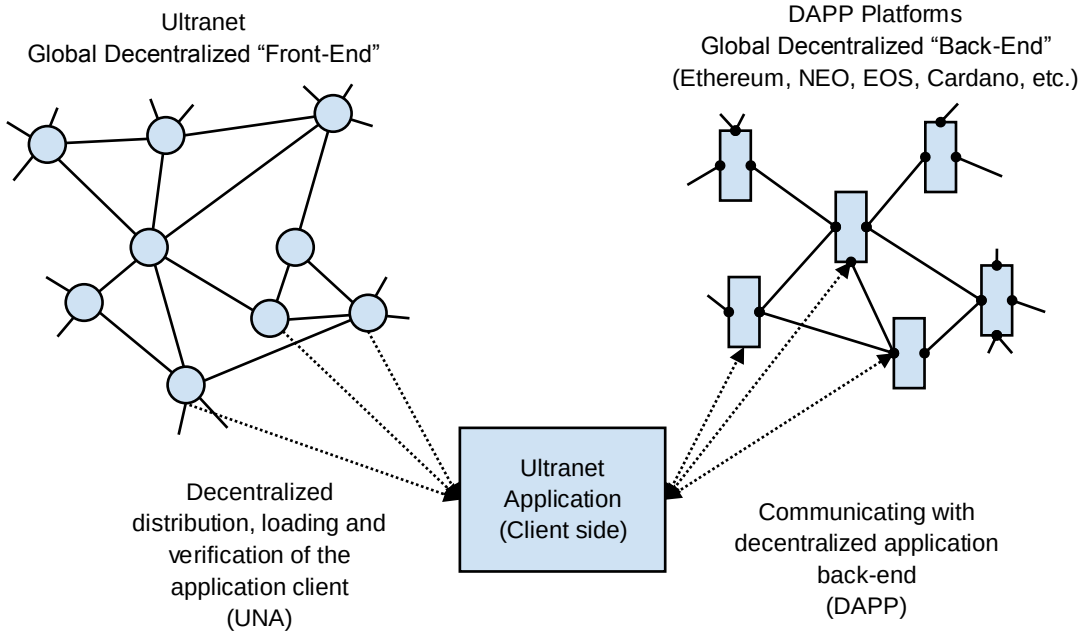


If the decentralized approach is used then when user requests to run UNA then the system do the following steps:

1. The user clicks on the link.
2. If the link protocol identifier is “una“ and URL is a DAPP address, then UOS uses a DAPP node to obtain the Meta Info address.
3. The DAPP Node call uses a special method of decentralized application to retrieve FDN address of the latest (or specific) Meta Info.
4. With the Meta Info address, the system can find the Meta Info file in the FDN.
5. Following downloading of Meta Info, the remaining steps are the same as for the first approach



Together with DAPP platforms, which would act as the back-end for UNA, this approach revolutionizes the way of how Internet works. There are no vulnerable centralized nodes, like web servers, anymore.



This leads to impossibility of DDOS attacks and censorship as all components do not rely on IP addresses or DNS records and so there are no physical targets to attack or block. The scaling is also not an issue for the front-end because FDN component of Ultranet is built on top of DHT (IPFS) technology that works similar to the Torrent networks which in turn are purposely designed to handle load at any scale. As for back-end, existing blockchain-based DAPP platforms have already shown a potential to address this kind of issues and there is a strong belief that they will be resolved in the very near future.

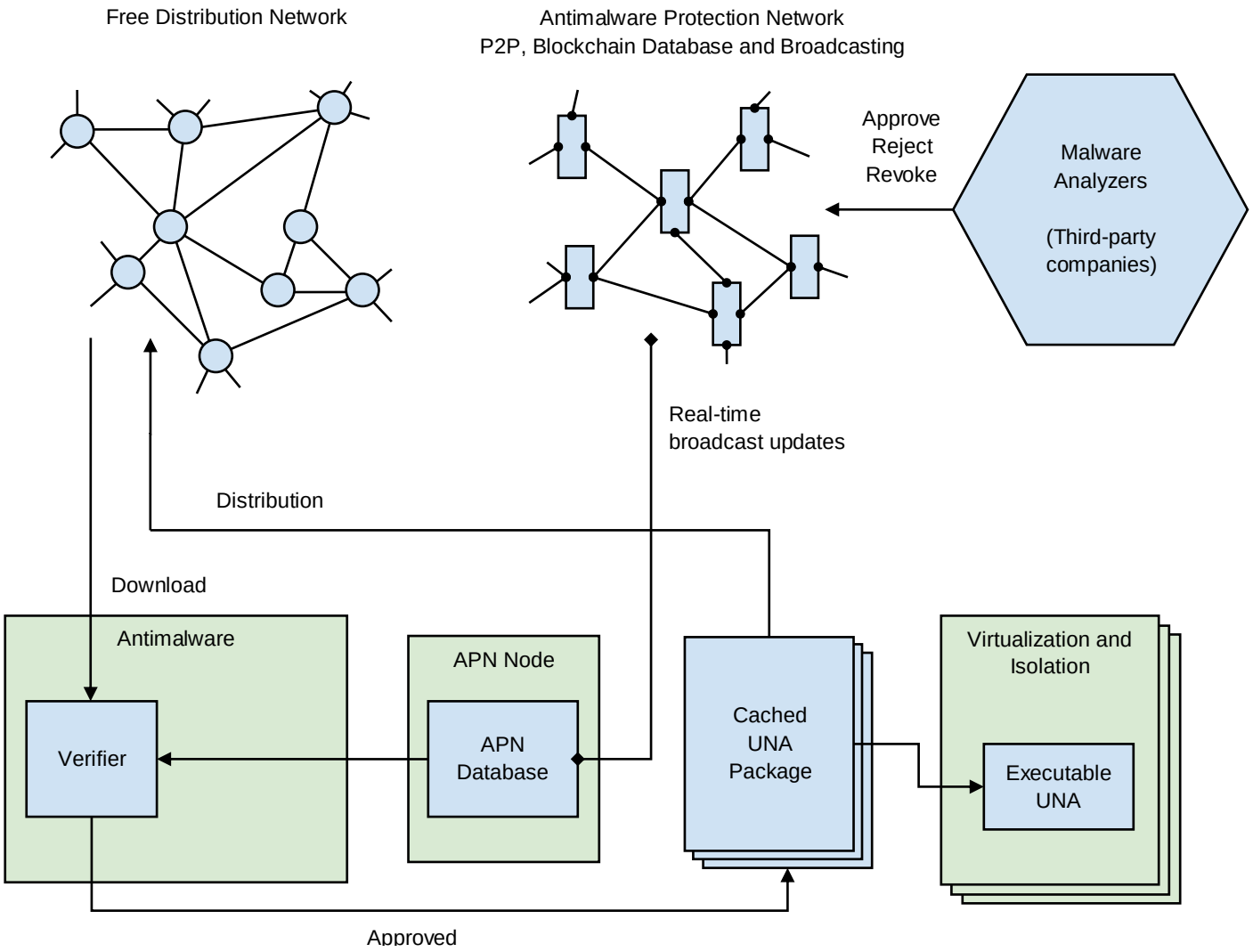
The technology provides an embedded mechanism to update applications so no special development efforts required to support auto-updating.

Using cloud (private, IPFS, Sia, Swarm, Storj or other) data storage, the technology enables the creation of semi-thin clients. This means that both applications and its data are stored remotely in a different distributed networks or clouds. Once a user loads his profile, all required applications and data download to a local device for followup.

Antimalware Protection Network

APN stands for a fast check just requested and cached UNAs for malicious code. It is a distributed database of approved application records, revoked approvals and a real-time signaling network for the rapid broadcasting of recently identified treats. Each UNA in this database is identified via its hash. A presence of approvals in the APN increases the level of trust for UNA. The more records issued for a particular application, the higher the trust level of the application. If no records are present, an unknown trust level occurs. This, in turn, means there is no pre-verification performed, and use of the application is risky.

The distributed database is built as a blockchain database with a hybrid Proof-of-Stake consensus mechanism.



To get approval, the UNA publisher asks the Malware Analyzer to check an application for malicious code and if nothing suspicious is found then MA creates a record with an application identifier (hash) and signs it with a company signature and initiates broadcasting. The signature ensures that the received broadcasted record is not spam and that it is issued by the known Malware Analyzer and stored in the distributed database. The list of trusted anti-malware vendors may vary depending on particular requirements. The Ultranet Organisation always provides the default list of certified Analyzers.

If the previously approved application is later found to be malicious, then MA can issue an approval revocation record. This type of record revokes previously issued approval for this application, and immediate broadcasting is initiated to let other nodes know about that change.

The Antimalware component is a standard or third-party software that acts as a regular antivirus and APN Node. It searches for MA approvals each time a new application is downloaded in a copy of the blockchain-like database of application approval/revocation records and participates in record broadcasting.

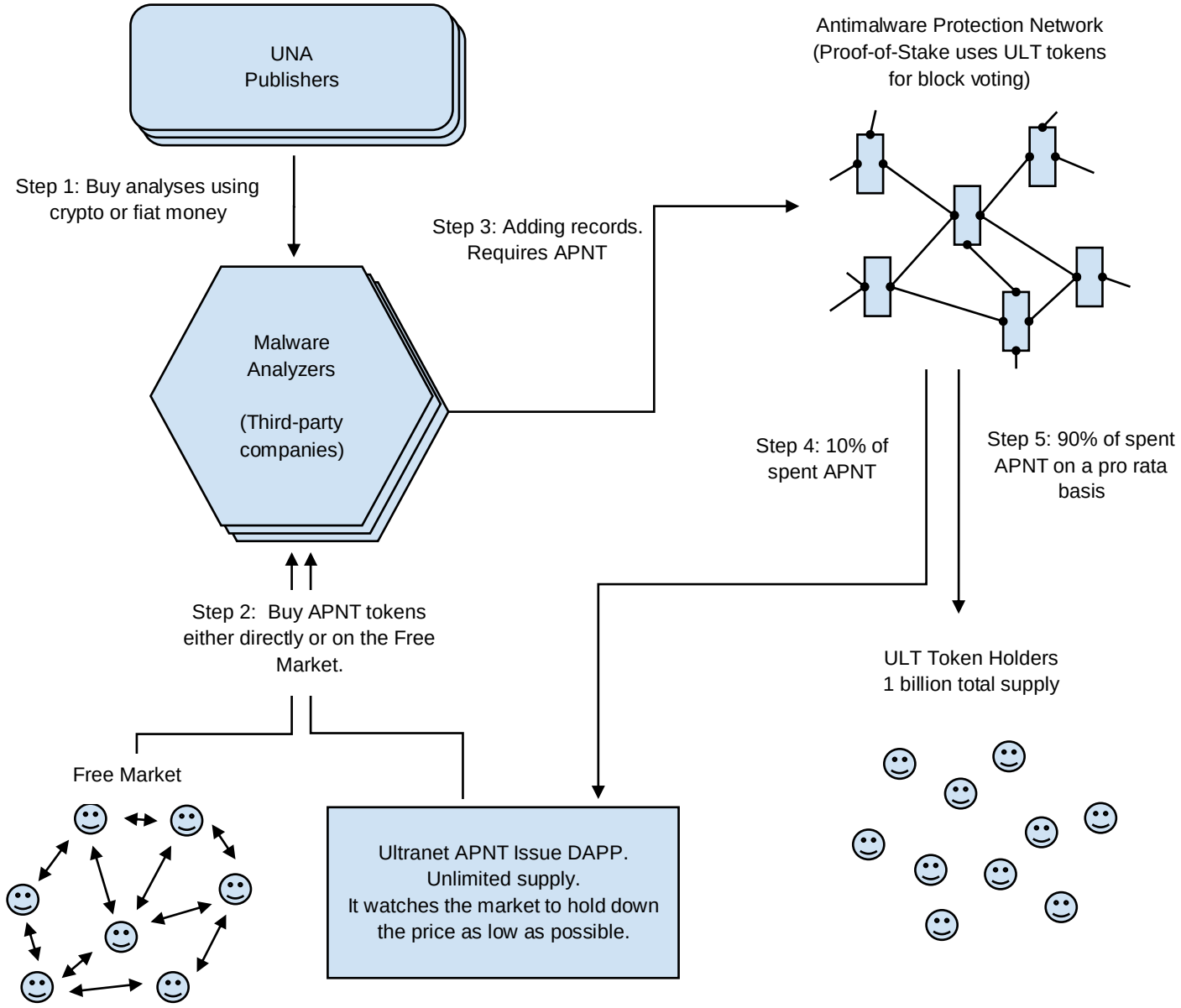
In the future, due to expected progress in AI, this network may transform into a distributed AI built as a decentralized application (DAPP) using Ethereum, NEO, EOS or other appropriate technology. This AI would collect treat-analytics data from all network nodes and produce application ratings which in turn would be used to approve or prevent application execution.

APNT and ULT Tokens

The general philosophy and key principles:

- The technology requires initial investments.
- The investors should benefit from their investments.
- The investors should be interested in development and the infrastructure support of the technology.
- The cost of securing approvals should be as low as possible to make the technology as affordable as possible for all UNA publishers.
- The future success of the technology should reward the Investors but should not affect affordability for the publishers.
- The publishing affordability should not depend on market speculations.
- The Ultranet Organization should have adequate income to support further research and development.

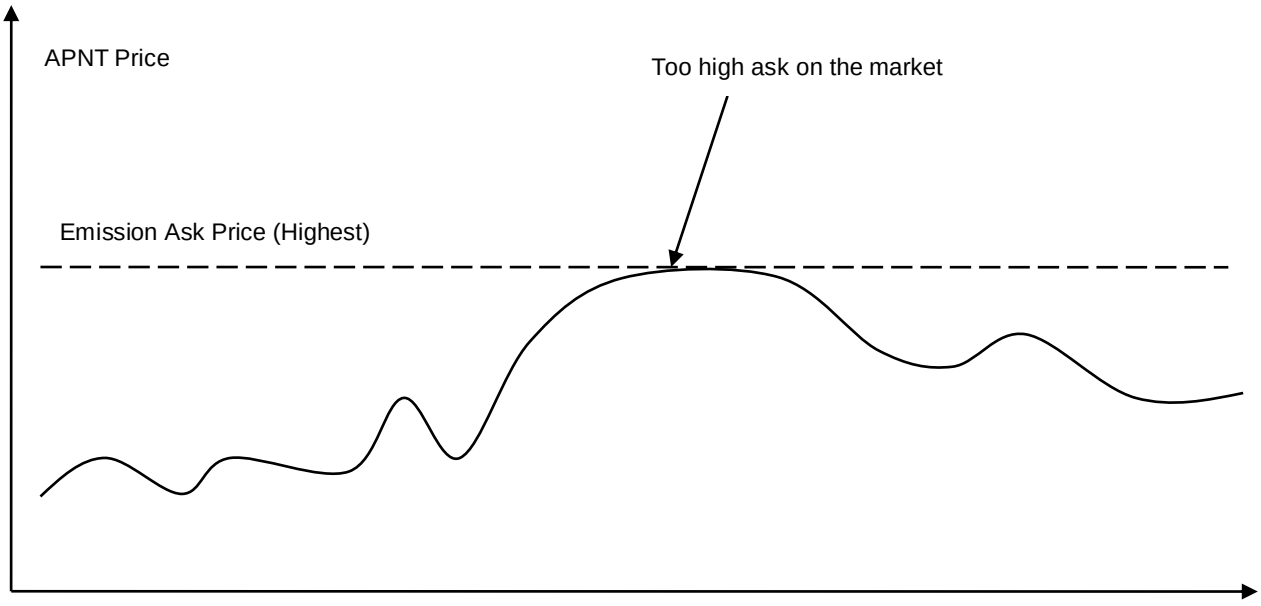
To meet all above criteria, we introduce two different tokens - ULT and APNT



APNT tokens allow Malware Analyzers to add approval, revocation and other records to the APN blockchain database. Another important function of this token is the protection of network from the spam. APNT tokens can be purchased either on the free market or directly in the Ultranet Organization. The Ultranet APNT Issuer is implemented as DAPP to reach maximum possible availability. To make the approval process affordable for any publisher, APNT token has unlimited supply and the Ultranet Organization watches the market to hold down its price as low as possible.

ULT is infrastructure supporting and governance token. It will be distributed via a token sale and have a maximum supply of 1 billion tokens. When ICO is finished this token can be purchased on the free market only.

The process of UNA approval begins when the Publisher asks the Malware Analyzer to verify its application and add approval to the APN database if no threats found. The publisher pays some amount of crypto or fiat money to the Analyzers for that service. Analyzers use funds to purchase APNT tokens either on the free market or directly in the Ultranet Organization. After purchasing tokens, Analyzers can use it to add records in the ratio one token to one record. Following the creation of a block of records in the APN blockchain, 10% of collected APNT tokens are sent to The Ultranet Organization account to support further development and to enable some contraction of token supply to not allow the price to fall too much. Another 90% is distributed among accounts with locked ULT tokens on a pro-rata basis so that all Ultranet supporters have continuous income proportionally to the amount of ULT they have locked in order to participate in the APN Proof-of-Stake consensus.



Analysis Wholesalers help to obtain approvals from many Analyzers with only a single request so that the Publishers don't need to deal with each Analyzer Company individually.

The Ultranet Organization is unable to discriminate against any publisher as it only sells anonymous tokens and any entity may purchase it for any purpose. An alternative transaction approach of a single token could lead to a POS problem of permanent nobility. However, with our separated tokens approach, participation in a block confirmation does not change proportions of stakeholder deposits, i.e., big deposits do not get bigger with time. The distribution and circulation of APNT is implemented via DAPP developed and governed by the Ultranet Organization. The APN nodes owners have the power to reject any protocol changes by not switching to a new version of APN DAPP if it disrupts the principles of decentralization.

Unified Operating Superstructure

UOS provides safe execution and exposes local system API for Ultranet Applications. A virtualized and isolated execution environment is another measure to minimize the impact of malicious software or incorrect behaviour of a running application. This security measure is similar to how applications work on iOS and Android. In this environment each application could be in one of the following three isolation states:

- Isolated - execution is allowed within a virtual environment
- System - execution is allowed, and virtualization is disabled so application can access any resources it needs even a data of other applications
- Blocked - low trust level is received from the APN or Antimalware found its code suspicious and confirmed by the user to prevent application from execution

The isolated state application has access only to the virtual file system. It has a single root, and other various sources can be mounted to it. At the OS level the file structure is going to be organized in the following way:

Applications
    Application [N]
        User [N]
            Local
            Global
                View[N]
            SystemLocal
            SystemGlobal
        <binaries and other app-specific files and folders>

- Global and SystemGlobal directories are used to store personal user data like documents, and this data may be synchronized with external cloud or distributed storage networks.
- SystemLocal and SystemGlobal directories may contain special files which are used to make application-specific changes to a local or global configuration using file merger method.
- View folders stand for storing view and layout settings on a per-device basis. It's used to have separate environment layout settings for 2D monitors, 3D gears, and mobile devices.

Unified Operating Superstructure provides access to a host OS via open standard cross-platform API Framework. Both native (C/C++) and LLVM API's are provided. Particular LLVM implementation is an extremely important task and subject to further research. MSIL, WASM are major candidates.

UNA developers can choose a type of code they want to distribute. The easiest way is to publish cross-platform code in the form of LLVM binaries. If a developer wants to provide the highest performance, then he can create an application using a cross-platform C/C++ approach and publish binaries for all major platforms. In the second case, when the application is requested to run, appropriate binaries will be downloaded and used according to the type of local platform.

No longer is there an installation process. To speed up loading of UNA, it is broken down into three stages. When the application is initially requested, the core and minimum required set of components are downloaded and default configuration is used to launch the UNA application. After application is started, standard components are downloaded in the background. And finally when the user needs some extensions, UNA can download and

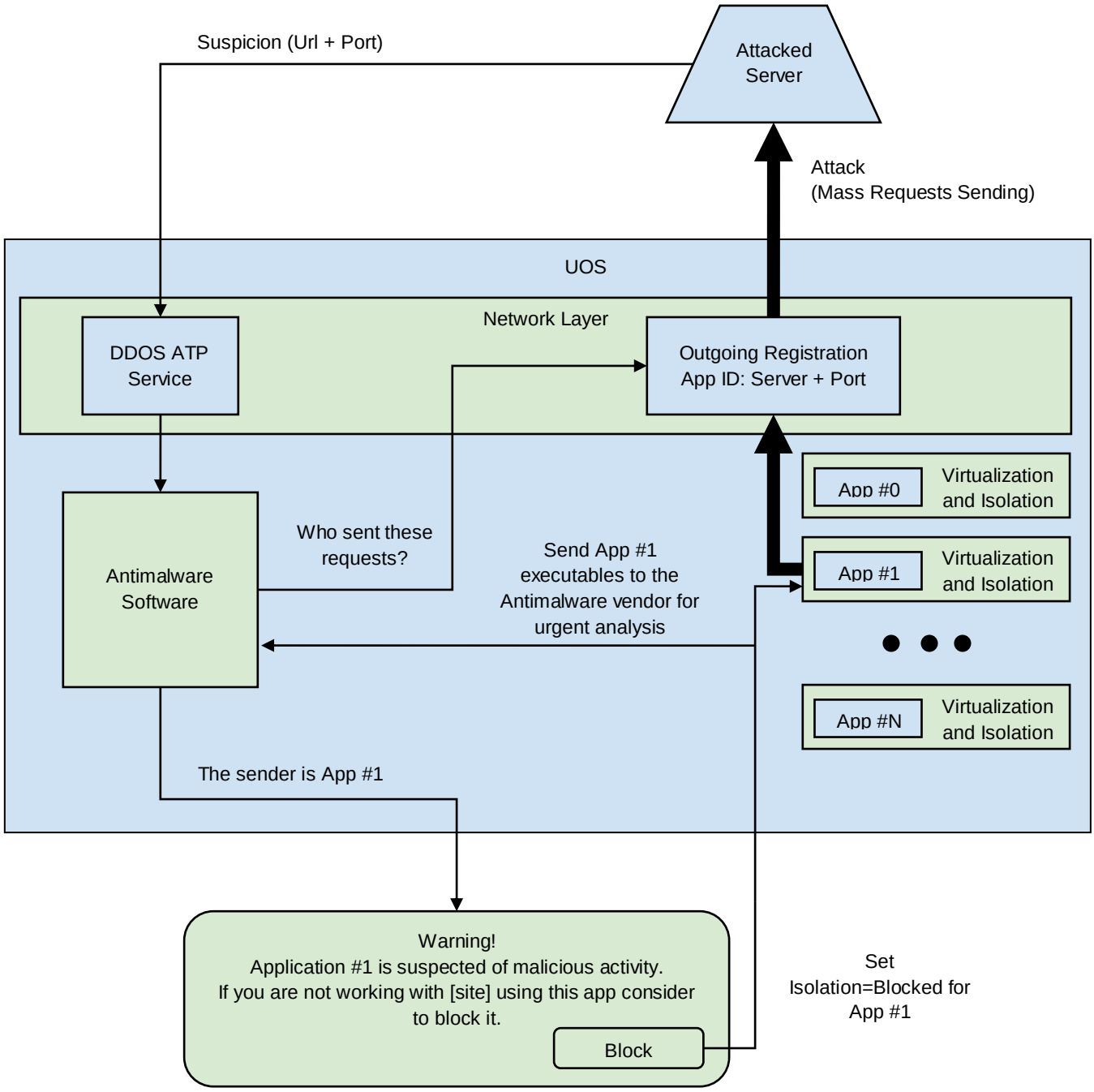
use them separately. So this means that UNA implements a component design a bit different way than traditional local applications.

Unified Operating Superstructure provides integrated support for web browsing. In this scenario, some well-known open source web engines like WebKit is used.

Distributed Denial of Service Active Termination Protocol (DDOS ATP)

UOS provides functionality to actively react when a local system is infected and participates in a DDOS attack. This process requires a server side to support this mechanism.

In case of DDOS, the attacked server analyzes incoming traffic and sends a special signal (SUSPICION) back to each node that sends too many requests. On the local side, UOS stores application-to-destination mapping information and can use it to identify the suspicious application, warn the user about its malicious activity and then block it if needed.



Once UOS identifies an infected application, it then asks Antimalware to send it to its vendor for urgent analysis. Then various anti-malware companies identify malicious code signatures, quickly update its virus databases and add rejection records to APN database or revoke approvals if they issued before.

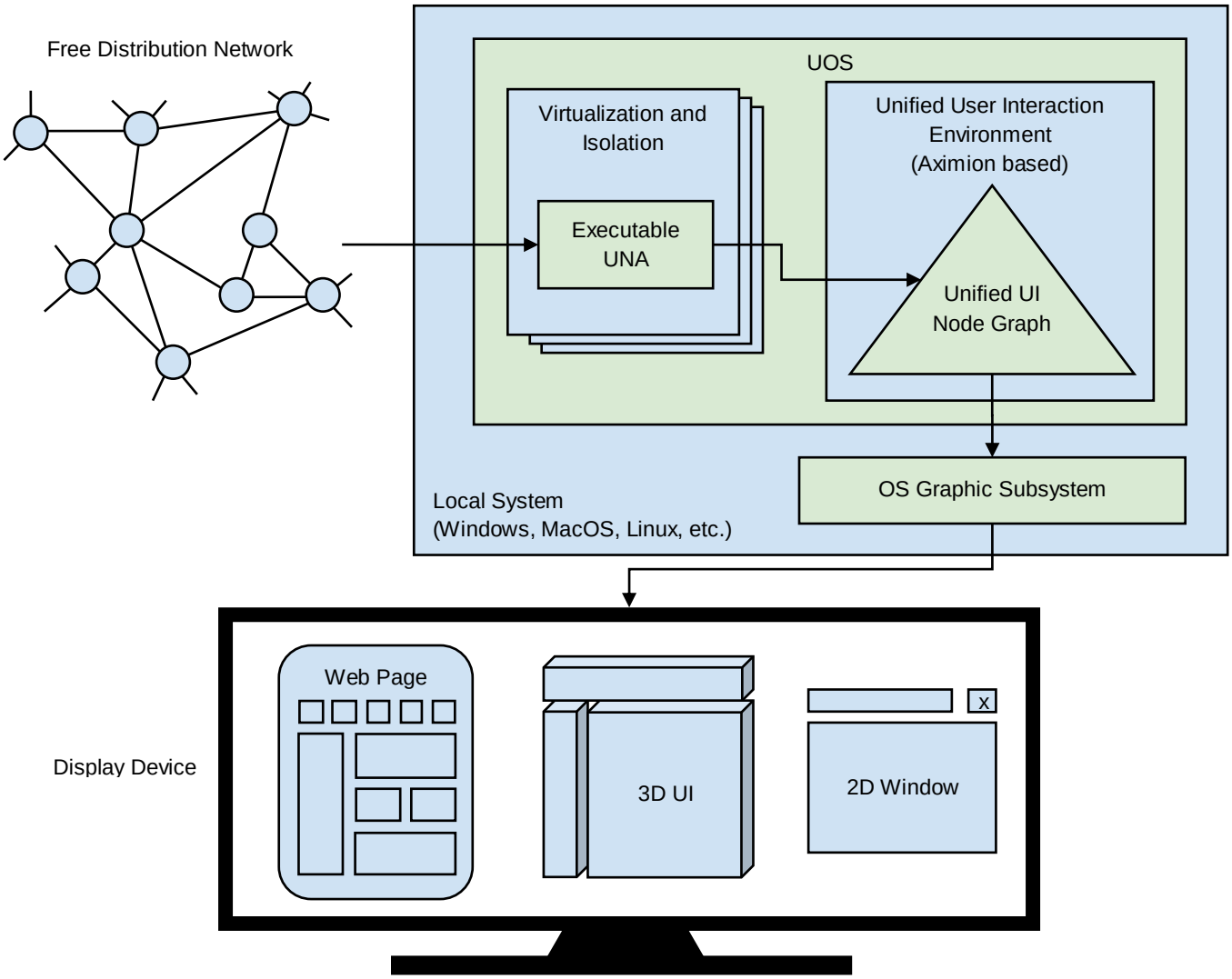
With this technology, an attacked server has the weapon not only to stop the DDOS attack but to shut down a botnet permanently.

Unified User Interaction Environment

The major ideas behind UUIE:

- The user does not run programs but instead opens applications from the Internet and constructs a unique environment
- The environment is independent of a particular physical device
- The technology can render any user interfaces, not only regular windows or web pages but also any possible 3D UI's in a single visual environment
- Support for regular display devices and utilize capabilities of modern and future VR gears
- Any object has a unique global address in the form of URL to support references

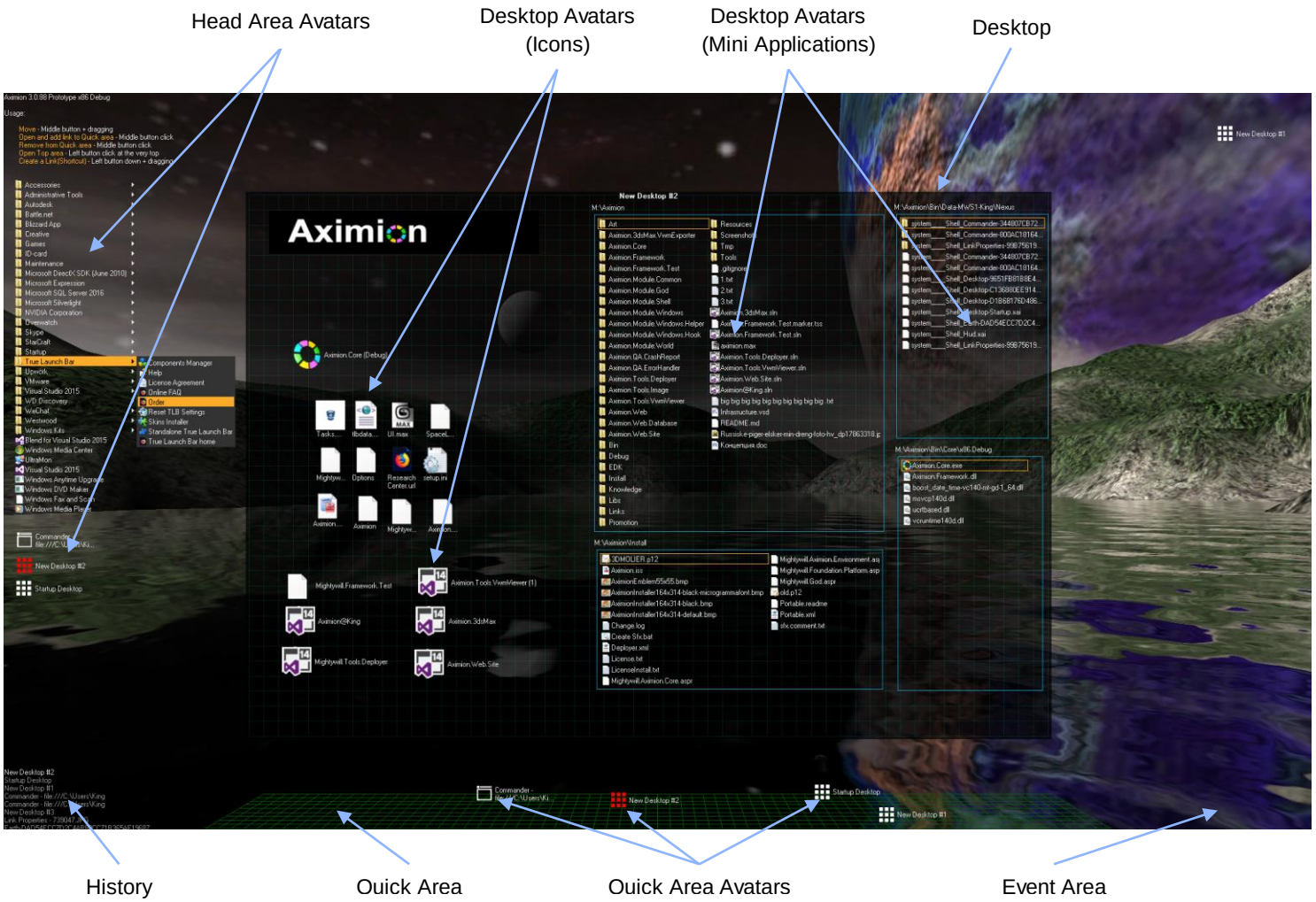
The concept is based on the Aximion project. The mission of Aximion is to do the next step in UI evolution by providing a new visual experience that inherits important features of classic windows GUI and web-like nature of Aximion objects with the capabilities of modern and future display devices and VR gears. The special open standard will be introduced to guarantee the same look-and-feel across all OS platforms. The development of this standard is still at a very early stage, but it will be something that looks like some hybrid of HTML and OpenGL.



Unlike any usual OS's in the Aximion, all root UI nodes have a global persistent unique address in the form of an URL. This gives UI objects capabilities intrinsic for web resources, like links, history, etc. and makes its life-cycle different from the one used in regular operating systems. With such ability, it's possible globally refer to any object using the semantics of URL. When a user creates something, it always has predefined or random unique address assigned to it, and all data related to that instance is kept until at least one reference to this object exists. So having these abilities user can now have links to any objects exposed by UNA the same way as he can for files, web pages, etc.

At the shell level, several UI concepts introduced in the Aximion to help users personalize and organize their environments.

The Aximion rethinks the idea of the Desktop. In the Aximion Desktop is a place where different aspects of a single project/context/interest can be grouped into a single visual entity. For example, if you have some project you may want to gather all related files, links, remote and other resources in one place and organize them accordingly to your preferences. The Aximion Desktops look similar to regular desktops but contain special elements - Avatars. Each type of Avatar represents some aspect of your particular project/context/interest. Their important feature is to change their look accordingly to its importance for the user or to other needs and this look may vary from simple icons to mini-applications with rich functionality. The user can create as many desktops as he needs.



Any application UI can be attached to another. For example, if a top-level UI element of application A is attached to a top-level UI element of application B then if a user opens B, A will show up automatically. The user then can create various application unions which reflect some corresponding contexts. For example, if the user is a software developer then he can create an individual desktop for each of his projects and also create several other desktops that would contain shared tools and resources required for all of his projects. Once any of these project desktops is open all attached desktops with shared tools will appear together with that.

The Quick Area is used to place links to needed objects for the current session, much the same way tabs are used in browsers. Middle button click helps the user to place avatars in the form of a link to that area automatically. They do not disappear when the user closes the object so when needed the object is available for reopening.

All avatars placed to the Head Area are always available for the user and always stay on top of other objects. Using VR gear this area acts as a Heads-up display, i.e., always in a field of view whenever a user's head moves.

History Area this is a history of user navigation. This function similar to browser history and together with Quick Area effectively replaces a standard Desktop Taskbar behaviour.

The Events Area stands for showing events and progress reports, similar to the Windows system tray but with advanced functions. It can show simple notifications about events that require user attention. Also, any user-initiated long time operations should place representation items in this area, so the user has a continuous report about current time-consuming processes.

Stages and Roadmap

Stage 0: ICO — Jun 2018

- ULT Token Sale
- ULT Token Wallet
- Listing on exchanges

Stage 1: Research and Alpha Standardization — Sep 2018

- Basic specifications of UOS API
- Basic specifications of UNA
- Basic specifications of FDN and Publishing
- Basic specifications of UUIE Protocol
- Basic specifications of APN Protocol

Stage 2: Alpha Development — May 2019

- UOS for Windows Platform Alpha
- UOS for MacOS Platform Alpha
- UOS for Linux Platform Alpha
- FDN Test Network
- Publishing DAPP Alpha
- APN Test Network

Stage 3: Alpha Testing and Beta Standardization — Sep 2019

- UOS API Standard Beta
- UNA Standard Beta
- FDN and Publishing Standard Beta
- UUIE Protocol Standard Beta
- APN Protocol Standard Beta

Stage 4: Beta Development and Final Standardization — May 2020

- UOS for Windows Platform Beta
- UOS for MacOS Platform Beta
- UOS for Linux Platform Beta

- UOS API Open Standard
- UNA Open Standard
- FDN and Publishing Open Standard
- UUIE Protocol Open Standard
- APN Protocol Open Standard

Stage 5: Beta Testing Sep 2020

- First Ultranet Applications

Stage 6: Version 1.0 May 2021

- UOS for Windows Platform v 1.0
- UOS for MacOS Platform v 1.0
- UOS for Linux Platform v 1.0
- Main FDN network
- Main APN network

Conclusion

Ultranet is called upon to solve long-standing problems of the web and local platforms and together with that introduces many new innovations to improve and revolutionize user interaction experience. By combining and utilizing the most advanced software technologies available Ultranet shapes the next era of the Internet.