RDN-based

Interaction Layer

(Continuously updated)

https://www.ultranet.org

All information provided is preliminary and subject to further research.

Abstract: RDN-based Interaction Layer (RIL) introduces an open decentralized technology for publishing, distribution, verification and delivery of any kind of software, particularly a new generation of non-Web-based Internet applications that blur the distinction between Web, mobile and desktop platforms. Three major components – RDN, FE and AMPP – cover the publication, delivery and protection aspects of RIL. The peer-to-peer and blockchain components of RIL revolutionize the way in which software is published and delivered to users. A distributed permission-less cryptography-protected database is in turn used to manage software distribution and may be considered as a hybrid of static content web server and DNS. A special protocol is also used as a decentralized verification service for protecting the whole ecosystem from malicious software and other threats.

Disclaimer: *This Technical White Paper is for information purposes only. Ultranet Organization does not guarantee the accuracy of, or the conclusions reached in, this white paper, and this white paper is provided "as is". Ultranet Organization does not make, and expressly disclaims, all representations and warranties, whether express, implied, statutory or otherwise, whatsoever, including, but not limited to (i) warranties of merchantability, fitness for a particular purpose, suitability, usage, title, or non-infringement; (ii) that the contents of this white paper are free from error; and (iii) that such contents will not infringe third-party rights. Ultranet Organization and its affiliates shall have no liability for damages of any kind arising out of the use of, reference to, or reliance on this white paper or any of the content contained herein, even if advised of the possibility of such damages. In no event will Ultranet Organization or its affiliates be liable to any person or entity for any damages, losses, liabilities, costs, or expenses of any kind, whether direct or indirect, consequential, compensatory, incidental, actual, exemplary, punitive, or special for the use of, reference to, or reliance on this white paper or any of the content contained herein, including, without limitation, any loss of business, revenues, profits, data, use, goodwill, or other intangible losses.*

Contents

Glossary

Blockchain a growing list of records, called blocks, which are linked using cryptography

dApp Decentralized application (dapp, Dapp, dApp, or DApp): a computer application that runs on a distributed computing system. DApps have been mostly popularized by distributed ledger technologies (DLTs), specifically Ethereum Blockchain, where dApps are often referred to as smart contracts.

DDOS Distributed denial of service: a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting the services of a host connected to the Internet. In a DDOS attack, the incoming traffic flooding the victim originates from many different sources. This effectively makes it impossible to stop the attack simply by blocking a single source.

GUI Graphical user interface: a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation

HTML Hypertext Markup Language: the standard markup language for creating Web pages and Web applications

HTTP Hypertext Transfer Protocol: an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, wherein hypertext documents include hyperlinks to other resources that the user can easily access.

IPFS InterPlanetary File System: a protocol and network designed to create a content-addressable, peer-to-peer method of storing and sharing hypermedia in a distributed file system.

OS Operating system: system software that manages computer hardware and software resources and provides common services for computer programs

TTI Time to interact: the performance metric that measures the time between a point when the user has requested the application and when the user can start to interact with it.

VR Virtual reality: an experience taking place within simulated and immersive environments that can be similar to, or completely different from, the real world.

The Challenges

A long-standing problem with the Internet is its server-centric architecture. Every Web request to every website is processed by Web servers. Because of this, they often suffer from heavy loads, but more importantly, this makes them ideal targets for censorship, DDOS, and hacker attacks. Decentralized application platforms (dApp) is a promising new technology that has the potential to eliminate the need for centralized servers by transforming the current Web architecture into a homogeneous peer-to-peer network. However, it only deals with the server applications layer and still has to rely on conventional infrastructure for application and UI delivery. There are also other issues related to creating a decentralized Web server:

- High node hardware requirements, as a result of which such networks degenerate into conventional highly centralized clouds.
- Difficulties with efficient load balancing and session synchronization
- Existing P2P solutions work for static content only.

The same problem concerns downloadable applications as well, as they rely on centralized locations from where they are published, downloaded and updated.

The Proposal

To overcome the above challenges and to achieve complete decentralization, we propose to move away from pure Web-based and traditional stand-alone application paradigms and proceed with a more advanced approach. Let's summarize all the requirements of a new technology:

- True decentralization of publishing, distribution and delivery
- Free for users; free or as cheap as possible for publishers
- High scalability
- High resistance to DDOS, censorship and other threats
- Minimizing malicious activity on infrastructure, local system, and user data
- Open platform to avoid any patent infringement cases, profit-oriented evolution strategy, and promotion of proprietary software

We hereby propose the technology that is designed to meet all these requirements. The following are the major concepts and components which are important for understanding this technology:

Data Layer A common term that defines both software and hardware infrastructure, centralized or decentralized, involved in processing and storing some business-specific data. This includes data processing servers, databases, stored procedures, business logic, web services, blockchains, smart-contracts and so on.

Interaction Layer A common term that defines both software components and hardware infrastructure, centralized or decentralized, that provide users a necessary means to interact with Data Layer. This includes client applications, mobile applications, web sites, web servers, web browsers and so on.

Account Represents ownership of an asset or permissions in RDN. An account is actually an abstraction for a cryptographic key which consists of two parts—a public key and a private key. This approach of having two keys is known as public key cryptography which is a widely used cryptographic systems. Account address can be defined in hexadecimal form:

0x0C1961854264BE24957E42D1893AF1D842DB1C56

(minimal form)

Or in fully-qualified URI form using "uaa" scheme:

uaa:0x0C1961854264BE24957E42D1893AF1D842DB1C56

Resource A record in RDN under some domain. Consists of name under domain and value, where value can be any content including file/directory/package/etc. address in FE. Value can be in dynamic or sealed state.

Dynamic means that any account with permissions can anytime change resource value.
Sealed means that nobody can change/clear value or delete a resource itself.

Each resource has its own unique address that can be defined in URI form following “ura” scheme:

scheme:rdn/uo/application/platform/0.0.3
(fully-qualified)

scheme:/uo/application/platform/0.0.3
(with default network)

uo/application/platform/0.0.3
(minimal form, with default network and scheme implied)

Domain A record in RDN that have a similar role as traditional web-domain. I.e., it has unique name and owner(account) and any kind of resources can be placed under it.

RDN Resource Distribution Network – a decentralized permission-less cryptography-based registry and MCV consensus protocol (DLT technology) that are used to manage resource publication. It allows publishers to register domain names, create its resources and publish various content in a decentralized manner.

FE File Exchange Network – a component of RDN, provides free and decentralized storing and distribution of application releases and any other content, implemented as p2p distributed file system simialar to IPFS and BitTirrent protocols.

AMPP Anti-malware Protection Protocol – a component of RDN, a protocol and special RDN functions that serve to minimize the impact of malicious activity on the ecosystem. Its primary purpose is to perform independent anti-malware verification of application releases and publish reports for all users in a trustworthy manner.

UOS Unified Operation Superstructure – a platform-independent local system layer that provides a safe execution environment with a unified UI and exposes a standard API for uApp applications.

uApp UOS Applications – a new class of platform-independent non-Web-based Internet applications that are distributed via RDN and FE, run under UOS, and must comply with the uApp Open Standard, whose specifications cover API, building, packaging, and distibution of UOS applications.

AMI Abstraction-based Morphable Interface – a concept of a unified UI for uApp applications that is able to adapt a platform-independent virtual user environment to desktop, mobile, and VR devices, and to leverage the full power of local hardware. All implementations must strictly comply with a special technical standard to guarantee the same look and feel across all platforms.

Resource Distribution Network

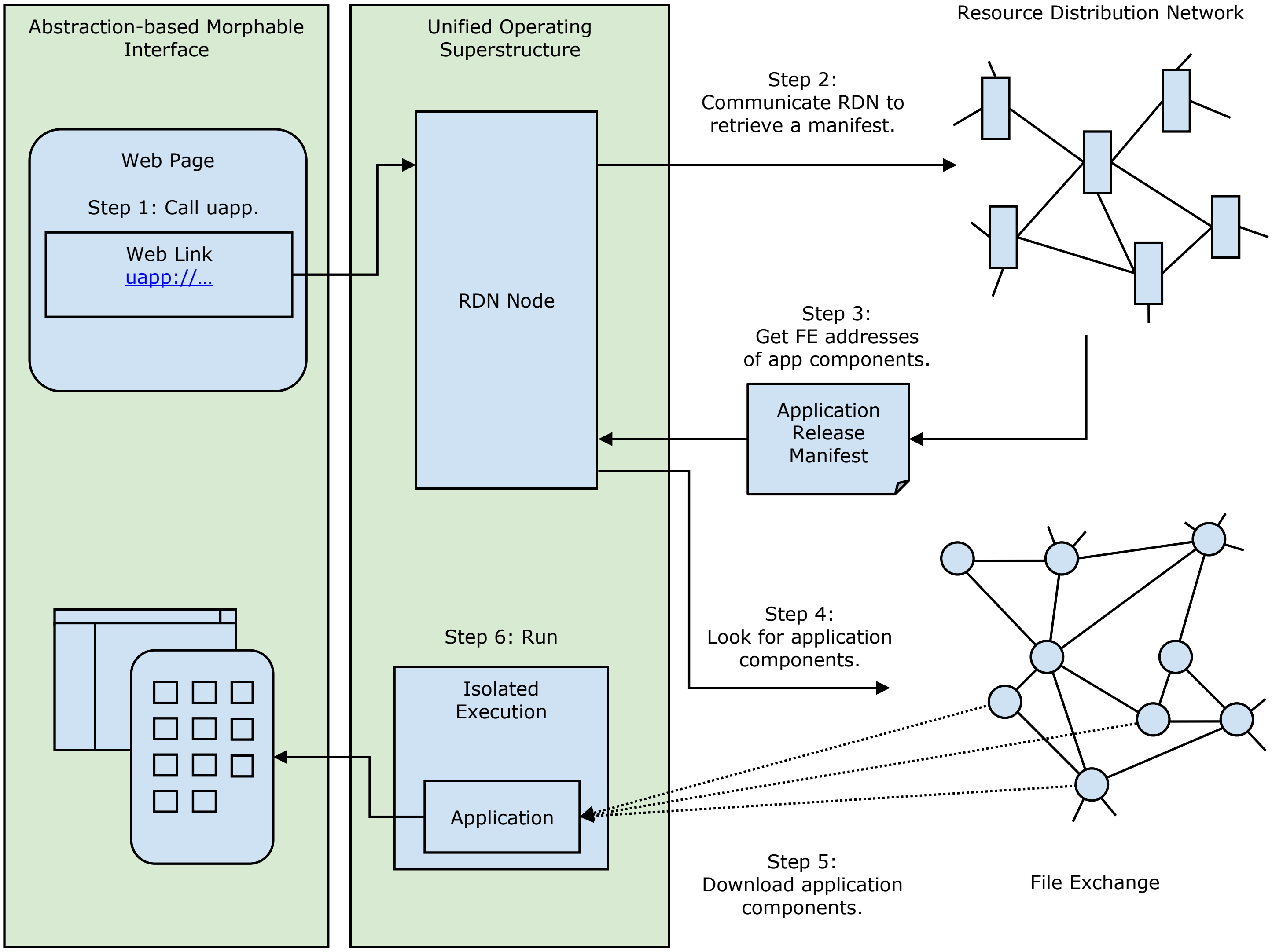
Resource Distribution Network (RDN) – a permission-less cryptography-based distributed registry and MCV consensus protocol that are used to manage resource publication. It allows publishers to register globally unique domain names (similar to Web domains), create resources under it and publish various content in a decentralized manner.

File Exchange (FE) – peer-to-peer protocol that allows participation in the storing and distribution of any data resources and is implemented using Distributed Hash Table (DHT) technology similar to IPFS and BitTorrent. Anyone can publish files there, and anyone can download and seed them. An FE address is not the same as a webpage URL – it strictly identifies a particular file and so cannot point to dynamic content. It is therefore guaranteed that the user always gets exactly what s/he requests. The technology is completely peer-to-peer and highly censorship- and DDOS-resistant. Not only standalone applications can be distributed in the FE, but any kind of resources, such as shared components or assets

When the user requests to run a app via web link, the system performs the following steps:

1. If the link protocol identifier is "uapp:", then UOS connects to a random RDN node to obtain the corresponding uApp manifest.
2. Depending on request parameters, RDN node may return a manifest of the latest or a specific release.
3. The system reads the manifest and shows a visual stub in the UI environment with notification of the download's progress.
   Simultaneously, the system reads identifiers of core and dependency components from the manifest and sends corresponding requests to the FE, looking for peers to download packages from. Anti-malware approval reports are sent as a header before downloading package content itself (explained in the next section).
4. 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.

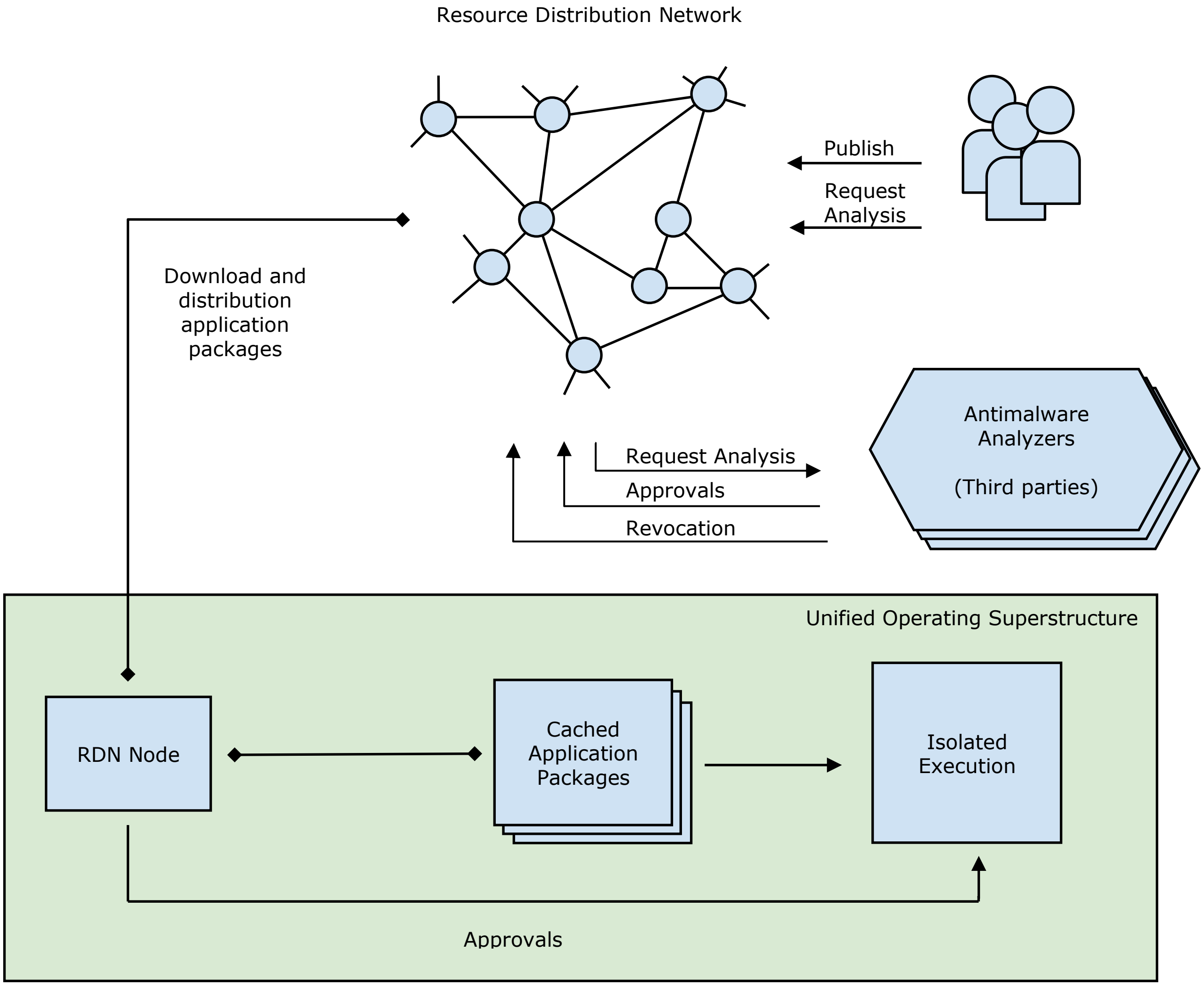
Figure 1. Decentralized Publishing, Distribution and Delivery


Abstraction-based Morphable Interface
Unified Operating Superstructure
Resource Distribution Network
Web Page
Step 1: Call uapp.
Web Link
uapp://...
RDN Node
Step 2: Communicate RDN to retrieve a manifest.
Step 3: Get FE addresses of app components.
Application Release Manifest
Step 4: Look for application components.
Step 6: Run
Isolated Execution
Application
Step 5: Download application components.
File Exchange

Anti-Malware Protection

The Anti-Malware Protection function carries out a fast check of just-requested applications for malicious code. It uses the RDN to store malicious analysis reports and adds to its protocol a real-time signaling for the rapid broadcasting of recently identified treats. Independent third parties perform malware analysis by a publisher request. Since RDN is a permissionless network, anyone can generate such reports, but use system doesn't need to check all of them. Using a list(s) of well-known and trusted Antimalware Analyzers (AAs), the anti-malware subsystem of the UOS checks verification reports of these entities only, ignoring the rest. The list of trusted anti-malware analyzers may vary depending on particular requirements. Ultranet Organization always provides the default list of certified analyzers.

Figure 2. Anti-malware Protection


Resource Distribution Network
Publish
Request Analysis
Download and distribution application packages
Antimalware Analyzers
(Third parties)
Request Analysis
Approvals
Revocation
Unified Operating Superstructure
RDN Node
Cached Application Packages
Isolated Execution
Approvals

Providing of approvals increases the level of trust for a particular application. The more records are issued for a particular application by trusted AAs, the higher overall trust level the application gains. If no approvals are provided, an unknown trust level is set. This, in turn, means there is no pre-verification performed and it is completely up to the user whether to allow an application to run or not.

In order to gain approval, the publisher requests an AA to check an application release for malicious code. The publisher prepays some amount of money to analyzers for that service. The resulting reports is published in RDN by each AA (no matter negative or positive it turned out to be) and associated with the release resource for users swift location and retrieval.

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

In the future, due to the expected progress in AI, this network may transform into some kind of distributed AI, which would collect and process analytics data from all network nodes and produce ratings that in turn could be used to estimate the safety of published products.

Time-to-Interact Optimizations

There is no longer an installation process – an application is started once all the required components are downloaded. To speed up the loading of the uApp applications, its architecture should be designed to consist of three levels of components:

Core When the application is initially requested, only the core components are downloaded, and the default configuration is used to launch the application.

Deferred After the application is started, the deferred loading components are downloaded in the background.

Optional Finally, when the user needs some additional features, s/he can ask the system to download corresponding components manually.

Additional Benefits

Below are some of the additional benefits that come with the technology:

Economy

- It is completely free to use for ordinary users.
- Publishers have to pay only a small fee for publishing and verification.
- The verification stage is optional but significantly increases trust in published software.
- It creates a big, new market for antivirus companies.

Freedom

- It is not possible to ban a particular publisher or application as it does not rely on a specific DNS record or IP address.
- Even publishers have no power to restrict users from using older versions if users are not satisfied with the latest ones.
- RDN cryptography replaces the need for code-signing certification.

Performance

- The technology does not require a high-performance BFT protocol -, as even existing solutions are already powerful enough for all its functions.
- Global unique identifiers of shared products ensure that downloading the same files more than once is avoided, thus minimizing TTI and Internet traffic in general.
- Users no longer need to have locally installed antivirus sacrificing the performance of their hardware.

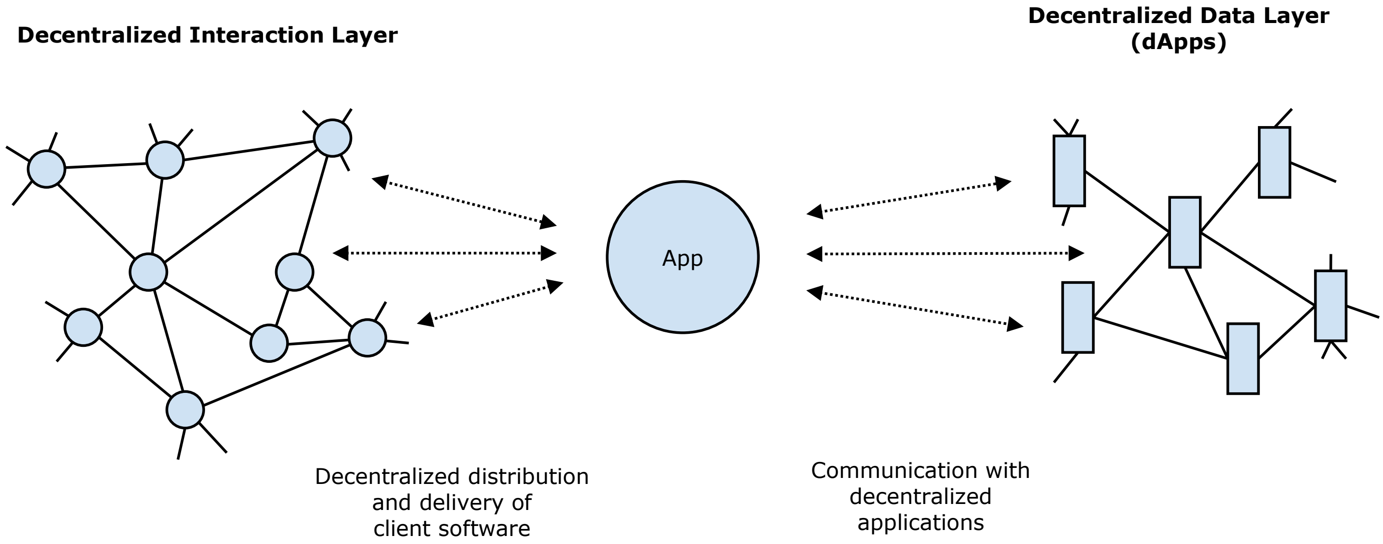
Opportunities

- Various decentralized software stores can be built on top of RDN.
- The technology provides a unified mechanism for updating applications so that no special development efforts are required to support auto-updating.
- UOS has the potential to create applications that work on various currently binary incompatible Linux distributions, which would give a new lease of life to the open-source community.
- RDN = Global Software Registry as a unified replacement for various package/component/library databases – not just for a code.
- Utilizing antimalware companies' specialized infrastructure makes it possible to perform a much deeper (even AI) and more comprehensive analysis of application code.
- Together with distributed file systems (cloud, Filecoin, Chia, Sia, Swarm, Storj, etc.), the technology enables server-less thin clients. In this case, both the applications and user data are stored remotely in a decentralized manner. Once a user is authorized in such a system and their profile is loaded, all the required applications and related data are downloaded to a local device for follow-up.

Conclusion

Utilizing various dApp platforms as the Data Layer and this technology as the Interaction Layer, together they give rise to a completely decentralized software stack that is invulnerable to DDOS attacks, censorship and other vulnerabilities inherent in traditional centralized solutions

Figure 1. Complete Decentralization


Decentralized Interaction Layer
Decentralized Data Layer (dApps)
App
Decentralized distribution and delivery of client software
Communication with decentralized applications

The scaling is also not an issue for this infrastructure, because p2p file exchange function of RDN, works in a similar way to the Torrent networks, which in turn can handle load at any scale by design. As for data layer infrastructure, some existing blockchain-based platforms have already shown their potential to bear heavy loads, and there is a strong expectation of major progress in this area in the very near future.

This technology is designed to revolutionize and disrupt vulnerable server-centric Internet architecture by decentralizing and improving the way in which applications and services are published and delivered to users. By combining and utilizing the most advanced network technologies available, Decentralized Interaction Layer is shaping the next era of the Internet.