UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ENS LABS LTD.

Petitioner

v.

UNSTOPPABLE DOMAINS INC.

Patent Owner

Inter Partes Review No.: IPR2024-00872

U.S. Patent No. 11,558,344

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 11,558,344

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

LIST OF EXHIBITS

Exhibit No.	Exhibit
1001	U.S. Patent No. 11,558,344 (“the ’344 Patent”)
1002	Declaration of James McDonald Regarding U.S. Patent No. 11,558,344
1003	Nick Johnson, ENS Documentation, Release 0.1, March 30, 2019. (captured on the Internet Archive, Dec. 13, 2019) https://buildmedia.readthedocs.org/media/pdf/ens/latest/ens.pdf (“Johnson”)
1004	Brantly Millegan, “ENS Now Supports Tor .Onion Address Resolution,” August 13, 2019 (captured on the Internet Archive, Aug. 15, 2019). https://medium.com/the-ethereum-name-service/ens-now-supports-tor-onion-address-resolution-9bb3bdff6217 (“Millegan”)
1005	Nick Johnson, “EIP-137: Ethereum Domain Name Service—Specification,” Ethereum Improvement Proposals, No. 137, Apr. 2016 (captured on the Internet Archive Mar. 29, 2018) https://eips.ethereum.org/EIPS/eip-137 (“EIP137”)
1006	Jim McDonald, “EthDNS: an Ethereum backend for the Domain Name System,” January 22, 2018 (captured on the Internet Archive, Feb. 8, 2018) https://medium.com/@jgm.orinoco/ethdns-an-ethereum-backend-for-the-domain-name-system-d52dabd904b3 (“McDonald”)
1007	USPTO – Patent Center Assignment Records for US Patent No. 11,558,344
1008	Richard Moore, “EIP-634: Storage of text records in ENS,” May 17, 2017 (captured on the Internet Archive March 13, 2019) https://eips.ethereum.org/EIPS/eip-634 (“Moore”)
1009	U.S. Patent Publication 2004/0249939 (“Amini”)
1010	U.S. Patent Publication 2013/0114432 (“Haynes”)
1011	U.S. Patent Publication 2005/0235031 (“Schneider”)
1012	Vaibhav Saini, “StoragePedia: An Encyclopedia of 5 Blockchain Storage Platforms,” August 13, 2018 (captured on the Internet Archive August 21, 2018). Available: https://hackernoon.com/storagepedia-an-encyclopedia-of-5-blockchain-storage-platform-8aa13c630ace (“Saini”)

1013	Prosecution History of the ’344 Patent
1014	(Reserved)
1015	U.S. Provisional Patent Application No. 63/084,493 (“the ’493 Provisional Application”)
1016	Entriken, William et al., "ERC-721: Non-Fungible Token Standard," *Ethereum Improvement Proposals*, no. 721, January 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-721 (“ERC721”)
1017	Affidavit of Nathaniel E Frank-White, The Internet Archive, February 22, 2024.
1018	Affidavit of Nathaniel E Frank-White, The Internet Archive, January 27, 2024.
1019	Ethereum Name Service, “.eth Permanent Registrar,” captured on the Internet Archive, May 4, 2019, https://web.archive.org/web/20190504155147/https://docs.ens.domains/contract-api-reference/.eth-permanent-registrar (“ENS Docs”)
1020	(Reserved)
1021	Curriculum Vitae of Jim McDonald
1022-1029	(Reserved)
1030	Lutkevich, Ben, et al., “Definition: domain name system (DNS), August 2021. Available: https://www.techtarget.com/searchnetworking/definition/domain-name-system.
1031	“What is DNS? How DNS works.” Available: https://www.cloudflare.com/learning/dns/what-is-dns/.
1032	Yasar, Kinza, Definition: IP address (Internet Protocol address), January 2023. Available: https://www.techtarget.com/whatis/definition/IP-address-Internet-Protocol-Address.
1033	IP Tracker. Available: https://www.ip-tracker.org/lookup.php?ip=nytimes.com.
1034	Scarpati, Jessica, et al., Definition: URL (Uniform Resource Locator), September 2021. Available: https://www.techtarget.com/searchnetworking/definition/URL.
1035	Coen, Emanuel, “What is the Ethereum Name Service and what is it good for?” (Almost 4 Years Ago). Available:

	https://cryptotesters.com/blog/what-is-the-ethereum-name-service-ens-and-what-is-it-good-for.
1036	CloudFlare, "DNS server types." Available: https://www.cloudflare.com/learning/dns/dns-server-types/
1037	Barney, Nick, et al., "Definition: blockchain," October 2023. Available: https://www.techtarget.com/searchcio/definition/blockchain.
1038	Ravikiran A S, "What is Blockchain Technology? How Does Blockchain Work? {Updated}, October 18, 2023. Available: https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology#:~:text=A%20blockchain%20is%20a%20distributed%2C%20immutable%2C%20and%20decentralized%20ledger%20at,a%20chronological%20chain%20of%20information.
1039	MetaMask, "Learn the basics of blockchains and Ethereum (miners and validators, gas, cryptocurrencies and NFTs, block explorer, networks, etc.)." Available: https://support.metamask.io/hc/en-us/articles/360015489611-Learn-the-basics-of-blockchains-and-Ethereum-miners-and-validators-gas-cryptocurrencies-and-NFTs-block-explorer-networks-etc.
1040	Kerner, Sean Michael, et al., "Definition: Ethereum," January 2022. Available: https://www.techtarget.com/whatis/definition/Ethereum#:~:text=Ethereum%20is%20a%20blockchain%2Dbased,a%20number%20of%20financial%20uses.
1041	Barney, Nick, et al., "Definition: smart contract," September 2023. Available: https://www.techtarget.com/searchcio/definition/smart-contract#:~:text=Smart%20contracts%20function%20on%20the,in%20the%20contract%20are%20executed.
1042	Barney, Nick, et al. "Definition: non-fungible token (NFT), September 2023. Available: https://www.techtarget.com/whatis/definition/nonfungible-token-NFT#:~:text=NFTs%20are%20created%20using%20smart,such%20as%20ownership%20and%20transferability.
1043	Amazon, "Non-Fungible Tokens (NFTs) Explained. Available: https://aws.amazon.com/blockchain/nfts-explained/#:~:text=Under%20the%20hood%2C%20a%20non,token%20to%20a%20new%20owner.

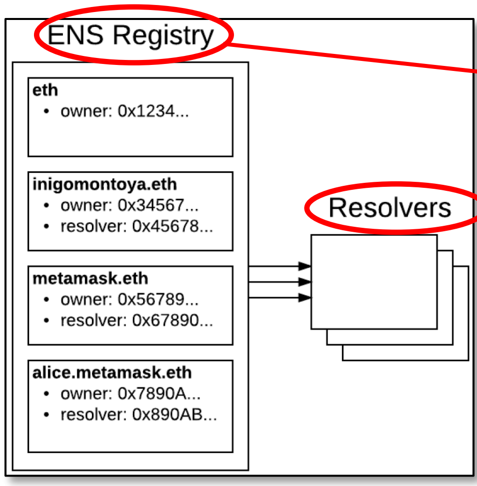
1044	Sharma, Rakesh, et al., "Cryptocurrency: Non-Fungible Token (NFT): What It Means and How It Works," January 28, 2024. Available: https://www.investopedia.com/non-fungible-tokens-nft-5115211#:~:text=Non%2Dfungible%20tokens%20(NFTs),are%20stored%20in%20other%20places.
1045	Wade, Jacob, et al., "Cryptocurrency – Strategy & Education: How to Create an NFT," December 9, 2023. Available: https://www.investopedia.com/how-to-create-an-nft-6362495.
1046	DeFi – Non-Fungible Tokens (NFTs): The Technical Structure of NFTs Explained," March 10, 2023. Available: https://www.gemini.com/cryptopedia/what-is-a-non-fungible-token-nft-crypto.
1047	Cooling, Sam, "What is ERC-721?" July 20, 2023. Available: https://www.techopedia.com/definition/erc-721.
1048	Ethereum, "ERC-721 Non-Fungible Token Standard," November 19, 2023. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-721/.
1049	(Reserved)
1050	Future Learn. "How do crypto-wallets work?" RMIT 2021. Available: https://www.futurelearn.com/info/courses/defi-exploring-decentralised-finance-with-blockchain-technologies/0/steps/254823#:~:text=In%20the%20context%20of%20blockchains,wallet%20does%20not%20hold%20tokens.
1051	"Ethereum wallets: A beginner's guide to storing ETH." Available: https://cointelegraph.com/learn/ethereum-wallets-a-beginners-guide-to-storing-eth.
1052	Ghobadi, Arya, "What is MetaMask?" March 15, 2022. Available: https://thedefiant.io/what-is-metamask.
1053	Shi, Z. et al., "Integration of Blockchain and Auction Models: A Survey, Some Applications, and Challenges," IEEE Communications Surveys & Tutorials, Vol. 25, No. 1, First Quarter 2023, pp. 497-537, https://doi.org/10.1109/COMST.2022.3222403. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9953944.
1054	Seike, H., et al., "Blockchain-Based Ubiquitous Code Ownership Management System without Hierarchical Structure," *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud*

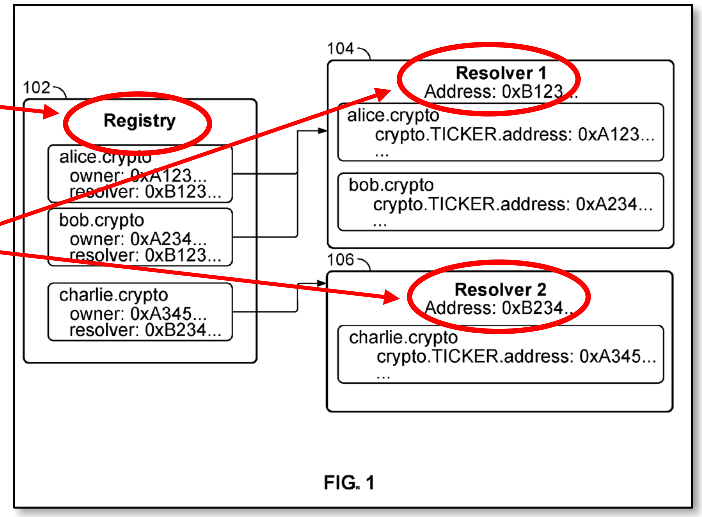
	& Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 2018, pp. 271-276, doi: 10.1109/SmartWorld.2018.00081. Available: https://ieeexplore.ieee.org/document/8560059.
1055	CryptoNinjas, "Ethereum Name Service (ENS) now supports Tor .onion addresses," August 14, 2019. Available: https://www.cryptoninjas.net/2019/08/14/ethereum-name-service-ens-now-supports-tor-onion-addresses/
1056	Cryptonomist, "Tor: the .onion domain on Ethereum Name Service, August 15, 2019. Available: https://en.cryptonomist.ch/2019/08/15/tor-on-ethereum-name-service/

Pursuant to 35 U.S.C. § 311 and 37 C.F.R. § 42.100, Petitioner ENS Labs Ltd. ("Petitioner") requests *inter partes* review ("IPR") of claims 1-20 ("the Challenged Claims") of U.S. Patent No. 11,558,344 ("the '344 Patent"). EX1001. Unstoppable Domains Inc. ("Patent Owner") is the assignee of record. EX1007.

I. INTRODUCTION

The Challenged Claims are directed to a method for resolving a domain name, e.g., 'alice.crypto,' on a blockchain. See EX1001, 4:27-29, FIG. 1. But this is nothing new. Several years before the September 28, 2020 effective filing date of the '344 Patent, various printed publications described the Ethereum Name Service ("ENS") system as resolving domain names with an '.eth' extension (i.e., ENS names) on the Ethereum blockchain. As explained herein, the '344 Patent relies on the same registry-and-resolver infrastructure described in the prior art published user documentation for the ENS system:





EX1003 at 6 (Johnson) EX1001 at FIG. 1 ('344 Patent)

Additional printed publications described the ENS system being used with a web browser to access websites and content from blockchain-based distributed storage networks, as well as various other modifications and add-ons for adapting the ENS system to interface with existing technology.

II. OVERVIEW OF THE TECHNOLOGY

The concept of resolving domain names on a blockchain was well known by the September 28, 2020 effective filing date of the '344 Patent.

Stepping back, resolving domain names, or domain name resolution, refers to the translation of human-friendly domain names into computer-friendly numeric addresses where retrievable content is stored. EX1002 at 25-27; EX1030 at 1-2. Domain name resolution is necessary because computers identify network addresses via numeric or alphanumeric strings (e.g., 192.168.1.1) that are more difficult for humans to remember than human-friendly words typed into browsers. EX1002 at 27; EX1031 at 1-2. Naming services handle domain name resolution by mapping network addresses to human-friendly domain names—similar to how a conventional phonebook maps a person's phone number to that person's name. EX1002 at 28; EX1031 at 1-2; EX1030 at 1-2.

A. Domain Name Service (DNS): The Phone Book of the Internet

Since the late 1980s, DNS has been used to look-up the Internet Protocol (IP) addresses[1] of computers by their associated website domain name, which is also referred to as a Uniform Resource Locator ("URL"),[2] for example, 'nytimes.com.' EX1002 at 29; EX1031 at 1-2; EX1030 at 1; EX1035 at 2-3. This process of looking-up an IP address by its domain name is called resolving the domain name, or domain name resolution. EX1002 at 29; EX1031 at 2; EX1030 at 2. In the digital world, DNS is the "phonebook of the Internet." EX1002 at 30; EX1031 at 1.

The basic DNS domain name resolution process is shown in Figure A below. When the user enters a website's domain name or URL, one or more DNS servers

---

[1] An IP address is a unique numeric or alphanumerical identifier of a device or network, such as '151.101.129.164,' which is the IP address for www.nytimes.com. EX1002 at 29; EX1032 at 1; EX1033 at 1.

[2] A URL is a unique identifier for an Internet resource (e.g., a webpage) that includes the domain name of the computer where the website or other content is located. EX1002 at 29; EX1034 at 1-2.

look-up the stored records for the domain name, e.g., nytimes.com, and return the corresponding IP address—e.g., '151.101.129.164'—for the computer hosting the website content. EX1002 at 30-31; EX1030 at 1.

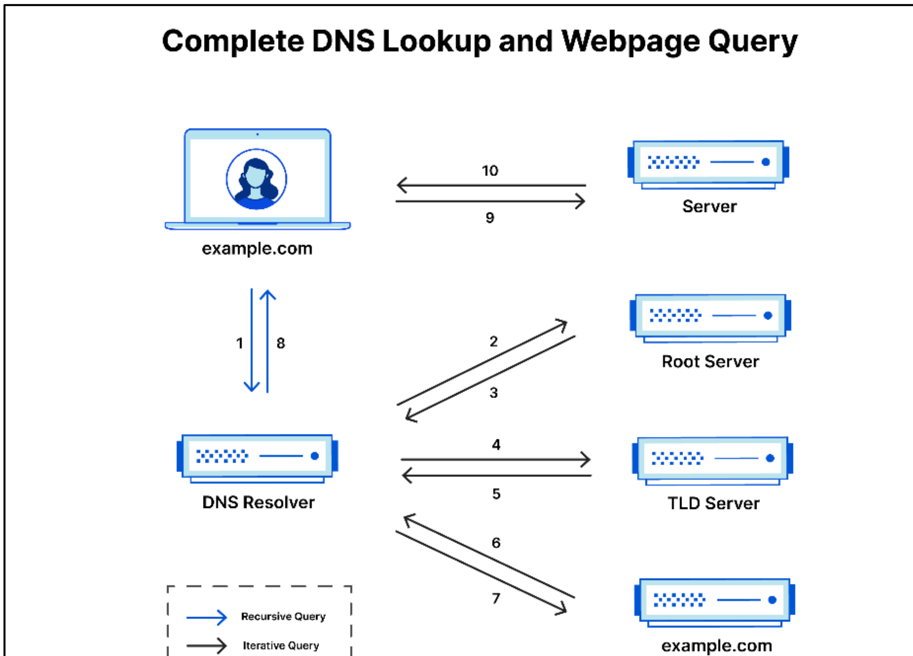


FIGURE A

EX1031 at 7.

Often, there are multiple DNS servers that participate in the DNS resolution process to resolve the domain name. As shown in Figure A above, the web browser sends a URL query (1) to a DNS resolver. The query requests the IP address associated with the URL (example.com). If the IP address is stored locally in the DNS resolver, the DNS resolver will respond to the web browser with the IP address (8) of example.com. EX1002 at 31-32; EX1031 at 6-7; EX1036 at 2. *Accord* EX1001, Col. 1:13-22. However, if the IP address is not stored locally in the DNS resolver, it will be resolved by querying other DNS servers. Accordingly, the DNS resolver may send sequential queries (2), (4), and (7) to a root server, TLD server, example.com nameserver and receives responses (3), (5) and (6) from the servers that identify each subsequent server by its respective IP address until finally obtaining the IP address associated with the example.com URL. EX1002 at 32; EX1031 at 6-7; EX1036 at 1-5.

When it receives the IP address from the example.com nameserver (6), the resolver sends the IP address (8) to the web browser, which uses the IP address to request (9) the website content from the computer hosting the website, *i.e*. the webserver. The webserver then returns the web content (10) to the web browser, which displays the website content. EX1002 at 33; EX1031 at 6-7; EX1036 at 1-5.

B. ENS: The Phone Book of the Blockchain

Similar to DNS being the phonebook of the Internet, Ethereum Name Service (ENS) is "the phonebook of the blockchain." EX1002 at 34; EX1035 at 2-3. But instead of resolving a URL by querying a series of DNS servers using respective IP addresses via the internet as in DNS, ENS resolves a URL via a smart contract on the Ethereum blockchain. EX1002 at 34; EX1035 at 3. Thus, it is helpful to review some aspects of blockchain technology—particularly Ethereum—before discussing domain name resolution on the blockchain.

1. Blockchain & Ethereum

A blockchain is a type of distributed ledger, which means that the data records are stored in duplicate in multiple locations in a peer-to-peer (P2P) network of computers, or nodes. The multiple nodes store, validate, and update the duplicate blockchain data records simultaneously. This makes data storage on the blockchain more resilient to threats and disruptions due to its redundancy. EX1002 at 36; EX1037 at 1-4; EX1038 at 3; EX1039 at 1-3.

As the name "blockchain" implies, the data records of a blockchain are called "blocks" and are stored in a format that "chains" the blocks in a sequence, where each block refers to the previous block in the chain. EX1002 at 37; EX1037 at 3-5; EX1038 at 3-6; EX1039 at 3_. Various types of data are stored in each block. EX1002 at 37; EX1037 at 3-5; EX1038 at 5-6. The computers storing the multiple copies of the blockchain execute consensus algorithms (e.g., Proof-of-Stake) that allow the computers to agree upon additional blocks to be added to the chain and authenticate the validity of the data stored in the blockchains. EX1002 at 37; EX1037 at 4-5, 12; EX1038 at 5, 8, 16; EX1039 at 4-5.

Ethereum is one type of a blockchain that was developed in 2015. EX1002 at 38; EX1040 at 1, 9; EX1039 at 4. In addition to supporting transactions of its cryptocurrency, known as "Ether," the Ethereum blockchain supports storing computer programs, known as “smart contracts.” EX1002 at 38; EX1040 at 1-2; EX1039 at 4-5.

2. Smart Contracts and Non-Fungible Tokens

A smart contract is a computer program stored on a blockchain. The program automatically executes in response to certain conditions being met. EX1002 at 39; EX1041 at 1-2; EX1039 at 4-5. Among other things, smart contracts are used to create and manage non-fungible tokens (NFTs). EX1002 at 39; EX1042 at 6.

An NFT is a type of blockchain-based data structure that establishes ownership of a unique digital asset. EX1002 at 40; EX1043 at 1-3; EX1041 at 5, 10. Each NFT includes a unique identifier that distinguishes the NFT from other NFTs, as well as an owner identifier that identifies who owns the NFT and the associated unique digital asset. EX1002 at 40; EX1043 at 1-3; EX1044 at 1-3; EX1045 at 1-2. The unique identifier may also be referred to as a token ID or “hash” that may be computed using a hash function. EX1002 at 40; EX1045 at 1-2. NFTs may also include an address of where the associated digital asset is stored, such as a physical address, an IP address, and a blockchain address, and other information about the asset. EX1002 at 40; EX1046 at 2-3.

On January 24, 2018, the ERC-721: Non-Fungible Token Standard was issued to standardize a set of requirements for NFTs on the Ethereum blockchain.

EX1002 at 41; EX1016 at 1-2; EX1047 at 1-2; EX1048 at 1-2; EX1039 at 7. While the ERC-721 standard has become one of the more popular NFT standards, NFTs existed previously on Ethereum and other blockchains—known as "deeds" and by other names. EX1002 at 41; EX1016 at 1, 7; EX1047 at 2.

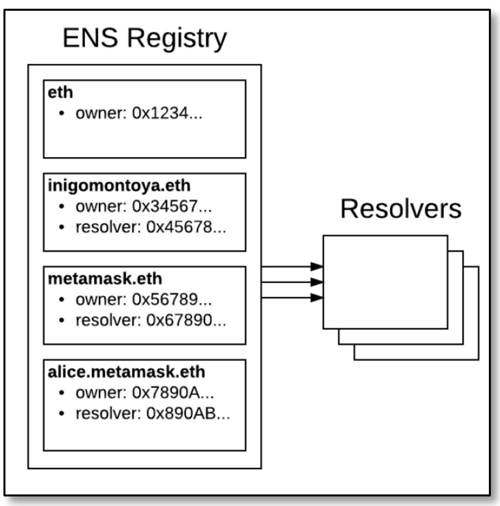
3. CryptoWallets: MetaMask for Ethereum

Interaction with a blockchain is via hardware and/or software often with a user interface referred to as a cryptocurrency wallet or crypto-wallet. EX1002 at 42; EX1050 at 1; EX1051 at 2. MetaMask is a crypto-wallet that is available as a web browser extension or mobile app. The MetaMask browser extension enables interaction with the Ethereum blockchain via traditional web browsers, such as Chrome, Firefox, and Edge. EX1002 at 42; EX1052 at 2; EX1039 at 7. Introduced in or around 2016, MetaMask was one of the first crypto-wallets available for interacting with the Ethereum network. EX1002 at 42; EX1052 at 4.

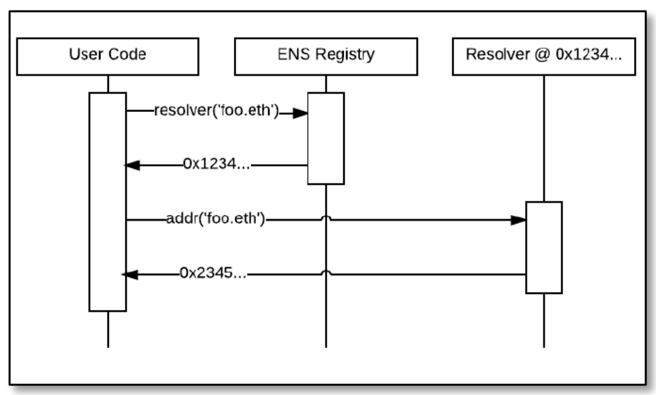
4. From DNS to ENS

Analogous to how DNS allows clients to request resolution of names into IP addresses and other results using data local to each DNS server, the Ethereum Name Service ("ENS") provides the same functionality via a smart contract to store and manage data on the Ethereum blockchain. EX1002 at 43, 47. Developed in 2016 and 2017, ENS is a decentralized naming protocol built on the Ethereum blockchain. EX1002 at 44, 47; EX1003 at 5. ENS performs a similar function as DNS: just as DNS maps a URL to an IP address, ENS maps an ENS name, e.g., 'alice.eth,' to a blockchain address where a retrievable resource is stored. EX1002 at 44, 47; EX1003 at 5, 18-19.

ENS relies on two principal components: resolvers and the registry. EX1002 at 45; EX1003 at 5. The resolvers are smart contracts that maintain records of ENS names and the blockchain address of resources associated with each ENS name, such as websites and content. EX1002 at 45; EX1003 at 18-19, 21. The registry is a smart contract that keeps track of which resolver is responsible for which ENS name. EX1002 at 45; EX1003 at 6.



EX1003 at 6



EX1003 at 18.

As shown in the above-right diagram, in order to resolve the ENS name, e.g., *foo.eth*, a request is sent by the user code to the ENS registry. EX1002 at 46; EX1003 at 18. The registry then looks-up the resolver responsible for the ENS name, e.g., *foo.eth*, and returns the address of the appropriate resolver. EX1002 at

46; EX1003 at 18-19. A request is subsequently sent to the appropriate resolver, which looks-up and returns the retrievable resource. EX1002 at 46; EX1003 at 19.

III. OVERVIEW OF THE '344 PATENT

A. The Specification and the Challenged Claims

The '344 Patent issued from U.S. Patent Application No. 17/486,512, filed on September 27, 2021, which claims priority to U.S. Provisional Patent No. 63/084,493 ("the '493 Provisional Application), filed on September 28, 2020. EX1001; EX1015. The '344 Patent incorporates by reference the contents of the '493 Provisional Application "for all purposes." EX1001, Col. 1:6-9.

The '344 Patent describes a method for resolving a domain name on a blockchain. The method involves using two types of smart contracts on the blockchain. EX1001, Col. 3:59-61; EX1002 at 49. FIG. 1 of the '344 Patent below depicts these smart contracts. A resolver smart contract maintains a record of domain name identifiers and the locations of resources associated with each domain name identifier, such as websites and content. EX1001, Col. 3:66-4:1; EX1002 at 49. A registry smart contract keeps track of which resolver smart contract is responsible for which domain name identifier. EX1001, Col. 3:64-66; EX1002 at 49.

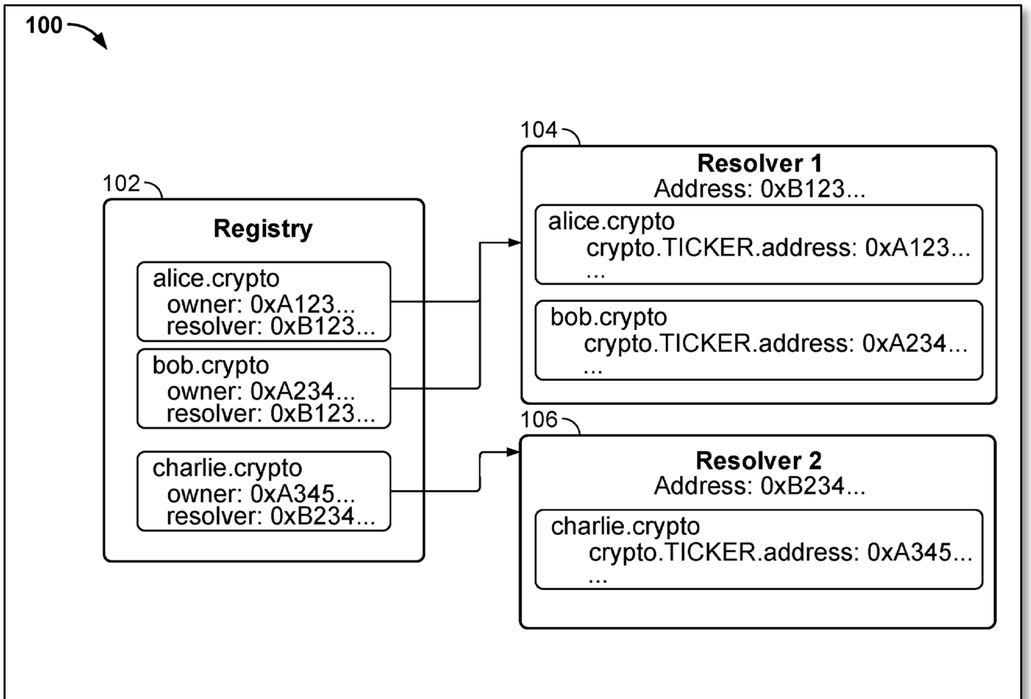


FIG. 1

In order to resolve the domain name identifier, e.g., 'alice.crypto,' a request is sent to the registry smart contract referencing a hash of a domain name identifier. EX1001, Cols. 11:65-12:1; EX1002 at 50. The registry smart contract looks-up the resolver smart contract responsible for the domain name identifier, e.g., 'alice.crypto,' and returns the address of the appropriate resolver smart contract. EX1001, Col. 10:41-55; EX1002 at 50. The request is sent to the appropriate resolver, which looks-up and returns the location of the resource associated with the domain name identifier. EX1001, Col. 10:56-60; EX1002 at 50.

The ‘344 Patent includes 20 total claims, with three independent claims each covering a method, a system and a computer program product. Independent claim 1 is representative of the challenged independent claims:

> 1. A method, comprising:
>
> receiving a request to resolve a name of a domain of an identifier of web content;
>
> automatically determining that the name of the domain is to be resolved using a blockchain;
>
> determining an identifier of a non-fungible token corresponding to the domain of the identifier of the web content;
>
> using the identifier of the non-fungible token to send a request to a smart contract of the blockchain to obtain one or more resolution records for the domain, wherein the blockchain stores the non-fungible token associating the domain to an account address of an owner of the domain;
>
> receiving the one or more resolution records of the domain; and
>
> utilizing the received one or more resolution records to resolve the name of the domain.

EX1001, Col. 21:25-42. Independent claims 19 and 20 respectively recite a corresponding system and computer program product. EX1001, Cols. 22:40 – 23:13. Dependent claims 2-18 recite well-known features related to resolving domain names and blockchain technology. EX1001, Cols. 21:43 – 22:39.

B. Overview of the Prosecution History

To overcome a prior art rejection during examination, the independent claims were amended to add the use of the non-fungible token. EX1013 at 70-73. Claim 1 was amended to add the step of “determining an identifier of a non-fungible token corresponding to the domain of the identifier of the web content;" and to add the limitations of "using the identifier of the non-fungible token to send …," "wherein the blockchain stores the non-fungible token associating the domain to an account address of an owner of the domain." EX1013 at 70. The Examiner subsequently allowed the claims. EX1013 at 24 (Notice of Allowance).

IV. PERSON HAVING ORDINARY SKILL IN THE ART ("PHOSITA")

A PHOSITA would have a Bachelor's degree (or international equivalent) in computer science, mathematics, or related fields, and at least 2 years of experience with software programming projects including at least six months experience with blockchain technologies. EX1002 at 56.

Alternatively, a PHOSITA would not need a Bachelor's degree (or equivalent) so long as that person had sufficient relevant industry experience, such as an additional two years more of experience with software programming projects (making the minimum for this person four years of such experience). This alternative PHOSITA would still need to have at least six months experience with blockchain technologies. EX1002 at 57.

V. CLAIM CONSTRUCTION

Claim terms in IPRs "shall be construed using the same claim construction standard that would be used to construe the claim in a civil action." 37 C.F.R. §42.100(b) (as amended Nov. 13, 2018); *Phillips v. AWH Corp*., 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc). Unless otherwise indicated, the terms of the '344 patent should be given their plain and ordinary meaning in light of the specification of the '344 patent. 37 C.F.R. § 42.200(b). Petitioner submits that all claim terms in the '344 Patent may be interpreted according to their plain and ordinary meaning.

VI. STATUTORY GROUNDS OF CHALLENGE

Ground	Claims	Statutory Basis	Prior Art
1	1, 2-4, 6, 7, 10, 15, 19, and 20	§103	Johnson in view of Millegan.
2	5	§103	Johnson and Millegan in further view of Schneider
3	9	§103	Johnson and Millegan in further view of Moore
4	8 and 11	§103	Johnson and Millegan in further view of EIP137
5	12-14	§103	Johnson, Millegan, and EIP137 in further view of Haynes
6	16-18	§103	Johnson and Millegan in further view of Saini

VII. DESCRIPTION OF THE PRIOR ART RELIED ON

With the exception of EIP137 (EX1005), none of the prior art documents relied on were considered by the Examiner during prosecution of the '344 Patent. EX1013.

A. Johnson—ENS Documentation, Release 0.1

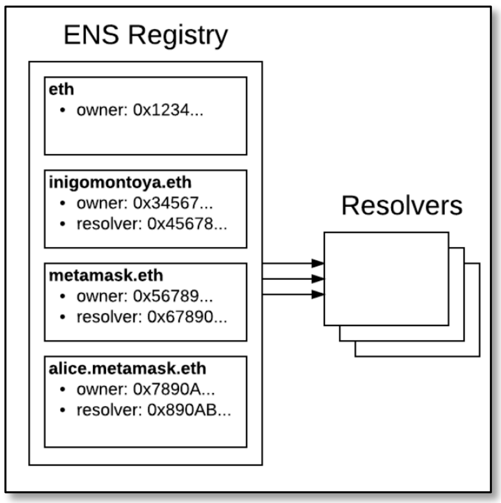
Johnson, entitled "ENS Documentation, Release 0.1," was published on March 30, 2019. EX1003. Johnson was captured and archived by the Internet Archive on December 13, 2019.[3] EX1003; EX1018. Johnson was publicly accessible to a PHOSITA as of that date, if not earlier. EX1002 at 58-60. Johnson is a printed publication and is prior art under 35 U.S.C. § 102(a)(1).

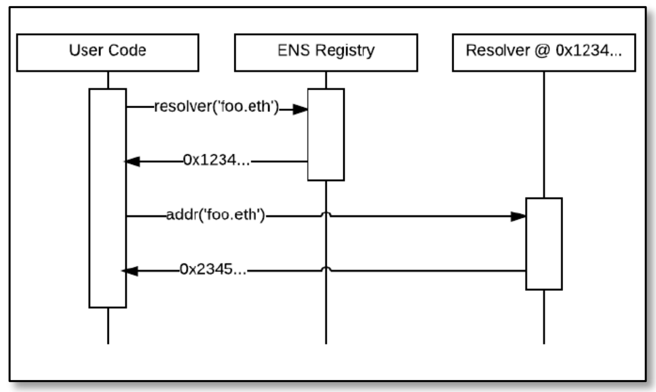
Johnson was published as a developer's guide or user documentation to describe the use and implementation of ENS. EX1002 at 58-60. Johnson states that ENS "map[s] human-readable names like 'alice.eth' to machine-readable identifiers such as Ethereum addresses, content hashes, and metadata." EX1002 at 61; EX1003 at 5. Johnson explains that ENS is analogous to the well-known DNS, but "has significantly different architecture, due to the capabilities and constraints provided by the Ethereum blockchain." EX1003 at 5. EX1002 at 61.

As shown in the below-left diagram, Johnson's ENS has two principal components: resolvers and the registry. EX1002 at 62; EX1003 at 5. The resolvers are smart contracts that maintain records of ENS names and the blockchain address of resources associated with each ENS name, such as websites and content. EX1002 at 62; EX1003 at 7, 19-20. The registry is a smart contract

[3] An affidavit from the Internet Archive, popularly known as "the WayBack Machine," is attached as EX1018 and certifies the public availability date.

that associates the resolver responsible for each ENS name. EX1002 at 62; EX1003 at 6-7.





EX1003 at 6 **EX1003 at 18.**

As shown in the above-right diagram, in order to resolve the ENS name, e.g., *foo.eth*, a request is sent from the user code to the ENS registry. EX1002 at 63; EX1003 at 18. The registry then looks-up the resolver responsible for the ENS name, e.g., *foo.eth*, and returns the address of the corresponding resolver. EX1002 at 63; EX1003 at 19. A request is subsequently sent to the corresponding resolver, which looks-up and returns the resource at the blockchain address associated with the ENS name. EX1002 at 63; EX1003 at 18-19.

In sending requests to the registry or the resolver, Johnson describes computing a hash of the domain and computing a hash of the ENS name, which are each a unique identifier stored in an associated NFT. EX1002 at 64-66; EX1003 at 19.

B. Millegan—ENS Now Supports Tor .Onion Address Resolution

Millegan, entitled "ENS Now Supports Tor. Onion Address Resolution," published at least as early as August 13, 2019. EX1002 at 67-69; EX1004 at 1. Millegan was captured and archived by the Internet Archive on August 15, 2019. EX1004; EX1018. This document was publicly accessible to a PHOSITA as of that date. EX1002 at 67-69. Millegan is a printed publication and is prior art under 35 U.S.C. § 102(a)(1).

Millegan discloses integrating ENS with a web browser extension that allows domain names to be entered into the browser and resolved with ENS on the Ethereum blockchain to retrieve website content. Millegan describes an example of using ENS to resolve a *.onion* domain name.[4] EX1002 at 70-71; EX1004 at 1. Millegan discloses using a web browser having the MetaMask browser extension installed, which "recognizes that [the inputted domain name] is an ENS name and so prevents the browser from treating it as a search or normal DNS name." EX1004 at 2; EX1002 at 72.

C. Schneider—U.S. Patent Publication 2005/0235031

Schneider, entitled "Hyperlink Generation and Enhanced Spell Check Method, Product, Apparatus, and User Interface System," was published on

[4] A *.onion* top level domain is a domain like '.com' or '.edu,' except that it is not recognized by the governing authority, The International Corporation for Assigned Names and Numbers (ICANN), and therefore is not resolvable by DNS. EX1002 at 71.

October 20, 2005. EX1011. Schneider is prior art under 35 U.S.C. §§ 102(a)(1) and 102(a)(2).

Schneider relates to DNS resolution and describes a technique to determine whether a highest-level domain (HLD) name is resolvable. EX1002 at 74; EX1011 at ¶ [0056]. Specifically, Schneider describes comparing a given domain name against a list of resolvable top-level domain names to determine whether the domain name can be resolved. EX1002 at 74; EX1011 at ¶¶ [0023], [0056].

D. EIP-137—Ethereum Domain Name Service-Specification

EIP137, entitled "EIP 137: Ethereum Domain Name Service—Specification," was published on April 4, 2016. EX1005; EX1002 at 75-76. EIP137 was captured and archived by the Internet Archive on March 29, 2018. EX1005; EX1018. This document was publicly accessible to a PHOSITA as of that date. EX1002 at 75-76. EIP137 is a printed publication and is prior art under 35 U.S.C. § 102(a)(1)).

EIP137 is a specification for an Ethereum improvement proposal that explains the basics of the ENS service for resolving domain names. EX1002 at 76, 78. EIP137 provides a proposal of ENS system, along with sample code snippets for implementing the registry and resolver smart contracts. EX1002 at 78; EX1005 at 7-12. EIP137 explains the type of resources that a resolver may return, including "a contract address, a content hash, or IP address(es) as appropriate.

EX1002 at 79; EX1005 at 3. EIP137 also explains that the resolution records included an IP address, a decentralized web content hash identifier, a plurality of target identifiers for different protocols, DApp gateway identifiers, hash values thereof and how it may use a protocol other than a decentralized web protocol. EX1002 at 80; EX1005 at 2, 6, 7.

E. Moore—EIP-634: Storage of text records in ENS

Moore, entitled "EIP-634: Storage of text records in ENS," was published on May 17, 2017. EX1002 at 81-82; EX1008. Moore was captured and archived by the Internet Archive on March 13, 2019. EX1008; EX1017. This document was publicly accessible to a PHOSITA as of that date. EX1002 at 81-82. Moore is a printed publication and is prior art under 35 U.S.C. § 102(a)(1).

Moore is a specification for an Ethereum improvement proposal describing "a resolver profile for ENS that permits the lookup of arbitrary key-value text data … allow[ing] ENS name holders to associate e-mail addresses, URLs and other informational data with a[n] ENS name." EX1008 at 1; EX1002 at 82, 84.

Moore highlights the flexibility of the ENS system in its ability to return a wide variety of textual data using custom-defined resolvers. EX1002 at 85. Some examples of the types of data that may be returned by an ENS resolver include, "an e-mail address," "a URL," "a URL to an image used as an avatar or logo," and "[a] description of the name," among others. EX1008 at 2; EX1002 at 85.

F. Haynes—U.S. Patent Publication 2013/0114432

Haynes, entitled "Connecting to an Evolved Packet Data Gateway," was published on May 9, 2013. EX1010. Haynes is prior art under 35 U.S.C. §§ 102(a)(1) and 102(a)(2).

Haynes is directed to communication between devices via the Internet using IP addresses. EX1002 at 87; EX1010 at ¶ [0036]. Haynes discloses selecting a protocol for communication based on a prioritized protocol list. EX1002 at 87; EX1010 at ¶¶ [0042]-[0044].

G. Saini—StoragePedia: An Encyclopedia of 5 Blockchain Storage Platforms

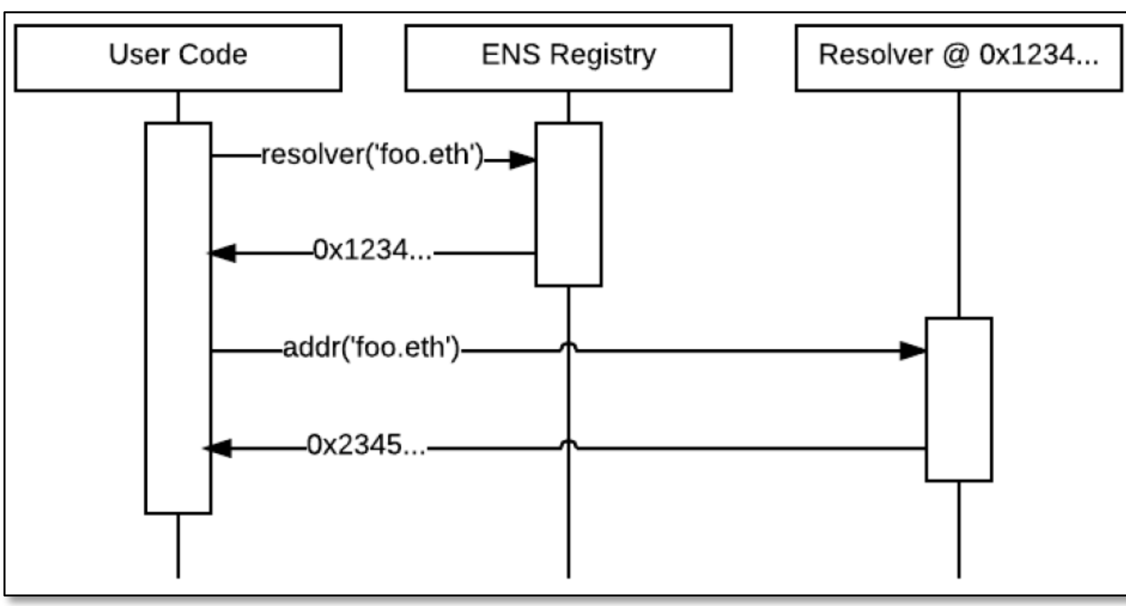
Saini, entitled "StoragePedia: An Encyclopedia of 5 Blockchain Storage Platforms," was published on August 13, 2018. EX1002 at 88-89; EX1012. Saini was captured and archived by the Internet Archive on August 21, 2018. EX1012; EX1017. This document was publicly accessible to a PHOSITA as of that date. EX1002 at 88-89. Saini is a printed publication and is prior art under 35 U.S.C. § 102(a)(1).

Saini provides an overview of five example blockchain-based distributed storage platforms, including Swarm. EX1002 at 91; EX1012 at 1-2. Saini explains that "Swarm is designed to deeply integrate with . . . the Ethereum blockchain for domain name resolution (using ENS). EX1012 at 3; EX1002 at 91. Further confirming the cooperation between Swarm and ENS, Saini states, "ENS is the system that Swarm uses to permit content to be referred to by a human-readable name, such as 'theswarm.eth.'" EX1012. p. 5; EX1002 at 91.

VIII. DETAILED EXPLANATION OF GROUNDS

A. <u>GROUND 1:</u> Independent claims 1, 19, and 20, and dependent claims 2-4, 6, 7, 10 and 15, are obvious under 35 USC § 103 over Johnson in view of Millegan.

Johnson and Millegan both relate to resolving ENS names, e.g., 'foo.eth' or 'duckduckgotor.eth,' to return resources that are located at Ethereum addresses associated with those ENS names. EX1002 at 93; EX1003; EX1004. In Johnson, the ENS name is input to and is used by a "user code" to look-up and return the stored resource (e.g., a website address) associated with the ENS name (i.e., to "resolve" the ENS name). EX1002 at 93; EX1003 at 18-19, 35. The following figure from Johnson illustrates this process:



EX1003 at 18-19. The "user code" queries an ENS registry to resolve the ENS name, e.g., *foo.eth*. EX1002 at 93, EX1003 at 18-19. The ENS registry then returns the Ethereum address of the ENS resolver responsible for returning the stored resource. EX1002 at 93; EX1003 at 18-19. The user code subsequently queries the ENS resolver (at its Ethereum address) for the stored resource. *Id*. The ENS resolver thereafter returns the stored resource to the "user code." *Id*. This process was generally referred to at the time of the invention as ENS resolution, although it was occasionally referred to as an ENS "look-up." EX1002 at 94.

Millegan refers to using this very same process to resolve ENS names and return an associated website address—except where Johnson relies on a generic "user code" and returns a generic resource, Millegan expressly describes (1) a modified web browser requesting the ENS resolution, and (2) a website address as the returned resource. EX1002 at 95; EX1004 at 2. Millegan indeed explains that the ENS name (i.e., *duckduckgotor.eth*) is entered into the modified web browser, which does an ENS "look-up" to retrieve a website address (e.g., an *.onion* address) associated with the ENS name. EX1002 at 95; EX1004 at 2.

A PHOSITA would have found it obvious to combine Johnson and Millegan so that the modified web browser of Millegan queries the ENS registry and ENS resolver of Johnson so as to resolve ENS names into website addresses for the web browser. EX1002 at 96. This combination is no more than a simple combination of known elements that yields predictable results. EX1002 at 96. *See KSR Intern. Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007) . Johnson indeed explains that the "user code" can be any software that accepts ENS names, as is illustrated by the "user code" of the above-figure accepting the ENS name for ENS resolution. EX1002 at 97; EX1003 at 17. Millegan provides such software in the form of its modified web browser that accepts ENS names for ENS resolution. EX1002 at 97; EX1004 at 2. Johnson further suggests this combination by referring to MetaMask as an example of software that accepts ENS names—the same software that provides the browser extension in Millegan. EX1002 at 98; EX1003 at 34; EX1004 at 1-2. A PHOSITA would have therefore been motivated by the express suggestions of Johnson and Millegan to use the MetaMask enabled web browser of Millegan as the "user code" of Johnson. EX1002 at 98. *See KSR*, 550 U.S. at 399.

Johnson further explains that the retrieved resource can be any resource, and Millegan describes the returned resource as a website address. EX1002 at 99, EX1003 at 19, 37; EX1004 at 2. Returning a website address is indeed a natural result of resolving ENS names entered into a web browser, because web browsers navigate to websites. EX1002 at 99. A PHOSITA would have therefore been motivated to return a website address as the resource retrieved via ENS resolution of the ENS name entered into the modified web browser of Millegan. EX1002 at 99.

The results of combining Johnson and Millegan would be predictable to a PHOSITA, because that is what Millegan expressly suggests. EX1002 at 100. *See KSR*, 550 U.S. at 416 . Millegan specifically refers to and builds on the ENS resolution process described in Johnson by explaining that its modified web browser "does a look-up on ENS on Ethereum, grabbing the *.onion* address for [the ENS name entered into the web browser] in the ENS record for [the ENS name] and returning it to the address bar." EX1002 at 100; EX1004 at 2. Thus, Millegan provides an explicit suggestion to combine it with the ENS resolution process of Johnson by illustrating the advantages of the combination. EX1002 at 100. *See KSR*, 550 U.S. at 399 (2007) .

1. Independent Claim 1:

[1-Pre] – A method, comprising:

The preamble is presumptively not limiting, but both Johnson and Millegan disclose methods for resolving an ENS name on the Ethereum blockchain. EX1002 at 101-102; EX1003 at 7, 18; EX1004 at 1-3.

[1a] – receiving a request to resolve a name of a domain of an identifier of web content

Millegan discloses "receiving a request to resolve a name of a domain of an identifier of web content" in the form of receiving an ENS name by a web browser for resolution to website address associated with the ENS name. EX1002 at 104-112; EX1004 at 1-2. The ENS name, e.g., *duckduckgotor.eth*, is entered into a web browser and is thereby received as a request for ENS resolution. EX1002 at 105; EX1004 at 2. The ENS resolution returns to the web browser a website address, e.g., a *.onion* address, stored in a content record of a public resolver. EX1002 at 105-107; EX1004 at 3. A PHOSITA would have understood that the ENS name and the returned website address are each "an identifier of web content." EX1002 at 105-108; EX1004 at 2-3.

As discussed in Sec. VIII.A., a PHOSITA would have found it obvious to modify Johnson to use the modified web browser of Millegan to receive ENS names for ENS resolution into website addresses. EX1002 at 109-112. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 110-112.

[1b] – automatically determining that the name of the domain is to be resolved using a blockchain

Millegan discloses that the web browser automatically determines that the entered ENS name is an ENS name for ENS resolution, explaining that "[a]fter you enter duckduckgotor.eth in the address bar of the Tor Browser, MetaMask recognizes that duckduckgotor.eth is an ENS name." EX1002 at 113-114; EX1004 at 2. The MetaMask extension then accesses the Ethereum blockchain and "does a look-up on ENS" to "grab[]" the website address. EX1002 at 114; EX1004 at 2.

As discussed in Sec. VIII.A., a PHOSITA would have found it obvious to modify Johnson to use the modified web browser of Millegan to receive ENS names for ENS resolution into website addresses—and to consequently determine that ENS name entered into the web browser is to be resolved via ENS resolution that uses the Ethereum blockchain, as described by Millegan. EX1002 at 115-118. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 115-117.

[1c] – determining an identifier of a non-fungible token corresponding to the domain of the identifier of the web content

Johnson discloses "determining an identifier of a non-fungible token" in the form of determining an identifier of a deed for a domain corresponding to the ENS name. EX1002 at 119-123; EX1003 at 14. A PHOSITA would have understood that the deed is an NFT, because the deed is a unique token that records ownership of the domain corresponding to the ENS name. EX1002 at 120-122; EX1003 at 14. Indeed, the ERC721 standard for NFTs on the Ethereum blockchain explains that deeds are another name for non-fungible tokens, stating, "non-fungible tokens, also known as deeds." EX1002 at 122; EX1016 at 1. Johnson discloses that the identifier of the deed is determined by applying a hashing algorithm,[5] referred to as the *sha3* function, to the domain to which the deed pertains. EX1002 at 121. For example, Johnson explains that the identifier of the deed for the domain "foo" is

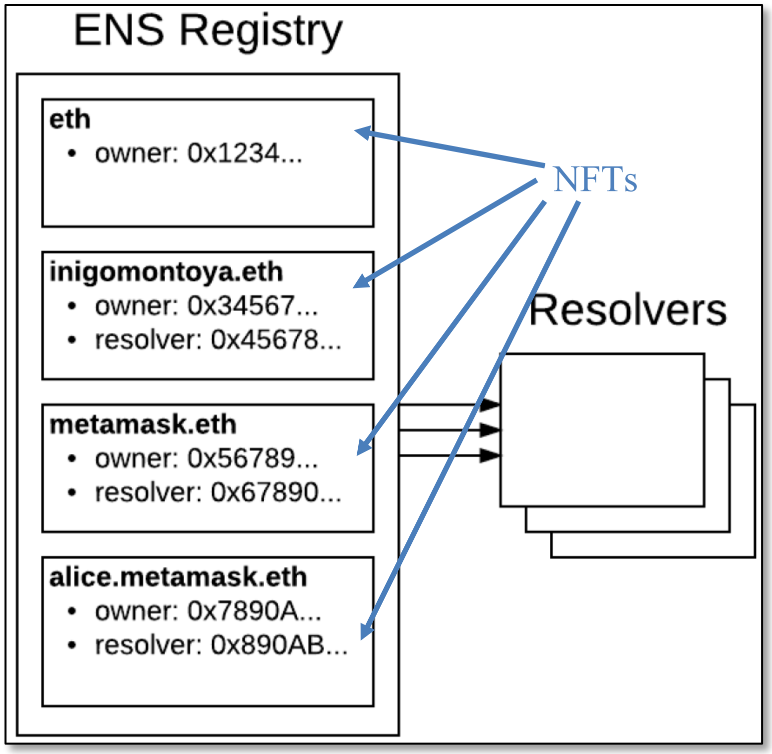
---

[5] A hashing function is a cryptographic function that translates an input into a fixed-length, hexadecimal value. EX1002 at FN3.

"*sha3(foo)*." Johnson thus describes "determining an identifier of a non-fungible token corresponding to the domain." EX1002 at 121.

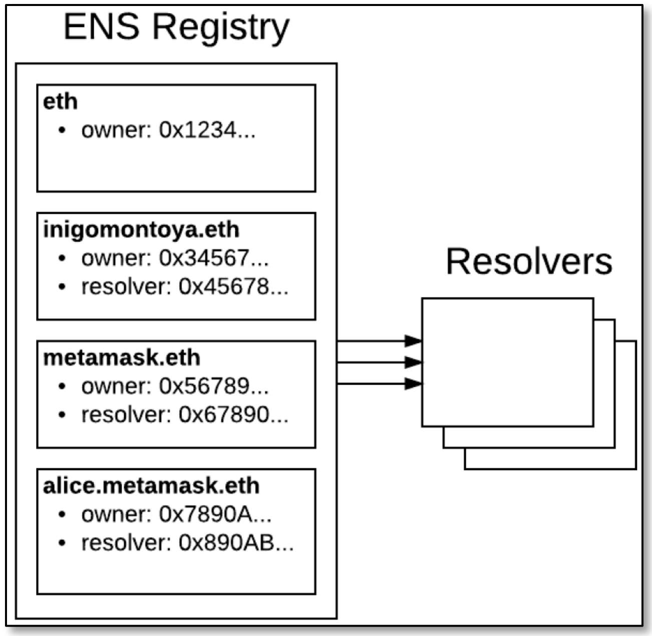
As discussed for *[1a]*, Millegan describes associating a domain with "an identifier of web content" in the form of a website address stored in the ENS record for the ENS name corresponding to the domain. EX1002 at 124. A PHOSITA would have found it obvious to combine Johnson and Millegan as described for *[1a]*. EX1002 at 124.

Johnson additionally discloses "determining an identifier of a non-fungible token corresponding to the domain," by determining a hash of the requested ENS name, e.g., *foo.eth*. EX1002 at 125; EX1003 at 18-19. More particularly, Johnson explains that a hashing algorithm called *namehash* is applied to translate the domain name into a hash value that is used to identify a data record stored in the ENS registry. EX1002 at 125; EX1003 at 22. Johnson indeed notes that "the output of the *namehash* function, [is] used to uniquely identify a name in ENS." *Id.*

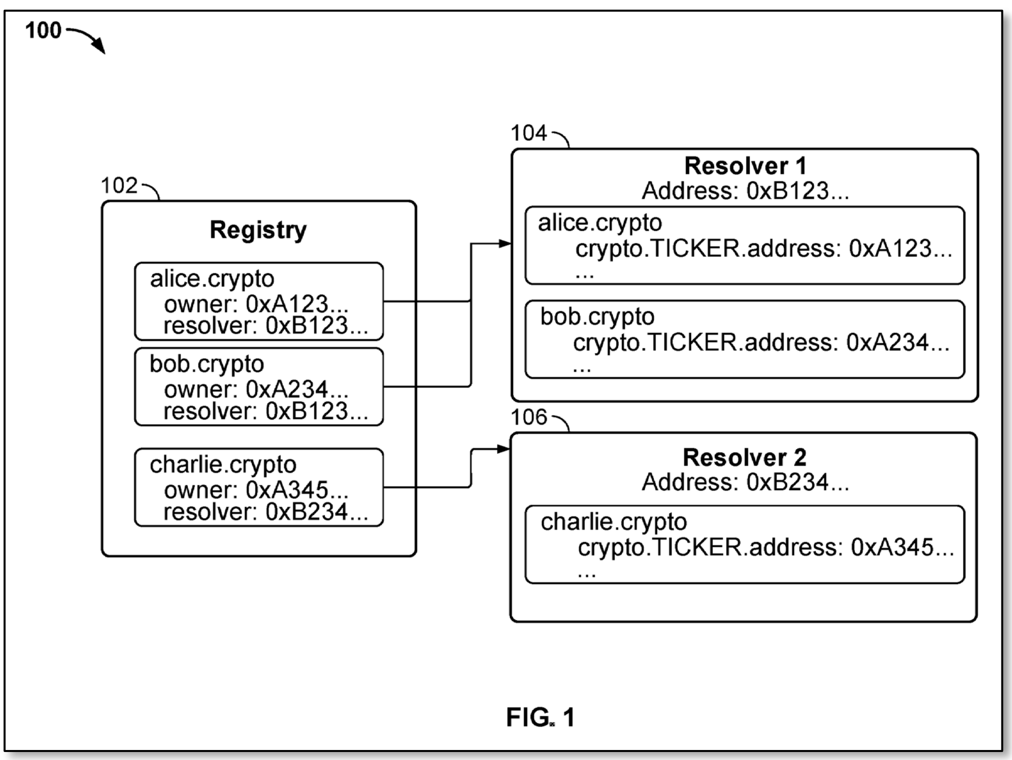


EX1003 at 6 (annotated)

A PHOSITA would have understood that the data record of the ENS name that is stored in the registry is also a non-fungible token, as identified in the above annotated figure. The data record of the ENS name is a non-fungible token because it: (1) has a unique identifier in the form of a unique hash of the ENS Name (e.g., the *namehash* of the *.eth* domain name), (2) has an associated owner, and (3) is stored on the Ethereum blockchain. EX1002 at 126; EX1003 at 5-6, 18, 22. This corresponds to the NFTs illustrated in the '344 Patent. EX1002 at 127. EX1001 at Col. 4:11-15, 43-55. The figure of Johnson, and Fig. 1 of the '344, are depicted below top-bottom to show the virtually identical architecture in the storage of NFTs in the registry.



EX1003 at 6 (Johnson)



EX1001 at FIG.1 ('344 Patent)

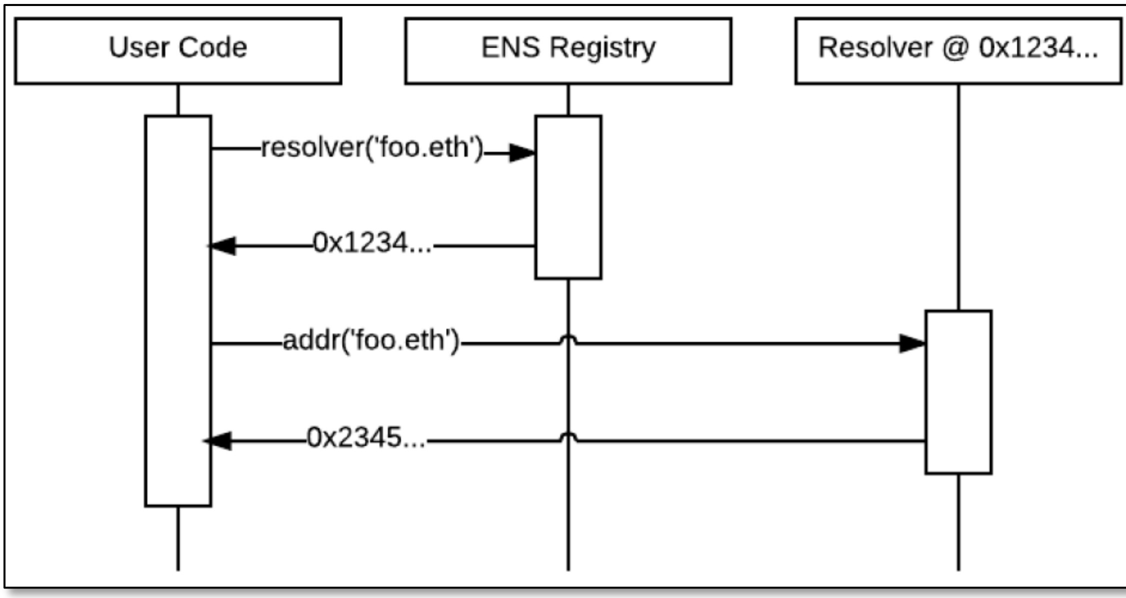
EX1001 at FIG. 1; EX1003 at 6.

Accordingly, Johnson discloses the above-limitation by both (1) determining the *sha3* hash of the domain *foo* as the identifier of the deed corresponding to *foo.eth*, and (2) determining the *namehash* of the ENS name *foo.eth* as the identifier of the data record of the ENS registry. EX1002 at 128-129.

[1d] – using the identifier of the non-fungible token to send a request to a smart contract of the blockchain to obtain one or more resolution records for the domain, wherein the blockchain stores the non-fungible token associating the domain to an account address of an owner of the domain

Johnson discloses "send[ing] a request to a smart contract of the blockchain to obtain one or more resolution records for the domain," by querying the registry using the *namehash* of the ENS name. EX1002 at 130-132; EX1003 at 19.

More particularly, as shown in the below figure, Johnson describes the user code querying the ENS registry by calling the *resolver(bytes32)* method on the ENS registry and passing it the *namehash* of the ENS name, e.g., *foo.eth*, being queried about:



EX1003 at 18-19. See also, EX1002 at 131. The *namehash* of the ENS name refers to a hash resulting from applying a *namehash* function on the ENS name. EX1002 at 131; EX1003 at 22. The *namehash* of the ENS name is thus used to query the ENS registry. EX1002 at 131; EX1003 at 19. The ENS registry then returns the address, e.g. "0x1234…," of the ENS resolver responsible for returning the resource associated with the ENS name. EX1002 at 132; EX1003 at 18-19. The user code then queries the ENS resolver by calling the *addr(bytes32)* method on the ENS resolver and passing it the *namehash* of the ENS name. EX1002 at 132; EX1003 at 18-19. The ENS resolver then returns the resource associated with the ENS name—which corresponds to the "resolution record." EX1002 at 132; EX1003 at 18-19.

Both the ENS registry and the ENS resolver are smart contracts. EX1002 at 133; EX1003 at 5-6, 18-19. Moreover, as discussed in *[1c]*, the *namehash* of the

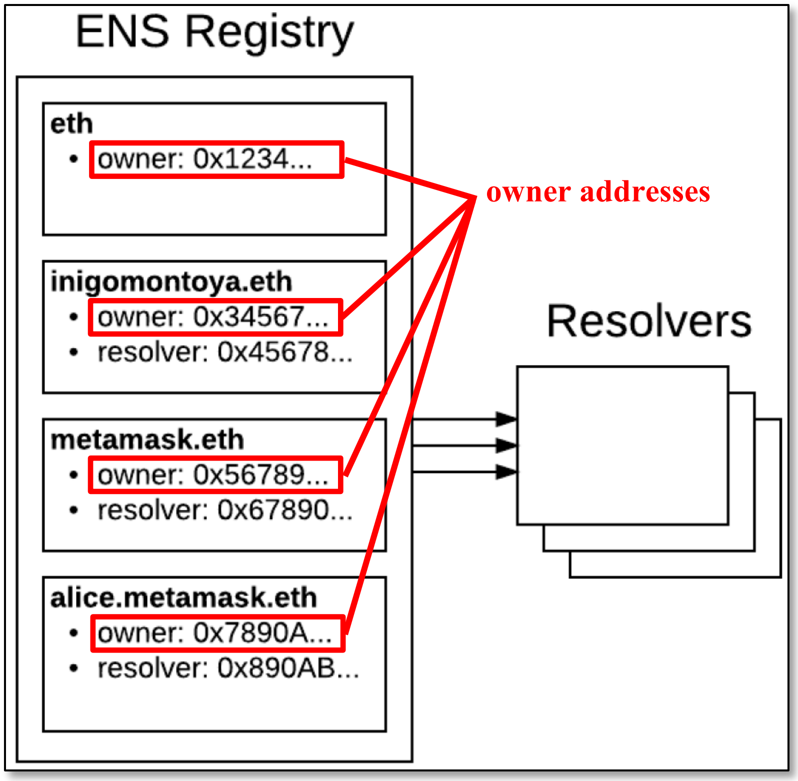
ENS name, e.g., *foo.eth*, corresponds to "the identifier of the non-fungible token." EX1002 at 133. Thus, Johnson discloses "using the identifier of the non-fungible token to send a request to a smart contract of the blockchain to obtain one or more resolution records for the domain" in the form of the user code using the *namehash* of the ENS name to query the ENS registry to resolve the ENS name so as to obtain the resource associated with the ENS name. EX1002 at 134.

Additionally, as discussed for *[1c]*, the *sha3* hash of the domain, e.g., *foo*, also corresponds to "the identifier of the non-fungible token" in that it is the identifier of the deed corresponding to the ENS name, e.g., *foo.eth*. EX1002 at 135. And Johnson further explains that the *sha3* function is used by the *namehash* function when querying the ENS registry and the ENS resolver. EX1002 at 136. Indeed, the below Python code, reproduced from Johnson below, illustrates that the *namehash* function partitions the inputted domain name (e.g., *foo.eth*) into labels, and each label (e.g., *foo* and *eth*) is hashed using the *sha3* function:

```
def namehash(name):
  if name == '':
    return '\0' * 32
  else:
    label, _, remainder = name.partition('.')
    return sha3(namehash(remainder) + sha3(label))
```

EX1003 at 22. See also, EX1002 at 136; EX1005 at 4. Because the *sha3(foo)* portion of *namehash(foo.eth)* corresponds to the identifier of the deed for the *foo.eth* domain, the identifier of the deed, i.e., *sha3(foo)* is used in the process of requesting resolution of the *foo.eth* ENS name. EX1002 at 137. A PHOSITA would therefore understand that Johnson's use of the *namehash* of the ENS name (e.g., *namehash(foo.eth)*) to query the ENS registry is "using the identifier of the non-fungible token to send a request to a smart contract of the blockchain to obtain one or more resolution records for the domain" in the form of the *sha3()* of the domain (e.g., *sha3(foo)*) that is used within the *namehash* of the ENS name (e.g., *namehash(foo.eth)*). EX1002 at 138.

Johnson also discloses "wherein the blockchain stores the non-fungible token associating the domain to an account address of an owner of the domain," in that both of the deed and the registry record "associat[es] the domain to an account address of an owner of the domain." EX1002 at 139. More particularly, as discussed for *[1c]*, Johnson discloses that the blockchain stores the deed, which associates the domain to an owner account address at least when transferring the deed for the ENS "name" to a new owner (i.e., newDeedOwnerAddress). EX1002 at 140; EX1003 at 14. As shown in the annotated figure below, the owners are identified by a hexadecimal address, e.g., 0x1234…, 0x34567…, etc., that corresponds to an Ethereum wallet of the owner.



EX1003 at 6; EX1002 at 140.

Additionally, as discussed for *[1c]*, Johnson discloses that the ENS registry smart contract stores hashed ENS names mapped to the respective ENS resolvers responsible for them, as well as to the owners of the ENS names. EX1002 at 141; EX1003 at 14. Thus, Johnson discloses the limitation. EX1002 at 142.

[1e] – receiving the one or more resolution records of the domain

Johnson discloses this limitation in the form of the user code receiving from the resolver the resource associated with the queried ENS name. EX1002 at 143-146; EX1003 at 18-19. A PHOSITA would have understood that the returned resource corresponds to the "resolution record," because (a) the resolution record is described in the ’344 Patent as what is returned from the resolver, and (b) the resource of Johnson is what is returned from its ENS resolver. EX1002 at 144-146. *Compare* EX1001 at Col. 17:29-55 and EX1003 at 19.

Moreover, as discussed for *[1a]*, Millegan also discloses receiving a website address (e.g., a *.onion* address) as a result of the ENS resolution of an ENS name entered into its web browser. EX1002 at 147-149; EX 1004 at 2. As discussed in Sec. VIII.A., *supra,* a PHOSITA would have found it obvious to modify Johnson to use the modified web browser of Millegan to receive, as a result of ENS resolution, website addresses associated with ENS names entered into the web browser. EX1002 at 147-149. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 147-149.

[1f] – utilizing the received one or more resolution records to resolve the name of the domain

As discussed in *[1a]*, Millegan discloses “utilizing the received one or more resolution records to resolve the name of the domain,” in the form of utilizing the retrieved website address (e.g., a *.onion* address) to access the website address in lieu of the ENS name entered into the browser. EX1002 at 150. Millegan thus discloses the limitation. EX1002 at 150-152.

As discussed in Sec. VIII.A., a PHOSITA would have found it obvious to modify Johnson to use the modified web browser of Millegan to receive ENS names for ENS resolution into website addresses. EX1002 at 151. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 151.

2. Claim 2

The method of claim 1, wherein the identifier of the web content includes a Uniform Resource Locator (URL).

As discussed in *[1a]* and *[1f]*, Millegan discloses the "identifier of the web content" as both the ENS name, e.g., duckduckgotor.eth, and the returned website address, e.g., the *.onion* address. EX1002 at 154; EX1004 at 2. A PHOSITA would have understood both *duckduckgotor.eth* and the *.onion* address to be URLs. EX1002 at 155-156.

As discussed in Sec. VIII.A., a PHOSITA would have found it obvious to modify Johnson to use the modified web browser of Millegan to receive ENS names for resolution into website addresses, which are URLs. EX1002 at 157. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 157.

A PHOSITA would have thus found the limitations of Claim 2 obvious based on the combined teachings of Johnson and Millegan. EX1002 at 158.

3. Claim 3

The method of claim 1, wherein the request is received at a web browser.

Millegan discloses the limitation, as discussed in *[1a]*. EX1002 at 160; EX1004 at 1-2. As discussed in Sec. VIII.A., *supra,* a PHOSITA would have found it obvious to modify Johnson to use the modified web browser of Millegan to receive ENS names for ENS resolution into website addresses. EX1002 at 161. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 161-162.

A PHOSITA would have thus the limitations of Claim 3 obvious based on the combined teachings of Johnson and Millegan. EX1002 at 162.

4. Claim 4

The method of claim 1, wherein the request is received at a Domain Name System (DNS) gateway configured to handle resolution of domains of a blockchain-based name service.

Johnson describes that ENS also supports "hosting DNS domains on the Ethereum blockchain via ENS." EX1003 at 31; EX1002 at 164-165. To achieve this functionality, the "domain's nameserver records [can be configured] to point to gateway DNS servers . . . [which] consult[] an ENS registry which points to resolvers containing the zone data for the relevant domain." EX1003 at 31, EX1002 at 165. This step configures the gateway DNS server to handle resolution of ENS names by consulting the indicated ENS registry address. EX1002 at 165-167.

Accordingly, the "gateway DNS servers," "automatically determin[e] that the name of the domain is to be resolved using a blockchain" by "consulting an ENS registry which points to resolvers [deployed on Ethereum] containing the zone data for the relevant domain." EX1003 at 31; EX1002 at 165-167. A PHOSITA would have understood that, in order to perform this function, the DNS gateway would receive the request to handle the resolution of the domain name and carry out resolution of the ENS name by interacting with the ENS registry and resolver smart contracts in the same way as the "user code." EX1002 at 165-167; EX1003 at 18-19.

Thus, Johnson discloses that "the request is received at a Domain Name System (DNS) gateway configured to handle resolution of domains of a blockchain-based name service." EX1002 at 164-168.

5. Claim 6

> ***The method of claim 1, wherein the determination that the name of the domain is to be resolved using the blockchain is performed using an extension of a web browser.***

As discussed for *[1b]*, Millegan discloses that the web browser automatically determines that the entered ENS name is an ENS name for ENS resolution, explaining that "[a]fter you enter duckduckgotor.eth in the address bar of the Tor Browser, MetaMask recognizes that duckduckgotor.eth is an ENS name." EX1002 at 170-171; EX1004 at 2. The MetaMask extension then accesses the Ethereum blockchain and "does a look-up on ENS" to "grab[]" the website address. EX1002 at 171; EX1004 at 2. Thus, Millegan discloses "the determination that the name of the domain is to be resolved using the blockchain is performed using an extension of a web browser." EX1002 at 171.

As discussed in Sec. VIII.A., a PHOSITA would have found it obvious to modify Johnson to use the MetaMask extension modified web browser of Millegan to receive ENS names for ENS resolution into website addresses. EX1002 at 172. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 172.

A PHOSITA would have thus the limitations of Claim 6 obvious based on the combined teachings of Johnson and Millegan. EX1002 at 170-173.

6. Claim 7

The method of claim 1, wherein the identifier of the non-fungible token is calculated using a hash function.

As discussed for *[1c]*, Johnson discloses the non-fungible token as both the deed and as the data record of the ENS registry. EX1002 at 175. The identifier of the deed is the *sha3* hash of the domain (e.g., *foo*) corresponding to the ENS name (e.g., *foo.eth*). EX1002 at 176; EX1003 at 14. The identifier of the data record is the *namehash* of the ENS name (e.g., *foo.eth*). EX1002 at 176; EX1003 at 22.

Thus, Johnson discloses that "the identifier of the non-fungible token is calculated using a hash function." EX1002 at 175-177.

7. Claim 10

> ***The method of claim 1, wherein the one or more resolution records include a decentralized web content hash identifier.***

Johnson discloses that the resource returned by the ENS resolver can include a "Swarm content hash." EX1002 at 179; EX1003 at 7. A PHOSITA would have understood that Swarm is storage system for decentralized web content. EX1002 at 180. Thus, a PHOSITA would have understood that a "Swarm content hash" is a hash identifier of decentralized web content. EX1002 at 179-180. Johnson thus discloses the limitations of Claim 10. EX1002 at 181.

8. Claim 15

> ***The method of claim 1, wherein utilizing the received one or more resolution records to resolve the name of the domain includes providing a response including at least a portion of the one or more resolution records to a web browser.***

As discussed for *[1a]* and *[1f]*, Millegan discloses utilizing the retrieved website address (e.g., a *.onion* address) to access the website address in lieu of the ENS name entered into the browser. EX1002 at 183. Thus, Millegan discloses this limitation. EX1002 at 183-186.

As discussed in Sec. VIII.A., a PHOSITA would have found it obvious to modify Johnson to use the modified web browser of Millegan to receive ENS names for ENS resolution into website addresses. EX1002 at 184-185. A PHOSITA would recognize that this is a simple combination of known elements with predictable results. EX1002 at 184-185.

A PHOSITA would have thus the limitations of Claim 15 obvious based on the combined teachings of Johnson and Millegan. EX1002 at 183-186.

9. Independent Claim 19

[19-Pre] – A system, comprising:

The preamble is non-limiting, but Johnson discloses a system for resolving a domain name using the Ethereum Name Service (ENS). EX1003 at 7, 18; EX1002 at 187-189. Johnson describes a system that uses a method analogous to that of claim 1. EX1002 at 187-201.

[19a] – a processor configured to:

Johnson discloses a variety of software-based computer operations, including executable code provided in different programming languages including Javascript and Python. EX1002 at 190, 194; EX1003 at 9, 11, 20-22, and 25. A PHOSITA would have understood these software-based operations and functions would be executed by a processor. EX1002 at 190-195. The limitations of *[19-Pre]* are thus disclosed by Johnson. EX1002 at 196.

[19b] – receive a request to resolve a name of a domain of an identifier of web content

[19c] – automatically determine that the name of the domain is to be resolved using a blockchain

[19d] – determine an identifier of a non-fungible token corresponding to the domain of the identifier of the web content

[19e] – use the identifier of the non-fungible token to send a request to a smart contract of the blockchain to obtain one or more resolution records for the domain, wherein the blockchain stores the non-fungible token associating the domain to an account address of an owner of the domain

[19f] – receive the one or more resolution records of the domain

[19g] – utilize the received one or more resolution records to resolve the name of the domain

The limitations of *[19b]*-*[19g]* track those of *[1a]*-*[1f]*, and such limitations are taught by the combination of Johnson and Millegan for the same reasons as explained above. EX1002 at 197.

[19h] – a memory coupled to the processor and configured to provide the processor with instructions

Johnson discloses a variety of software-based computer operations and functions (i.e., instructions), including executable code provided in different programming languages including Javascript and Python. EX1002 at 198; EX1003 at 9, 11, 20-22, and 25. A PHOSITA would have understood these software-based operations and functions to be instructions that would be executed by a processor. EX1002 at 198. A PHOSITA would have further understood that software executed by the processor would be stored in a memory coupled to the processor and thereby be provided to the processor for execution. EX1002 at 198-200. The limitations of *[19h]* are thus disclosed by Johnson.

10. Independent Claim 20

[20-Pre] – A computer program product embodied in a non-transitory computer readable medium and comprising computer instructions for:

Johnson discloses computer instructions for resolving a name using the Ethereum Name Service (ENS), which is based on the Ethereum blockchain. EX1002 at 201-207; EX1003 at 5, 18. The computer instructions in Johnson are provided in Javascript and Python. EX1002 at 201-207; EX1003 at 25. A PHOSITA would have understood that such computer instructions are embodied in a non-transitory computer readable medium. EX1002 at 201-206. The limitations of *[20-Pre]* are this disclosed by Johnson. EX1003 at 207.

[20a] – receiving a request to resolve a name of a domain of an identifier of web content

[20b] – automatically determining that the name of the domain is to be resolved using a blockchain

[20c] – determining an identifier of a non-fungible token corresponding to the domain of the identifier of the web content

[20d] – using the identifier of the non-fungible token to send a request to a smart contract of the blockchain to obtain one or more resolution records for the domain, wherein the blockchain stores the non-fungible token associating the domain to an account address of an owner of the domain

[20e] – receiving the one or more resolution records of the domain

[20f] – utilizing the received one or more resolution records to resolve the name of the domain

The limitations of *[20a]*-*[20f]* track those of *[1a]*-*[1f]*, and such limitations are taught by the combination of Johnson and Millegan for the same reasons as explained above. EX1002 at 208.

B. **<u>GROUND 2:</u>** **Claim 5 is obvious under 35 U.S.C. §103 over Johnson in view of Millegan and further in view of Schneider**

1. Claim 5:

> ***The method of claim 1, wherein determining that the name of the domain is to be resolved using the blockchain includes determining that a top-level domain included in the name of the domain matches an entry in a list of domains to be resolved using the blockchain.***

As discussed in *[1b]*, Millegan discloses that the web browser automatically determines that the entered ENS name is an ENS name (e.g., .eth) for ENS resolution. EX1002 at 210; EX1004 at 2.

Schneider discloses determining whether a domain name is resolvable by comparing it against a list of resolvable top-level domains. EX1011 at ¶ [0056]; EX1002 at 211. A PHOSITA would have understood that this feature allows a DNS server or gateway to quickly determine whether the request to resolve the domain name requires alternative processes to resolve. EX1002 at 212.

Thus, it would have been obvious to a PHOSITA to implement this same technique of Schneider to an ENS name resolving system using the blockchain as described by Johnson and Millegan. EX1002 at 213. Importantly, an ENS name is a domain name like traditional domain names on DNS servers, except that it resolves to a resource located on the Ethereum blockchain instead of locally on a server, for example. EX1002 at 214. Schneider is therefore analogous art in the same field of domain name resolution and teaches the added benefit of rapidly determining whether a domain name is resolvable by using a list of resolvable top-level domains. EX1002 at 214-215. Combining the teachings of Schneider with Johnson and Millegan would apply a known technique (comparing a candidate domain name with a list) to improve similar systems (blockchain-based and DNS-based name resolution services) in the same way. *See KSR*, 550 U.S. at 417 . EX1002 at 215-216.

A PHOSITA would have thus found the limitations of Claim 5 obvious based on the combined teachings of Johnson, Millegan and Schneider. EX1002 at 216.

C. <u>GROUND 3:</u> Claim 9 is obvious under 35 U.S.C. §103 over Johnson in view of Millegan and further in view of Moore

1. Claim 9:

The method of claim 1, wherein the one or more resolution records include a redirect address.

As discussed in *[1d]-[1f]*, Johnson and Millegan disclose receiving from the resolver the "resolution record" in the form of the returned resource associated with the queried ENS name. EX1002 at 218; EX1004 at 2.

Moore discloses that the stored resource returned by the resolve can be any "arbitrary key-value text data," such as "a website URL." EX1002 at 219; EX1008 at 2. A PHOSITA would have understood that a URL, rather than an IP address, resolved from an ENS name is a redirect address, at least because the web browser is redirected to the URL via the ENS resolution. EX1002 at 219. That a PHOSITA would have understood that resolving a domain name to another URL instead of a website address constitutes redirection to the website address of the URL instead of the original domain name is supported by the patent publications at the time. EX1002 at 220. For example, Amini (EX1009) explains that "DNS can provide redirection when the client device attempts to resolve an IP domain name into an IP address" and that "the DNS server can resolve this name to an alternate, or alias, domain name. . . [which] is often referred to as DNS CNAME (canonical Name) redirection." EX1009 at ¶¶ [0004], [0005]; EX1002 at 220.

A PHOSITA would have combined the teachings of Moore, as evidenced by Amini, which are analogous art in the field of domain name resolution, with the ENS system as disclosed by Johnson and Millegan at least because Moore expressly relates to improvements in ENS in "defin[ing] a resolver profile for ENS that permits the lookup of arbitrary key-value text data." EX1008 at 1; EX1002 at 220-222. A PHOSITA would readily identify an ENS name that resolves to an URL as a redirect address based on the similarities to CNAME redirection found in conventional DNS systems. EX1002 at 220-222.

A PHOSITA would have thus found the limitations of Claim 9 obvious based on the combined teachings of Johnson, Millegan and Moore. EX1002 at 222.

D. <u>GROUND 4:</u> Claims 8 and 11 are obvious under 35 U.S.C. §103 over Johnson in view of Millegan and further in view of EIP137.

1. Claim 8:

> ***The method of claim 1, wherein the one or more resolution records include an Internet Protocol (IP) address.***

As discussed in *[1d]*-*[1f]*, Johnson and Millegan disclose receiving the "resolution record" in the form of the returned resource associated with the queried ENS name. Moreover, as discussed in *[1a]* and *[1f]*, Millegan describes the returned resource as a website address (e.g., a *.onion* address). EX1002 at 224; EX1004 at 2.

EIP137 describes ENS resolvers that "are responsible for performing resource lookups for a name – for instance, returning a[n] . . . IP address(es) as appropriate." EX1005 at 1, 3; EX1002 at 225.

In view of the teaching of EIP137 that the ENS resource returned by the resolver includes IP addresses, a PHOSITA would have found it obvious to return, as the resource of Johnson and Millegan, the website address in the form of an IP address. EX1002 at 226. Indeed, website addresses are commonly represented by their IP addresses. EX1002 at 29.

Moreover, a PHOSITA would have referred to the teachings of EIP137 that discloses features proposed for the ENS system as disclosed by Johnson and Millegan at least because Johnson expressly refers to the EIP137 document. EX1002 at 227. For example, Johnson expressly references EIP137 as a source of information for "a list of currently recognised [*sic*] resource types" and as a "Resource" for further reading. EX1003 at 19 and 7. Accordingly, a PHOSITA would have combined Johnson and Millegan with the teachings of EIP137—as directed by Johnson—to incorporate the additional functionality of the ENS system described by Johnson, such as adding the return of IP addresses as disclosed by EIP137. EX1002 at 227-229.

A PHOSITA would have thus found the limitations of Claim 8 obvious based on the combined teachings of Johnson, Millegan and EIP137. EX1002 at 229.

2. Claim 11:

The method of claim 1, wherein the one or more resolution records include a plurality of target identifiers for different protocols.

As discussed in *[1d]-[1f]*, Johnson and Millegan disclose receiving from the resolver the "resolution record" in the form of the returned resource associated with the queried ENS name. EX1002 at 231; EX1004 at 2.

EIP137 discloses that the resource returned by the resolver may include a plurality of target identifiers for different protocols, including "a DApp hosted in

Swarm, a Whisper address, and a mail server." EX1005 at 2; EX1002 at 232. Additional example target identifiers are provided in the table for returning a contract address, reverse ENS name lookup, contract application binary interface (ABI), and SECP256k1 public keys lookup. EX1005 at 6-7; EX1002 at 232.

A PHOSITA would have found it obvious to combine the teachings of EIP137 with Johnson and Millegan such that the one or more resolution records include a plurality of target identifiers for different protocols, as described by EIP137. EX1002 at 233. A PHOSITA would have been motivated to make this combination at least because Johnson expressly refers to EIP137. EX1002 at 234. For example, Johnson references EIP137 as a source of information for "a list of currently recognised [*sic*] resource types" and as a "Resource" for further reading. EX1003 at 19 and 7. A PHOSITA would also appreciate that including a plurality of target identifiers for different protocols "allow the same domain [name] to reference different resources. . . . [such as resolving] 'mysite.swarm' to the IP address of its server . . . [or] resolv[ing] the same address to a mail server." EX1005 at 1. Accordingly, a PHOSITA would combine Johnson and Millegan with the teachings of EIP137 as directed by Johnson to improve the functionality of the ENS system, such as by using a plurality of target identifiers for different protocols and services as described by EIP137. EX1002 at 235-235.

A PHOSITA would have thus found the limitations of Claim 11 obvious based on the combined teachings of Johnson, Millegan and EIP137. EX1002 at 236.

E. <u>GROUND 5:</u> Claims 12-14 are obvious under 35 U.S.C. §103 over Johnson in view of Millegan, EIP137, and further in view of Haynes

1. Claim 12:

> ***The method of claim 11, wherein utilizing the received one or more resolution records to resolve the name of the domain includes selecting one of the plurality of target identifiers based on an ordered list of protocol preferences.***

As discussed in *[1a]* and *[1f]*, Johnson in view of Millegan discloses utilizing the retrieved website address (e.g., a *.onion* address) to access the website address in lieu of the ENS name entered into the browser. EX1002 at 238. Moreover, as discussed in Ground 3 with respect to claim 11, EIP137 discloses that the resolution records of the resolver may include a plurality of target identifiers for different protocols and services. EX1002 at 238; EX1005 at 2, 6, and 7.

Haynes, in the field of wireless internet communications, further discloses "select[ing] a tunneling protocol from the list of tunneling protocols based on the prioritized list." EX1010 at ¶ [0044]; EX1002 at 239-240. The list "specifies an order in which user device 110 should use [one of several] IPsec tunneling protocol components." EX1010 at ¶ [0033]; EX1002 at 239-240. In this way, the user device 110 may improve the likelihood of successfully establishing a connection with the evolved packet data gateway (ePDG) device using an IPsec protocol that is supported by the ePDG device to retrieve applications from an IP address. EX1010 at ¶¶ [0036] and [0044]; EX1002 at 239-240.

A PHOSITA would have found it obvious to incorporate the concept of an "ordered list of protocol preferences" as disclosed in Haynes into the ENS system of Johnson, Millegan, and EIP137 to prioritize the "plurality of target identifiers for different protocols." EX1002 at 241. Prioritizing the list target identifiers has the advantage to increase the likelihood of successfully connecting with a target device using a protocol that is known to be supported and/or compatible. EX1002 at 242. Such a modification amounts to applying a known technique (the prioritized protocol list of Haynes) to improve similar devices (computer devices that communicate via IP addresses) in the same way (by selecting a protocol from the prioritized list to improve the likelihood of a successful connection with another device). EX1002 at 243-243.

A PHOSITA would have thus found the limitations of Claim 12 obvious based on the combined teachings of Johnson, Millegan, EIP137 and Haynes. EX1002 at 244.

2. Claim 13:

> ***The method of claim 12, wherein the ordered list of protocol preferences is at least in part specified in the one or more resolution records received using the blockchain.***

As discussed in *[1e]*, Johnson and Millegan disclose receiving from the resolver, via ENS resolution on the Ethereum blockchain, the resource associated with the queried ENS name—which corresponds to the "resolution record." EX1002 at 246. Moreover, as discussed above with reference to Claim 12, the combination of Johnson, Millegan, EIP137, and Haynes provides for the use of an ordered list of protocols in the one or more resolution records of the resolver. EX1002 at 247.

A PHOSITA would have found it obvious to include the ordered list of protocol preferences in the resolution records of Johnson, Millegan, and EIP137 because EIP137 teaches that the target identifiers are in the resolution records of the resolver already. EX1002 at 248-249; EX1005 at 6-7. A PHOSITA would have been motivated by the suggestion in Johnson to maintain efficient data storage and quick data retrieval by having the ordered list of protocols as taught by Haynes in the same data records where the target identifiers are stored, as EIP137 teaches the target identifiers are in the resolution records. EX1002 at 249. This combination applies a known technique to a known system ready for improvement to yield predictable results. *See KSR*, 550 U.S. at 417 . EX1002 at 249.

Additionally, only a finite number of locations exist within the ENS system where the ordered list of protocol preferences could be stored, including within the resolution records of the resolver, within the registry, or within the user code (e.g., web browser). EX1002 at 250. A PHOSITA could choose from any one of these identified, predictable storage locations with a reasonable expectation of success, which also supports a conclusion of obviousness. EX1002 at 250-251.

A PHOSITA would have thus found the limitations of Claim 2 obvious based on the combined teachings of Johnson, Millegan, EIP137 and Haynes. EX1002 at 251.

3. Claim 14:

> ***The method of claim 12, wherein selecting the one of the plurality of target identifiers based on the ordered list of protocol preferences includes evaluating each protocol specified in an order of the ordered list to determine whether there exists a specific target record for the protocol.***

As discussed above with reference to Claim 12, the combination of Johnson, Millegan, EIP137, and Haynes provides for the use of an ordered list of protocols. EX1002 at 253. Haynes discloses the remaining limitations in the form of evaluating whether the protocol "listed first in the prioritized list of tunneling protocols . . . [provided that the protocol] has not been already unsuccessfully used by user device 110 to attempt to establish the connection." EX1002 at 254; EX1010 at ¶¶ [0044] and [0045].

A PHOSITA would have found it obvious to include evaluating the target identifiers in the resolution records of EIP137 in a specified order to determine whether a target record exists for the protocol as taught by Haynes. EX1002 at 255-258. A PHOSITA would have been motivated to evaluate the target identifiers of EIP137 in a prioritized way as taught by Haynes to minimize processing of target records that do not exist. EX1002 at 255-258. This combination applies the known technique of Haynes to the known system of Johnson, Millegan, and EIP137, which is ready for improvement, to yield predictable results. *See KSR*, 550 U.S. at 417 . EX1002 at 255-258.

A PHOSITA would have thus found the limitations of Claim 14 obvious based on the combined teachings of Johnson, Millegan, EIP137 and Haynes. EX1002 at 258.

F. <u>GROUND 6:</u> Claims 16-18 are obvious under 35 U.S.C. §103 over Johnson in view of Millegan and further in view of Saini

1. Claim 16:

> ***The method of claim 1, wherein utilizing the received one or more resolution records to resolve the name of the domain includes providing an identifier of a decentralized web gateway.***

As discussed in *[1e]*, the combination of Johnson and Millegan discloses receiving from the resolver the resource associated with the queried ENS name. EX1002 at 260; EX1003 at 18-19.

Saini discloses the remaining limitations in the form of describing that Swarm is "designed to deeply integrate with . . . the Ethereum blockchain for domain name resolution (using ENS)." EX1002 at 261; EX1012 at 3. Saini further explains that "ENS is the system that Swarm uses to permit content to be referred to by a human-readable name, such as 'theswarm.eth.'" EX1012 at 5; EX1002 at 261. Saini also explains that Swarm provides a "public gateway [that] can be found at https://swarm-gateways.net," which would have been understood by a PHOSITA to correspond to an identifier of a decentralized web gateway. EX1012 at 5; EX1002 at 261. Saini discloses that Swarm content referred to by the example name "theswarm.eth," may be accessed through "major browsers such as Chrome, Firefox or Safari" using ENS. EX1012 at 5; EX1002 at 261. To do so, the Swarm public gateway can be provided in conjunction with the ENS name; e.g., "https://swarm-gateways.net/bzz://theswarm.eth/". EX1012 at 5; EX1002 at 261.

A PHOSITA would have combined the teachings of Saini with Johnson and Millegan such that resolving the ENS name includes providing an identifier of a Swarm gateway. EX1002 at 262. There is express motivation for this combination in both Johnson and Saini. EX1002 at 263. As previously explained with reference to claim 10, Johnson describes that the resolution records of ENS can include a "Swarm content hash." EX1002 at 263. Saini expressly describes using

ENS with Swarm for the advantage of using a web browser to access Swarm using decentralized web gateway. EX1012 at 3, 5, and 6; EX1002 at 263. Accordingly, a PHOSITA would have found it obvious to utilize a decentralized gateway to access the Swarm content, as taught by Saini, as the received resource. EX1002 at 264.

A PHOSITA would have thus found the limitations of Claim 16 obvious based on the combined teachings of Johnson, Millegan and Saini. EX1002 at 265.

2. Claim 17:

> ***The method of claim 16, wherein in response to a request made to the decentralized web gateway using the provided identifier of the decentralized web gateway, the decentralized web gateway obtains from a decentralized content network, decentralized web content referenced by a decentralized web content hash value included in the one or more resolution records of the domain, and stores and provides the obtained decentralized web content.***

The combination of Johnson, Millegan, and Saini provide for the limitations of Claim 17 that are present in Claim 16, as discussed with respect to Claim 16. EX1002 at 266.

Saini also discloses the remaining limitations in the form of a request made to the public Swarm gateway using the ENS identifier that corresponds to the Swarm content hash. EX1002 at 267; EX1012 at 5. In response, the Swarm public gateway obtains decentralized web content from the Swarm network as "chunks of data" that are reassembled locally. EX1012 at 5; EX1002 at 267. Saini further teaches that the decentralized web content is referenced by the "Swarm hash [value] of the content," (i.e., a decentralized web content hash value) which is "content to be referred to by a human-readable name, such as 'theswarm.eth'" using the ENS system. EX1012 at 5; EX1002 at 267.

As discussed for Claim 10, Johnson teaches that ENS can be configured to return, via the resolver, a Swarm content hash value of the human-readable domain name within one or more resolution records. EX1003 at 7, 18; EX1002 at 268. Thereby, the decentralized web content is retrieved and may then be stored and provided to the browser using the Swarm public gateway. EX1012 at 5; EX1002 at 268.

Accordingly, because Johnson, Millegan, and Saini expressly disclose combining its features with the ENS system of Johnson according to known predictable methods, a PHOSITA have found it obvious to retrieve data from a Swarm site using a decentralized web content hash value included in the resolution records of the domain as taught by Johnson and Millegan in view of Saini. *See KSR*, 550 U.S. at 417 ; EX1002 at 269-271.

A PHOSITA would have thus found the limitations of Claim 17 obvious based on the combined teachings of Johnson, Millegan and Saini. EX1002 at 271.

3. Claim 18:

> ***The method of claim 17, wherein the decentralized web gateway obtains the content referenced by the decentralized web content hash value using a decentralized web protocol but provides the obtained decentralized web content using a protocol that is not the decentralized web protocol.***

The combination of Johnson, Millegan, and Saini provide for the limitations of Claim 18 that are present in Claim 16, as discussed with respect to Claim 16. EX1002 at 272.

Saini discloses the remaining limitations in the form of using the Swarm public gateway on a web browser to retrieve, via the bzz protocol, content referenced by the Swarm content hash value (the decentralized web content hash value). EX1002 at 273; EX1012 at 4. The browser then communicates with the Swarm public gateway and provides the decentralized content using the HTTP protocol, which is not the Swarm-native bzz protocol. EX1012 at 5; EX1002 at 273.

Accordingly, because Millegan, and Saini each expressly disclose combining its features with the ENS system of Johnson according to known predictable methods, a PHOSITA have found it obvious to retrieve data from a Swarm site using a decentralized web protocol and providing the content using a different protocol as taught by Johnson and Millegan in view of Saini. EX1002 at 274-275.

A PHOSITA would have thus found the limitations of Claim 18 obvious based on the combined teachings of Johnson, Millegan and Saini. EX1002 at 276.

G. NO SECONDARY CONSIDERATIONS

Petitioner is not aware of any secondary considerations that would overcome the grounds of obviousness presented above. In particular, there is no evidence of secondary considerations relied on during prosecution. EX1013. As such, the strong showing of unpatentability presented in the above Grounds is not overcome by secondary considerations. EX1002 at 277-279.

H. NO BASIS FOR DISCRETIONARY DENIAL

In evaluating whether to exercise its discretion under § 325(d), the Board considers: (1) whether the Petition presents the same or substantially the same art/argument previously presented; and, if so, (2) whether the Office materially erred in its prior evaluation. *Advanced Bionics, LLC v. Med-El Elektromedizinische Gerate GmbH*, IPR2019-01469, Paper 6 at 8 (Feb. 13, 2020) (precedential) ("*Advanced Bionics*").

Step (1) is not satisfied here. None of the grounds relied upon are the same or substantially the same art/arguments presented during the examination of the 344 Patent. Only one piece of prior art relied upon here, EIP137, was of record during examination, and it was not relied upon in the rejections of the pending claims. Even so, only Grounds 3 and 5, which are directed to dependent claims, rely upon EIP137.

The combinations of prior art presented in this Petition were therefore not previously presented to the PTO, nor is it reasonably cumulative with the prior art of record. *See Intex Recreation Corp. v. Team Worldwide Corp*., IPR2018-00871, Paper 14, at 13 (PTAB Sept. 14, 2018) (cited references not discussed or used in rejection "weighs against exercising discretion to deny under § 325(d)" (citation omitted)); *see also Apple Inc. v. Qualcomm Inc.,* IPR2018-01315, Paper 7, at 22-26 (PTAB Jan. 18, 2019).

Because the result of Step (1) of the *Advanced Bionics* inquiry is that the Petition does not present the same or substantially the same art/argument previously presented to the PTO, the inquiry ends. Nevertheless, Step (2) is also not satisfied here because the PTO materially erred by concluding that the limitations: "determining an identifier of a non-fungible token corresponding to the domain of the identifier of the web content," and "using the identifier of the non-fungible token to send a request to a smart contract of the blockchain to obtain one or more resolution records for the domain," were not in the prior art. EX1013 at p. 31. But in making its determination, the Examiner did not have Johnson and Millegan to consider nor did the Examiner have the benefit of Dr. McDonald's expert analysis. Indeed, as set forth above, Johnson and Millegan disclose or render obvious these limitations—and alone or in combination with one or more of the remaining references presented herein discloses or renders obvious all the limitations of each of the challenged claims. *See generally* Section VIII. *See also, e.g.*, *Advanced Energy Industries v. Reno Technologies*, IPR2021-01397, Paper 7, 7 (Feb. 16, 2022); *Lyft, Inc. v. RideShare Displays, Inc*., IPR2021-01602, Paper 7, 14-15 (April 11, 2022); *Samsung Electronics Co., Ltd. et al. v. Power2B, Inc*., IPR2021-01190, Paper 11, 19 (Jan. 6, 2022).

IX. MANDATORY NOTICES UNDER 37 C.F.R. §42.8(A)(1)

A. Real Party-in-Interest

For purposes of 35 U.S.C. §312(a)(2) and 37 C.F.R. §42.8(b)(1) only, Petitioners identify the real-parties-in-interest as ENS Labs Ltd.

B. Related Matters

Petitioners are not aware of any related litigation. Petitioners are not aware of any reexamination certificates or reissue applications pending for the '344 Patent. Petitioner is aware of the following related patents and/or patent applications: U.S. Patent No. 11,875,774; and U.S. Patent Application No. 18/513,214.

C. Lead and Back-Up Counsel

Lead Counsel	Back-Up Counsel
Robert Mallin (Reg. No. 35,596) rmallin@crowell.com Crowell & Moring LLP Suite 3600 NBC Tower 455 Cityfront Plaza Drive Chicago IL 60611-5599 T: 312-321-4200, F: 202-628-5116	William Frankel (Reg. No. 30,337) wfrankel@crowell.com Marc V. Richards (Reg. No. 37,921) mrichards@crowell.com Crowell & Moring LLP Suite 3600 NBC Tower 455 Cityfront Plaza Drive Chicago IL 60611-5599

Lead Counsel	Back-Up Counsel
	T: 312-321-4200, F: 202-628-5116 Kainoa Asuega (Reg. No. 64,271) kasuega@crowell.com Crowell & Moring LLP 3 Park Plaza, 20th Floor Irvine, CA 92614 T: 949-798-1379, F: 949-263-8414 Matthew David (Reg. No. 77,058) mdavid@crowell.com Crowell & Moring LLP 1001 Pennsylvania Ave, NW Washington, DC 20004 T: 202-624-2500, F: 202-628-5116

D. Service, Payment and Deposit Account Information

Please address all correspondence to the lead and backup counsel at the contact information above. Petitioners consent to service by email at federalcourts@crowell.com, rmallin@crowell.com, wfrankel@crowell.com, mrichards@crowell.com, kasuega@crowell.com and mdavid@crowell.com.

Petitioners authorize the Patent and Trademark Office to charge Deposit Account No. 05-1323 for the fees required by 37 C.F.R. §42.15(a) for this Petition, and authorize payment of any additional fees to be charged to this Deposit Account.

X. GROUNDS FOR STANDING UNDER 37 C.F.R. §42.104(A)

Petitioners certify that the ’344 Patent is available for IPR and that Petitioners are not barred or estopped from requesting IPR on any ground.

XI. CONCLUSION

For the foregoing reasons, Petitioners request institution of an *inter partes* review to cancel claims 1-20 of the ’344 Patent.

Dated: May 2, 2024

Respectfully submitted,

/Robert Mallin/

Robert Mallin, Reg. No. 35,596
Lead counsel for Petitioner

CERTIFICATE OF COMPLIANCE

The undersigned certifies that this brief complies with the type-volume limitations of 37 CFR §42.24(a)(1)(i). This brief, not including mandatory notices and certificates, contains 12,702 words as calculated by the "Word Count" feature of Microsoft Word 2019, the word processing program used to create it.

The undersigned further certifies that this brief complies with the typeface requirements of 37 CFR §42.6(a)(2)(ii) and typestyle requirements of 37 CFR §42.6(a)(2)(iii). This brief has been prepared in a proportionally spaced typeface using Microsoft Word 2019 in Times New Roman 14-point font.

DATED: May 2, 2024

CROWELL & MORING LLP

By: */Robert Mallin/*

Robert Mallin
Lead Counsel for Petitioner
Reg. No. 35,596
Crowell & Moring LLP
Suite 3600 NBC Tower
455 Cityfront Plaza Drive
Chicago IL 60611-5599
T: 312-321-4200
F: 202-628-5116

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on May 2, 2024, a complete and entire copy of this Petition for *Inter Partes* Review, and all supporting exhibits, were served EXPRESS MAIL® to the Patent Owner by serving the correspondent of record, as indicated below:

Patent Owner's Address of Record:
VAN PELT, YI & JAMES LLP
1 1ST ST, STE 12
LOS ALTOS, CA 94022

Dated: May 2, 2024

/Robert Mallin/

Robert Mallin (Reg. No. 35,596)
William H. Frankel (Reg. No. 30,337)
Marc V Richards (Reg. No. 37,921)
Kainoa Asuega (Reg. No. 64,271)
Matthew David (Reg. No. 77,058)
Crowell & Moring LLP
Suite 3600 NBC Tower
455 Cityfront Plaza Drive
Chicago IL 60611-5599
T: 312-321-4200
F: 202-628-5116
Attorneys for Petitioner