Blockchain-based
Web-scale Data Ownership Management Technology

Universal Resource Auth (UR-Auth)

Protocol and System Architecture

URI + DID = UR-Auth

You are authenticated / authorized / authentic
with your **UR-Auth** on public blockchains

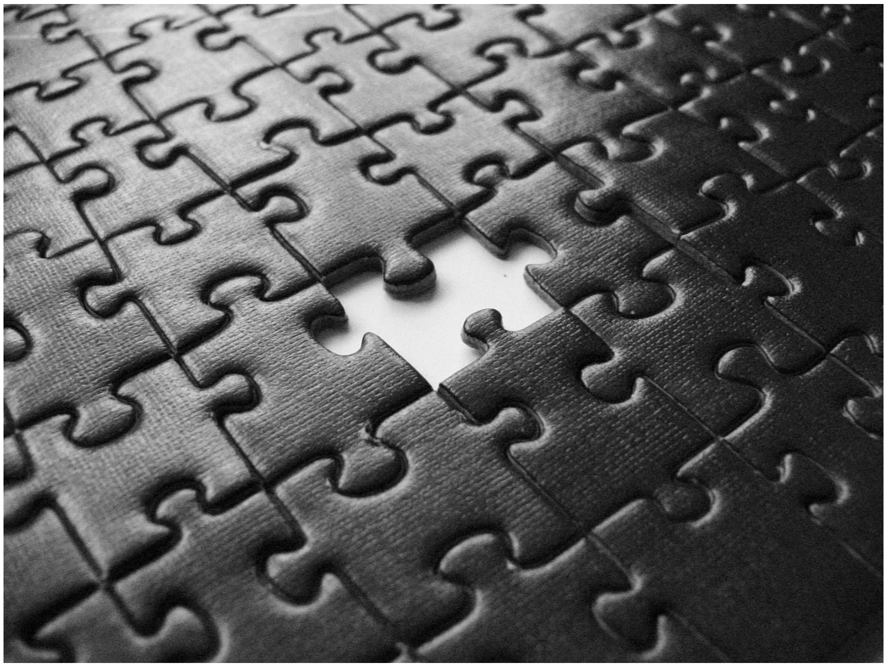
`Technical White Paper v0.7c`

July 2023

Blockchain Labs

Table of Contents

1. The Missing Puzzle Piece of Current Web



AI products such as ChatGPT, Bard, LLaMA, Dall-E, Midjourney implemented with generative AI technologies[1] from big tech AI companies are gaining global attention, and the AI technology and market are showing rapid growth. However, behind the impressive results shown by the rapidly growing AI technology, there is the serious data copyright problem in which large amounts of online/offline data are being used without permission for AI machine learning. The machine learning processes of Large-scale language models (LLM)[2] and image generation AI models require massive amounts of human-created text, image, and video data. AI technology companies have collected large volumes of data, such as news articles, blogs, social media posts, image and video content sites, online communities, books, papers, and programming source codes, without permission from the data copyright holders, to create commercial generative AI models. Large-scale lawsuits and opposition movements against generative AI continue to arise, leading to the preparation of regulatory bills related to the data provenance of AI models from worldwide

[1] Generative artificial intelligence - https://en.wikipedia.org/wiki/Generative_artificial_intelligence
[2] Large language model - https://en.wikipedia.org/wiki/Large_language_model

nations. The European Union (EU) and the United States have taken the lead in proposing AI regulatory bills. In the case of the EU, they will strongly regulate the use of copyright data by mandating the disclosure of the sources and copyrights of data used in generative AI model training.[3]

The fast and free circulation of information has been realized through internet technology; however, the current web technology lacks standard infrastructure technology that enables authentication of data ownership, secure transaction of data, and tracking data distribution. While simply entering a URL into a web browser allows easy access to web content pointed by that URL, there exists no standardized technical method to verify copyright for the data referenced by that URL, and pay appropriate compensation to copyright holders for using the data. Even if AI regulations are enacted, generative AI companies are in a situation where they lack the technological means to verify copyright for collected data, pay fair price for data usage, and comply with these regulations.

Blockchain technology can enable web-scale data ownership management and secure data distribution and transactions. This technical white paper introduces the Universal Resource Auth (UR-Auth) protocol that allows web data and copyrighted tangible and intangible assets referenced by Uniform Resource Identifiers (URI)[4] to be publicly registered on the internet using blockchain technology and Decentralized Identifiers (DID)[5] technology. Data producers such as website owners and web service users (such as SNS users) can create their own blockchain ID (DID) and register ownership, copyright information, data access rules and prices on the blockchain for their data being served on the web. They can also directly register newly created documents, images, audio, and video data on the blockchain for copyright protection. This enables anyone to transparently verify data ownership, access rules, and transaction history registered on the blockchain. The records of data ownership, data access and transaction history on the blockchain can serve as legal evidence in case of disputes related to data ownership.

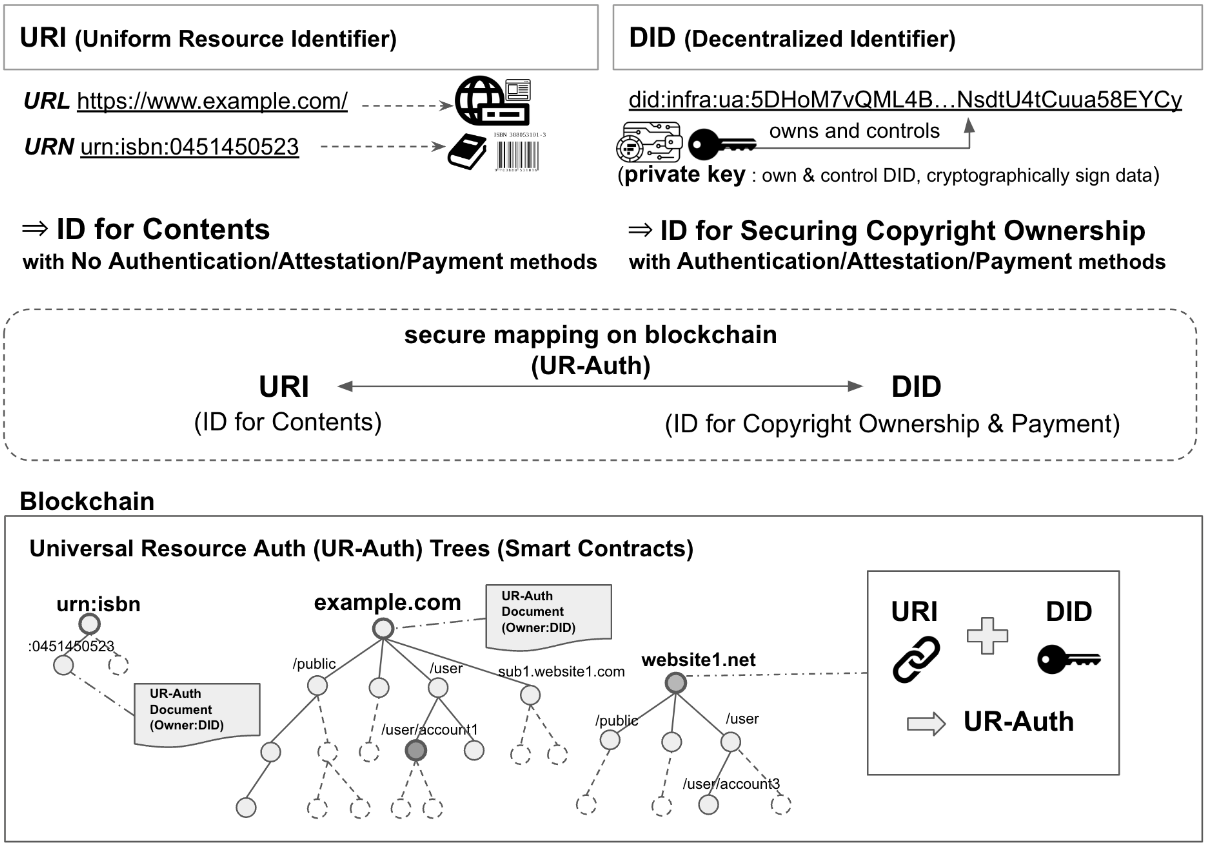
[3] Regulation of artificial intelligence - https://en.wikipedia.org/wiki/Regulation_of_artificial_intelligence
[4] Uniform Resource Identifier - https://en.wikipedia.org/wiki/Uniform_Resource_Identifier
[5] Decentralized Identifiers (DIDs) v1.0 - https://www.w3.org/TR/did-core/

Data consumers, such as AI technology companies and search engines, must access/purchase data in compliance with copyright information and data access rules published on the blockchain. They should record data access history transparently on the blockchain, and pay data access/purchasing costs directly to copyright holders using blockchain technology. Additionally, a blockchain-based data market where pre-packaged datasets refined for AI model training and similar purposes are registered and traded with data ownership information on the blockchain is implemented. This enables data consumers to easily search and safely use high-quality data without the risk of disputes.

InfraBlockchain technology, a public multi-blockchain without native cryptocurrency, is utilized to implement the UR-Auth protocol. Trusted entities such as news, media companies, relevant associations, government agencies, and financial institutions participate as validators of blockchain networks. These trusted entities can issue stable tokens based on legal currency to process blockchain transaction fees and handle payments for data transactions, creating a public blockchain network for data copyright management and transactions. To realize web-scale data ownership registration and data access/purchase management, a multi-blockchain architecture in which connected multiple blockchains operate simultaneously is required to enable concurrent large-scale blockchain transactions.

2. Universal Resource Auth (UR-Auth) Protocol

2.1 URI + DID = UR-Auth


URI (Uniform Resource Identifier)
URL https://www.example.com/
URN urn:isbn:0451450523
⇒ ID for Contents
with No Authentication/Attestation/Payment methods
DID (Decentralized Identifier)
did:infra:ua:5DHoM7vQML4B…NsdtU4tCuua58EYCy
owns and controls
(private key : own & control DID, cryptographically sign data)
⇒ ID for Securing Copyright Ownership
with Authentication/Attestation/Payment methods
secure mapping on blockchain
(UR-Auth)
URI
(ID for Contents)
DID
(ID for Copyright Ownership & Payment)
Blockchain
Universal Resource Auth (UR-Auth) Trees (Smart Contracts)
urn:isbn
:0451450523
UR-Auth Document (Owner:DID)
example.com
UR-Auth Document (Owner:DID)
/public
/user
sub1.website1.com
/user/account1
website1.net
/public
/user
/user/account3
URI
DID
UR-Auth

[Figure 1] URI, DID, UR-Auth

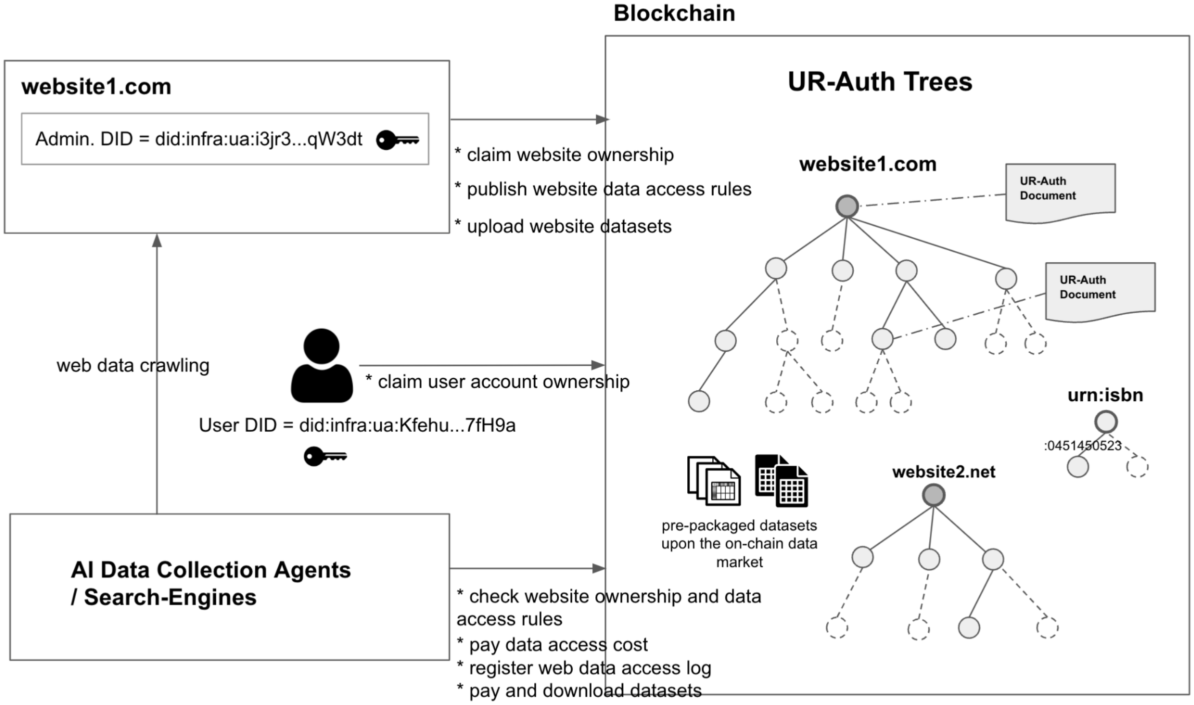
URI is the identifier scheme commonly used in web technology to uniquely identify logical and physical resources both online and offline. URI is divided into URL (Uniform Resource Locator) and URN (Uniform Resource Name). A URL such as https://www.example.com/public/page.html can identify the content being served on a web server, while a URN such as urn:isbn:1803241063 can identify the published book identified by its ISBN (International Standard Book Number)[6]. Simple identifiers such as URI do not have inherent authentication capabilities (there is no way to prove that the content owner possesses that identifier), so functions like digital signatures and payments cannot be implemented using the URI identifier scheme alone. A URI is just a pointer to a logical or physical resource online or offline. Therefore, to authenticate ownership of the resources that URIs point to, separate authentication systems operated by a third party (e.g. ID/Password-based authentication on a web server system) is required in the current internet.

Decentralized Identifiers (DID)[7] enable authentication, cryptographic digital signatures, and payment functions within the identifier scheme itself using public-key cryptography technology[8], without relying on centralized third-party authentication systems. Instead of using ID/Password stored on a centralized server, users create their DID directly on their devices (generating a (Private Key, Public Key) pair through the user device's blockchain wallet). The Public Key of the generated DID is either encoded into the DID identifier itself or stored in a DID Document registered on the blockchain, making it accessible for anyone to retrieve the Public Key of the DID. The Private Key is stored only on the user's device (digital blockchain wallet), allowing users to independently generate cryptographic digital signatures and perform authentication based only on the blockchain without relying on a third-party authentication system. With the Private Key, users can also create blockchain transactions to modify information on the blockchain or process payments.

Currently, URIs used on the web serve the purpose of content identification, yet they lack authentication capabilities, so there is no technical method to manage ownership of content identified by URIs. By integrating self-sovereign authentication functionality of DID into the URI identifiers through the blockchain-based *Universal Resource Auth* protocol, it's possible to implement ownership management, access control and blockchain-based

[6] ISBN, International Standard Book Number - https://en.wikipedia.org/wiki/ISBN
[7] Decentralized Identifiers (DIDs) v1.0 - https://www.w3.org/TR/did-core/
[8] Public-key Cryptography - https://en.wikipedia.org/wiki/Public-key_cryptography

resource transactions for web contents and tangible/intangible offline resources identified by URIs without relying on centralized third-party authentication systems.


Blockchain
UR-Auth Trees
website1.com
Admin. DID = did:infra:ua:i3jr3...qW3dt
* claim website ownership
* publish website data access rules
* upload website datasets
website1.com
UR-Auth Document
UR-Auth Document
web data crawling
* claim user account ownership
User DID = did:infra:ua:Kfehu...7fH9a
urn:isbn
:0451450523
website2.net
pre-packaged datasets upon the on-chain data market
AI Data Collection Agents / Search-Engines
* check website ownership and data access rules
* pay data access cost
* register web data access log
* pay and download datasets

[Figure 2] Blockchain-based URI Ownership and Data Access Rule Registration and Query Protocol

The *Universal Resource Auth* protocol implements a web-scale data ownership and copyright management system by linking URIs and DIDs through the blockchain. Ownership information and data access rules for resources represented by URIs (web page URLs) on the internet are registered on the blockchain, allowing anyone to transparently verify web data copyright information for each URL. Generative AI or search engines that want to collect web data can collect web data in compliance with data access rules registered on blockchain by each URL owner. They can also pay data access costs to URL owners and record detailed web data access logs on the blockchain.

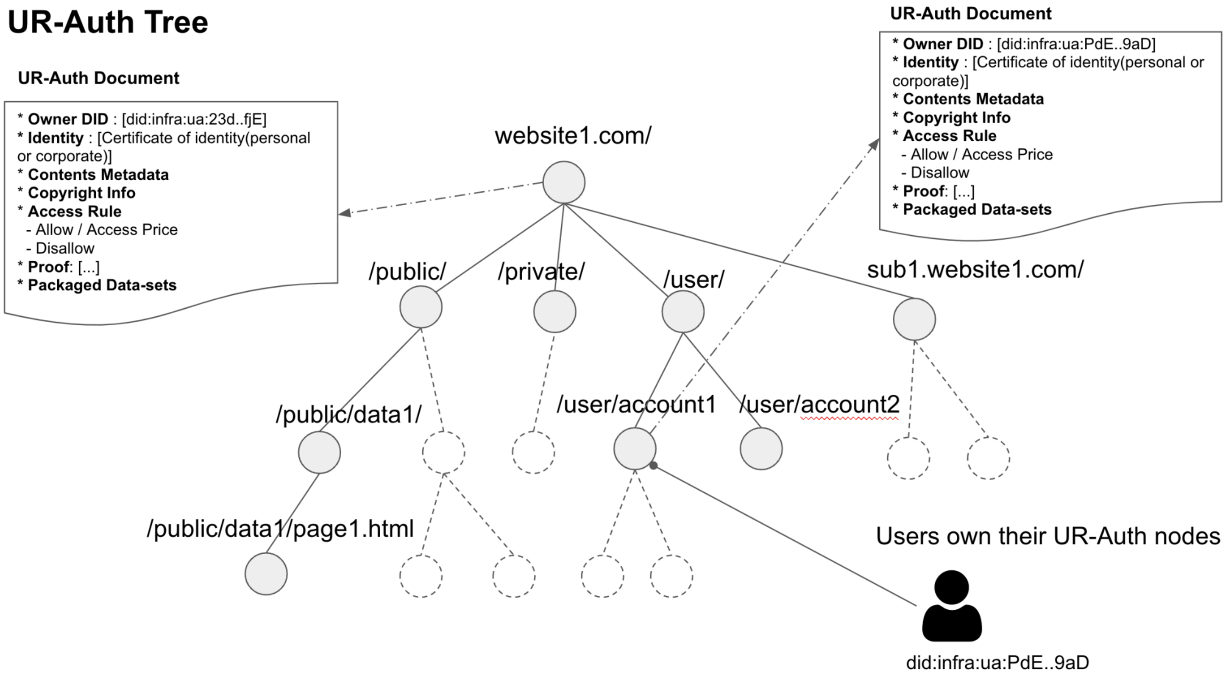
While the conventional method of storing *robots.txt*[9] or *ai.txt* in the root directory of a website domain provides web data crawling robots with data access rules, registering

[9] Robots Exclusion Protocol, robot.txt - https://en.wikipedia.org/wiki/Robots.txt

ownership information represented by the blockchain ID (DID) and data access rules on the blockchain offers transparency, integrity, and security for ownership and access rule information. This approach also provides a blockchain-based authentication mechanism for website owners and data accessors, allowing blockchain-based payments for data access and transactions.

Web data copyright holders (website domain owners, website users, etc.) can register their self-created blockchain ID (DID) as the owner of a URI on the blockchain after undergoing processes to verify ownership of that URI (such as domain ownership verification, web service user login authentication, etc.). Changes in ownership and modifications to data access rules for registered URIs require the cryptographic digital signature of the copyright holder, generated with the private key of the blockchain ID (DID) registered as the owner of the URI on the blockchain.

2.2 Universal Resource Auth (UR-Auth) Tree



[Figure 3] Universal Resource Auth (UR-Auth) Tree Data Structure

The website URL and sub-page URLs in a website can be expressed as nodes in the UR-Auth Tree data structure implemented as smart contracts on the blockchain. After undergoing processes to verify ownership of the resource represented by the URI, the resource owner can register ownership of the resource identified by the URI at each node. A single *UR-Auth Tree* corresponds to a single website domain and the data identified by URIs within that domain, and each node of the UR-Auth Tree stores *UR-Auth Document* that defines the owner (DID) information, copyright information, and data access rules of the URI. In the example of [Figure 3], the owner of the *website1.com* domain generated his/her DID (*did:infra:0x:23d...fjE*) on his/her blockchain wallet, registered the DID as the owner of the root node of tree, published copyright information for the website, and data access rules that web crawlers/scrapers[10] should comply with *website1.com*. Web data crawlers must refer to the publicly available ownership information and data access rules of *website1.com* on the blockchain before crawling the web pages served by *website1.com*. After accessing the website's data, the data access cost specified in the data access rules should

[10] Web scraping - https://en.wikipedia.org/wiki/Web_scraping

be paid to the owner DID of the *website1.com* node by using blockchain-based tokens for payments, and the detailed access history including the whole list of crawled URLs should be recorded on the blockchain by the web crawler. Since the size of detailed access history data can be large, only hashes of access records can be stored on the blockchain, while the access log data itself is stored on off-chain distributed storage.

Domain name owners can register the root node for the new UR-Auth Tree, and child nodes can be registered by the owner (DID) of the parent node. In the example of [Figure 3], the owner of the *website1.com* node can register the *website1.com/user* child node by submitting a blockchain transaction signed by the owner DID, and register the child node's ownership as the DID of a website user (e.g. a journalist on a news website)

For URIs whose corresponding *UR-Auth nodes* do not exist on the *UR-Auth Tree*, the *UR-Auth Document* for the URI is matched with the *UR-Auth Document* of the closest ancestor node in the tree. For example, in [Figure 3], ownership information and data access rules for *website1.com/public/data2/page5.html* follow the information stored in the *UR-Auth Document* of the closest ancestor node, *website1.com/public*. It is not necessary to register all URLs present in the website in the UR-Auth Tree, only descendant nodes having different ownership information or data access rules need to be registered.

2.3 Universal Resource Auth (UR-Auth) Document

Each node of the UR-Auth Tree holds a UR-Auth Document that records information such as ownership information, copyright details, and data access rules applied to the corresponding URI and all its sub-URIs. The content of the UR-Auth Document can only be modified by the owner DID of that node. Below is an example of a *UR-Auth Document*.

JavaScript

```
{
  "@context": [
    "https://infrablockchain.net/ns/ura/v1"
  ],
  "@id": "did:infra:ua:JewF..f2D",
  "uri": "website1.com",
  "ownerDID": ["did:infra:ua:5DHo..EYCy"],
  "identityInfo": ["eyJ0eXAiOiJ..Q07XDeGoqmw"],
  "contentMetadata": { "data" : "ipfs://qafyb...fyavhwq" },
  "copyrightInfo": "©2023 All Rights Reserved. Website1 is a registered trademark of ...",
  "accessRules": [
    {
      "path": "/public",
      "rules": [
        {
          "userAgents": ["*"],
          "allow": {
            "typesPrices": [
              [ "Image", "10 DUSD/MB" ],
              [ "Text", "20 DUSD/MB" ],
              [ "Code", "30 DUSD/MB" ]
            ]
          },
          "disallow": {
            "types": [ "Video" ]
          }
        },
        {
          "userAgents": [ "GPTBot/did:infra:ua:5PG..Feq", "Googlebot" ],
          "allow": {
            "typesPrices": [
              [ "*", "40 DUSD/MB" ]
            ]
          },
          "disallow": {
            "types": [ "Video", "Image" ]
          }
        }
      ]
    },
    {
      "path": "/private",
      "rules": [
        {
          "userAgents": ["*"],
          "disallow": {
            "types": [ "*" ]
          }
        }
      ]
    }
  ],
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2023-07-26T20:52:49Z",
    "verificationMethod": "did:infra:ua:5DHo..EYCy#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z58DAdFfa9Skq..FPP2oumHKtz"
  }
  "accessLogs": [
    "ipfs://wefo...bvoieu",
    "ipfs://nqop...fbeuhd", ...
  ]
  "datasets": [
  ],
}
```

[Code 1] A sample UR-Auth Document

- **@context** : The *UR-Auth Document* follows the *JSON-LD*[11] data format. The @context field indicates the scheme information of the *UR-Auth Document*.
- **@id** : The ID of the corresponding *UR-Auth node*
- **ownerDID** : List of owner DIDs for the corresponding *UR-Auth node*
- **identityInfo** : (optional) List of copyright holder identity information (Verifiable Credentials containing identity proofs)
- **contentsMetadata** : (optional) Registering the hash link of associated content metadata file for the corresponding URI stored on off-chain IPFS distributed file storage (can be used when the URI is not a URL and content data needs to be directly registered)
- **copyrightInfo** : (optional) Copyright information published by the URI owner, in free format
- **accessRules** : (optional) List of data access rules published by the URI owner URI (ref. 2.3.1 Data Access Rules)
- **proof** : Cryptographic signature for the *UR-Auth Document* signed by the owner DID (following the specifications of DIDs and Verifiable Credentials Proofs[12]), the data in the *accessLogs* and *datasets* fields is not included in the message to be signed.
- **accessLogs** : (optional) List of access history and payment information for the web data corresponding to the *UR-Auth node*, including dataset sales history.
- **datasets** : (optional) List of datasets containing web data corresponding to this *UR-Auth node* (ref. 4. Blockchain-based Data Market)

[11] JSON-LD 1.1, A JSON-based Serialization for Linked Data - https://www.w3.org/TR/json-ld/
[12] Verifiable Credentials Data Model v1.1 - https://www.w3.org/TR/vc-data-model/#proofs-signatures

2.3.1 Data Access Rules

The data copyright holder who owns the corresponding *UR-Auth node* can register data access rules and data pricing information in the *UR-Auth Document* for data accessors such as web crawlers/scrapers, data collection agents of AI and search engines to adhere to when accessing the copyrighted data.

URL	website1.com			
Path	User Agents	Allow		Disallow
		Type	Price	Types
/public	* (All)	Image	10 DUSD/MB	Video
		Text	20 DUSD/MB	
		Code	30 DUSD/MB	
	GPTBot Googlebot	*	40 DUSD/MB	Video Image
/private	*	-	-	*

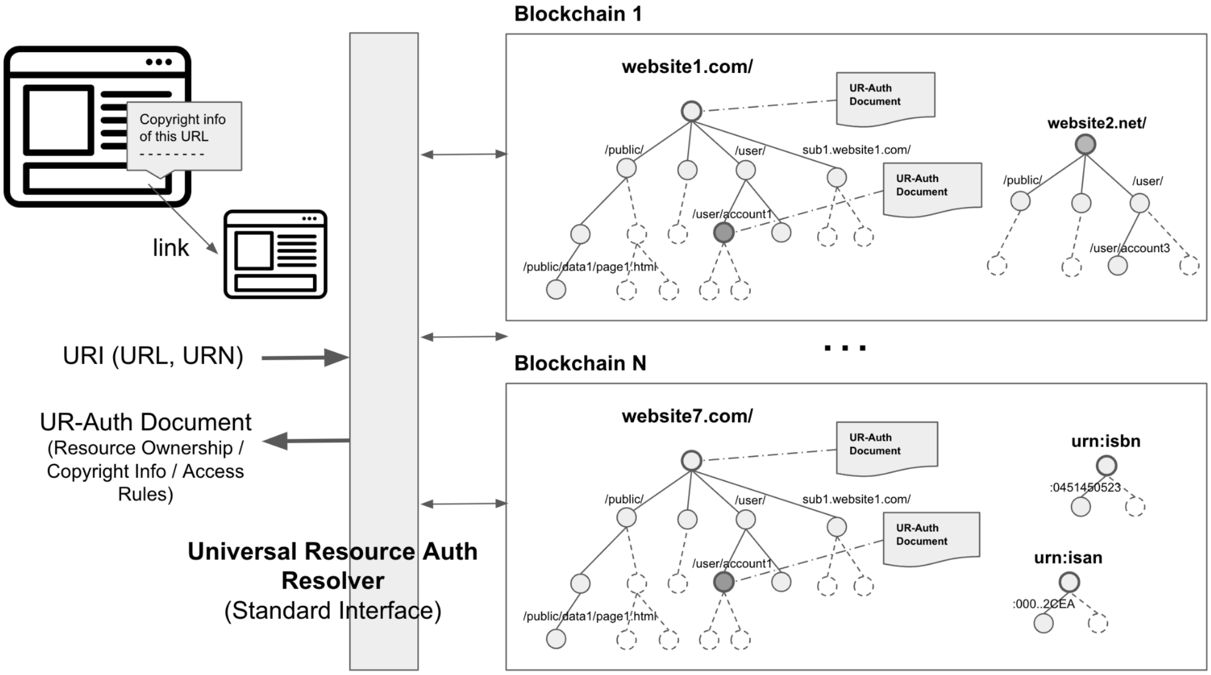
[Table 1] An example of Data Access Rules specified on an UR-Auth Document

Data access rules for an *UR-Auth node* can be composed of permission('allow') and disallowance('disallow') rules per data accessor type (*User Agents* of http request) and per data type, applicable to the corresponding URL and all its sub-paths. Access permission('allow') rules can include data access price setup. In the example shown in Table 1, the owner of the *website1.com* domain has allowed data access for *Image* data type at a price of $10 per 1MB, *Text* data type at a price of $20 per 1MB, and *Code* data type at a price of $30 per 1MB for all web contents within the *website1.com/public* directory and its subdirectories. However, access to *Video* data type has been disallowed for all data accessors. For GPTBot and Googlebot, separate access rules have been specified for website1.com/public path and its subpath. Specifically, access to Video and Image data types has been disallowed, while other data types have been allowed with a price of $40 per

1MB. Access to data within the website1.com/private path and its subpaths has been disallowed for all.

Web data crawlers/scrapers accessing data on the *website1.com* must download only allowed data in compliance with the data access rules specified in *UR-Auth Document*s on the blockchain. In case of commercial use of the data, they must pay the specified data access cost to the owner DID by transferring blockchain-based tokens.

2.4 Universal Resource Auth Resolver



[Figure 4] Universal Resource Auth Resolver

Ownership, copyright information, and data access rules for web data and intellectual properties represented by URIs (URLs, URNs) can be stored on various blockchain networks implementing the *Universal Resource Auth* protocol. The *Universal Resource Auth Resolver* standard interface is provided allowing users to input a URI and retrieve the corresponding UR-Auth Document from the blockchains. The standard interface specification and reference API server implementations are made available as open-source. Through the publicly available *Universal Resource Auth Resolver* API services, anyone can easily retrieve ownership information, copyright information and data access rules for web data represented by URLs and resources represented by URNs. Utilizing open-source implementations, anyone can easily build and service *Universal Resource Resolver* API servers.

universal-resource-auth-resolver(URI) → <UR-Auth-Document>

Unset

```
GET /ur-auth/resolve?uri=https%3A%2F%2Fwebsite1.com HTTP/1.1
Host: www.standard-universal-resource-auth-resolver.net
User-Agent: Nutch Crawler
Accept: application/json

HTTP/1.1 200 OK
Date: Fri, 28 Jul 2023 01:49:30 GMT
Server: ur-auth-resolver
Content-Type: application/json
...
{
  "@context": [ "https://infrablockchain.net/ns/ura/v1" ],
  "@id": "did:infra:ua:KdE..3iq",
  "uri": "website1.com",
  "ownerDID": ["did:infra:ua:5DHo..EYCy"],
  ...
}
```

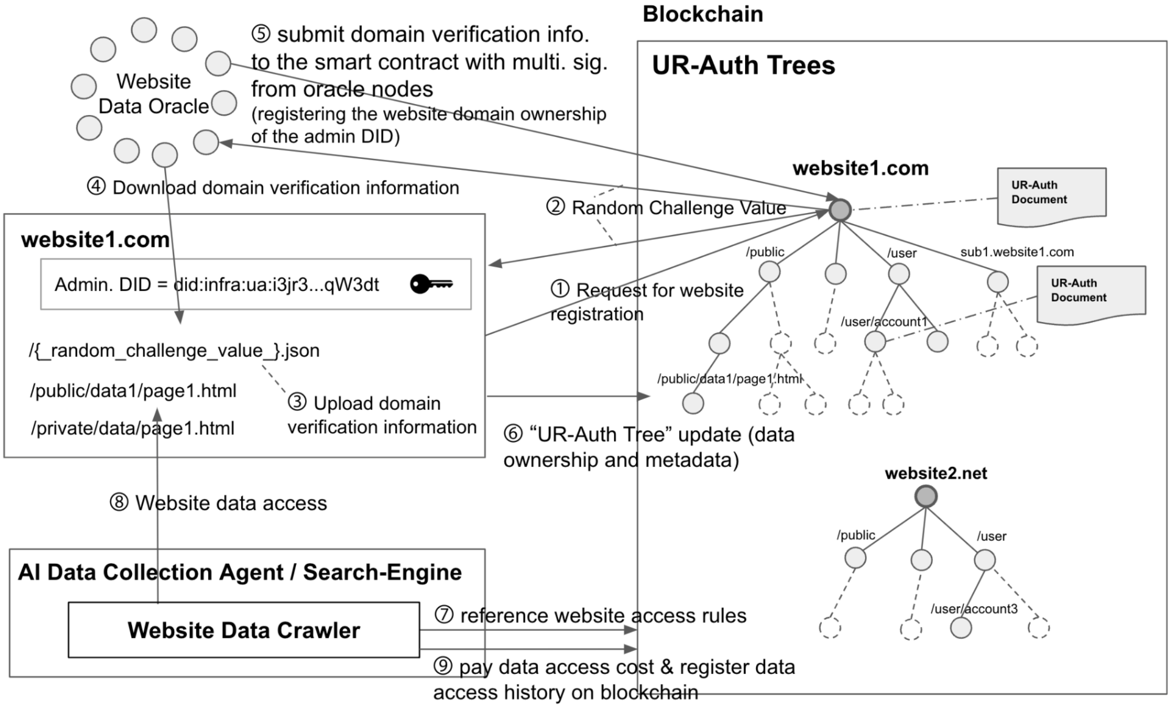
[Code] Universal Resource Auth Resolver HTTP Request Response

It is a mechanism similar to DID Resolution defined in the DID standard., which outputs a *DID Document* that stores information related to DID (public key, etc.) from blockchains when inputting DID. Using the *UR-Auth Resolver* standard interface, anyone can query any URI for registered ownership, copyright information. When integrated into web browsers, the UR-Auth Resolver allows users to easily verify ownership and copyright information of any web resource they access on browsers.

3. Claiming Universal Resource Auth (UR-Auth) Nodes

The ownership of a specific resource identified by a URI is represented by a single *UR-Auth Tree node* on the blockchain. To register ownership of the UR-Auth node on the blockchain, the resource owner needs to undergo a valid resource ownership verification process based on the type of resource. After successfully completing the proper ownership verification process, the resource owner can register their DID in the UR-Auth node on the blockchain.

3.1 Claiming the Ownership of Website Domain



Blockchain
UR-Auth Trees
Website Data Oracle
⑤ submit domain verification info. to the smart contract with multi. sig. from oracle nodes
(registering the website domain ownership of the admin DID)
④ Download domain verification information
② Random Challenge Value
① Request for website registration
website1.com
Admin. DID = did:infra:ua:i3jr3...qW3dt
/{_random_challenge_value_}.json
/public/data1/page1.html
/private/data/page1.html
③ Upload domain verification information
⑥ "UR-Auth Tree" update (data ownership and metadata)
⑧ Website data access
AI Data Collection Agent / Search-Engine
Website Data Crawler
⑦ reference website access rules
⑨ pay data access cost & register data access history on blockchain
website1.com
UR-Auth Document
/public
/user
sub1.website1.com
UR-Auth Document
/user/account1
/public/data1/page1.html
website2.net
/public
/user
/user/account3

[Figure 5] Website Domain Ownership Registration Process

After the domain ownership verification process, the website domain owner can register his or her own DID as the owner of the root node of the UR-*Auth Tree* corresponding to the domain name.

Blockchain-based Domain Ownership Verification and Ownership Registration Process

(1) The website owner generates their DID and sends a blockchain transaction requesting the registration of the *UR-Auth Tree* to the *UR-Auth* smart contract.

requestRegisteringNewURAuthTree(domainName, ownerDID)

(2) When the *UR-Auth Tree* registration request transaction is executed, the *UR-Auth* smart contract generates a random challenge value and publishes it on the blockchain.

(3) The website domain owner creates a domain verification information JSON file containing the domain name to be registered, the website owner's DID(*adminDID*), the randomly generated challenge value obtained from the blockchain, and the current timestamp cryptographically-signed by the website owner's DID using the private key of the DID. The filename of the domain verification information JSON file is set to the same random challenge value obtained from the smart contract. This JSON file is then uploaded to the root directory of the website domain, allowing anyone to download the domain verification information JSON file for verification.

website1.com/__random_challenge_value__.json

JavaScript

```
{
  "domain" : "website1.com",
  "adminDID" : "did:infra:ua:i3jr3...qW3dt",
  "challenge" : "__random_challenge_value__",
  "timestamp": "2023-07-28T10:17:21Z",
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2023-07-28T17:29:31Z",
    "verificationMethod": "did:infra:ua:i3jr3...qW3dt#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "gweEDz58DAdFfa9.....CrfFPP2oumHKtz"
  }
}
```

Uploading the domain verification information JSON file, which includes the DID's signature and is named as the random challenge value from the blockchain, to the root directory of the respective domain can be achieved by only the entity who actually own the website domain name and possess the private key of the *adminDID* specified within the domain verification information JSON file. Therefore, when the domain verification information is delivered to the *UR-Auth* smart control on the blockchain, domain ownership can be verified transparently and safely and the owner DID can be registered on the UR-Auth node.

(4) Oracle's role in the blockchain is to bring external data into the blockchain. In the *UR-Auth* system, web data oracle nodes periodically check if a domain verification information JSON file named *__random_challenge_value__.json* has been registered in the root directory of the domain's web server when a new domain registration request is registered on the *UR-Auth* smart contract. The oracle nodes download the domain verification information JSON file, and then submit the file with cryptographic signature of each oracle to the *UR-Auth* smart contract. Off-chain oracles are not responsible for the domain ownership verification, they are only responsible for just downloading the domain verification information file, relaying that file to the smart contract which executes the domain ownership verification code on chain.

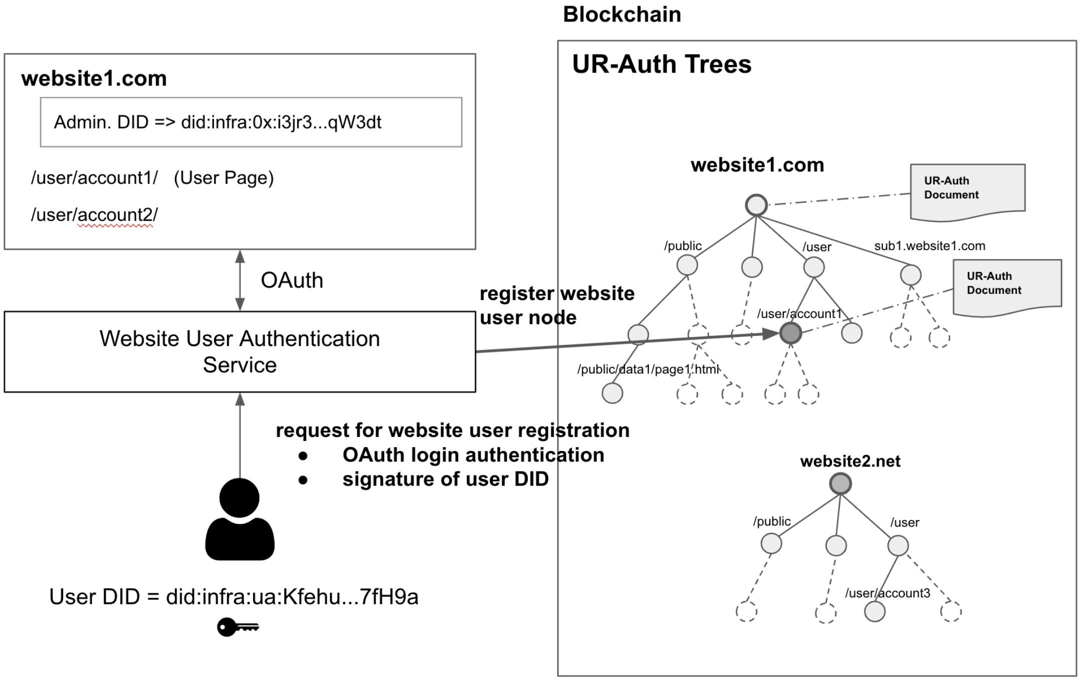
(5) The *UR-Auth* smart contract verifies that a certain number (e.g., 10 (2/3) out of 15 oracles) of DIDs registered as web data oracles have all submitted the same domain verification information JSON file for the requested domain. It also checks if the random challenge value in the submitted domain verification information JSON file matches the value generated by the *UR-Auth* smart contract during the domain registration request, and *adminDID* equals the *ownerDID* of the original domain registration request. Additionally, the cryptographic signature of the *adminDID* within the JSON file is correctly generated. If these conditions are met, the *UR-Auth* smart contract sets the owner DID in the UR-Auth Document of the root node of the

UR-Auth Tree corresponding to the domain to the *adminDID* from the submitted domain verification information JSON file.

Through the above process, the website domain owner proves their actual ownership of the domain using blockchain without the intervention of any centralized verification system, and registers that the owner of the website domain is the domain owner's DID on the blockchain.

Once the domain ownership registration process is completed, the website owner can perform *UR-Auth Document* modification actions such as registering copyright information and web data access rules for the website data.

3.2 Claiming the Ownership of Web Service Account


Blockchain
UR-Auth Trees
website1.com
Admin. DID => did:infra:0x:i3jr3...qW3dt
/user/account1/ (User Page)
/user/account2/
OAuth
Website User Authentication Service
register website user node
request for website user registration
● OAuth login authentication
● signature of user DID
User DID = did:infra:ua:Kfehu...7fH9a
website1.com
UR-Auth Document
/public
/user
sub1.website1.com
UR-Auth Document
/user/account1
/public/data1/page1/.html
website2.net
/public
/user
/user/account3

[Figure 6] Registering the Ownership of Web Service User Account on UR-Auth Tree

Web service users can register the ownership of their content within their web service accounts on the UR-Auth Tree on the blockchain after undergoing user authentication processes. Many major global web services provide user authentication based on the OAuth[13] standard protocol, allowing third-party application services (OAuth Clients) to easily authenticate users of the web service that support OAuth and access web data resources allowed by the authenticated users. Using this, when implementing a new web service, the user registration process can be implemented using the OAuth-based social login function of the existing OAuth-compatible web services used by the user without requiring users to create a new ID/Password. For web services that offer OAuth, the ownership of web service user accounts can be quickly and securely registered on the *UR-Auth Tree*. Each web service user account is provided with a unique personal account URL within the website (e.g., *website1.com/user/account1*). The ownership of these user account URLs can be registered on the *UR-Auth Tree*, allowing users to manage ownership information and data access rules for their web service account's data. The web service user authentication service (an OAuth Client) that has been granted permission to register user URLs on the *UR-Auth Tree* of the web service domain can authenticate the web service users through OAuth (social login), and set the user-created DID as the owner of the corresponding *UR-Auth node*. Web service users with registered ownership can register access rules for their web service account's data on the blockchain by making blockchain transactions signed by their DID.

3.3 Registering Copyrights of Data Files

[13] OAuth, Open Authorization Protocol - https://en.wikipedia.org/wiki/OAuth

The copyright of newly created data files (not existing on a website) can be registered on the blockchain. Users who wish to register data files can create their own DID and register ownership of text, documents, images, audio, video, and other data files by specifying their DID as the owner of these data files on the UR-Auth Document on blockchain.

A URL as follows is issued for a data file that is to be registered in the *UR-Auth Tree* :

urauth://file/{CIDv1}

CIDv1[14] : Content Identifier Version 1, self-describing content-addressed identifier

The CID (Content Identifier) is determined by the cryptographic hash value of the content of the data file to be registered. Even a single bit difference in data will result in completely different CID values, allowing CID to uniquely identify the content of a data file. The URL assigned to the data file is stored in the blockchain as a *UR-Auth node*, and this node contains the *UR-Auth Document* containing ownership and metadata information of the data file. The copyright information of the registered data file in the blockchain can be verified by anyone, and ownership of the data file can be traded on the blockchain.

In order to use a copyrighted data file registered on blockchain, you need to pay the owner of the data file through the blockchain. The data file can be stored in off-chain distributed file storage either as an unencrypted public file or as an encrypted file that only the data owner can decrypt. To access the encrypted data file, a data consumer needs to pay the data file owner on the blockchain and can then download the encrypted data file that only the data file purchaser can decrypt.

[14] CID (Content IDentifier) Specification - https://github.com/multiformats/cid

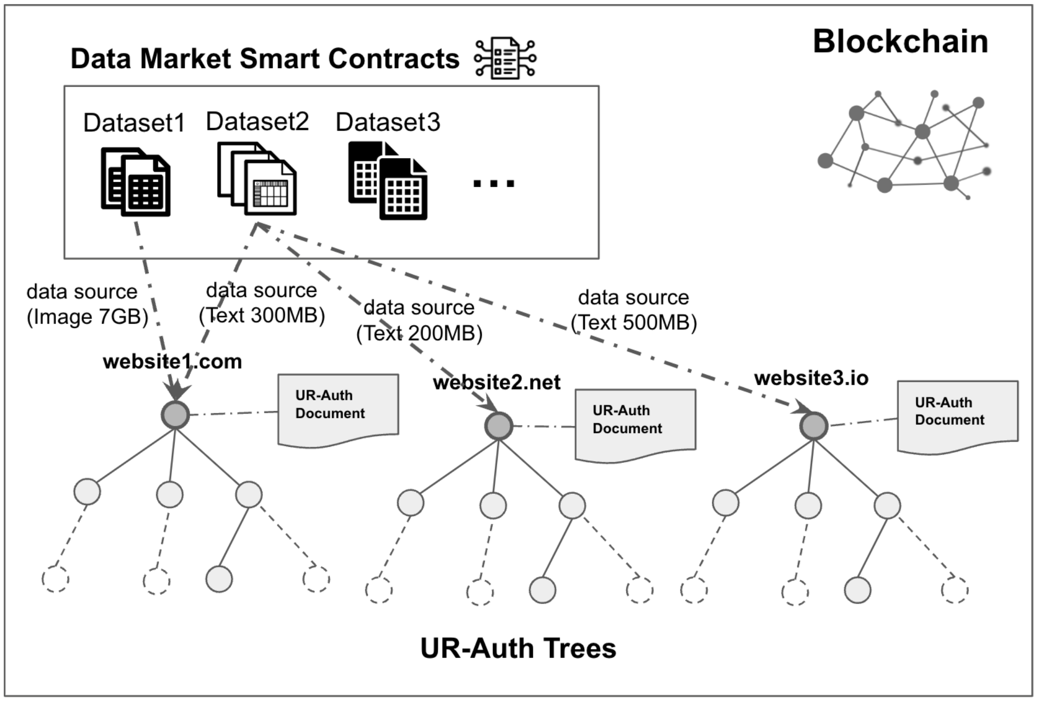
4. Blockchain-based Data Market

4.1 Pre-packaged Datasets

It is common for data consumers who want to collect web data (for training AI models) to acquire packaged datasets that have been pre-crawled and refined, instead of directly crawling or scraping all the necessary data. Pre-curated web crawling datasets available in open repositories like Common Crawl[15] have been significant sources of data for ChatGPT[16] AI model training. Many AI model development companies download datasets registered on Hugging Face[17] and use them for AI machine learning purposes. However, this existing method of distributing web datasets disregards data copyrights and indiscriminately allows web data to be used for commercial purpose AI machine learning, leading to serious infringements of data copyrights of numerous websites. This problem can be solved by allowing datasets to be registered and traded on the *UR-Auth* blockchain where web data copyright and data access rules are registered and managed.

[15] Common Crawl - https://commoncrawl.org/
[16] OpenAI ChatGPT - https://openai.com/chatgpt
[17] Hugging Face datasets - https://huggingface.co/datasets

4.2 Data Market on UR-Auth Blockchain



[Figure 7] Datasets Traded on Data Market Smart Contracts, Stored on Blockchain Cloud P2P Network

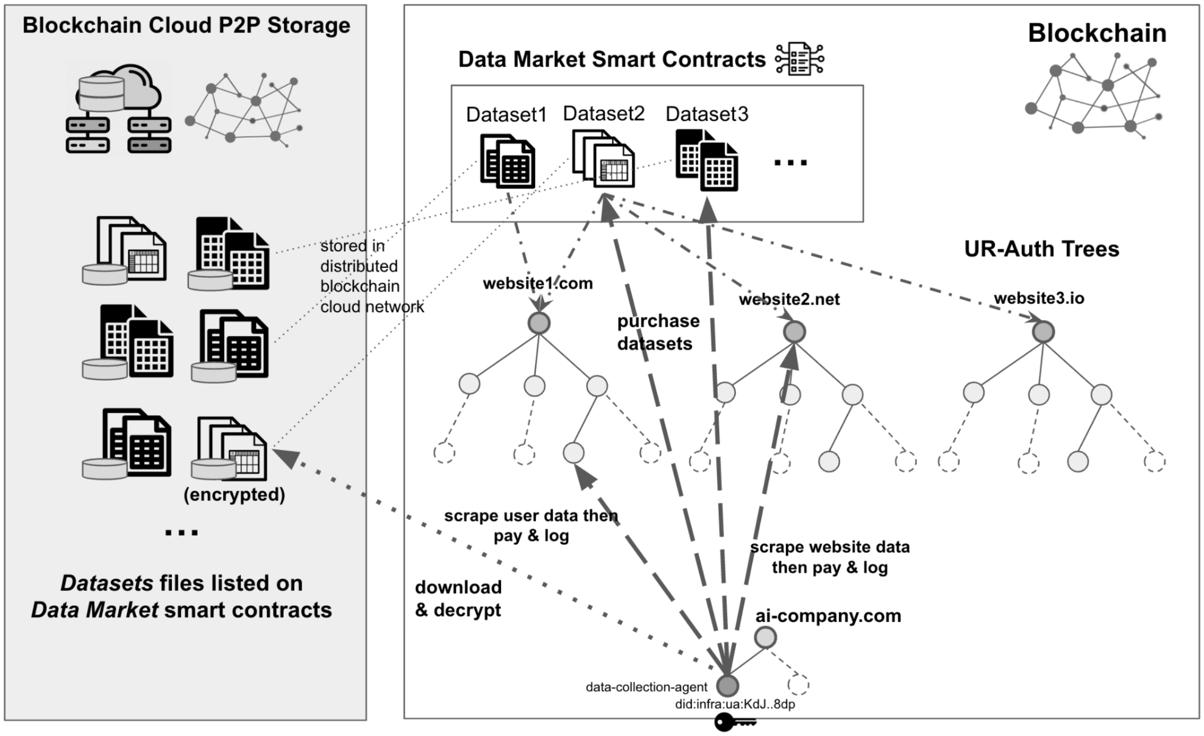
Pre-packaged web datasets can be registered and traded on the data market smart contracts on the blockchain implementing UR-Auth protocol which manages web data copyright information, so that web data consumers can pay web data owners for legitimate data access to download pre-packaged organized datasets and use them for commercial purposes such as AI machine learning. This solves the data copyright problem.

The metadata for the dataset, dataset sales information, and source information for the data included in the dataset (references to *UR-Auth Tree* nodes of data source websites) are all registered on the data market smart contract on blockchain. Large-size dataset files and all source URLs for all data items included in the dataset can be stored in off-chain distributed storage.

The *UR-Auth Document* of the *UR-Auth node* designated as a data source for a dataset contains ownership information and data access rules for the website data used in that dataset. These rules must be adhered to by the registered datasets too. If datasets on the data market do not comply with the data access rules, the web data owners (*UR-Auth node* owners) can request the cancellation of dataset registration on the data market contract. Website owners can register pre-packaged datasets created by crawling their own website data and sell them at their desired price on the data market. In the example of *Figure* 7, the owner of *website1.com* has registered *Dataset* 1 on the data market, which includes image data from their own website. Dataset 2, containing text data collected from *website1.com*, *website2.net*, and *website3.io*, has been registered by a specialized company that builds AI training datasets, which includes references to the *UR-Auth nodes* of each data source site. The price of *Dataset* 2 can be dynamically calculated based on the access rules specified in the *UR-Auth Document*s of each source website.

Web data consumers can search for and purchase web datasets that match their desired criteria in the data market service listing the datasets registered on the data market smart contracts on blockchain. Dataset purchases are made by generating a dataset purchase blockchain transaction that transfers blockchain-based tokens corresponding to the dataset price from the data consumer's blockchain account (DID) to the data market smart contract. Data sales revenue is automatically distributed by the data market smart contract to data copyright holders and dataset producers by referring to *UR-Auth Tree*s. Datasets purchased by data consumers on the blockchain are provided to dataset buyers in a way that can only be decrypted by the buyer's blockchain account (DID).

4.3 Datasets Stored on Blockchain Cloud P2P Storage Network


Blockchain Cloud P2P Storage
stored in distributed blockchain cloud network
(encrypted)
...
Datasets files listed on Data Market smart contracts
Data Market Smart Contracts
Blockchain
Dataset1
Dataset2
Dataset3
...
UR-Auth Trees
website1.com
website2.net
website3.io
purchase datasets
scrape user data then pay & log
scrape website data then pay & log
download & decrypt
ai-company.com
data-collection-agent
did:infra:ua:KdJ..8dp

[Figure 8] Data Market and Blockchain Cloud P2P Storage

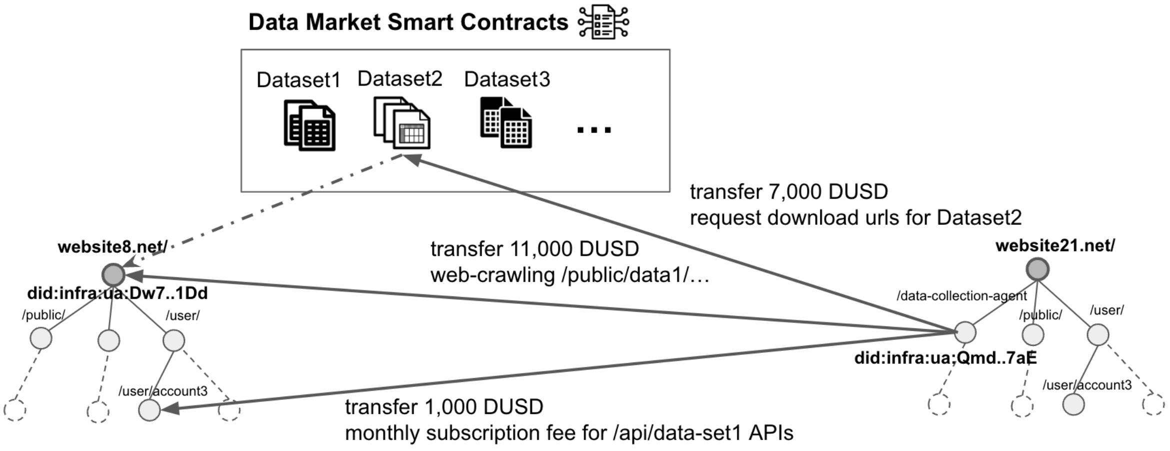
The Data Market smart contract stores metadata about datasets and the hash values of dataset files. Storing large-size dataset files directly on the blockchain storage layer managed by the contract would be highly inefficient. Dataset files containing actual data are stored in off-chain distributed storage networks based on *blockchain cloud technology* that operates outside the blockchain. These dataset files are encrypted and stored in a distributed manner. The list of all source URLs for web data contained in a dataset, and sample data from a dataset are not encrypted and is publicly available in the distributed storage for search and data distribution tracing purposes. Datasets purchased by data consumers on the blockchain are re-encrypted in a way that only the buyer's blockchain account (DID) can decrypt, stored in blockchain cloud storage, and provided to the buyer.

The *blockchain cloud* technology for storing data files involves distributing storage engine nodes across various public cloud infrastructure services (such as AWS, Azure, GCP, etc.). Data files are encrypted by the blockchain ID (DID) of the data owner to ensure that only the owner can decrypt the files. These encrypted data files are then divided into pieces using *erasure coding* technology and distributed across the storage nodes in the public cloud services. Each public cloud service in which encrypted pieces of data are stored cannot restore by itself the original data owned by the data owner DIDs. However, data owners holding the private key of the owner DID can recover the original data if they can collect more than a certain percentage of erasure-coded data pieces of the original encrypted data even if they do not have access to some cloud services or some erasure-coded data pieces are lost. Through *blockchain cloud* technology, it's possible to protect the data registered for data sales on the data market from being publicly accessible, and protect the storage system from being dependent on a specific cloud service.

5. Tracking Data Distribution

Through the UR-Auth blockchain platform, data consumers can conveniently collect safe and well-refined data without the possibility of ownership disputes, and data owners can transparently track how their data is distributed and used in a legitimate way through the blockchain.

5.1 Data Access Log and Payment on Blockchain



[Figure 9] Data Access Log and Payment on Blockchain

Data consumers can collect copyrighted data legitimately through various methods using the UR-Auth blockchain platform, and can directly compensate data owners for the fair use of the data. By utilizing web crawling and scraping libraries integrated with the *UR-Auth* platform, they can collect website data according to the data access rules specified in the *UR-Auth Document* on the blockchain. They can register data access history (complete list of data-collected URLs) on the blockchain and pay for data access costs directly to the data owner's DID account by transferring tokens on the blockchain. (The complete list of URLs for the data access history may be large, so it is stored in off-chain distributed storage, and the hash value of the list is stored on the blockchain.) Instead of directly crawling website data, data consumers can purchase refined datasets from the blockchain-based data market by conducting purchase transactions using blockchain tokens. Purchased datasets include the complete list of URLs for the data resources included in the dataset, so it is possible to trace all URLs used by the data consumer. Datasets can be periodically updated to reflect the latest information from websites, and data consumers can purchase a paid subscription for these datasets. Datasets traded in the data market can be provided through API interfaces instead of data files, allowing data consumers to access dataset APIs with charges based on data usage or through paid subscriptions for a specific period. In these ways, data consumers can easily collect high-quality web data through various methods using the *UR-Auth* platform, ensuring a safe and secure approach that avoids copyright disputes.

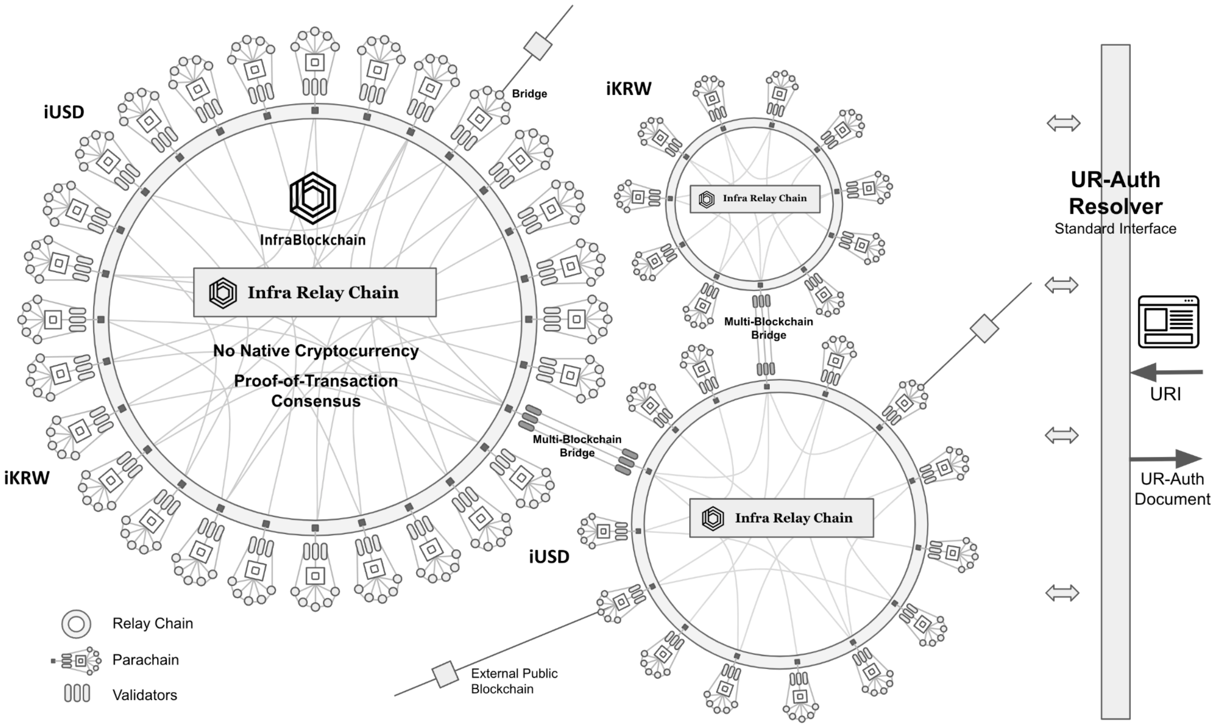
5.2 Tracking the Usage of Web Data

If data consumers comply with data access rules, pay for data usage, and record data access history through the *UR-Auth* platform, data owners can transparently track how their web data, social media account content, and registered data files are being circulated. Media companies can track articles posted on their news sites, and individual users can track how their images and messages on their blog and SNS are used by data consumers such as AI companies. And they can track in detail how their data is included in datasets traded in the data market. Data consumers such as AI companies can explore and safely use high-quality data through the UR-Auth platform, and blockchain can prove that they have collected AI learning data through legitimate channels and paid a fair price. Data owners can use the data access history records on the *UR-Auth* platform to verify any unauthorized data usage. In case of data ownership disputes, these blockchain-based data access records can serve as legal evidence in lawsuits.

6. Underlying Blockchain Networks Implementing UR-Auth

Data ownership information is stored and managed through smart contracts implementing the *UR-Auth* protocol on the blockchain network. The blockchain technology enables that the data copyright information can be securely stored in an immutable form, and anyone with internet access can verify it transparently. The blockchain networks can serve as a web-scale data copyright registration system. Any public blockchain can be used if a smart contract complying with the *Universal Resource Auth* protocol can be implemented, but the most suitable blockchain infrastructure that can serve as a global web-scale data copyright registry should be selected. *InfraBlockchain*[18], a public multi-blockchain network without native cryptocurrency, is the optimal blockchain network technology for implementing the *UR-Auth* protocol.

6.1 InfraBlockchain, Enterprise Public Multi-Blockchain Networks Without Native Cryptocurrency



[Figure 10] Enterprise Public Multi-Blockchain Networks Without Native Cryptocurrency

[18] InfraBlockchain Technology - https://infrablockchain.net/technology

Cryptocurrency-based public blockchains, which are strictly regulated in many countries and have very high price volatility, are difficult to be used freely by government agencies and companies. InfraBlockchain utilizes a stable token based on legal tender issued by trusted authorities, avoiding the issuance of its own native cryptocurrency. This stable token is used to process blockchain transaction fees (gas fee) and handle payments for asset trading on the blockchain. Through the Proof-of-Transaction consensus algorithm, trusted entities and blockchain service providers can be elected as the validators to operate the public blockchain. *InfraBlockchain* enables governments and companies to operate nationwide large-scale blockchain services without issuing and/or using cryptocurrencies.

A single blockchain with limited transaction processing capacity is insufficient to track data ownership and data distribution of all web data on the internet. A highly scalable multi-blockchain architecture is required, where numerous blockchains operate concurrently. To achieve web-scale data ownership management and tracking, multiple blockchains per data types, web domains, and countries need to operate in parallel. *InfraBlockchain is multi-blockchain technology that* connects various service-specific, industry-specific, and country-specific service blockchains (*Parachains*) to a central *Infra Relay Chain* that serves as a hub interconnecting numerous blockchains in parallel. The multi-blockchain architecture can be further expanded by having one service chain (parachain) slot on the *Relay Chain* serve as a *Bridge* to connect with the *Relay Chain* of another multi-blockchain network, thus creating a network of multi-blockchains. Through the *Network of Multi-Blockchain* architecture, infinite blockchain scalability and interoperability can be achieved. Trusted entities such as media companies, content production associations, government agencies, and financial institutions aiming to establish a blockchain-based data copyright management ecosystem can participate as validators in their blockchain network. They can build and service a multi-blockchain network that facilitates web-scale data copyright management, data distribution, and transactions.