AIRBLOC PROTOCOL

Technical White Paper

Version 2.0

Published on Oct 4, 2018

ab180 Inc.

Table of Contents

1. Why Airbloc Protocol is Needed

A new paradigm for personal data management

Recent years have seen an increased emphasis on stricter regulation with regards to consumer data privacy and personal data ownership. Issues of personal data privacy infringements, malicious usage of personal data, and collection of data without explicit consent from data owners are the foremost reasons behind this wave of new data regulations.

As evinced from the sweeping General Data Protection Regulation (GDPR) for personal data privacy which came into legally binding effect in the European Union on 25 May 2018, the world is indeed witnessing a movement towards safeguarding individuals' personal data privacy and data sovereignty. The GDPR will not just have an impact within EU, but will have a strong cascading effect on other jurisdictions' data-related regulations too.

The essence of such data-related regulations seems to impose new procedural obligations for organizations processing personal data to return more rights to data owners.

Blockchain and the GDPR share similar goals and motivations. Both aim at decentralizing data control, and returning sovereignty of control from centralized entities back to the individual. Aside from sharing similar goals, blockchain can effectively help organizations, data-centric applications and data owners with the GDPR and other data-related regulations.

This is where Airbloc positions itself.

2. Problems in Today’s Data Industry

User Data Privacy Infringement

The data industry today is dominantly controlled by centralized services that aggregate data illegally and sell them to other enterprises to generate huge revenues. However, these data-centric enterprises infringe on users' data privacy. Users' data today are collected, monetized and utilized without users' consent or even awareness.

Applications Face Difficulties in Collecting and Monetizing Data

Because there are no existing legitimate data markets for applications to monetize their users' data for additional revenue, applications often sell users' data illegally to other services that further aggregate data and sell it to other enterprises. In the process data owners are excluded from opportunities to earn from their data flows.

Enterprises Lack Legitimate and Transparent Data in Quality and Quantity

There are no existing data marketplaces that allow enterprises to purchase high quality, insightful and legally acquired data at reasonable prices for their business intelligence, research or targeted marketing purposes. Provenance of the data on these marketplaces is opaque as well.

Opaque Data Flows Creates Distrust

Users are not willing to give their data as much as enterprises want their data because data flows are opaque. Users do not know how their data is used and by whom since the methods of data collection by enterprises are usually undisclosed.

3. Introducing Airbloc Protocol

3.1. High Level Overview

Airbloc Protocol redefines how data is collected, monetized and utilized. Leveraging blockchain technology and token economics, it seeks to facilitate more transparent data flows between data owners, data providers and data consumers.

Ultimately, it aims to return data ownership back to data owners, provide applications with tools to collect and monetize data legitimately, and allow data consumers to purchase explicitly consented data with an auditable source of provenance for their business intelligence, research, and targeted marketing purposes.

To this end, Airbloc Protocol will develop a fully auditable data supply chain through User, Application and Enterprise oriented services that are interconnected with each other.

- Personal data is collected after explicit user consent is obtained through Airbloc's blockchain-based data authorization protocol called DAuth. The data is controllable and traceable by the data owner on a public blockchain.
- Airbloc intends to build a token ecosystem that incentivizes various actors to sustain and grow the Airbloc data ecosystem. Ecosystem participants who provide and process data in Airbloc will receive a revenue share from the data asset. Participants who contribute via data processing, verification or replication will receive additional contribution rewards from the network.
- To ensure that the data on Airbloc is treated as an asset, data access should only be granted to rightful data consumers after a purchase has been made. Data owners' data privacy and data access are protected through our proprietary **Privacy Shield** technology that encrypts, anonymizes and secures personal data even though the data is traded on a public blockchain.

Features in the protocol can be used as building blocks to build a shared data economy between data owners, data providers and data consumers. Applications can be built using Airbloc Protocol, such as applications with data-monetization business models, data marketplaces, or data management platforms (DMP) that aggregate customers' data.

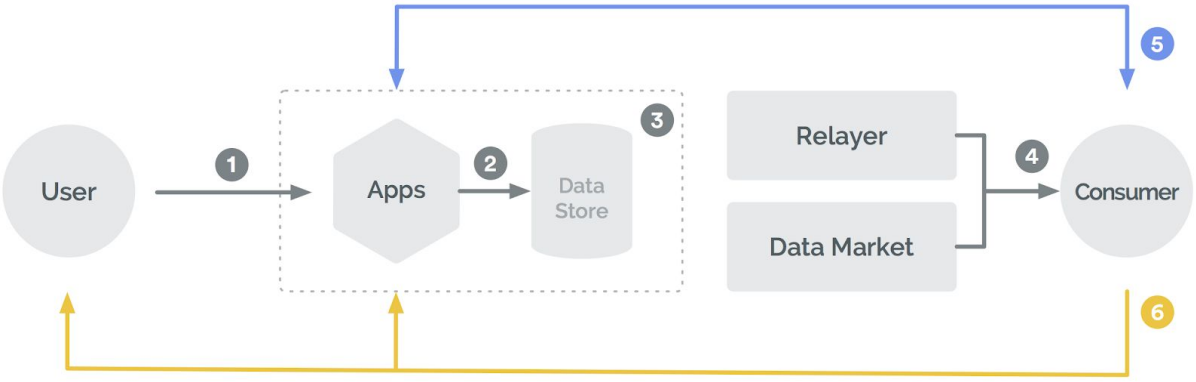

Figure 1: High-level overview of the data collection, registration, storage, and trading process in Airbloc

1. **Data Collection:** Data providers (applications) collect data on behalf of users through a data collection authorization process called DAuth. DAuth helps applications to seek for data owners' explicit permission to collect and monetize their data. This protects users' data rights and provides an auditable data provenance trail as data that were not authorized through DAuth cannot be registered on Airbloc.
2. **Data Registration:** Data providers need to register collected data to Airbloc for data validation and cleansing before it can be stored.
3. **Data Storing:** Data will be encrypted and stored by the data provider. The integrity and availability of the stored data is verified by data validators who can issue a data availability challenge if there is a data availability problem with a data provider.
4. **Data Discovery:** Data consumers can search for their required or demanded data, add filters, specifications or conditions in the data market to narrow down and gather the specific data through a data relayer.
    - **Data Market:** Data providers can bundle the collected data in the form of data sets and segments, set the desired price, and register it on the data market for sale.
    - **Data Relayer:** Relayer is a node that supports user segmentation and data retrieval from multiple data providers. Relayer first constructs a queryable user profile using anonymized data aggregated from various data providers. When a data consumer requests for a particular data segment, the relayer queries from the profile and returns the requested data to the data consumer.

5. **Data Exchange:** When data providers accept the purchase contract presented by the data consumer, data will be exchanged and the data consumer will be given access to the requested data.

6. **Data Purchasing:** Once the exchange concludes, the data consumer pays for the data through ABL tokens. The paid ABL will be automatically converted to AIR tokens and be respectively distributed to the data owners and providers.

3.2. Airbloc Ecosystem Stakeholders

There are various stakeholders and participants who constitute the Airbloc network.

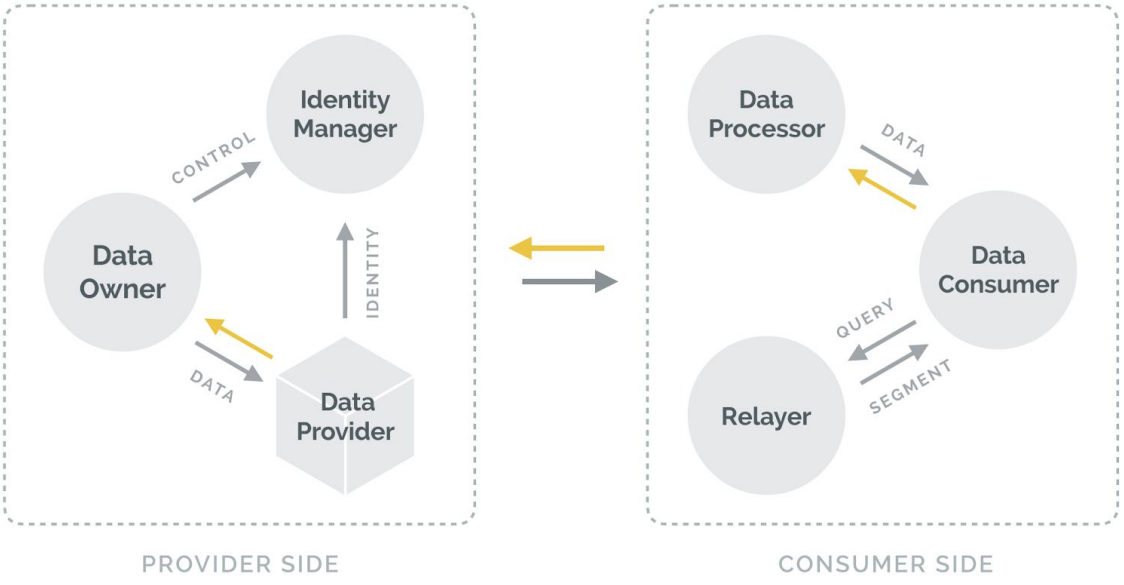

Figure 2: Ecosystem Stakeholders in Airbloc

Stakeholders who provide data in the supply side are:

- **Data Owner**: Individual users who are the source of the personal data on Airbloc. They can track and control their own data flows and receive incentives when they provide their personal data.

- **Data Provider**: Applications which can collect and sell the data upon data owners' consent and share the revenue generated by the data with data owners. Applications can be any kind of services on top of any device such as mobile phone, personal computer (PC), smart devices like smart watch, IoT devices like beacon, digital machines like POS or platform such as native applications, websites, etc. Most of the time these applications can

cover multiple devices and multiple platforms at the same time. For example, an application can have services supporting mobile native applications on Android and iOS, and on website well.

- **Identity Manager**: A data tracker service or personal information management service which stores identity data on behalf of the user. Identity managers can receive contribution rewards by storing identity data of data owners, and notifying them when their data is purchased by data consumers. Identity managers are also able to provide identity matching services.

Stakeholders who deliver and consume data in the demand side are:

- **Data Consumer**: Enterprises, research institutes or organizations interested in purchasing data for the purpose of business intelligence, research or targeted advertising.
- **Data Processor**: Data processors are data analysis services or data science companies specializing in refining raw data into valuable data. Anonymized raw data can be provided by a data provider to be further processed by a data processor.
- **Data Relayer**: Data relayer is a node which supports data discovery by querying on the data marketplace. It helps data consumers find and purchase their specific subsets of data under certain filters by aggregating the data from multiple data providers.

The stakeholder who verifies data and maintains the network is:

- **Data Validator Node**: A node that acts as a "policeman" to verify the data availability of data providers. Data validator nodes can verify, challenge and replicate the data to receive contribution rewards.

4. Defining Personal Data

4.1. Characteristics of Personal Data

Airbloc Protocol primarily deals with personal data.

Personal data are pieces of information that, when aggregated together, explains or helps infer certain traits or aspects of a specific person. In today's digital world, personal data is automatically generated in massive quantities from digital devices such as mobile phones, personal computers, or IoT devices like smart watch, and even smart cars as well. Every single activity interfaced with digital devices leaves a trail of personal data. Even though majority of these data are generated from digital devices, direct personal interest statements from these persons can be also counted as personal data.

Examples of personal data include: email address, phone number, name, age, home address, location data, IP address, internet cookie ID, mobile advertising ID, installed application list of a device, device application usage history, and even behavioral data such as personal interests and preferences, etc.

An example of personal data that Airbloc is capable of collecting from mobile devices looks like the following:


```
{
  "OS": "Android",
  "eventCategory": "InApp Event",
  "eventTimestamp": "2018-09-30 21:34:48",
  "goalCategory": "User Signin",
  "platform": "APP",
  "rawData": {
    "app": {
      "appId": null,
      "packageName": "com.sample.application"
    },
    "device": {
      "clientIP": "155.123.24.74",
      "deviceModel": "SM-N950N",
      "deviceUUID": "441bd0ef-a8a5-4951-a979-22161e57f0e1",
      "gaid": "10ead579-bbcc-465f-8459-fee6092d7c82",
      "limitAdTracking": false,
      "manufacturer": "samsung",
      "osNameVersion": null,
      "osVersion": "8.0.0",
```

```
      "timezone": "Asia/Seoul"
    },
    "eventCategory": null,
    "eventData": {
                  ... (abbreviated) ...
    },
    "eventTimestamp": 1538310888354,
    "eventUUID": "f8b19ca0-be9f-4a33-ba99-579c13f05d67",
    "requestTimestamp": 1538310893059
  },
}
```

4.2. Components of Personal Data

Any kind of personal data can be parsed and divided into two further data components. The first being Identity Data and the other being Payload.

Identity Data

Identity data can be broadly defined as any kind of identifiers (or IDs) such as email address, phone number, mobile advertising ID like GAID, IDFA, hashed email address, hashed phone number, etc. that satisfy the conditions below:

1. The identifier should point to or be connected to a person who is the owner of the data that has been generated.
2. The identifier should preferably be in one-to-one direct relationship to one unique person.

To illustrate, a "GAID" (Google Adverting ID[1]) with the value of `"5299ed79-b9ca-422a-ae24-bb0de50edd45"` belongs to the identity data component of personal data since GAID is given to each unique mobile device and in most cases one unique person will be holding onto one mobile device.

There are three types of identity data and the reason why we categorize them in Airbloc is to draw a clear line when dealing with personal data to safeguard data owners' privacy.

1. **Personal Identifiable ID (PIID)**: PIID contains any kind of identifiers that are directly associated with the identity of a specific person. Social security numbers, email addresses, phone numbers can be categorized into PIID. Email and phone numbers are deemed as having 'direct' relationship

[1] "Advertising ID | Android Developers." http://www.androiddocs.com/google/play-services/id.html.

because a data consumer can engage a call with a person or email the person with these PIIDs.

2. **De-Identified ID (DIID):** DIIDs are any kind of identifiers that are not directly associated with the identity of a specific person, but are still loosely connected to a person's identity in a sense that it can be used for exposure to targeted advertisements. DIID is not 'direct' since data consumers cannot directly engage the person through DIID. However data consumers can use DIIDs to target certain group of people to increase advertisement exposures in modern digital advertising platforms such as Facebook or Google Ads.

   Also some PIIDs can be converted into DIIDs when hashed by certain algorithms like SHA-256. For example, "my@email.com" when hashed into `"2ea651891d71a513fe6cacb304de3cbae"` is irreversible back to "my@email.com", but still can be used to point to one unique person who is the owner of that email address. Mobile advertising IDs like GAID, IDFA, hashed email addresses and hashed phone numbers all belong to DIID.

3. **Anonymized ID (ANID):** ANID is a unique ID for each unique data owner in Airbloc's Ecoystem. It is used for purposes of distinguishing unique data owners on Airbloc during intermediate data handling processes. For example, a data owner with an ANID of `"dedcdd8e-35ae-414c-8ab1-4bd8558fbd93"` is clearly not the same person as a data owner with an ANID of `"34cb399d-7ada-4096-8220-cd0a1ca1d689"`.

   Solely with ANIDs, direct or indirect relationships cannot be formed with the data owner since no means of communications can be established with ANID.

This way even if in an unlikely scenario that the payload data is compromised, the more important identity-related information will not be compromised and this plays a crucial role in protecting data owners' privacy.

Though ANID represents one unique person, multiple ANIDs may be generated per one unique person and distributed to multiple stakeholders in the ecosystem respectively. For example ANID X will be given to the data relayer A and ANID Y will be given to the data relayer B, but X and Y will both point to Alice.

To protect data owners' data privacy, Airbloc Protocol is designed to automatically identify and erase or hash identity data from the personal data as a whole during the data registration process. For example, if PIID is detected in newly registered data, it is either removed (e.g. social security numbers) or hashed (email addresses or phone numbers) and converted into DIID.

Even though DIIDs are loosely connected to a person's identity, it can still be used for targeted advertisements. Also, DIIDs are commonly used by multiple big advertising platforms and they may already be connected with PIIDs in these respective advertising platforms because of user profiling.

Thus, to further protect data owners' privacy, Airbloc will mask DIIDs with ANIDs and use only ANIDs when handling data for back-end intermediate processes during a data exchange.

However, ANIDs should ultimately still be connectable to the root DIIDs or PIIDs so that when actual data purchase is made by the data consumer, ANIDs can be replaced and converted back to the corresponding DIIDs and PIIDs for actual usage by data consumers. Further information on this can be found in Section 6.3.

The stakeholder responsible for the connection and conversion of these three types of IDs is called "Identity Manager Node" and further details about identity manager nodes can be found later in this paper.

Payload Data

Payload data are the actual contents of the personal data. Examples of payload data are device installed application list, shopping cart history, device application usage history, personal interests and preferences, etc. Payload data is stored and managed by the data provider.


```
{
  "OS": "Android",
  "eventCategory": "InApp Event",
  "eventTimestamp": "2018-09-30 21:34:48",
  "goalCategory": "User Signin",
  "platform": "APP",
  "rawData": {
    "app": {
      "appId": null,
      "packageName": "com.sample.application"
```

```
    },
    "device": {
      "clientIP": "155.123.24.74",
      "deviceModel": "SM-N950N",
      "deviceUUID": "441bd0ef-a8a5-4951-a979-22161e57f0e1",
      "gaid": "10ead579-bbcc-465f-8459-fee6092d7c82",
      "limitAdTracking": false,
      "manufacturer": "samsung",
      "osNameVersion": null,
      "osVersion": "8.0.0",
      "timezone": "Asia/Seoul"
    },
    "eventCategory": null,
    "eventData": {
                  ... (abbreviated) ...
    },
    "eventTimestamp": 1538310888354,
    "eventUUID": "f8b19ca0-be9f-4a33-ba99-579c13f05d67",
    "requestTimestamp": 1538310893059
  },
}
```


In the example above, any data except for "deviceUUID" or "GAID" are categorized as payload. In other words, any data that are devoid of identity data belong to payload. Normally in standardized personal data that is collected from digital devices, there can be multiple components of payload and some of the important components are featured below:

1. User (or Individual or Person)
2. Device
3. Operating system
4. Application (or service)
5. Event (action or conversion)
6. etc.

The data set that can be constructed using the above mentioned components can be the following: a user aged 24 used an iPhone8 with OS version 12.1 installed an application called "Crypto News" on 1 October 2018. This piece of information derived

from payload data can reveal about the user's preferences and interests in the cryptocurrency and blockchain industry.

Solely with such payload data, data consumers cannot directly engage the data owners by emailing them or giving phone calls. Also, data consumers cannot expose them to targeted advertisements.

Without PIID or DIID and with only ANID, this payload data is can be only used for non-commercial purposes such as research or partly for commercial purposes like business intelligence.

4.3. GDPR Compliance in Personal Data Management

Airbloc Protocol is designed to be GDPR compliant with regards to personal data management. Airbloc Protocol can considered as a *Personal Information Management System* (PIMS), as outlined in the GDPR. Applying GDPR's specifications in the strictest sense, Airbloc allows data owners (as data subjects) to have ownership and control their data through Airbloc, and applications (as data controllers) in Airbloc to process, use, or monetize data owners' data upon receipt of explicit consent.

Data owners in Airbloc can manage where their data is stored and, if necessary, export or delete it. In essence, Airbloc grants data owners:

- The right to access data
- The right to object to data collection or usage
- The right to correct errors in data
- The right to delete data
- The right to move data to grant data portability

5. Standardizing Data Schema Formats

5.1. Data Format Mismatch

One of the problems that can arise when exchanging data is that it is difficult to exchange data when the data schema formats are different.

Consider the following scenario: Enterprise A and Enterprise B tries to exchange their phone number data. Enterprise A uses "phoneNumber" as a key, however, Enterprise B uses "telephoneNumber" as a key instead. This leads to a schema mismatch. In other words, different schema for each data taxonomy makes exchanging data difficult.

Fortunately, since most of the personal data are coming from digital devices, much of data can be standardized. First, events can be categorized into certain types. For example, in mobile applications, there are standardized events such as “application install', 'launch', 'page view', 'sign up', 'purchase', etc. Second, payload is always identical in every event collected. For example, device-related information like device manufacturer, OS-wise information like OS name and version will be collected in the same format repeatedly.

5.2. Data Schema Registry

To allow for easier data exchange between enterprise data systems with different data formats, Airbloc will integrate and streamline data schema for each data taxonomy. A common universal data schema registry called *Data Schema Registry* will be introduced. All data registered in Airbloc should follow the common schema expressed in the Data Schema Registry.

Be that as it may, there is still a possibility that data providers may still register different data schema onto Airbloc. To address this issue, Airbloc introduces an incentive scheme based on a reputation system to encourage the use Airbloc's standardized data schema registry. Data providers will receive more reputation points when they adhere to the data schema standards as outlined in the registry. Higher reputation points allows data providers to yield more Contribution Rewards when they register data onto Airbloc.

6. Privacy Shield

Since Airbloc Protocol deals with personal data, highest priority is placed in protecting data owners' privacy. Four major measures will be taken to achieve this goal:

1. De-identification of Identity Data
2. Masking
3. Decoupling and anonymizing personal data
4. Private Identity Matching

6.1. De-identification of Identity Data

De-identification is the process of substituting PIID with DIID. PIID, unless otherwise authorized by the data owner, will be hashed immediately upon data collection by Airbloc SDK on the client-side, and double-checked by Airbloc itself once the data is registered. This process is designed to prevent unnecessary private information to be passed from the client-side to the Airbloc network.

For example, if raw email address string such as "my@email.com" is collected, but not explicitly authorized by the data owner during DAuth, then this will be automatically hashed to `"41B19A2B09E17D4FA9A97C63ADC44BF2"` before it is registered onto Airbloc.

6.2. Identity Masking

Masking is the process of substituting both PIID and DIID with ANID (Anonymized IDs).

Masking is necessary because if data owners were to use PIIDs or DIIDs directly, they will be vulnerable to the risk of ID back-tracking since PIID and DIID are closely associated with a person's identity. When these data are compromised, malicious attackers can expose targeted advertisements, send emails, and even make phone calls, thus there might be direct infringement in personal privacy.

After masking, ANIDs will be coupled with payload data to constitute the actual data that will be processed in Airbloc.

ANIDs must meet the following attributes:

1. **Anonymity**: Except for identity manager nodes, all other stakeholders in Airbloc with ANIDs should not have any information regarding the person's original identity such as PIID or DIID.

2. **In-Exchangeability**: Different stakeholders should have different ANIDs even though they are connected to the same PIID or DIID. This is to prevent stakeholders from colluding and exchanging data without purchasing data from the marketplace.

3. **Pseudo-Identifiability**: Within the same stakeholder, ANID should remain the same across one unique person and ANID should be distinguishable between different unique people

4. **Verifiability**: Identity manager nodes should be able to verify that such ANID are indeed connected to a certain PIID or DIID

To illustrate, Alice's ANID appears as X data for Enterprise A, and Alice's ANID appears as Y data for Enterprise B. X and Y both are referring to Alice, but Enterprise A and B has no way of knowing that the data owner is Alice. Also, since Bob's ANID appears as H for Enterprise A and K for Enterprise B, enterprises will be able to distinguish between each unique person via ANID differentiation.

ANIDs can be created through encryption techniques such as hashing or digital signatures. However, there is a security trade-off for using such simplistic techniques. For instance, to prove that ANID is the same as certain PIID, this PIID has to be revealed to the public network in the process.

Thus, Airbloc uses **Zero-Knowledge Proof** for the verification of an ANID by disclosing only the zero-knowledge proof about the ANID to the network instead of information regarding the PIID or DIID. This way, it allows ANIDs to be matched and verified with the PIID or DIID without disclosing these sensitive corresponding IDs to the public network.

6.3. Decoupling and Anonymizing Personal Data

Decoupling is the process of separating the storage and management of PIID and DIID with that of ANID combined with the payload.

Specifically, when data is registered to Airbloc network (through smart contract) by the data provider, identity data including PIID & DIID and ANID & payload will be separately stored and managed. PIID and DIID are stored and managed by identity manager nodes.

Only identity manager nodes will be empowered to store and manage PIIDs and DIIDs. All other identity data used in the entire data pipeline on Airbloc are ANIDs.

Payload data by its nature is deemed as already anonymized when registered onto Airbloc. This means data relayers or data processors will only handle anonymized data especially since only ANIDs will be coupled with the corresponding payload data.

Since identity and payload data are stored separately, they can only be purchased separately on Airbloc. For instance, a data consumer can purchase only identity data such as "Advertising IDs of 500 males over age 30" or purchase only anonymized data such as "installed applications list of 500 anonymous users".

If data consumers are seeking to purchase more quality and complete data, they may rely on delegated data relayers to combine these two data sets.

6.4. Private Identity Matching

Identity matching is needed because one unique person's data as a whole is collected from multiple different data providers and each data provider will be assigned one unique ANID and the person is likely to be connected to multiple other identifiers like mobile advertising ID or email address as well. Thus in order for data consumers to purchase a complete user profile, identity matching is necessary to pull all the related data to that person.

For example, some part of Alice's personal data is coming from an application A with one ANID and mobile advertising ID, the other part of Alice's data is coming from application B with another unique ANID and email address. In this example there are four different IDs from two data sources all pointing to one person. Suppose Enterprise A has an email address of Alice and wants to complete her user profile, then identity matching is used.

In this identity matching process, any participant in Airbloc can make a request for identity matching by submitting any identifier that is endorsed by the Data Schema Registry to be returned of ANID or other identifiers.

In terms of privacy protection, it should be noted that the identity matching process must be done in a private manner since transferring identity data across multiple stakeholders and participants may lead to the exposure of sensitive identity data.

Airbloc uses **zero-knowledge proof** to prevent the exposure of the sensitive identity information such as PIID or DIID during the identity matching process. Therefore, identity matching is performed by the following procedures:

1. Alice uses the arbitrary value r as Salt and computes h which is the hash of the identity information i to match.

2. Alice propagates h and r to the Identity Manager Network.

3. Bob, an identity manager which has the same i_{Bob} as i, verifies that hash i_{Bob} matches h that Alice has propagated. Confirms, $hash(i_{Bob} + r) = h$.

4. Bob sends i_{Bob} to Alice via a private channel, which certifies Alice has the identity i to match.

5. Bob returns the AID, not the actual user ID, as a result of the match. $u_A = AID(u)$

6. Bob does not just return u_A to Alice, but reveals a zero-knowledge proof π that $hash(u)$ value corresponds to u_A. However, for the above process value of $hash(u)$ must be registered on the blockchain in advance.

7. Alice proves π, and when it succeeds, identity matching is accomplished.

Simply put, this process ensures that there is indeed the existence of valid identifiers by the requestor by sending the hashed version of those identifiers to the verifiers.

7. Identity Manager Nodes

Identity manager node is a public node that any participant can install and run in Airbloc. A participant can run one or more identity manager nodes under the same group and the nodes together provide various services. All identity manager nodes as a whole would constitute the decentralized p2p network and communicate with Airbloc Standard Nodes to receive and respond to requests.

There are two considerations when handling identity data in Airbloc:

1. It is possible that the same identity data can be collected multiple times from different data providers if a data owner completes multiple DAuths on multiple data providers' applications. Therefore, if data providers store identity data like the payload, it could lead to unnecessary data duplication and data storage inefficiencies. Thus, it makes sense to delegate the task of storing identity data to identity manager nodes.
2. Since identity data is the data that can directly affect data owners' privacy, data owners must be in primary control of their identity data flows. Hence, identity manager nodes can be run by data owners themselves.

Identity manager nodes will provide three services:

1. Storing and management of PIIDs, DIIDs, and their relations to ANIDs
2. Identity matching service by matching one identifier to another identifier: a service provided to data consumers and data providers
3. Data owner account management service that helps data owners sign up for the Airbloc data control center with a temporary account that was assigned to the data owner during the DAuth process. This service also assists data owners in tracking and controlling the permission levels of data collected.

During DAuth process, data owners can select and delegate authority to an identity manager to manage their identity data along with account management. However since this process might be complicated for data owners, Airbloc provides an interface in Airbloc SDK to let data owners choose Airbloc data control center (aka "Airbloc Tracker") as their default identity manager node in a simple manner.

7.1. Running an Identity Manager Node

To run an identity manager node, certain amount of ABL tokens are required to be staked as collateral. The amount of collateral increases logarithmically - directly proportionate to the number of users' identity data the identity manager nodes chooses to store, as follows:

$$C = \gamma log_{10}(n_U)$$

where n_U is the maximum number of data owning users (Data Owners) that can be stored, and γ is a constant parameter.

Since anyone can run identity manager nodes, data providers can also stake ABL tokens to run their own identity manager nodes, and configure the use Airbloc SDK in a way to ask their users to use their identity managers to more strictly protect their users' data. Data owners can run identity manager nodes as well.

Since identity manager nodes are contributing to the maintenance of the entire ecosystem, identity manager nodes can receive contribution rewards as incentives for managing identity data. The rewards increase proportionally to the quantity of identity data they manage and the quantity of identity data managed is determined by the ABL token staked by the identity manager nodes following the equation above.

8. Data Providers

To hasten and maximize the data acquisition process on Airbloc, data on Airbloc will be primarily provided by data providers such as mobile, website, smart device applications, etc.

Data providers can register and monetize the data collected from their underlying users to Airbloc through the data collection authorization process DAuth. Payload data is stored by the Data Provider and the data availability is ensured through the data availability challenge (explained below). For security reasons, the access control of the data relies on re-encryption proxy (explained below).

8.1. Application Registration

To collect data and register data on Airbloc, data providers must first register their application on Airbloc Network. Registration of the application can be done by making a request to the Airbloc standard nodes. For the registration, the data providers are required to stake a certain amount of ABL tokens as a collateral. The maximum number of users' data that a data provider can upload on the network proportionately increases logarithmically to the ABL amount the application is staking.

The formula is as follows:

$$C_P = \alpha log_{10}(n_U)$$

where n_U stands for number of unique users provided to the network by the data provider.

This is to prevent malicious applications from attacking the network through Sybil attacks and/or data generation attack, and to ensure legitimate data availability.

8.2. Data Collection Registration

Before collecting data from users, data providers would need to follow a data collection policy which includes data schema conformity and incentive policy structure; Data providers declare the types of data they wish to collect and the Incentive Policies (flat revenue or profit sharing data monetization structure) on the blockchain. After registration, the types of the data for collection and incentive policies can then be presented to users through DAuth process.

8.3. Incentive Policy

Incentive policy defines how data providers incentivize and reward data owners for their provision of personal data. The incentive structure will be recorded on smart contract to ensure transparency in payment. Data Providers can choose from the following options:

- **Flat Data Collection Fee** The data provider pays the data owner a flat fee for data collection and all future revenue generated from the data will be solely accrued to the data provider and not shared with the data owner.

- **Revenue Share** The data provider does not pay a flat fee to the data owner upon data collection. Instead, the data provider will share a certain percentage of revenue generated from the data to the data owner each time the data is sold. This process is guaranteed by smart contract.

9. Data Collection Authorization via DAuth

9.1. DAuth Process

DAuth is a consumer-facing data collection authorization process for data providers to seek for data owners' explicit consent before collecting and monetizing their data on Airbloc. A data provider can collect data from a user only after one or more of their data types have passed through at least one DAuth authorization. This is to allow users to agree on the data collection and determine the type of data to be collected.

Data that has not been authorized by DAuth is filtered during the data cleansing process and prevented from being registered on Airbloc.

The detailed process of DAuth is as follows:

1. Application shows DAuth interface to users.
    - Shows users the data collection information; types of data the application wishes to collect and the incentive policy
2. User agrees or declines to each type of data.
3. User enters Airbloc Account ID to receive rewards.
    - Account ID is usually Email, but other identifiers can be also used. Application can automatically provide (or fill) the account ID data using the user information known by the application.
    - Even if the user does not have any Airbloc Account, the user can still receive rewards by creating a Temporary Account.
    - Account creation and management process are governed by identity managers.
4. Airbloc records the authorization on the blockchain
    - Applications can only monetize data which users authorized.
    - Unpermitted data will be filtered during the data cleansing process.

The DAuth is built on a similar architecture to that of OAuth's Three-Legged Authorization[2]. OAuth (Open Authorization) is an open standard for token-based authentication that is widely used in many services such as Facebook login system.

[2] "Three-legged OAuth flow - IBM." https://www.ibm.com/support/knowledgecenter/en/SS9H2Y_7.6.0/com.ibm.dp.doc/oauth_threeleggedflow.html.

9.2. Temporary Account Creation

Users are allowed to do DAuth and receive a reward without an Airbloc account. This is similar to traditional point reward systems — which only requires customers' phone number to save their points.

Likewise, on Airbloc, users can authorize data collection and receive rewards by entering only basic identity information (e.g. email address, phone number, etc.), without any account registration.

Users can register later if they want to control their data flows or check their reward balance. This is done to improve the UX (user experience) for the users and the application.

However, since the account does not have a password yet, there is a possibility that these temporary accounts may be attacked by identity manager nodes with malicious intent.

To prevent this, Airbloc locks the ability to change data permission and withdrawal of ABL tokens until the user switches the temporary account to an Airbloc account through the following process:

1. User signs in to Airbloc by creating a temporary account in DAuth process.
    a. User enters identity data i (e.g. Email Address, Phone Number, etc.) as the ID used for the temporary account.
2. Application hashes the identity data $h_i = hash(i)$, and sends it to the identity manager.
3. Identity manager stores the hashed identity data and the basic account information that is collected by the respective identity manager nodes.
4. Airbloc hash-locks the account on the blockchain.
5. To unlock the account (sign-in into & turn it from temporary to the active account), the user must type in the original ID i to unlock the hashlock.
    a. To further protect the user's privacy, double-hashing of the original ID can be considered.
6. User receives the reward, but withdrawal is still locked.
7. To withdraw, users must switch the temporary account to an Airbloc account.
    a. Users are required to pass an authorization process or 2FA (Two-Factor Authorization) using the original ID i to prove their ownership of the account. 2FA process can be done by sending

authentication codes via email address or phone number that user has entered into.

b. User reveals the original ID i and register her/his own wallet account.

8. Airbloc unlocks the account if $hash(i) = i$.

10. Data Storage, Security and Access Control

10.1. How Data is Stored

All data in Airbloc is stored off-chain, and data providers are responsible for storing the data.

Storing data off-chain reduces storage costs and allows data providers to store data by themselves, rather than storing it on a public data storage network. This has the benefit of preventing a data generation attack that overloads the system through the generation of massive sets of false raw data. Furthermore, storing data off-chain provides scalability as it enables Airbloc to support not only static data sets, but also dynamic data subscription services for data consumers.

Storing data off-chain may result in a possibility where data providers store data once initially and not maintain the availability and integrity of the data properly. This issue is known as *Data Availability Problem*.

Airbloc addresses this problem through the Data Availability Challenge. Data validator nodes can periodically verify that the data is being served correctly and can challenge if there is a problem with data availability.

Since payload data which accounts for most of the actual data itself is not stored and managed by Airbloc, but stored and managed by data providers off-chain, Airbloc only manages the data availability and access information; whereas identity data is stored and managed by identity manager nodes.

For this purpose, Airbloc provides a **Decentralized Data Warehouse** where the data can be registered and accessed regardless of the storage types, and uses re-encryption proxy for access control of the data to protect users' data privacy in the decentralized environment.

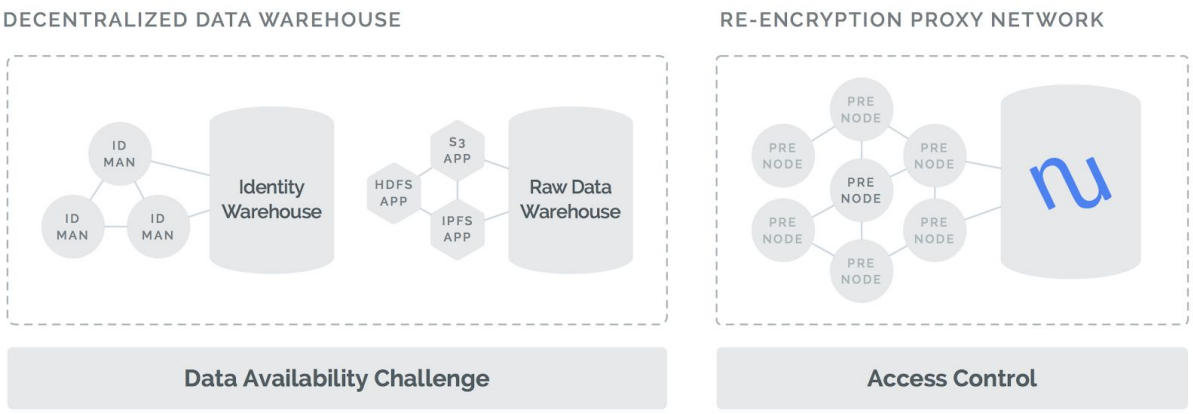

Figure 3: Decentralized Warehouse and Re-Encryption Proxy Network

10.2. Decentralized Data Warehouse

Decentralized Data Warehouse is an abstract storage layer which locates URIs (Uniform Resource Identifier) of stored data, performs encryption and decryption, and monitors data availability.

The Decentralized Data Warehouse is composed of two parts:

- **Identity Data Warehouse** stores identity data, and is maintained by Identity Managers.
- **Raw Data Warehouse** stores payload data, and is maintained by Data Providers.

To store data on the DW (Data Warehouse), data providers must first store the data on their own storage. Data providers are allowed to use any storage such as IPFS[3], Swarm[4], S3[5], and HDFS[6].

Then, data providers register the information of the data to the data registry on the blockchain. The information includes the hash of the data, collection timestamp, etc. To minimize the use of the on-chain storage, the data is registered in bulk, and only the *Merkle Root* of the bulk data—a root hash of the merkle tree built on bulk data—is stored on-chain, and the rest of the information is stored in a meta-database. The details are elaborated in Section 16.2.

[3] "IPFS is the Distributed Web." https://ipfs.io/.
[4] "GitHub - ethersphere/swarm: swarm docs." https://github.com/ethersphere/swarm.
[5] "Cloud Object Storage | Store & Retrieve Data Anywhere | Amazon ...." https://aws.amazon.com/s3/.
[6] "What is HDFS? - Apache Hadoop® Distributed File System | IBM ...." https://www.ibm.com/analytics/hadoop/hdfs.

After registering the data, the availability and integrity of the data will be periodically monitored and verified through the Data Availability Challenge, and access to the data is controlled by Re-Encryption Proxy.

10.3. Data Availability Challenge

Since the data is stored by the data provider, there is a problem if the data provider does not provide the data to the stakeholders who are supposed to have access to the data (e.g. data consumers who bought the data and data owners). Therefore, surveillance of data providers is required.

In order to ensure data provider's availability of the data, we introduce the **Data Availability Challenge**, a challenge protocol that penalizes data providers who do not provide the data upon demand. Data validator nodes can issue a challenge to the data, and they can be incentivized if the challenge is correct.

A data availability challenge is initiated by data validator nodes based on the claims by data consumers. When a data consumer has a problem accessing the purchased data, then the data consumer can make a validation request to the data validator node. Data validator node checks the availability of the validation-requested data by requesting data to the data provider and checking integrity by comparing the hash. If the node judges that there is a problem, then the node can initiate a challenge.

There are two phases in Data Availability Challenge: *Reporting* and *Replication*

- During the **Reporting** phase, the network requests data providers (Including replica nodes, described later) to reveal the data within the reporting period t.
    - Within the reporting period t, data providers need to be able to recover or retrieve the data and broadcast the data to the data validator network.
    - Reporting will end successfully if the k data validator nodes claim that they have the data, by submitting a proof-of-possession. After the successful challenge, the k nodes will be incentivized by a Contribution Reward.
    - If k nodes do not claim possession of the data until the end of the reporting period t, then it is considered that data providers have not been able to restore data availability and they will be **penalized** by having their stake slashed.

- After reporting phase ends successfully, the **Replication** phase will begin. The k validator nodes who claimed that they have the data need to replicate the data, by storing the data on its own storage and register to Airbloc Network as a replica of the data.
    - By possessing a replica, the data validator node is considered as one of the data providers from the Decentralized Data Warehouse's perspective. Therefore, the replica needs to provide the data if the existing data providers do not maintain the data availability the next time.
    - When the rest of data providers has a data availability problem (e.g. Network Fault, Collusion of bad data providers) and the replica successfully provides the data, rewards will be given to data validator nodes that possess the replica.
    - If all data providers including replicas have a data availability problem, then another challenge can arise and the replica can be also penalized.
    - This process can reduce the likelihood of data availability issues by allowing replicas of the data to store the data. This also aims to distribute the costs of maintaining availability of data that presents the problem more frequently.

The detailed mechanisms of the Data Availability Challenge such as proof-of-possession, incentive mechanisms, methods to encourage participation will be covered in the next version of the technical whitepaper.

10.4. Data Access Control

Data stored in the Decentralized Data Warehouse must be accessible only to authorized users (e.g. Data Consumers who have purchased the data, Data Owner, etc.). In a centralized environment, access control of the data can be granted by the data provider. However, in a decentralized system, it is a complicated problem because there are no trusted authorities.

If we trust data providers to perform the access control, there will be a possibility of a data withholding attack where the data providers refuse to grant access to the data providers even though they actually paid for the data. Also, if the data exists, but is not decrypted after purchase, then it is considered to not be available.

One solution to this attack vector is to delegate the access control to a third party. However, allowing third parties to manage the access to the data can infringe on users' data privacy since the data can be accessed and may be manipulated or maliciously used by the third party.

Therefore, Airbloc uses *Proxy Re-Encryption*[7] to perform access control of the data to prevent any data privacy infringements by third parties. Proxy Re-Encryption re-encrypts the intermediate encrypted data to the accessor's (referring to data consumer) private key and delegates access control to a third-party proxy. Re-encryption can be done without exposing the original content of the data or the private key to third parties. With these characteristics, access control can be performed securely in a decentralized environment.

Instead of using a single proxy for re-encryption, using a decentralized re-encryption proxy network is required in a decentralized environment. Therefore, Airbloc uses **NuCypher**[8] as a re-encryption proxy network to perform secure access control. NuCypher can securely manage the re-encryption key on the distributed network using threshold secret sharing mechanism, and ensure the security of the access.

The process for accessing data using NuCypher as a re-encryption proxy is as follows:

1. Once the transaction of a data is settled, the data provider generates a re-encryption key using its own private key and the data consumer's public key to delegate access to the data consumer.
2. Data provider propagates the generated key to the NuCypher network and notifies the data consumer's ownership of that key.
3. After the key is propagated, the data consumer requests NuCypher network for the access to the data.
4. NuCypher network checks the permission level granted to the data consumer and re-encrypts the data for the data consumer.
5. Data consumer downloads the data and decrypts it using its own private key.

[7] "Proxy re-encryption - Wikipedia." https://en.wikipedia.org/wiki/Proxy_re-encryption.
[8] "NuCypher." https://www.nucypher.com/.

11. Data Relayer

11.1. Data Marketplace Requirements

In general, data marketplaces need to meet the following necessary conditions:

1. Data consumers should be able to check the size of the data (or the number of the records) each time they add filters to narrow down the target data. Also, the size of the filtered data should be computed in near real-time (NRT)

2. Data consumers should be able to export and actually retrieve the data that they have targeted to purchase

For example, when a data consumer constructs a query of creating a segment data, which is a group of filtered users, that points to `"gender = male" and "age > 30" and "city in ("Berlin", "Seoul", "New York")" and "Uber in installed applications list"` then in near real-time the size of the users who satisfy these filters should be displayed on the marketplace's dashboard. Also when the data consumer makes a purchase and requests for data querying, then the list of ANIDs of the users should be retrieved and exported.

11.2. Why Data Relayer is Required

Unlike centralized data marketplaces, Airbloc Protocol's registered data as a whole are all distributed around multiple data providers and all payload data are encrypted. Thus it is not technically feasible to query encrypted data, and especially since all the data storage locations are fragmented across data providers.

For this purpose, Airbloc Protocol assigns a data relayer to construct a queryable user profile using anonymized data aggregated from and delegated by multiple data providers; data is anonymized since only ANID and payload data is provided to the relayer.

Data relayer is a node which supports data discovery by querying on the data marketplace. It helps data consumers find and purchase their specific subsets of data under certain filters; the subset as a query result would be the list of ANIDs or the list of data IDs. Data relayers themselves can be developed into a decentralized data marketplace when designed for commercial purposes.

When a data consumer requests for a particular subset of data, the relayer queries from the pre-constructed user profile and locates the requested data to the data consumer.

Also the data provider authorizes which relays the data can be retrieved, and the relayer who is authorized to retrieve can access the anonymized data of that data provider.

Meanwhile, anonymized data can have certain value since it can be used for business intelligence or research purposes. So if the data provider requires a cost, the relays will also have to trade with each other and purchase anonymized data.

Also, since a data relayer is a query delivery service and uses computing power to aggregate data from different data providers' storages, the fee can be billed to the data consumer when the data is purchased and will be rewarded accordingly for the relaying efforts in aggregating the subset of data that the data consumer has requested to purchase.

12. Data Exchange Methods and Processes

12.1. Data Discovery

Before the data consumer buys data, they need to know the data IDs (e.g. Data set, User Segment containing ANIDs of users) for the data they want to buy. The process is called Data Discovery. Data consumers can discover the data using the following methods:

- **Via Public Data Market:** Data providers can create a unique data set using their collected data and register it on the public data market with a desired price. This way, it makes the data discovery process simple for data consumers. However, data consumers would not be able to purchase specified portions of the data as the data set has to be bought as a whole.
- **Through Data Relayers:** Since data relayers are able to construct a queryable data profile through locating the data IDs to the data consumer, discovering of the data can be done by by data relayers according to the specific conditions or filters the data consumer wants. This way, data consumers would be able to purchase data under specific conditions or filters, unlike purchasing it on the public data market where data is sold as a whole data set.
- **From Data Providers:** Data providers can directly give data IDs to data consumers. This can be on the second-party data exchange. Enterprises or data providers who do not want their data to be traded on the public data market can still exchange the data through a mutual agreement with the data consumer.

12.2. Data Exchange Process

Once the data consumer has data IDs, the data consumer can purchase data through mutual agreement. The exchange process is as follows:

1. The data consumer offers the price of the data to data providers.
    - Alternatively, data consumers can offer ricardian contracts or smart contracts including data exchange conditions if they want a more complicated form of the exchange. For example, the data consumer can offer a discount coupon to data providers.
2. Data providers can then accept or decline the offer.

3. If the data provider accepts the offer, then the data provider must authorize the data consumer so that the consumer can be granted access to the purchased data.
    - Data provider creates re-encryption key using the data consumer's public key, and broadcast it to the NuCypher network to delegate the access to the data consumer.
4. ABL tokens will be paid from the data consumer's account, and will be converted and distributed as AIR tokens. AIR Token is a virtual token, and can be exchanged with ABL on 1:1 ratio. Details about AIR are further explained in the 15.2. Airbloc Reward (AIR) section.

After the transaction is completed, the data exchange record is stored on the blockchain and data owners can be notified their data flows.

13. Data Processors

Data processors are entities who have professional expertise in data engineering and data science that can add value to raw data through the following methods:

1. OLAP (Online Analytical Processing)[9]
2. Statistical analysis
3. AI-powered techniques or machine learning techniques

Using these methods, data processors can generate the following, but not limited to, value-added commodities that can be traded in the data marketplace:

1. Reports : reports can aggregate certain category raw data and present the aggregated numbers to give useful insights (e.g. market intelligence report)
2. Inferred users attributes : inferred user attributes can enrich the sparse user-profiles and this helps businesses have better understandings of the users (e.g. "age" and "gender" of the users can be inferred from the list of installed applications)
3. Private or public API : certain statistical, AI-powered or machine-learning algorithms can be wrapped into private or public API to be served in the marketplace (e.g. lookalike engines)

The amount and scope the work that data processors are capable of doing cannot be confined to a few examples. Be that as it may, this technical paper will illustrate two common examples which are lookalike engines and market intelligence reports.

Lookalike Engine

Data processors can build lookalike engines and wrap it into private or public APIs. The APIs can be traded on the marketplace via API call or subscription.

Lookalike engines help identify new set of users with similar attributes to the original set of target users. Lookalike engine is highly important in modern digital advertising because it helps expand the size of the targeting audience. For example, with lookalike engine, services are able to calculate and target the mobile devices that are similar to those high-value existing users. Some approaches to building lookalike engine include k-means clustering, frequent pattern mining, collaborative filtering, etc.

[9] "Online analytical processing - Wikipedia." https://en.wikipedia.org/wiki/Online_analytical_processing. 접속일자: 4 10월. 2018

Market Intelligence Reports

Just like data relayers, data processors are only authorized to handle ANID and payload data. Also if the data provider requires a fee, the data processors will also have to trade with the data provider to purchase the anonymized data.

Using powerful OLAP engines like Druid[10], data processors can build reports on different markets. For example, they can build a report on the mobile application market regarding which mobile application is trending and which one is losing popularity in each category and nation. Some examples of these market intelligence reports include, but are not limited to:

1. Consumer report showing consumer goods (e.g. Shampoo, Toothpaste) trend
2. Mobile app market intelligence report showing the performance of mobile applications in a multi-dimensional way
3. Mobile SDK intelligence report showing the performance of SDKs of different SaaS (Software-as-a-Service) tools and advertising platforms (Publisher-Side)
4. Website intelligence report showing the up-to-date traffic rankings of websites

[10] Yang, et al. "A Real-time Analytical Data Store - Druid." http://static.druid.io/docs/druid.pdf.

14. Contribution Graph

14.1. Contribution Graph

Airbloc uses a system called Contribution Graph to incentivize network participants. Contribution Graph measures the contribution of participants on the network by analyzing the works of the network participants and the reputation of the entities (e.g. Data, Application, Schema) associated with the participants. Then, Contribution Rewards are distributed according to the contribution levels. Therefore, participants earn rewards in proportion to their contribution.

$$c_{pu} = \max(0, \log r_u * \log p_{pu})$$

Contribution can be expressed as the product of **Reputation** and **Participation** as in the above formula. Reputation represents the degree of network effect and indirect reliability of each participant's activity, and participation represents the amount of work a participant has done in the network during the 7 day contribution period. Therefore, to earn the rewards, a participant need to work on the network, no matter how high the reputation is.

14.2. Reputation

Reputation represents the degree of network effect and indirect reliability of the participants' activities. The metrics used for estimating one's reputation are as follows:

- Amount of AIR held
- The amount of ABL collateral held by the data provider or the identity manager
- The number of Identifier Data provided
- The number of actual usage of the created data schema

The reputation system can judge which participant in the network has contributed the most work to the network. For example, the amount of AIR held by participants can be attributed to the amount of contributions so far, since AIR can only be earned by doing work on the network.

The reputation system also serves to incentivize participants to behave in a manner that benefits the overall ecosystem. For example, to ensure that data can be easily traded between different systems, the schema of the data on Airbloc needs to be

integrated and standardized with a common data schema. Therefore, Airbloc integrates reputation system into the data schema registry to encourage the use of the existing data schema which is likely to be higher reputation.

14.3. Participation

Participation increases by performing work for the network. Below are examples of network participation methods:

- Paying for a data (or being paid for own data)
- Operation and validation of the Plasma Chain
- Performing Data Availability Challenges
- Participation to the on-chain governance

Participation does not have to be done directly. For example, if a data consumer purchases a data, the data owner and data provider are also considered to have participated to the network. Even if the activity "purchasing" is not done by themselves, they are linked to the activity because they participated in the process to facilitate the process of data purchase.

The "work" that can increase participation is limited to crypto-economically provable objective work. For example, simply running a data validator node is not considered as a work because the behavior of running a node cannot be proven objectively or technically. Work has to be done while running a node, for example, paying for data during a data trade or performing data replication or data availability challenge. This is similar to the Cryptographic Proof concept used in the Ocean Protocol's Proofed Curation Market[11].

The period for participation is same with the contribution period. Once a contribution period (7 days) has elapsed, participation is reset to maintain participation and contribution. So for participants can be rewarded with a continued contribution to the network.

14.4. Contribution Reward

Contribution rewards calculate the contribution of a participant over a seven-day contribution period and are paid out to participants according to their contribution at the end of the contribution period. The block rewards and fees collected during the

[11] "Curated Proofs Markets: A Walk-Through of Ocean's Core Token ...." 19 April. 2018, https://blog.oceanprotocol.com/curated-proofs-markets-a-walk-through...

contribution period are collected in the fee pool, and the ABL in the pool is allocated proportionally to the contribution of each participant.

The contribution reward allocation formula is as follows:

$$R_{pu} = b_p * \frac{c_{pu}}{\sum c_p}$$

where b_p stands for the amount of the reward mined in that period, and c_{pu} stands for the contribution of the user in that period.

15. Airbloc Token

Airbloc offers two types of tokens: **Airbloc (ABL**) and **Airbloc Reward (AIR)**:

- **ABL (Airbloc)**: Tradable ERC20 token that can be bought on an exchange. It is mainly used as a means of participating in the network such as payment settlement by data consumers for data exchange, and staking to register and maintain a node.
- **AIR (Airbloc Reward)**: A virtual token that cannot be traded. It is only one-way convertible back to ABL. It is used primarily as a means of providing rewards to participants on the network.

When a data consumer pays for data with ABL, it is converted to AIR and the data provider receives it as a reward. AIR can be converted back to ABL, but ABL cannot be converted to AIR.

15.1. Airbloc Token (ABL)

ABL is used as a means of payment, settlement, and participation in Airbloc. To obtain ABL, one must either buy it on an exchange or convert AIR to ABL. The use of the ABL is as follows:

- To purchase data, data consumers need to pay with ABL.
- To participate in the network and be rewarded for being a data provider, a data validator node, or an identity manager mode, a certain amount of ABL is required to be staked as a collateral.

15.2. Airbloc Reward (AIR)

AIR is a virtual token native to Airbloc ecosystem which cannot be traded on exchanges. It is used as a reward for productive behavior within Airbloc. ABL tokens that data consumer paid for data are converted to AIR tokens and rewarded to data owners and data providers of the data via smart contract. AIR is a non-transferrable token, so it needs to be converted to ABL before transferring it.

AIR is used to assess a participant's reputation and contribution. The amount of AIR depends on the level of participation that adds value to the network — such as

providing data or participating in Data Availability Challenge.

AIR has an indirect effect of controlling the supply of ABL. Participants with higher reputation can receive more rewards from participating in the network. From a business perspective, data consumers are more willing to purchase data from data providers with higher reputation. Therefore, rather than swapping AIR directly to ABL, this incentive may induce participants to hold AIR tokens. This concept is similar to Steem Power in Steem Network.

AIR can be converted to ABL, but ABL cannot be exchanged for AIR. The mechanism of converting AIR to ABL is much more convenient than the Power Down process in Steem, the reason being that there is no extra time delay in exchanging with ABL. However, the amount of AIR can be exchanged from the network to ABL per hour is limited to 0.1% of the amount of AIR generated in the current network.

15.3 Mining Contribution Reward

400,000,000 ABL tokens are initially issued. Tokens are mined through the initial 2.5% annual inflation rate and -10.9% annual depreciation rate. This reward token distributes as a reward to network participants by means of a contribution graph.

Based on the time required to form the ecosystem of the initial Airbloc, block halving period (half-life) for an additional reward token is set to six years. This allows more rewards for early network participants. The ABL is not issued indefinitely and designed to reach the convergence of the total supply at about 33% of the initial supply, so there is no risk of a decline in value due to an unlimited supply of tokens.

16. Considerations

16.1. Scalability Problems: Plasma Sidechain

Airbloc uses Ethereum. However, Ethereum is currently not efficient to handle large amounts of requests from DApps due to the low transaction throughput and operational costs due to high transaction fees. To solve this problem Airbloc uses Plasma.

Airbloc builds a sidechain using Plasma. The chain will be mounted on top of Ethereum, and most of the business logic is done in the plasma sidechain to reduce the processing load on the main chain.

The main chain is used as minimally as possible to ensure the security of the plasma sidechain, and the scalability of Airbloc can be ensured because all behavior in Airbloc will be executed on the plasma sidechain, regardless of the scalability of the main chain.

The plasma sidechain of the Airbloc is free of charge and operated by Proof-of-Authority (PoA) consensus mechanism. Nonetheless, because of plasma's structure, the security the sidechain can be ensured by the parent chain, through the periodic block header commits and Plasma Exit challenge systems.

Even if the sidechain turns faulty, users still can withdraw their assets from the sidechain safely. Nevertheless, there are some attacks that the plasma sidechain does not solve by itself:

- **Denial-of-Service (DoS) Attack**
  It is possible for an attacker to attack the blockchain by intentionally generating a large number of transactions to occupy resources since Airbloc AERO does not have any transaction fees. To counter this, Airbloc uses a Resource Allocation method similar to Steem. Users are given storage and transactional resources proportional to the amount of ABL and AIR tokens they hold and the amount of transactions they can send per unit of time is managed via Quality-of-Service (QoS) control.

- **Block Withholding Attack**
  If the operator, which is the block producer of the plasma chain, generates malicious transactions and does not provide the necessary block information to prove that the transaction is wrong— this can lead to an attack, making normal Plasma Exits impossible. Although Airbloc AERO has no room for the attack scenario described above, Airbloc is currently

working on a solution to this problem, and this will be left as a future study.

It should be noted that Plasma, an approach to solve the scalability problem, is currently still on undergoing research and development by the community. Therefore, other approaches solving scalability problem can be adopted in the future.

16.2. Storing Metadata: BigchainDB

By default, all personal data in Airbloc is stored directly by the data provider. In addition to personal data, Airbloc handles a variety of additional information (e.g. Data Registration Information, App Information, ...) called metadata which uses a meta-database for storage to ensure that only a minimum amount of information related to the smart contract or the proof of data is uploaded to the blockchain.

Metadata is stored in BigchainDB for easier information retrieval and lower management cost. Currently, BigchainDB uses Practical Byzantine Fault Tolerance-based consensus[12] (PBFT); only permissioned operators can operate BigchainDB. This makes absolute architectural decentralization difficult. Therefore, the metadatabase is governed by Airbloc and its ecosystem partners.

Since only the metadata is stored in BigchainDB, it has no privacy-related risks. The hash value and the merkle proof can be added to the blockchain to minimize using on-chain storage while maximizing security. Furthermore, once BigchainDB's decentralized structure is fully operating according to their roadmap, Airbloc will transfer the governance of the meta-database to the public so that individuals can maintain the metadatabase nodes and receive contribution rewards in a permission-less manner.

[12] "BigchainDB 2.0 is Byzantine Fault Tolerant – The BigchainDB Blog." 27 Mar. 2018, https://blog.bigchaindb.com/bigchaindb-2-0-is-byzantine-fault-tolerant-5ffdac96bc44.