RESEARCH

Wibson Protocol for Secure Data Exchange and Batch Payments

Daniel Fernandez[1], Ariel Futoransky[2], Gustavo Ajzenman[1], Matias Travizano[1], Martin Minnoni[1] and Carlos Sarraute[1*]

*Correspondence: charles@wibson.org

[1]Wibson Limited, 28 Irish Town, Gibraltar, UK

Full list of author information is available at the end of the article

Abstract

Wibson is a blockchain-based, decentralized data marketplace that provides individuals a way to securely and anonymously sell information in a trusted environment. The combination of the Wibson token and blockchain-enabled smart contracts hopes to allow Data Sellers and Data Buyers to transact with each other directly while providing individuals the ability to maintain anonymity as desired.

The Wibson marketplace provides infrastructure for individuals to securely sell personal information without sacrificing personal privacy. Data Buyers receive information from willing and actively participating individuals, known as Data Sellers, with the benefit of knowing that the personal information should be accurate and current.

We present here two different components working together to achieve an efficient decentralized marketplace. The first is a smart contract called Data Exchange, which stores references to Data Orders that different Buyers open in order to show to the market that they are interested in buying certain types of data, and provides secure mechanisms to perform the transactions. The second is used to process payments from Buyers to Sellers and intermediaries, and is called Batch Payments.

Keywords: information market; data marketplace; blockchain; smart contract; data privacy

DISCLAIMER:

This document (v2) supplements the "Wibson Whitepaper v1 dated June 1, 2018" (v1) available at `https://wibson.org/wp-content/uploads/2019/04/Wibson-Technical-Paper-v1.1.pdf` and should be read subject to and in conjunction with v1. Where there are inconsistencies between v2 and v1, v2 will supersede v1. We would recommend you also read v1.

1 Introduction

We are proposing a new market-based approach that leverages the latest developments in blockchain, cryptography and market design to connect data consumers (companies, organizations) and data owners (essentially, individuals).

Wibson is a blockchain-based, decentralized data marketplace that provides the infrastructure for individuals to securely and anonymously sell personal information that is validated for accuracy. Wibson is built on a set of core principles: transparency, anonymity, fairness, censorship resistant, and the individual's control over the use of their personal information.

The design and price of information in data markets is an active field of study [1, 2]. The perception of value of data has changed in the last years, while economists have suggested that data should be considered as labor [3]. In the Wibson marketplace that we describe herein, citizens will be able to participate in an efficiently functioning, decentralized marketplace that provides both financial incentives and control over their personal information.

The essence of blockchain lies in its ability to support trusted transactions via networked computation in place of human control [4]. By leveraging trust mechanisms arising from blockchain, the Wibson marketplace enables users to share their personal information without sacrificing privacy.

One of the biggest challenges that was presented after implementing the first version of the Wibson protocol [5] was how to reduce the gas that was being paid due to the large number of transactions that were occurring between Buyers and Sellers.

We decided to implement a separate smart contract that batches several payments and then performs a single transaction to pay multiple users simultaneously. This contract is called Batch Payments, and was introduced to reduce the gas costs associated with the operations in the Ethereum blockchain. In addition, part of the logic for paying Notaries was included in this new smart contract. As a consequence, the Data Exchange contract was also considerably modified – its new version is presented in Sections 7 and 8.

The remainder of this paper is organized as follows. We recall the marketplace definitions in Section 2. We define the participants of Wibson in Section 3 and its main components in Section 4, including the message structure for all off-chain communications. Section 5 explains the querying system for Buyers, and describes the Data Ontology and Data Order structure. Section 6 briefly introduces the Batch Payments system. Section 7 explains the mechanism used to perform atomic data exchange operations in the marketplace. Section 8 puts all the pieces together and describes the new version of the whole Wibson protocol, and Section 9 concludes the paper.

2 Marketplace Definitions

We recall the notion of Decentralized, Privacy-Preserving Data Marketplace (dPDM) which was introduced in [5].

Definition 2.1 A *Data Marketplace* (DM) is a platform for the trade of information which provides:

- The infrastructure for a market whereby parties engage in exchange: sellers offering their data in exchange for money from buyers.
- Allow any data tradeable item to be evaluated and valued.
- Incentives for all players to be honest, and an enforcement system to take actions if a dishonest behavior is chosen.
- Incentives for all players to ensure that data is trustworthy, and to provide quality data with the addition of an enforcement system to ensure this.

Definition 2.2 A *Decentralized Data Marketplace* (dDM) is a DM where:

- There is no central authority which regulates the participants of the market.
- There is no central data repository. The users who generate the data are the owners of the data, and keep their data in their own devices having full control of their data assets.
- There is no central funds repository, therefore providing a trustless system where actors do not have to entrust their funds to a third party.

Definition 2.3 A *Decentralized Privacy-Preserving Data Marketplace* (dPDM) is a dDM which allows users to sell private information, while providing them the following privacy guarantees:

- Participants anonymity: the identity of the Sellers and Buyers is not revealed, without their consent. In particular, the identity of the Data Seller is not revealed to the Data Buyer, without the consent of the Data Seller.
- Transparency over Data usage: the Data Seller has visibility on how his Data is used by the Buyer.
- Control over Data usage: the Data Seller can modify the rights over its Data at any time.

3 Wibson Participants

Wibson provides a marketplace inspired by the dPDM definition, while allowing for trade-offs that are practical and necessary to ensure a market adapted to today's industry structure. Part of the protocol is executed on-chain, and part off-chain.

3.1 Marketplace Participants

The protocol is conducted by four types of participants: Data Seller (Seller), Data Buyer (Buyer), Notary and Delegate.

Data Seller: The Seller owns data and has rights to sell that data. The typical case is an individual selling his/her personal data.
We denote the set of Sellers as $\mathbf{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$.

Data Buyer: Any entity who wants to purchase data.
We denote the set of Buyers as $\mathbf{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_n\}$.

Notary: We introduce the role of Notary as a verification system to verify participants' information when required, verify data quality and trustworthiness when required, and arbitrate in case of conflict between Data Sellers and Data Buyers.
We denote the set of Notaries as $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_p\}$.
To qualify as a Notary, the Notary must have access to *ground truth* information with respect to the data being exchanged in the marketplace. In other words, the Notary will be an entity with information from their own files on the Data Sellers and will be able to verify that information.
We expect all Notaries to have public identities and an off-chain reputation. We give in Section 3.3 real-world examples of Notaries that can participate in the marketplace, and explain in each case the *ground truth* information that enables them to act as Notaries.

Delegate: The Delegate is any entity that participates by sending transactions on behalf of another participants in exchange of earning an agreed fee. This new role was added in order to assist users that do not have Ether in their balances to submit transactions to Ethereum, and is part of the Batch Payments system.
We denote the set of Delegates as $\mathbf{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_q\}$.

We say that the marketplace is **decentralized** since any participant which qualifies can enter the marketplace as Data Seller, Data Buyer, Notary or Delegate. There is no central authority which controls the participation in the market, or gives/denies permission to act in the market.

There are other types of participants in the Batch Payments system, such as the Monitor who generates challenges, as a mechanism to ensure no fraudulent transactions are being made in the market (for more information see [6]).

3.2 Requirements for Participants of the Wibson Protocol

Data Seller: In order to participate in the Wibson protocol, a Data Seller $\mathcal{S}$ is required to have:

- Master Ethereum address to send and receive payments [7, 8].
- Public/private keys for signing transactions and encrypting data.
- Audience attributes (see Section 5).
- The Seller must be registered in the Batch Payments smart contract. Upon registration, he/she receives an identifier $\mathcal{S}_{ID}$.

Data Buyer: In order to participate in the protocol, a Data Buyer $\mathcal{B}$ is required to have:

- Ethereum address to send and receive payments.
- Public/private keys for signing transactions and encrypting data.
- Off-chain address to receive Data Responses. The off-chain address may be a URL or an IP address in the implementation of the protocol.
- The Buyer must be registered in the Batch Payments smart contract. Upon registration, he/she receives an identifier $\mathcal{B}_{ID}$.

Notary: In order to participate in the protocol, a Notary $\mathcal{N}$ is required to have:

- Ethereum address to send and receive payments.
- Public/private keys for signing transactions and encrypting data.
- Public URL address to receive data.
- The Notary must be registered in the Batch Payments smart contract. Upon registration, he/she receives an identifier $\mathcal{N}_{ID}$.
- Mandatory: The Notary $\mathcal{N}$ must reveal his public identity, by publishing his Ethereum address and public key in a publicly verifiable place.

The Notary must register himself on the Data Exchange smart contract, and publish his Public URL. This establishes a link between the on-chain and off-chain worlds.

Additionally, in the Notary's public URL, he publishes his notarization fees (by type of data, per person) and terms of service.

Delegate: In order to participate in the protocol, a Delegate $\mathcal{D}$ is required to have:

- Ethereum address to send and receive payments.
- Register in the Batch Payments smart contract. Upon registration, he/she receives an identifier $\mathcal{D}_{ID}$.
- The Delegate must have capital in order to operate in the Batch Payments system.

3.3 Examples of Real World Market Participants

We illustrate the market roles with two examples.

3.3.1 Bank Credit Card Transactions

Suppose that the Seller is a client of a Bank, who offers on the market his (anonymized) credit card transactions. The Buyer can be any entity requiring transactional data to train its machine learning models.

In this example, the Bank is the ideal Notary since:

- The Bank can verify that the Seller is actually a client of the Bank, by requiring the Seller to provide information that authenticates him/her.
- The Bank can act as a Notary in case of conflict, and verify whether the information of credit card transactions sent by the Seller to the Buyer is valid and trustworthy (in particular, by comparing with the Bank's own records of the client's credit card transactions).

3.3.2 Location Data

Suppose that the Seller is a client of a Telecommunications company, who offers on the market his (anonymized) records with location information. The Buyer can be any entity requiring location data to train its machine learning models. In this example, the Telecommunications company (Telco) is the ideal Notary since:

- The Telco can verify that the Seller is actually a client of the Telco, by requiring the Seller to provide information that authenticates him/her.
- The Telco can act as a Notary in case of conflict, and verify whether the location information sent by the Seller to the Buyer is valid and trustworthy (in particular, by comparing with the Telco's own records of the client's location when he used mobile phone services).

4 Wibson Main Components

These are the main components of the Wibson protocol. We describe them in more detail in the following sections.

Querying System for Buyers. *On-chain:* Buyers communicate their data requirements by placing Data Orders on the blockchain. We describe this system in Section 5.

Data Pricing Mechanism. *On-chain:* The Buyer publishes on the blockchain the price offered for each Data Order. After selecting Data Responses, the Buyer publishes on the blockchain the selected Sellers, the price paid, and the hashes of the data.

Payment System. *On-chain:* The initial implementation uses Ethereum smart-contracts [7, 8] and an ERC20 token [9] using the Zeppelin's implementation *Standard Token*. The initial supply is 9,000,000,000 tokens with nine decimals. The second part of the payment system is Batch Payments, which we describe in more detail in Section 6.

Incentive System. The system provides mechanisms and incentives to certify participants, verifies that data is trustworthy, and incentivizes honest behavior of marketplace participants.

Off-chain: Verification is performed by Notaries based on their proprietary information. Notaries audit transactions by signing them off-chain. The result of Notary audits is sent to the Buyer and used to determine payments to honest participants. Notaries earn tokens by verifying participants' information and validating data.

On-chain: Unlock of payments is performed on-chain by the Notary.

Data Exchange. Wibson implements ideas presented in the Secure Exchange of Digital Goods [10]. Section 7 explains the mechanism to perform atomic data exchange operations.

Off-chain Messages

Every message exchanged between parties in the Wibson Protocol must follow the Off-chain Message structure, comprised of:

1 The payload field: has the contents of the message.
2 The signature field: contains the sender's signature of the payload field.

The message is encrypted with the Public Key of the recipient using Elliptic Curve Digital Signature Algorithm (ECDSA) before being sent [11].

5 Querying System for Buyers

The Data Buyers communicate their data requirements by placing Data Orders on the blockchain. The Buyer indicates in the Data Order the intended audience and requested data, both specified using the Data Ontology. This section explains these concepts.

5.1 Data Ontology for Audiences and Data Requests

Definition 5.1 The *Data Ontology* is a publicly available document that formalizes naming, definition, structure and relationships [12] for the marketplace's data and can be used as a reference to generate Audiences and Data Requests.

The Data Ontology is comprised of a comprehensive variable list that defines available *Data Entities*, *Data Query Models for each variable type*, and *Audience Query Models* to filter available Data Sellers. Each particular implementation must define variables available in each category.

Given the publicly available definition of Data Ontology, a Data Buyer requests a particular Data Entity (e.g., browsing history) with additional parameters defined in the Data Query Model (e.g., two days of history) from an audience defined in the Audience Query Model (e.g., men who reside in Spain).

Data Entity: Specific data owned by a Data Seller $\mathcal{S}$.

- Example 1: Browsing history.
- Example 2: Historical credit card transactions.
- Example 3: Seller's mobile phone Ad ID.

Data Query Model: A set of parameters that a Data Buyer $\mathcal{B}$ can use to define the specific data amount, quality and type requested within a particular Data Entity.

- Example 1: Two days of Data Seller browsing history beginning on January 1, 2017.
- Example 2: All online credit card transactions.

Audience Query Model: A set of variables and values (or value ranges) that a Data Buyer $\mathcal{B}$ can use to request data from relevant Data Sellers.

- Example: Gender=Women, Age≥40, Income≥$200,000, Current Residency=Spain.

5.2 Data Order

The Data Order (DO) is placed on-chain by the Data Buyer and includes:

(i) audience $\mathcal{A}$,
(ii) requested data $\mathcal{R}$,
(iii) price p,
(iv) hash of the terms and conditions of data use: $H(\mathsf{tc})$,
(v) public URL to upload Data Seller's responses and encrypted data via HTTPS post for this particular DO: $U_{\mathcal{B}}$.

Audience $\mathcal{A}$: The audience is a filter of potential sellers for the data order, written in terms of the Audience Query Model defined in a publicly published data ontology.

- Example: Gender=Women, Age≥40, Income≥$200,000, Current Residency=Spain.

Data Requested $\mathcal{R}$: The data requested is a list of Data Entities with certain parameters (defined in the Data Query Model), in addition to the Audience. The data requested can be empty.

- Example 1: Credit card transactions over the last seven days.
- Example 2: Desktop browsing history over the last thirty days
- Example 3: Seller's mobile phone Ad ID.

Address to Upload Data Seller's Responses and Data $U_{\mathcal{B}}$: Public URL address where the Buyer publishes additional information, and where the Buyer can receive data.

The public URL contains additional information about each specific Data Order:

(i) the Data Buyer's public key $PK_{\mathcal{B}}$,
(ii) the name of the Buyer,
(iii) a description of the Buyer,
(iv) the Buyer's logo,
(v) the complete text of the terms and conditions of data use tc, whose hash $H(\mathsf{tc})$ matches the hash published in the DO,
(vi) intended use of data (chosen between the predefined categories),
(vii) list of accepted notaries for this DO, along with their fees, terms of services and signatures $\mathcal{L} = \{\mathcal{N}_1, \ldots, \mathcal{N}_s\}$. Buyers specify Notaries who are eligible to audit transactions, based on the match between Data Requested and Notary's verification capabilities.

6 Batch Payments

Batch Payments is an intermediate smart contract to reduce gas costs associated with operating with existing ERC20 tokens on the Ethereum blockchain. It is a proxy scaling solution for the transfer of ERC20 tokens [9]. It is suitable for micropayments in one-to-many and few-to-many scenarios, including digital markets such as the Wibson market.

In Batch Payments many similar operations are bundled together into a single transaction in order to optimize gas consumption. In addition, some costly verifications are replaced by a challenge game, pushing most of the computing cost off-chain. This results in a huge gas reduction in transfer costs. In addition, it includes many relevant features, like meta-transactions for end-user operation without ether, and key-locked transfers for atomic exchange of digital goods.

We provide here a brief overview of the Batch Payments functionality. These are the main characteristics:

1 Registration of all parties involved, where 32-bit account ids are used.
2 Buyers must add tokens to their Batch Payments account in order to make payments.

3 Buyers initiate payments by issuing a **registerPayment** transaction, which includes a per-destination amount and a somewhat-compressed list of seller-ids. In this step, the accounts of the Buyers and Sellers are updated within the Batch Payments system.
4 Sellers wait to accumulate enough payments, then initiate a **collect** transaction specifying a range of payments and a total amount corresponding to their account.
5 After a challenge period, the requested amount is added to the Seller's balance. The Seller may withdraw the tokens in his account after this step.
6 In the case of a dispute, the Seller lists the individual payments in which he/she is included. The challenger selects a single payment and requests a proof of inclusion. The loser pays for the verification game (stake).

7 Atomic Data Exchange

Here we propose a solution to the problem of trading real-world private information using cryptographic protocols and a public blockchain to guarantee the fairness of transactions. This solution is inspired by the *Secure Exchange of Digital Goods* (SEDG) protocol between a Data Buyer $\mathcal{B}$ and a Data Seller $\mathcal{S}$ [10]. The protocol relies on a Notary $\mathcal{N}$, which also plays the role of a trusted third party.

This protocol converts the exchange of data into an atomic transaction where three things happen simultaneously:

- The Buyer $\mathcal{B}$ gets access to the Data, by learning the key that enables him to decrypt C (previously received encrypted data).
- The Seller $\mathcal{S}$ gets paid for his Data when his key is revealed.
- The Notary gets paid for his services at the very same time that the Seller gets paid for his data.

This variation of the SEDG protocol allows the Notary to be paid for his services simultaneously with the Data Seller $\mathcal{S}$. The solution is described in Table 1.

The protocol begins with a **setup phase**, wherein the Notary generates a master key M that will be used to encrypt all the Sellers' keys. He also generates a $Lock = H(\mathcal{N}_{ID}||M)$ that is used by Batch Payments. To complete the setup for this transaction, he sends the $Lock$ to the Buyer.

In the following **transaction phase**, the Seller generates a random key $K_{\mathcal{S}}$, that will be used to encrypt the data only for this transaction. Then the Seller sends to the Buyer his/her data encrypted using the key $K_{\mathcal{S}}$: $C_{\mathcal{S}} = E_{K_{\mathcal{S}}}(\text{Data}_{\mathcal{S}})$

The Seller also sends the encrypted data to the Notary, along with the encryption key $K_{\mathcal{S}}$. The Notary can thus compute $\text{Data}_{\mathcal{S}} = D_{K_{\mathcal{S}}}(C_{\mathcal{S}})$ and verify the data if required.

After the Buyer confirms that he is interested in the data of $\mathcal{S}$, he sends the Notary the address of $\mathcal{S}$ and $h_{\mathcal{S}}$ (the hash of the Seller's encrypted data).

After the Notary performs the notarization and approves the data, he encrypts the Seller's key using the master key M. Here we present a simplified version, since the

Table 1 Atomic Exchange of Digital Goods

Seller $\mathcal{S}$	Notary $\mathcal{N}$	Buyer $\mathcal{B}$
	$M = \mathsf{Random}()$ $Lock = H(\mathcal{N}_{ID} \| \| M)$ $\mathsf{Send}_{Buyer}(Lock)$	
$K_{\mathcal{S}} = \mathsf{Random}()$ $C_{\mathcal{S}} = E_{K_{\mathcal{S}}}(\mathsf{Data}_{\mathcal{S}})$ $\mathsf{Send}_{Buyer}(C_{\mathcal{S}})$ $\mathsf{Send}_{Notary}(C_{\mathcal{S}}, K_{\mathcal{S}})$		
		$h_{\mathcal{S}} = H(C_{\mathcal{S}})$ $\mathsf{Send}_{Notary}(\mathcal{S}_{ID}, h_{\mathcal{S}})$
	Verify $H(C_{\mathcal{S}}) = h_{\mathcal{S}}$ $c_{K_{\mathcal{S}}} = E_M(K_{\mathcal{S}})$ $\mathsf{Send}_{Buyer}(c_{K_{\mathcal{S}}})$	
		$\mathsf{RegisterPayment}(\mathcal{S}_{ID}, Lock)$
	$\mathsf{UnlockPayment}(\mathcal{N}_{ID}, M)$	

master key M is used for the complete set of sellers approved in each transaction (the general setting is described in Section 8). The Notary sends the Buyer the key $K_{\mathcal{S}}$ encrypted with master key M: $c_{K_{\mathcal{S}}} = E_M(K_{\mathcal{S}})$.

The Data Buyer calls the **RegisterPayment** method of Batch Payments, specifying the address of the Seller, and the Lock which depends on the Notary's ID $\mathcal{N}_{ID}$ and the master key M.

Finally, the Notary reveals the master key M by calling the UnlockPayment method.

When the Notary publishes the master key M, the Batch Payments contract transfers the payments to the Seller and to the Notary. At the same time, the Buyer can compute the following:

(i) Verify the master key published M' by checking that $H(\mathcal{N}_{ID}||M') = Lock$.
(ii) Use M' to compute $K' = D_{M'}(c_{K_{\mathcal{S}}})$.
(iii) Use K' to compute Data $= D_{K'}(C)$.

After the transaction is completed, the Buyer $\mathcal{B}$ uses the master key to gain access to the Seller's data. This mechanism, wherein certain content is maintained private until a particular event (the publication of M) occurs, is reminiscent of the family of cryptographic primitives called Secure Triggers [13].

8 Wibson Protocol

We present here the mechanisms and flow of operations of the Wibson protocol. The normal flow of operations is represented in Figure 1.

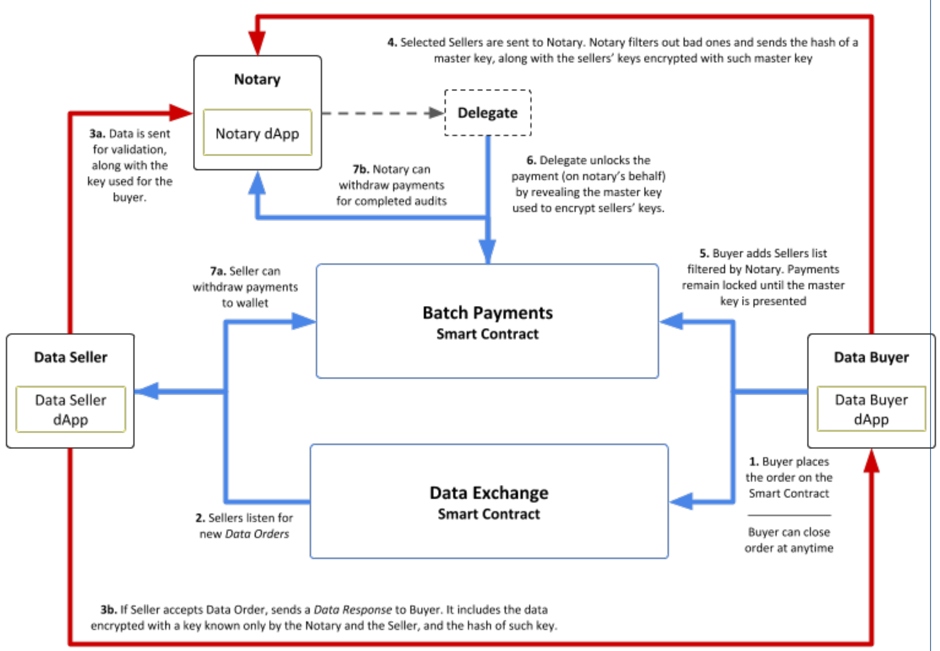

Figure 1 Wibson's normal flow of operations.

1 The Data Buyer $\mathcal{B}$ creates a Data Order query $\mathsf{DO} = \langle \mathcal{A}, \mathcal{R}, p, H(\mathsf{tc}), U_{\mathcal{B}} \rangle$ in the Data Exchange, obtaining an ID for it in return. The DO includes:
   (a) audience $\mathcal{A}$,
   (b) requested data $\mathcal{R}$,
   (c) price p,
   (d) hash of the terms and conditions of data use $H(\mathsf{tc})$,
   (e) public URL to upload Data Seller's responses and encrypted data via HTTPS post $U_{\mathcal{B}}$ for this particular DO. The public URL contains additional information for this specific Data Order, including the Data Buyer's public key $PK_{\mathcal{B}}$ and the list $\mathcal{L} = \{\mathcal{N}_1, \ldots, \mathcal{N}_s\}$ of accepted notaries for this DO.

   The creation of the DO will emit an event so that everyone is aware of this new Data Order.

2 Sellers monitor Data Orders and look for opportunities where they:
   (a) match Data Order's audience $\mathcal{A}$,
   (b) agree on data requested $\mathcal{R}$,
   (c) accept the Data Order price p,
   (d) accept terms and conditions of data use tc,
   (e) accept one of the suggested Notaries in $\mathcal{L}$,

3 If the Data Seller $\mathcal{S}$ accepts a Data Order, he or she sends a signed Data Response (Off-chain Message) directly to the Data Buyer's public URL $U_{\mathcal{B}}$, and an Off-chain Message directly to the Notary's public URL $U_{\mathcal{N}}$.

To encrypt the data, the Seller generates a random key $K_{\mathcal{S}}$, that will be used only for this transaction.

The Data Response $\mathsf{DR} = \langle \mathsf{DO}_{ID}, \mathcal{S}_{ID}, \mathcal{S}_{eth}, \mathcal{N}_{eth}, C_{\mathcal{S}}, \mathsf{nbr}\rangle$ that he sends to the Data Buyer includes:

(a) Data Order ID: DO_{ID},
(b) Batch Payments Seller ID: $\mathcal{S}_{ID}$,
(c) Ethereum Seller address $\mathcal{S}_{eth}$ (if the Data Seller has no Batch Payments Seller ID),
(d) Notary Ethereum address $\mathcal{N}_{eth}$. This is the selected Notary $\mathcal{N} \in \mathcal{L}$ who is included in the list of suggested Notaries $\mathcal{L}$.
(e) Data encrypted using the key $K_{\mathcal{S}}$: $C_{\mathcal{S}} = E_{K_{\mathcal{S}}}(\text{Data}_{\mathcal{S}})$,
(f) NeedsBuyerRegistration nbr: boolean (for the Batch Payments registration). In the case that the Seller is not registered in Batch Payments, he can request the Buyer to register him automatically for this transaction.

The Seller also sends the following Off-chain Message directly to the Notary's public URL:

(a) Data Order ID: DO_{ID},
(b) Batch Payments Seller ID: $\mathcal{S}_{ID}$,
(c) Ethereum Seller address $\mathcal{S}_{eth}$ (if the Data Seller has no Batch Payments ID),
(d) Data encrypted using the key $K_{\mathcal{S}}$: $C_{\mathcal{S}} = E_{K_{\mathcal{S}}}(\text{Data}_{\mathcal{S}})$,
(e) Key $K_{\mathcal{S}}$ used to encrypt the data sent to the Data Buyer.

4 The Data Buyer $\mathcal{B}$ selects the set of Data Sellers T from whom he wants to buy, and sends the list of Seller IDs or addresses to the Notary in the following form:

(a) Data Order ID: DO_{ID},
(b) Callback URL to receive the response from the Notary,
(c) A list of the sellers to be notarized (in the normal flow, these are all the sellers selected in T).

For each seller $\mathcal{S}$ in the list T, it includes:

(i) Batch Payments Seller ID: $\mathcal{S}_{ID}$,
(ii) Seller Ethereum address $\mathcal{S}_{eth}$,
(iii) Hash of the key $K_{\mathcal{S}}$: $h_{\mathcal{S}} = H(K_{\mathcal{S}})$.

5 The Notary will then create a master key M to encrypt and lock the data.
The Notary builds a list of notarization results for the Seller's data. The list contains for each Seller $\mathcal{S}$:

(a) Seller ID: $\mathcal{S}_{ID}$,
(b) Seller address: $\mathcal{S}_{eth}$,
(c) Notarization result specifying one of the following scenarios:
   (i) The data will not be notarized,
   (ii) The data was notarized and is valid (approved),
   (iii) The data was notarized and is invalid (rejected).
(d) Key $K_{\mathcal{S}}$ encrypted with master key M: $c_{K_{\mathcal{S}}} = E_M(K_{\mathcal{S}})$.

Then the Notary sends in response:

(a) Data Order ID: DO_{ID}.
(b) The list of Sellers containing the notarization results and encrypted keys.
(c) Notarization fee for the service performed.
(d) Notarization percentage applied (percentage of sellers to be notarized that were actually audited). The Notary decides which Data Sellers are notarized.
(e) Notary Ethereum address: $\mathcal{N}_{eth}$.
(f) Hash of the pay data payDataHash to be sent to Batch Payments (list of Seller IDs written in an efficient way, see [6])
(g) Lock to be sent to Batch Payments of the form: $Lock = H(\mathcal{N}_{ID}||M)$.

6 The Data Buyer $\mathcal{B}$ calls the **registerPayment** method of Batch Payments to add the filtered list of Data Sellers (filtered by removing the rejected Sellers) using the following parameters:
(a) Data Order ID: DO_{ID},
(b) Batch Payments Seller IDs,
(c) $Lock = H(\mathcal{N}_{ID}||M)$,
(d) Notarization fee for the service performed,
(e) Notary Ethereum address $\mathcal{N}_{eth}$,
(f) Notary signature received in previous step.

7 The Notary reveals the master key M by calling the **unlockPayment** method on the Batch Payments contract using the following parameters:
(a) Batch Id,
(b) Master Key M.
This triggers payments for the Sellers and the Notary.

8 At any time, Data Sellers or Notaries can withdraw payments by calling the Batch Payments contract.

9 Conclusion

Our aim with the Wibson decentralized marketplace presented here is to restore the individuals' ownership over their personal information. The Wibson protocol will benefit consumers by providing them the ability to control and monetize their personal information. It will also give access to high quality and verified data to organizations which need to train Machine Learning algorithms and models, as well as an explicit consumer consent mechanism which will be absolutely critical as new privacy regulations are coming into effect.

In addition to the marketplace protocol, Wibson also provides primitives to solve efficiently the problem of fair exchange [14] by providing an efficient zero-knowledge contingent payment mechanism [15, 16, 17, 18], reminiscent of the *secure triggers* cryptographic primitive [13].

By supporting the principles of transparency, anonymity, fairness and control, we believe that the Wibson marketplace will gain the people's trust needed to develop a vibrant data marketplace, that represents a fundamental change in the way organizations collect and use personal information for their data science and business analytics needs.

Acknowledgements

The authors thank Nicolás Ayala, Nahuel Santoalla, Agustín Modugno and Juan Carrillo for their work on the implementation of the Wibson platform.

Author details

[1]Wibson Limited, 28 Irish Town, Gibraltar, UK. [2]Disarmista, Buenos Aires, Argentina.

References

1. Bergemann, D., Bonatti, A., et al.: Markets for information: An introduction. Technical report, Cowles Foundation for Research in Economics, Yale University (2018)
2. Bergemann, D., Bonatti, A., Smolin, A.: The design and price of information. American Economic Review **108**(1), 1–48 (2018)
3. Arrieta-Ibarra, I., Goff, L., Jiménez-Hernández, D., Lanier, J., Weyl, E.G.: Should we treat data as labor? moving beyond "free". In: AEA Papers and Proceedings, vol. 108, pp. 38–42 (2018)
4. Zhao, J.L., Fan, S., Yan, J.: Overview of business innovations and research opportunities in blockchain and introduction to the special issue. Financial Innovation **2**(1) (2016). doi:10.1186/s40854-016-0049-2
5. Travizano, M., Sarraute, C., Ajzenman, G., Minnoni, M.: Wibson: A decentralized data marketplace. In: Proceedings of SIGBPS 2018 Workshop on Blockchain and Smart Contract (2018)
6. Mayer, H., Bejarano, I., Fernandez, D., Ajzenman, G., Ayala, N., Futoransky, A.: BatchPay: a gas efficient protocol for the recurrent micropayment of erc20 tokens (2019)
7. Buterin, V.: Ethereum: A next-generation smart contract and decentralized application platform. Ethereum White Paper (2014)
8. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper **151** (2014)
9. Vogelsteller, F., Buterin, V.: ERC-20 Token Standard. `https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md`. Accessed: 2018-01-25
10. Futoransky, A., Sarraute, C., Waissbein, A., Travizano, M., Minnoni, M.: Secure Exchange of Digital Goods in a Decentralized Data Marketplace (2019)
11. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ecdsa). International journal of information security **1**(1), 36–63 (2001)
12. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2), 199–220 (1993)
13. Futoransky, A., Kargieman, E., Sarraute, C., Waissbein, A.: Foundations and applications for secure triggers. ACM Transactions on Information and System Security (TISSEC) **9**(1), 94–112 (2006)
14. Cleve, R.: Limits on the security of coin flips when half the processors are faulty. In: Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, pp. 364–369 (1986). ACM
15. Campanelli, M., Gennaro, R., Goldfeder, S., Nizzardo, L.: Zero-knowledge contingent payments revisited: Attacks and payments for services. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 229–243 (2017). ACM
16. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy (SP), pp. 459–474 (2014). IEEE
17. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable, transparent, and post-quantum secure computational integrity. Cryptol. ePrint Arch., Tech. Rep **46**, 2018 (2018)
18. Bitansky, N., Chiesa, A., Ishai, Y., Paneth, O., Ostrovsky, R.: Succinct non-interactive arguments via linear interactive proofs. In: Theory of Cryptography, pp. 315–333. Springer, Berlin, Heidelberg (2013)