ØRIGIN

Whitepaper
Version 4

The Sharing Economy Without Intermediaries

Matthew Liu, Joshua Fraser
Founders of originprotocol.com

Abstract

Origin Protocol is a set of protocols that allow developers and businesses to build decentralized marketplaces on the blockchain, with a focus on the sharing economy. These protocols enable buyers and sellers of fractional use goods and services (car-sharing, service-based tasks, home-sharing, etc.) to transact on the distributed, open web. Using the Ethereum blockchain and Interplanetary File System (IPFS), the platform and community are decentralized, allowing for the creation and booking of services and goods without traditional intermediaries.

We define protocols for representing fractional usage assets on the blockchain and enabling buyers and sellers to transact in a completely trustless and distributed fashion. We also describe additional standards to self-govern and regulate the platform, promoting free commerce while disincentivizing bad actors.

This whitepaper:

- Examines our technical proposals for designing the decentralized app (DApp) and shared data layer that powers the Origin platform
- Introduces the Origin cryptographic token and its vital role in the smooth functioning of the Origin platform

For a comprehensive examination of the market need and product features of Origin, please see our Product Brief.

Table of Contents

Introduction

The decentralized sharing economy

Sharing economy global bookings are expected to top $335B by 2025[1], with buyers and suppliers meeting on marketplaces like Uber, Airbnb, Postmates, Doordash, GetAround, Fiverr, and TaskRabbit. These middlemen are expected to extract $40B of platform fees annually by 2022[2].

22% of US adults have participated as suppliers of services and goods[3] in the sharing economy, and this number is expected to rise. As value exchange becomes more distributed, these buyers and sellers need a decentralized way to match and meet without the middlemen. We propose cutting out these intermediaries with the Origin platform and its new standards for buying and selling fractional usage assets and services.

Technical design overview and philosophy

Origin is built on top of several existing open-source libraries, protocols and distributed systems. It is this prior work that makes Origin possible today.

Origin is built on the Ethereum platform, the leading cryptocurrency platform that enables smart contracts to execute on the blockchain. Critical transactional data such as pricing and availability are stored directly on the blockchain. Other metadata such as descriptions, images, reputation, and reviews are stored on the Interplanetary File System (IPFS) and cryptographically linked to the contract. This allows for better scaling and minimizes the expensive computing and storage costs associated with doing everything on chain. When a data object is created in the frontend DApp (e.g. a freelance engineering listing object or a customer profile object) and stored on IPFS, a unique IPFS content hash is created to reference this data. This hash is then stored on the Ethereum blockchain.

IPFS is a content-addressable, distributed file system, allowing us to trust the integrity of the data even though it is stored outside of the Ethereum network. Storage on the IPFS network is also expected to be significantly cheaper than on the

---

[1] The sharing economy is estimated to grow from $14 billion in 2014 to $335 billion by 2025 - Brookings Institution (https://www.brookings.edu/wp-content/uploads/2016/12/sharingeconomy_032017final.pdf)

[2] The new research, Sharing Economy: Opportunities, Impacts & Disruptors 2017-2022, forecasts that the sharing economy will reach $40.2 billion in 2022, in terms of platform provider revenues, up from $18.6 billion in 2017. - Juniper Research (https://www.juniperresearch.com/press/press-releases/sharing-economy-revenues-to-double-by-2022)

[3] TIME's poll of 3,000 people, conducted by Penn Schoen Berland in late November, found that 22% of American adults, or 45 million people, have already offered some kind of good or service in this economy. - Time (http://time.com/4169532/sharing-economy-poll/)

blockchain. The launch of Filecoin will add an important incentive system to ensure the longevity of this data on the IPFS network.

We anticipate several major advances in both Ethereum (e.g. Plasma and sharding) and IPFS (e.g. use of Filecoin as incentive to increase network speeds and reliability) prior to network launch, and will complete the platform with the most current and tested technologies.

We have three overarching goals in our architecture design. First, we want to keep everything as distributed and trustless as possible. We want to avoid single points of failure of our architecture like relying on a single centralized provider like Amazon Web Services. No single entity, including Origin, should have control over the network. Second, we want to stand on the shoulders of giants and avoid reinventing the wheel whenever possible. Lastly, we want to always carefully balance performance and computation efficiency with user experience. Having the best architecture in the world is pointless if no one uses it.

Engineering architecture

Origin listings can be created using a frontend DApp to publish a JSON data object to any publicly writable IPFS gateway. This JSON data object must conform to a set of standards and validation rules to be considered valid on the network. Users can optionally sign their listings cryptographically to verify their identity using publicly auditable proofs or trusted third parties. The IPFS node will publish the listing to the IPFS network making the listing instantly available via hundreds of distributed computers around the world to anyone who knows the content hash. The content hash of the listing is then sent to a smart contract which formally publishes the listing and stores pricing and availability information along with any specified booking rules and policies.

Listings can easily be searched, browsed, and booked via the frontend DApp. Since we anticipate having too many listings to reasonably parse in a browser, the frontend DApp connects to an open-source indexing server of the user's choosing, making it possible to search and filter the entire public corpus of listings. Once a listing has been selected, a user can make a booking by sending payment to the booking smart contract along with the IPFS hash of the chosen listing and the desired interval to book. The smart contract will verify that the booking is valid and handle the transfer of tokens between the buyer and the seller, including the escrow of funds when applicable.

We anticipate most sellers will prefer to list their prices in fiat currencies which often have less volatility than digital currencies. To solve this challenge, both the booking smart contract and the indexing servers will use a common set of oracles and a shared algorithm to determine the exchange rate to be used. This allows prices to be shown to end users in their preferred fiat currencies while the correct amount of digital tokens are sent during the booking. A diverse set of oracles will be chosen to avoid introducing single points of failure into the system. We will also be closely watching the development of stable coins as a possible solution to the volatility problem.

Sellers are responsible for disclosing their preferred messaging channels in their listings through which buyers can contact them before, during, or after a transaction. Buyers can similarly indicate their preferred messaging channels when they complete a booking. Non-transactional communication between buyers and sellers will occur off-chain, and both parties are encouraged to only use secure and verifiable communication channels. For transactions that have a possibility of needing arbitration, a multisignature messaging channel should be chosen that includes the arbitrator in all communications.

All listing and transaction data is public by default, but sensitive information such as a mailing address or phone number can be encrypted and only released to confirmed buyers. Delegated access can be granted via proxy re-encryption using a secure and decentralized service like NuCypher[4].

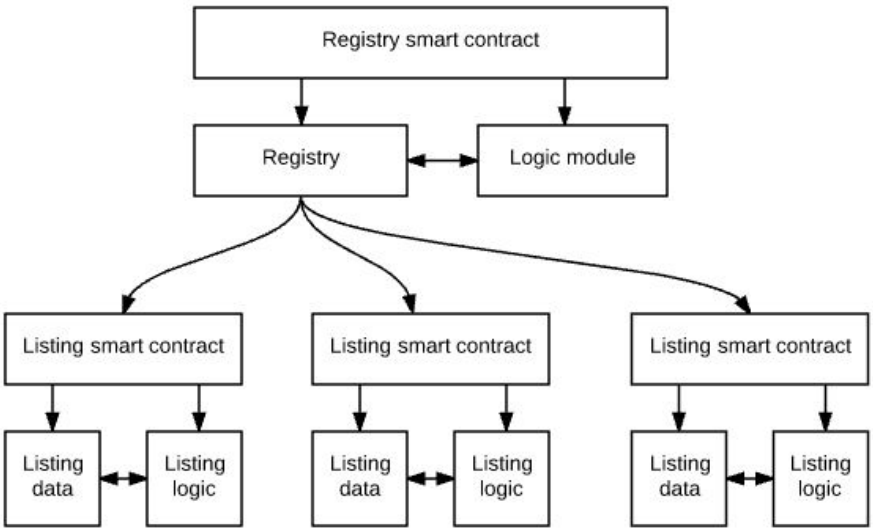
Once a transaction is complete, users are encouraged via economic incentives to leave feedback about the interaction in the form of a rating or review. Once again, the content is stored on IPFS and only the content hash is stored on Ethereum. Users are able to establish their reputations over time with verified transactions, building a unified reputation across multiple listing verticals. Buyers can use different wallets with varying levels of identity attached for certain transactions, or choose to only reveal their true identity to the seller while using a single-use wallet.

Listing policies around escrow, refunds, required deposits, and cancellations are set by the seller and are strictly enforced by the booking smart contract. Any exceptions to the policies must be handled directly off-chain by the two parties.

Core components

Solidity contracts

A series of smart contracts written in Solidity act as both the distributed database and the authoritative source of truth of all Origin listings.



These smart contracts will be used to publish and manage supplier listings, make bookings, leave reviews, and perform other interactions. We will use smart contract

[4] https://www.nucypher.com/

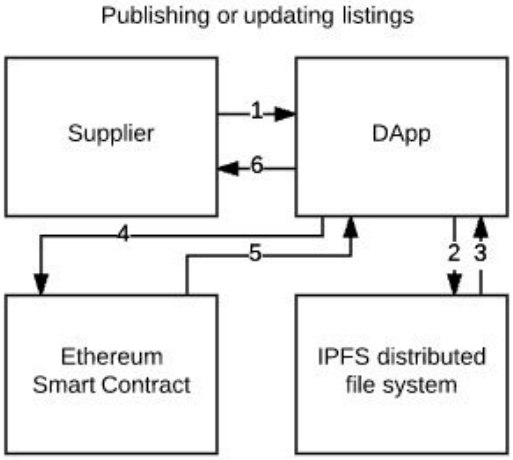
abstraction layers to enable continuous integration for deploying code updates. Each smart contract will have a wrapper contract that lives at a fixed, publicly advertised address. These wrapper contracts will import the smart contracts containing the latest business logic and listing data. Previous version contract locations are logged in a version control mapper so people can reference old contract addresses and use them directly if desired. Each Origin listing will have its own set of smart contracts which will be recorded in a single registry.

Frontend DApp

The Origin DApp is an open-source HTML and JavaScript application that connects and interacts with the Ethereum network, IPFS network, and the indexing server of your choice. The DApp allows sellers a user-friendly way to create, manage, validate and publish listings. The DApp will use js-ipfs[5] for connecting to the IPFS network and web3.js[6] for smooth integrations with popular clients like Mist[7], MetaMask[8], and Toshi[9], with fallback instructions for those who wish to transact manually. The Origin DApp will be capable of creating or booking any Origin listing, but we expect developers to create alternative versions which offer a better user experience for specific use cases or verticals. While we envision competing frontend applications, it's important to remember they will all read and write to the same shared data layer.

When a supplier creates or updates a listing:

1. A supplier connects to any Origin-enabled DApp.
2. The DApp enables the supplier to create or update a JSON object that represents their listing. The DApp validates that the submitted JSON object conforms to all of the validation rules of the selected JSON schema and then pushes the listing object to the IPFS network.
3. The IPFS network publishes the listing and returns the content hash.

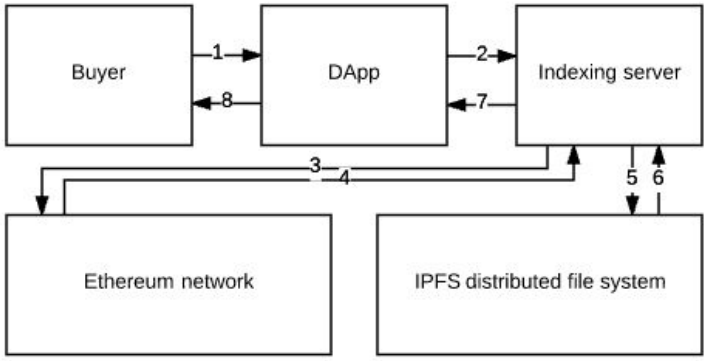


[5] https://github.com/ipfs/js-ipfs
[6] https://github.com/ethereum/web3.js/
[7] https://github.com/ethereum/mist
[8] https://metamask.io/
[9] https://www.toshi.org/

4. The DApp sends this content hash to the smart contract on Ethereum along with pricing, availability, and booking rules.
5. The smart contract returns an Ethereum transaction ID.
6. The DApp monitors the pending Ethereum transaction and notifies the user of whether or not their posting submission has been successful.

In a similar manner, buyers can search and browse listings with the addition of an indexing server for faster performance:

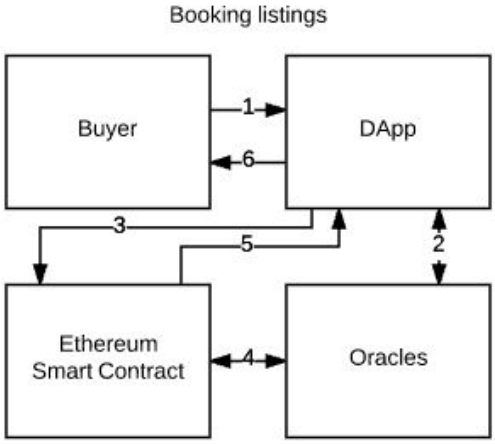
1. A buyer connects to any Origin-enabled DApp.
2. The DApp connects to a selected indexing server (either Origin hosted or a third-party alternative).
3. The indexing server requests the contents of the listings registry.
4. The smart contract returns a list of IPFS content hashes.
5. The indexing server requests each of those content hashes from the IPFS network.



6. The indexing server stores the results in a cache for future requests.
7. The indexing server returns listing results to the DApp.
8. The buyer can then browse all listing results.

Buyers can book listings as well:

1. A buyer connects to any Origin-enabled DApp along with the content hash of the desired listing and usage interval.
2. The DApp checks with a set of oracles to determine the current fair exchange rate.
3. The DApp sends the booking request to the smart contract.
4. The smart contract checks with the same oracles to make sure the correct amount of tokens were sent.
5. Assuming availability and the correct amount was sent, the smart contract will send confirmation of the booking to the



DApp and reserve that interval for the buyer.
6. The DApp will inform the buyer of the successful purchase or display any errors.

A similar architecture will be employed for less common operations such as cancellations, requesting a refund, or involving an arbitrator.

Listing indexer server

The listing indexing server is an open-source, server-side application that continually fetches the list of content hashes from the registry smart contract. It then fetches those listings from IPFS and indexes them so they can be quickly searched and filtered by DApps. The indexing servers play an important role in providing scalability to the network. The Origin indexing server will offer basic search and filtering capabilities across the entire public corpus of listings. We also envision custom DApps connecting to forked versions of our indexing server which specialize in offering custom functionality for specific verticals. For example, a marketplace for sporting event tickets would likely have very different needs from a regional car-sharing service.

Public IPFS gateway

Origin currently runs a public IPFS gateway[10] as a community service and intends to continue doing so to help bootstrap the network. Our frontend DApp will connect to this sponsored gateway by default, but users can choose to use any IPFS gateway they wish. Our sponsored gateway automatically pins all valid Origin listings that it discovers (regardless of whether it was originally published on our gateway or not) to help seed the network. It should be noted that Origin fully intends to comply with United States law and may be legally compelled to stop serving specific content at any time. If and when that happens, the availability of those listings will be dependent on at least one other node on the IPFS network being willing to serve them.

Developer tools

The Origin team plans to continue developing and open-sourcing tools to make it easy for other developers to create products that extend and improve the Origin platform.

- Origin.js[11] - We are currently developing a JavaScript library for interacting with Origin listings that provides an abstraction layer for connecting with the underlying decentralized networks.

---

[10] Our public IPFS gateway is currently hosted at http://gateway.originprotocol.com

[11] Documentation for https://github.com/OriginProtocol/origin-js can be found at http://docs.originprotocol.com/

- Listing Schemas[12] - Since many unrelated developers will be reading and writing to the same data layer, it is essential that everyone adhere to common standards. We will publish and maintain the rules for what constitutes a "valid Origin listing" as well as a library of inheritable JSON schemas for fields commonly used on listings, such as email addresses, URLs, GPS coordinates, international street addresses, international phone numbers and other metadata. These schemas are also easily extensible, enabling the creation of new product categories and the support for internationalization or other languages.
- Validators - We will create various tools for developers building on the platform to check for consistency and validity of listings.
- Notification tools - We will build open-source tools that monitor the smart contracts and send out emails, texts or other notifications to alert users when they receive relevant bookings, messages or feedback.

Decentralized service and product listings

The foundational data model for Origin is the service/product listing. Suppliers will be able to create a listing (e.g. car rental or office space lease). Buyers will be able to browse and search the listing corpus through the frontend DApp and listing indexer server. Then, they will be able to exchange Ether and/or ERC20 tokens for use of the service or asset.

Schema definition with JSON Schema

Origin uses the Mozilla project, JSON Schema[13], as a standardized structure for sharing data between DApps. JSON Schemas can be inherited and extended, allowing for a wide variety of unforeseen listings to be created on the platform. One of the important features of JSON Schemas is that they can also include validation and presentation rules so unaffiliated and even competing services can all build tools that use the same underlying data structures. The examples below are for illustrative purposes, as the formal specifications are still under active development and subject to change.

Every JSON object representing a valid listing will share a number of core components. For example, every listing object will contain metadata about the language and locale of the listing as well as a link to the Origin JSON Schema that should be used to validate it. Although prices can be denominated in fiat reserve currencies like US dollars or other digital currencies like Bitcoin, the ultimate payment will be made in Ether or an accepted ERC20 token:

```
"meta": {
```

[12] https://github.com/OriginProtocol/listing-schemas
[13] http://json-schema.org/

```
        "$schema": "https://gateway.originprotocol.com/QmPhnvn...",
        "locale": "en-US",
        "currency_symbol": "USD"
    }
```

Every listing will be required to have a name, description, and basic pricing information:

```
    "data": {
        "listing": {
            "name": "Tickets to see the SF Giants",
            "desc": "See the Giants play in AT&T park in San Francisco",
            "purchase_rules": {
                "price": "25",
                "units_available": "2"
            }
        }
    }
```

Listings can include a long list of additional fields like datetime objects, physical addresses, pictures, GPS coordinates, or other metadata. These fields all use inherited schemas to enable consistent validation across all Origin DApps.

Listings can define detailed policies around escrow, refunds, required deposits, and cancellations that are all strictly enforced by the booking smart contract:

```
    "policies": {
        "use_arbitration_escrow": true,
        "refund_policies": [{
            "days_before": 14,
            "refundable": true,
            "refund_percentage": 100
        }, {
            "days_before": 7,
            "refundable": true,
            "refund_percentage": 50
        },
        {
            "days_before": 1,
            "refundable": false
        }
        ],
        "deposit": 500
    }
```

Listings can include links to the seller's identity proofs:

```
"identities": [{
                "provider": "Facebook",
                "proof": "https://www.facebook.com/joshfraser/posts/10103717058342208"
        },
        {
                "provider": "Twitter",
                "proof": "https://twitter.com/joshfraser/status/892654623240052737"
        }
]
```

If any identity proofs are included in the listing, the entire `data` section of the listing object should be signed using the corresponding private key.

The simplest listings are single, one-time, for-sale by owner transactions. For these, the listing needs to include the required price and whether or not the item is still available. We can express this simply in our JSON listing object as:

```
"purchase_rules": {
        "units_available": 1,
        "price": 42
}
```

When an order is executed, the smart contract will decrement an internal variable counting the number of units available so everyone can query the blockchain and see that it is now sold out. The smart contract will now reject and refund anyone who tries to make a purchase of that listing. As detailed by the schema, the supplier can also increase the number of units available:

```
"purchase_rules": {
        "units_available": 5,
        "price": 42
}
```

In this example, the smart contract would decrement an internal `units_available` counter on the smart contract with each booking and store a mapping of each buyer that made a purchase until no more units are available.

It is important to understand that Solidity contracts will not publish any changes to IPFS. Instead, the smart contracts have their own internal, space-efficient representations of each field that is required to verify transactions on-chain. As such, DApps must query the Ethereum network, not IPFS, when requesting pricing rules, availability and policies. The following table shows the authoritative source of truth for each type of field:

IPFS:	Ethereum:

- Name
- Description
- Pictures
- Other listing details
- Interval definitions:
    - Initial timestamp
    - Interval size
- Listing IPFS hash
- Pricing rules
    - Base price
    - Recurring rules
- Availability & bookings (interval array with mappings to buyers)
- Policies (escrow, refunds, deposits, cancellations & arbitration)

Fractional usage, scheduling, and allocation of assets

iCalendar/jCal

Scheduling rules for fractional usage are oftentimes complicated. For example, a hotel might offer rooms on a nightly basis while requiring a 2-night booking minimum. In addition, their prices may increase on weekends and holidays. This information needs to be parsable by DApp frontend clients and enforceable by the booking smart contract. Our goal is to maximize interoperability with existing tools while minimizing storage costs on the blockchain.

There are several protocols we can use as a starting point for our work, including iCalendar (RFC 5545[14]) and it's JSON cousin jCal (RFC 7529[15]). The iCalendar format is widely supported in applications created by Google, Microsoft, and Apple, and there are client libraries available in most programming languages. In particular, the Recurrence Rule (RRULE) section of the iCalendar specification is a concise and widely adopted format for displaying recurring events on a calendar and is ideal for defining bookable intervals for Origin listings.

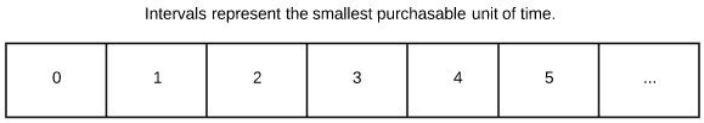
Origin uses a simplified version of jCal since our listings are already in the JSON format. Conveniently, there are already numerous conversion tools to convert jCal into the iCalendar format and vice-versa. By using the iCalendar format, sellers are easily able to view their bookings on any calendar tool that supports RFC 5545. We'll support a subset of RFC 5545 that we have extended for our own purposes.

Intervals

Listings for services or assets that are made available to different parties over time (e.g. fractional usage of an office building's conference rooms), must first be broken into intervals that represent blocks of time representing when the asset can be

[14] https://tools.ietf.org/html/rfc5545
[15] https://tools.ietf.org/html/rfc7529

booked. For example, a freelance designer may offer their services by the hour, while a hotel would offer their rooms by the day. These intervals are a fundamental building block for how we think about managing bookings on the blockchain.



Intervals are represented with whole numbers and are non-divisible since they represent the smallest purchasable unit of time on Origin. Intervals are numbered starting from 0 and are calculated relative to the interval start time (a unix timestamp) of a listing.

When a buyer books fractional usage of an asset, the smart contract does not need to understand how large of a window of time that interval represents. It only needs to know whether that interval is available and how much it costs. Listings that offer fractional usage of assets must define a constant size (in seconds) of each interval as well as the initial timestamp of the first interval. In this manner, desired windows of usage can be calculated by the DApp using basic algebra and time-zone adjustments. We intentionally handle almost all of the scheduling logic off-chain to keep storage and execution costs to a minimum.

By default, the minimum purchasable interval (interval_minimum) is 1, but it can be set to any integer greater than or equal to 1. For example, a hotel may require a minimum stay of two nights per booking.

Interval Examples

Listings offering fractional usage of assets can use RRULEs to set pricing and availability on a recurring basis. RRULEs are evaluated sequentially allowing later rules to supersede previous rules. Below are a few examples of how iCalendar RRULEs can be incorporated into Origin listings to set pricing and availability.

Example: A hotel increases prices on weekends:

```
{
        "interval_size": 86400,
        "interval_minimum": 1,
        "interval_start_time": 1514764800,
        "interval_rules": [{
                "rrule": "FREQ=WEEKLY;BYDAY=SU,MO,TU,WE,TH;INTERVAL=1",
                "available": true,
                "price": 180
        }, {
                "rrule": "FREQ=WEEKLY;BYDAY=FR,SA;INTERVAL=1",
```

```
                "available": true,
                "price": 200
        }]
}
```

Example: A home cleaner shows their availability in 30 minute intervals between 9am-4:30pm on weekdays:

```
{
        "interval_size": 1800,
        "interval_minimum": 1,
        "interval_start_time": 1514764800,
        "interval_rules": [{
                "rrule": RRULE:FREQ=MINUTELY;INTERVAL=30;BYHOUR=9,10,11,12,13,14,15,16",
                "available": true,
                "price": 30
        },{
                "rrule": "FREQ=WEEKLY;BYDAY=FR,SA;INTERVAL=1",
                "available": false
        }]
}
```

Example: A car owner offers to share their car in 15 minute intervals, with a minimum booking time of 1 hour:

```
{
        "interval_size": 900,
        "interval_minimum": 4,
        "interval_start_time": 1514764800,
        "interval_rules": [{
                "rrule": "RRULE:FREQ=MINUTELY;INTERVAL=15",
                "available": true,
                "price": 6
        }]
}
```

When a seller creates or updates their listing, the DApp converts the jCal format and associated RRULEs into a series of numbered intervals the are indexed from the `interval_start_time`. In the examples above, the `interval_start_time` of `1514764800` is the Unix timestamp for January 1st 2018 UTC. Therefore, in the hotel example, interval 0 represents Monday, January 1st, and interval 1 represents Tuesday, January 2nd, etc. The hotel's recurring weekend price increases can be represented on the blockchain using modulus functions, keeping the storage and computation required on the Ethereum network to a minimum:

```
if (interval % 7 == 4) or (interval % 7 == 5):
```

```
            price = 200
        else:
            price = 180
```

When a buyer sends a booking request for interval 12 (January 13th), both the DApp and the smart contract are aware that the higher pricing is in effect.

We recognize that dealing with dates and timezones can often be tricky. We will develop tools to help developers with the conversion process between these various scheduling formats. We realize that while this system is adequate for most common situations, it may turn out to be overly simplistic for scenarios not yet foreseen. Given the challenge of replicating calendaring software in Solidity, this is a compromise we're presently willing to make. There is more research to be done around this topic, especially on the best way to handle leap years, leap seconds, daylight savings time, and other quirks of our Gregorian calendar.

Identity management

Identity will be a core component of the Origin platform. Identity is opt-in and both buyers and sellers are free to transact pseudonymously as long as the other party consents. Both buyers and sellers are free to choose how they want to identify themselves, and which identity providers they will accept as valid from the other party.

We will encourage users to verify their identities in two keys ways.

First, we will encourage users to identify themselves on other platforms using publicly auditable proofs. For example, an Origin user can post their public key on Facebook, Twitter, or website and then cryptographically sign their listing using their private key. Users can then include links in their listings to the Facebook post, tweet, or website that displays their public key. In this manner, anyone can independently verify the poster's identity or at least confirm that they control those accounts or domain. The Origin DApp will simplify this process by making it easy to generate and to verify the proofs of users. Keybase[16] users will be familiar with this process and will be able to reuse their existing proofs to sign their listings. As people share their identity proofs on Facebook and other social networks, it will help create network effects as friends learn about Origin and decide to participate.

Secondly, we will allow users to collect verifications from trusted third-parties like Civic[17], uPort[18], and others. These third-party providers are particularly helpful for

[16] https://keybase.io/
[17] https://www.civic.com/
[18] https://www.uport.me/

identity verification that interfaces with the offline world. For example, a third-party identity provider may help confirm a physical address by sending a postcard with a special code to that address and then have the user enter that code on a website. Similar methods can be used to confirm control of a phone number or email address. Trusted third-parties can also verify government IDs like drivers licenses and passports which are required for certain types of listings like car rentals.

Origin Token

The Origin cryptographic token will be introduced to create cryptoeconomic incentives on the Origin platform.

As an ERC20 token, it will take full advantage of the Ethereum network's built-in wallets, developer tools, and resulting ease of use. It will be a fungible asset for marketplace participants that want to buy/sell the token for use on the platform.

The Origin token serves three key functions on the platform: ensuring platform security and health by creating both positive and negative incentives, enabling network governance, and facilitating buyer/seller transactions as a transfer of value.

Platform security and incentives

Designed to be an open platform without a centralized authority, Origin will launch with the appropriate mechanisms to self-regulate the community. The Origin token will be instrumental in maintaining platform security, creating disincentives to create fraudulent or inappropriate listings, accounts, and transactions.

Further, the Origin network is starting years after established incumbents like Craigslist, Airbnb, and Etsy. To compete, Origin will have a better-than-free[19] model for various buyer and seller interactions (beyond the actual transaction of services). In addition, we want to encourage a vibrant developer community and will allow developers to earn Origin token through multiple business models and/or developer grants. We believe the Origin token will provide promising positive socioeconomic incentives that spur the engagement, promotion, and development of the network.

Fraud and spam prevention

In many two-sided marketplaces, central moderators maintain large customer service, fraud, and spam prevention teams to prevent fraudulent listings and accounts from overrunning the networks. Origin explicitly lacks the trusted middleman, and as such, will deter undesired behavior with economic disincentives.

We will employ the popular Deposit-Challenge-Vote mechanic to ensure data integrity and cleanliness. This requires users to deposit Origin tokens when creating a listing, filing a complaint, etc. Provided the data is valid, the deposit is eventually

[19] After the early kinks are worked out, the token launch model will provide a technically feasible way for tech companies (and open-source projects in general) to spread the wealth and align their user base behind their success. This is a better-than-free business model, where users make money for being early adopters. - Balaji Srinivasan (https://news.21.co/thoughts-on-tokens-436109aabcbe)

returned to its original owner. However, if the data is determined to be spam or is fraudulent, the deposit will be taken from the owner.

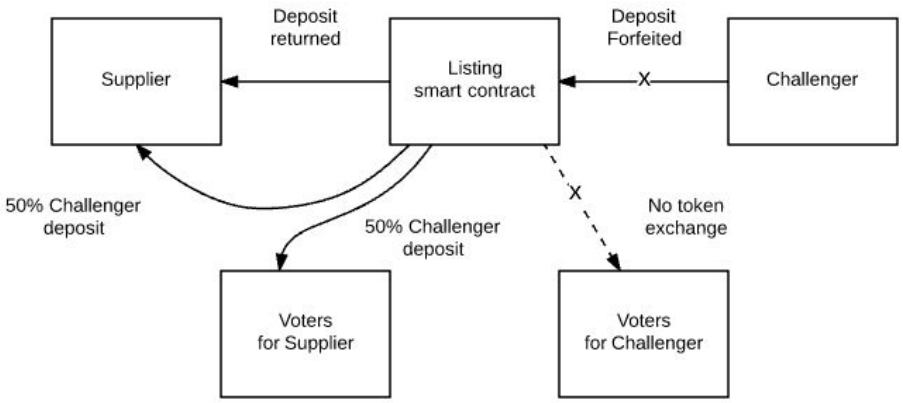
Because an initial deposit is required to publish to the Origin data layer, we believe even nominal deposit values will deter gratuitous spammers from creating outright fake listings.

For example, an independent photographer wishes to list her wedding album package. To create this listing, she deposits a requisite amount of Origin token during the listing creation process. This deposit will be held in escrow for a short period of time by the listing smart contract, and the listing will be publicly available for the entire community to see. If any other member of the community (e.g. a previous customer who was not given purchased photographs) believes her listing to be fraudulent during this challenge period, they can issue a challenge by staking an equivalent amount of Origin token against the submitted listing and providing a reason for their challenge.

Once a challenge occurs, a broader vote is opened up where other community members can in turn cast their votes with Origin token to determine whether or not the listing is valid. Once the vote is completed, the losing side (either the photographer or her challenger) forfeits their token. Both the winning side and voters for that side are given incentives in the form of a portion of the forfeited token by the loser. It is important to note that none of the tokens used by voters for either the depositor or the challenger are forfeited, as we want to give community members incentive to participate in the self-regulation of the network.

Deposit and Challenge:

Vote in favor of Supplier:



The deposit amounts for data submission will vary depending on the importance of the data as well as the likelihood of abuse. For example, more token will be required in a deposit for creating a £1,000 listing for renting a vacation home than is required for creating a $50 listing for a logo design. The deposit values will be calculated programmatically and enforced by smart contracts based on the fair market value of Origin token, the values of listings, the frequency of challenges, and the percentage of challenges that are successful.

Given the potential for fraud, participation will be limited to users with verified identities.

Arbitration

The most important version of the Deposit-Challenge-Vote mechanic is its use in arbitration after a transaction has occurred. If, for example, a renter has damaged a supplier's vehicle, the supplier may seek damages from the renter. Using photos and the written complaint submitted by the supplier, members of the community will make a decision as to whether or not damages will be paid to the supplier.

As the renter would have deposited Origin token to the transaction smart contract, this deposit would be forfeited to the supplier. In more advanced cases, the buyer may have staked additional token at the user account level (for token-verified account status as described below) and this token is also available to be used to pay for damages.

In cases where there is not enough Origin token deposited to pay for the assessed damages and the supplier has opted into an insurance policy, the initial Origin insurance pool (allocated at the token genesis event) will be used to make up the difference.

Token-verified accounts

Both buyers and sellers can increase the trustworthiness of their accounts by staking Origin token as a second way of increasing their reputation rating on the platform. Not to be confused with identity verification (described previously), this is a way for participants to assign economic value as a personal guarantee of their good behavior on the platform. This second type of verification will earn buyers and sellers additional privileges. For example, sellers will potentially show up higher in browse and search results. Buyers will be able to instantly book without having to go through a back-and-forth messaging process with some sellers. Malicious behavior (e.g. failing to return a rented asset, creating fraudulent listings) will result in the possible forfeiture of this token that is staked at the account level. In this way, the account level token deposits serve as a backup mechanism to the interaction-specific deposits listed above. These are entirely optional, but will give buyers and sellers preferential treatment on the platform. Staked funds can be withdrawn anytime, assuming the user has no pending transactions.

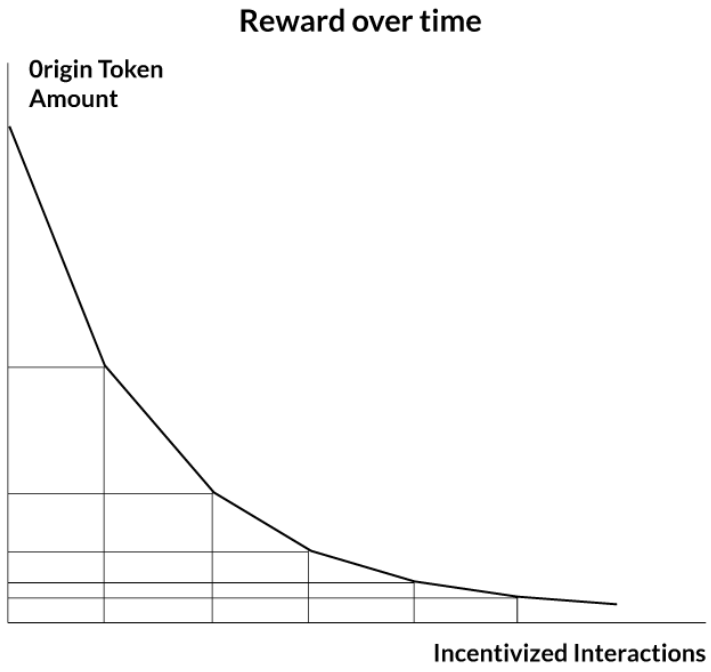
"Rewards points" for early adopters

To support the better-than-free business model and promote early user engagement, Origin platform will provide small amounts of Origin token to ecosystem participants that list their products and services, rate other users, write reviews, and verify their identities. In the listing example, this means that users will initially deposit Origin token to publish to the blockchain and IPFS (per fraud prevention above), and will receive their deposit and an additional amount of Origin token upon successful acceptance of their submission without losing a challenge.

When buyers purchase a service or usage of an asset, there will be a built in "rewards points" mechanism that will credit early users that purchase on the platform in an effort to increase transaction volume. This algorithmic token reward as a percentage of transaction value will decline over time just as a block reward for mining Bitcoin declines over time. Eventually, the reward will decrease to zero.

Referrals

Similarly, we want early buyers and suppliers to promote the platform to their peers. A referral program that offers gradually decreasing token award sizes as verified referrals are completed will be implemented to incentivize individuals and businesses to engage in grassroots marketing to increase the total number of network participants. In this way, early advocates of the platform will get a larger stake of Origin token than later advocates as they are doing the hard work of building up the buyer and seller user bases. Again, we will put the requisite verification and fraud prevention measures in place to make sure this is a healthy, honest rewards system.



Developer business models

A third major contributor to the platform will be the entrepreneurs, developers, and organizations that will contribute to the Origin platform with features, code improvements, and policy creation/monitoring. As stated previously, we encourage others to develop frontend interfaces to browse and consume Origin data. Developers of these third-party DApps and add-ons may require users to pay with Origin token with additional business models (e.g. a monthly subscription or per use fee). Another example of a novel business model is promotional placement of supplier listings within third-party DApps. A DApp developer may allows suppliers to show up in a Recommended Listings section based on the supplier's willingness to bid on a CPC or CPM basis (similar to how Amazon or Google show sponsored product listings).

We hope that third-party modules for localization and internationalization, support of many fiat currencies, and other add-ons will be built by the developer community and rewarded by the marketplace with Origin token.

Governance

An important property of the Origin token is its use in platform governance.

Origin is an open-source platform. As a community-driven project, the eventual goal is to allow Origin token holders to self-govern the direction of both software development and business/operational initiatives. We intend to allow Origin buyers, sellers, developers, and other participants to shape the direction of the project based on their ownership of Origin tokens. This is also one of the primary reasons that Origin token will be created and issued. Governance using Ether as a form of voting would give large Ether holders too much influence over the Origin platform even if those holders are not actively involved in the project.

Because account balances are managed on the public smart contract, a full record of all Origin token holders and balances is available for anyone to see. For major platform upgrades and feature updates, the community will create one or more proposals for voting by the entire ecosystem.

Prior to a vote, every holder of Origin token will receive an equivalent amount of a one-time use proposal token for each proposal. These proposal tokens will have no utility other than to cast votes. Origin stakeholders will then send proposal tokens to special smart contracts representing each proposal to cast their support. Any proposal tokens that are not used to cast votes are burned after the vote.

For example, an Origin developer may propose an initiative to add a new form of offline identity verification to improve security of the marketplace. The proposal will be communicated publicly for community members to review.

Prior to the voting period, a proposal smart contract is published that references two child contracts, one representing a Yes vote and the other representing a No vote. A snapshot of accounts and balances is taken at the designated Ethereum block in the smart contract, and one-time use tokens for approving/disapproving the identity verification initiative are deposited into each wallet that holds Origin token. An Origin token holder can then send some or all of their proposal token to the Yes vote or No vote contracts. At the end of the vote, the parent proposal contract will execute based on which child contract has received more tokens from community members. In this scenario, if 60% of the proposal is cast to the Yes vote contract, the Origin Community Fund will greenlight the new project with the software development grant.

Transactions

Buying and selling services and goods

While we expect Ether to be the primary transfer of value on the network, the third use of the Origin token is as an optional method of value transfer between buyers and sellers on the network when services and goods are exchanged. As a simple example, a

supplier of freelance video production services can be compensated in the form of Origin token upon successful completion of a marketing video.

In a more complex scenario, an initial deposit of Origin token will be made by a buyer of per use scooter membership service to a smart contract that then decrements the balance and pays the scooter supplier each time the buyer takes a ride.

Currency exchange intermediary

We understand that a new user to a DApp conforming to the Origin protocols may not have been exposed to cryptocurrencies and tokens, and may be most comfortable transacting in fiat currencies like the US dollar.

To attract and retain mainstream users, the platform will have to support on-the-fly conversions between fiat currencies and cryptocurrencies. The end user will not need to maintain a balance of Ether (or ERC20 tokens like Origin token) if they prefer to hold other currencies.

The Origin platform will be capable of integrating with various decentralized exchange protocols and stable coins to enable this improved user experience. In cases where currency pairs do not have adequate reserves, Origin token can be used as settlement currency that is transferred between buyer and seller.

Price stability

Buyers and sellers want a stable unit of account to describe the value of a good or service, and to that end we plan to allow listings in fiat denominations. Users can easily comprehend that a one hour bike rental will cost $1 USD. In contrast, there is much higher cognitive load to understand that the same listing costs 0.0033 Ether (at a $300 USD/1 Eth rate) or 1 Origin token (at a $1.00 USD/1 Origin token rate).

During the booking of the bike rental, the appropriate amount of Ether or ERC20 token will be held in escrow for eventual delivery to the supplier.

Given the high volatility of cryptocurrency assets like Ethereum and tokens, it is necessary to create a fair way to lock in the transaction at a fair and stable exchange rate.

Our initial approach will be to average out the exchange rate over different markets and time. As an example, assume that a buyer and seller are using Ether as the medium of exchange. Market prices of Ether across major exchanges in the last 24 hour period will be used to calculate the fair market price of listings at the time of booking.

Specifically, we will take the volume-weighted average price of Ether for the past 24 hours for major exchanges (e.g. Binance, Huobi, Bithumb, Kraken, Bittrex, Gdax). We will ignore both the highest and lowest values on each exchange to minimize the chance of either intentional or accidental price manipulation. We will further volume-weight each of those calculated prices based on the trading volume on each exchange. This will smooth out any pricing anomalies that may be observed in particular exchanges over that time period (e.g. the price of all assets on Kraken are trading lower than Bittrex due to downtime with the trading API).

$$Price = \frac{V_a P_a + V_b P_b + V_c P_c ... V_n P_n}{V_a + V_b + V_c ... V_n}$$

V = Volume weighted trading volume over 24 hours
P = Volume weighted price over 24 hours
a,b,c … n are the filtered exchanges (minus outliers)

Prior to the Origin Protocol network launch, we will integrate with "stable coins" that attempt to remove the volatility of cryptocurrencies. Unlike asset-backed cryptocurrencies like Tether[20] which require you to trust a centralized entity, new projects like Basecoin[21] and Fragments[22] are attempting to implement programmatic central banks that manage the inflation and deflation of the currency.

[20] https://tether.to/
[21] http://www.getbasecoin.com/
[22] https://www.frgcoin.com/

Summary

With the emergence of the Ethereum platform and IPFS, the fundamental building blocks are now in place to enable decentralized commerce in the sharing economy. Origin is launching a set of open-source protocols and standards to allow buyers and sellers to connect without rent-seeking middlemen.

Our initial approach is to use IPFS as a distributed data store of user profiles, listings, reputation information, etc. IPFS content hashes are then referenced in Ethereum smart contracts that allow for actual transactions and transfer of value.

The Origin token is being introduced as a utility token that serves multiple purposes. Both positive and negative incentives will be created by the token to ensure platform security, data validity, engagement, and growth. Further, we allow the community to participate in network governance through the Origin token. Finally, it will be an optional unit of exchange between buyers and sellers.

Origin is focused on bringing change and innovation to the sharing economy. We're excited by the opportunity to lower fees, increase innovation, free customer and transaction data, and decrease censorship and unnecessary regulation.

We are building a platform that invites other interested parties including developers, entrepreneurs and early believers to build this technology and community with us, altogether working to create the sharing economy of tomorrow. We hope you'll join us on this exciting journey.