Devery.io: An open-source protocol for verification services on the Ethereum network.

Andrew Rasheed, Antoine Najjarin, Chironjit Das

18th November, 2017

Abstract

Devery.io is developing the Devery Protocol, a decentralized verification platform that enables marking and tracking over the Ethereum network. The protocol allows manufacturers, brands, retailers and any other party to assign unique signatures to any products, services or digital goods sold, issued and traded online. The unique signatures are stored on the Ethereum network and can be queried to determine contextual data (including location, date, manufacturer/point-of-origin and the identification of the verifying party). Verification is not limited to the sale of physical goods and services, and can be extended to verifying the authenticity and legitimacy of any digital goods and services (such as certificates and courses).

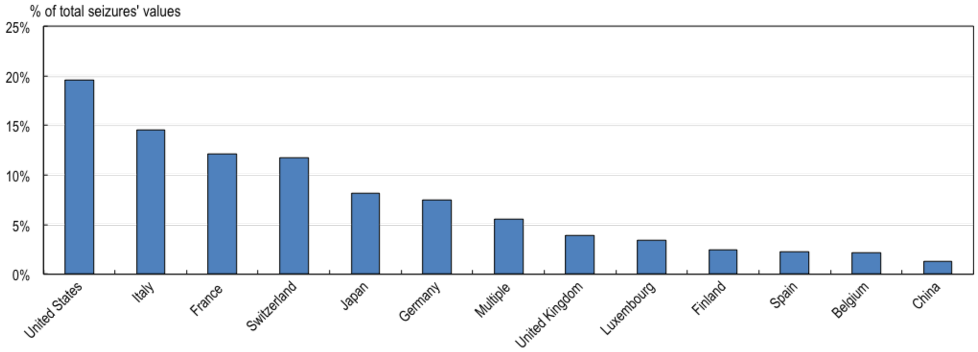
The protocol is the base layer of the Devery ecosystem. It can be used to build application level verification services and can be integrated with any existing e-commerce stores, applications or services. This fosters a competitive market of third-party verification services for specialty commercial markets, such as the clothing and apparel industry, technology, food markets, raw materials, education and other digitally sold goods and services.

An operational token, the Entry Verification Engine (EVE), is the engine that powers the protocol. The EVE token is required to generate unique signatures and contextual data on the protocol. Any application that builds on top of the Devery protocol requires the user to spend EVE tokens, which are received by the owner of the application as a fee for their verification services.

Contents

1 Background

According to a 2016 joint report by the Organization for Economic Co-operation and Development (OECD) and EU Intellectual Property Office (EUIPO), imports of counterfeit and pirated goods amount to approximately half a trillion dollars a year or around 2.5% of global imports[1].



xw

Figure 1- Data provided by the OECD.

These figures are steadily increasing each year, with the United States and Europe affected the most by its impacts. The OECD Deputy Secretary-General, Doug Frantz, states:

> "*The findings of this new report contradict the image that counterfeiters only hurt big companies and luxury goods manufacturers. They take advantage of our trust in trademarks and brand names to undermine economies and endanger lives*" [2]

Counterfeit products that have been seized range from luxury goods (such as handbags, perfumes and watches) to fake products that have the capacity to endanger lives ? toys that harm children, pharmaceuticals that do not treat the recipient, baby formula that contains dangerous ingredients, auto parts that fail and medical instruments that provide inaccurate data. There are two primary causes for the growth of the counterfeit market:

i. It is difficult to track products that moves through complex trade routes where free trade zones exist or there is weak governance; and

ii. Counterfeit products are designed to mislead and deceive, and as a result consumers are unable to distinguish between real and fake products.

2 Market Players

There are three key players that are affected by counterfeit goods and this paper will explore the impacts to each.

2.1 The seller

The seller (which may be the supplier, retailer or otherwise) is a profit-making entity and relies upon the quality of the produce sold and the reliability of the brand name developed. As the OECD states:

> "*The essential component that the commercial supply of counterfeit products relies on is 'free riding' on the economic value associated with a given intellectual property right*" [1]

Counterfeits abuse the brand name and intellectual property of other sellers, which hurts the overall reputation and profitability of a business. Further, the damage to consumer confidence affects the potential customer base of any business affected by the counterfeit products.

2.2 The purchaser

The purchaser is deceived into purchasing fake products that do not serve the expected purpose and/or may contain harmful side-effects. The consumer confidence in the online retail space is damaged due to the inability to inspect and differentiate legitimate and fake products. The element of trust is a significant factor when purchasing goods and it is routinely exploited.

2.3 The mediator

The mediator often takes the form of law enforcement seeking to seize and prevent counterfeit goods. The mediator is faced with exceedingly high costs of tracking and investigating the movement, procurement and sale of counterfeit products.

3 Devery Protocol and Ecosystem

The underlying Devery Protocol will enable developers to easily create verification applications without a thorough understanding of the blockchain. The Devery Protocol will abstract the complexities of interacting with smart contracts by deploying pre-developed smart contracts for an improved developer experience. The end result is an ecosystem of verification applications that communicate and interact with each other through the Devery Protocol.

3.1 Decentralized protocol layer

The protocol layer consists of 3 main data structures that interact via the Ethereum mapping method within the DeveryRegistry.sol and the DeveryTrust.sol contracts.

DeveryRegistry.sol
The application layer specifies an address to register the application's unique identifier on the protocol alongside the application's name and fee account. This allows the application to collect fees from users of third party verification applications built on the protocol.

```
struct App {
    address appAccount;
    string appName;
    address feeAccount;
    bool active;
}
```

Brands input their public key address to specify their corresponding brand name, and this is stored alongside the `appAccount` which will thereby be used to store the brand information and the products verified.

```
struct Brand {
    address brandAccount;
    address appAccount;
    string brandName;
    bool active;
}
```

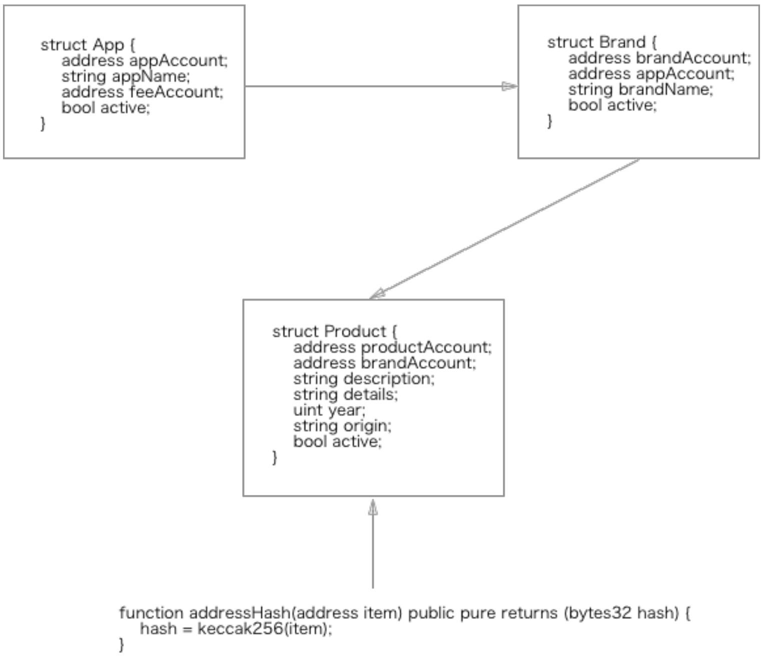
Products store the associated brand account as well as product information.

```
struct Product {
    address productAccount;
    address brandAccount;
    string description;
    string details;
    uint year;
    string origin;
    bool active;
}
```

This information is then hashed via the `addressHash(address item)` function and marked with the corresponding product public address, which uses the hashed address as a reference for lookup. This is the individual identifier for each product stored on the blockchain, and allows lookup via the `check(address item)` method.

DeveryTrust.sol
The Devery trust contract allows ethereum addresses to 'vouch' for other addresses. This allows third parties to become trusted intermediaries.

Ethereum addresses can revoke and apply vouches for brand and application keys via the `approve(address brandKey)` and `revoke(address brandKey)` public methods.



```
struct App {
    address appAccount;
    string appName;
    address feeAccount;
    bool active;
}
```

```
struct Brand {
    address brandAccount;
    address appAccount;
    string brandName;
    bool active;
}
```

```
struct Product {
    address productAccount;
    address brandAccount;
    string description;
    string details;
    uint year;
    string origin;
    bool active;
}
```

```
function addressHash(address item) public pure returns (bytes32 hash) {
    hash = keccak256(item);
}
```

Figure 2 - Visual representation of data structures.

The protocol smart contracts can be found at:
`https://github.com/devery/devery_contracts`

3.2 Devery Toolset

Open source frameworks will be provided to developers to ensure the developer experience is user friendly. Developers can opt to use utilize these frameworks within their commercial verification applications to reduce the need to interact directly with the smart contracts.

Devery.js

Devery.js provides a npm packaged Javascript framework that is built on top of Web3.js and the protocol layer to create an abstracted, developer-friendly tool to build on the Devery protocol. Using this layer, developers can opt to interact with the blockchain through Javascript and build commercial verification applications without the need to interact directly with the protocol's smart contracts.

devery_keygenerator

We have also provided a simple key generation tool to allow developers to generate public key addresses and associated QR codes for input into their applications.

The repository can be found at:
`https://github.com/devery/devery_keygenerator`

3.3 Decentralized application layer

Applications will build on top of the protocol to form the decentralized application layer. This will enable commercial applications to build and charge for services whilst using the Devery protocol.

> For Example, Bevery may choose to use the Devery protocol to verify individual beverages on the blockchain. Using the Devery protocol, developers will interact via the Devery.js framework or choose to directly develop via the smart contracts deployed on the blockchain.

3.4 Entry Verification Engine (EVE)

EVE is used to fuel the verification process. Applications will recieve EVE token as payment for hosting verification applications on the Devery protocol. Consumers using these applications will require EVE to mark on the blockchain, this EVE is then transferred to the application host as payment for hosting the applications via the protocol.

'Bokky's Token Teleportation Service' (BTTS), allows users to send EVE tokens without required Ethereum as a gas cost. Users can send message to third-party services to process their transactions and pay the gas cost on their behalf. In return, the user pays the service provider a percentage of EVE tokens. This provides a practical means of using the Devery protocol without requiring retailers and others users to hold EVE and Ethereum.

The repository can be found at: `https://github.com/bokkypoobah/BokkyPooBahsTokenTeleportationServiceSmartContract`

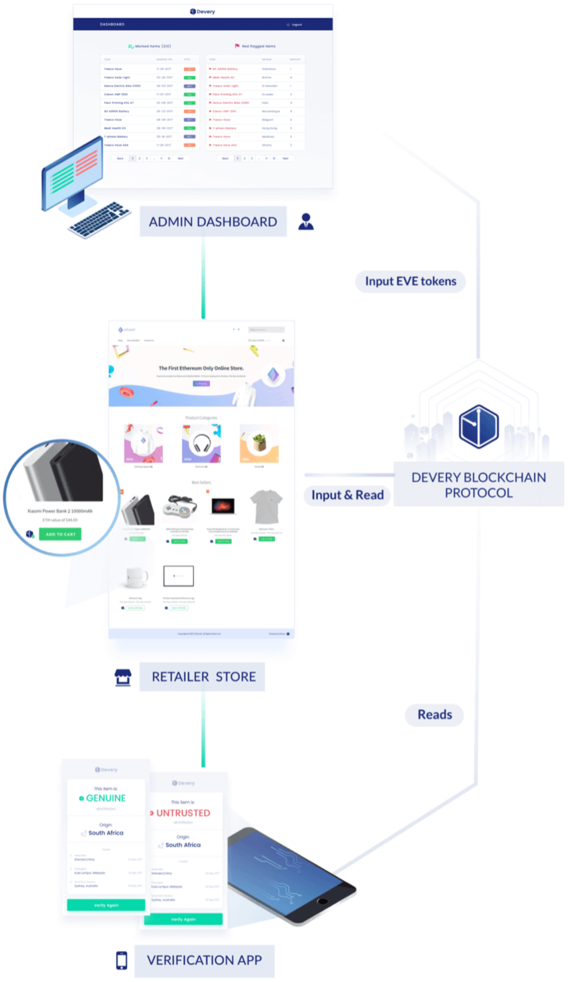
4 Use Cases

This paper will describe a few example use cases of the Devery Protocol. This is not an exhaustive list.

4.1 Online Product Verification

The Devery Protocol enables e-commerce retailers to verify the authenticity of any products or services they sell online. Retailers can assign unique ID signatures to each product sold online with a third-party verification application built on top of the Protocol. The retailer can then display unique one-time-use hashes generated from this ID to any potential customers that wish to verify the authenticity of a product. Consumers then log onto the application and input the code marked on the product in order to identify its authenticity. As well as this, origins and manufacturing can be disclosed dependent on the brand's preference.

An example of what this may look like is illustrated in the diagram below:



4.2 Digital Signing

The Devery Protocol can be used to verify that digital goods and services are issued from a legitimate source. An example would be digital certificates from online courses, colleges or universities. A certificate can be assigned a unique ID signature that can be verified via an application built on top of the Devery Protocol. The recipient of the certificate and any potential employer that wants to check its legitimacy can verify the certificate through this application. Further, details regarding the recipient?s results, behavior or other academic details can be stored on the chain.

4.3 Physical Signing

NFC and RFID chips, as well as barcodes and QR codes are compatible with the Devery Protocol. Unique ID signatures generated from the protocol can be stored into a physical marker and attached to a product. As the product moves across the supply chain, each party that handles the product can verify its source via an application on the Devery Protocol and update details (such as the location through which the product moves, timing and other conditions). The consumer can scan the hardware device to verify the movement of the product along the supply chain.

5 Roadmap

Q4 2017:Complete - Release of Devery Protocol alpha
First iteration of the Devery protocol will be released.

Q4 2017: Token sale commences
Token sale will be conducted.

Q1 - 2 2018: Onboard partners
Trial software with prexisting and new partnerships.

Q 3 2018: Release v1.0 of the Devery Protocol.
Version 1.0 of the protocol is slated to be released at this date.

2019+ : Assist with the development of verification applications for specific markets.

6 Token Sale

We will be selling 60,000,000 EVE tokens (out of a total of 100,000,000). The pre-sale commences on the 14th of December and it will require a 20 Ethereum minimum contribution. The presale provides a 5% bonus. The crowd-sale will commence on the 12th of January 2018, and will end either when the cap is reached or on 10 February 2018. All tokens will be distributed 1 week after the conclusion of the sale. Any unsold tokens will be burned.
The token sale excludes the United States of America, China, Canada, Australia and New Zealand. Acceptance of Terms of Sale, KYC and geoblocking are implemented.

References

[1] Michal Kazimierczak and Piotr Stryszowski *Trade in Counterfeit and Pirated Goods - Mapping the Economic Impact* 2016.

[2] Doug Frantz *OECD Integrity Forum* https://www.oecd.org/cleangovbiz/oecd-integrity-week.htm