

Smart Contract based API usage tracking on the Ethereum Blockchain

Patrick Holl¹ Elena Scepankova² Florian Matthes³

Abstract: API service providers usually charge their customers based on internally kept usage protocols. The whole process is highly intransparent for consumers because they are dependent on the providers' honesty. Using smart contracts on the Ethereum blockchain to log API usage creates an immutable and trustless single source of truth between consumers and providers. Leveraging the blockchain makes the whole process of API usage logging more transparent and comprehensible.

Keywords: Smart Contract; Ethereum; Blockchain; Logging; API; Usage

1 Motivation

Cloud computing and Software as a service (SaaS) is a fast-growing market. One segment of cloud computing includes provisioning of service providers' APIs. Those APIs are often used by companies or individuals who want to implement certain functionalities without developing the existing services on their own. Nowadays, there are APIs for almost everything, from general purpose machine learning to use case tailored APIs. In today's API service provider based economy, subscriptions and usage-based payment models are the defacto standard [Ma07]. Usually, consumers pay certain fees to get access to specific resources. Mostly, those resources are limited by the number of requests or the amount of transmitted data. To keep track of the consumed resources, providers log all requests associated with respective API keys. However, this process is highly intransparent for the consumer. There is currently no way to verify usage records in a trustless manner. Service consumers are required to trust the log files kept by the API providers without having the possibility to verify them. This is especially a problem when internally kept usage logs at the consumer side do not match with the ones provided by the service providers.

2 Idea & Prototype

To make API usage tracking more transparent, we are currently working on a system that leverages smart contracts on the Ethereum blockchain to store usage logs. The use of an

¹ Technical University of Munich (TUM), Department of Informatics (sebis chair), Boltzmannstraße 3, 85748 Garching bei München, Germany patrick.holl@tum.de

² TUM, Department of Informatics (sebis chair) elena.scepankova@tum.de

³ TUM, Department of Informatics (sebis chair) matthes@tum.de

immutable public and decentralized data structure like the Ethereum blockchain establishes a trustless environment between API providers and consumers. The service providers identify each request via a message that is signed with the private key of a consumer. Afterwards, the request is logged via the smart contract and written to the blockchain. Providers and consumers are always synced and share the same state since the data is publicly visible and immutable.

Authentication on conventional API requests is usually done via a specific API key and a secret. To prevent manipulation, we require a wallet signature on each request. Only those consumers who hold the private key to a certain wallet (i.e. consumers who pay for the API service) are authenticated. The smart contract enforces that only signed messages can be added to the log entries of a consumer. This prevents the API provider (or any other party) from adding requests that never occurred to the usage logs. This whole process is illustrated in Fig 1.

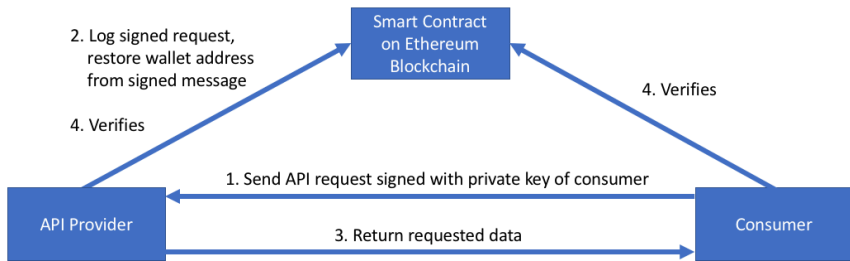


Abb. 1: Lifecycle of an API request

We are currently implementing a prototypical system which offers the stated functionality and characteristics. Primarily, small API providers profit from the proposed solution since they don't have to build their own API logging infrastructure. The smart contract represents a shared state of usage data between the API provider and the consumer.

3 Current limitations

A major reason for using Ethereum is that it's one of the most established blockchains which currently supports smart contracts [Wo14]. However, there are several challenges that are related to Ethereum. First, the transaction fees are rather high, at the time of writing the costs of a single transaction⁴ are around 0.0008 ETH which is ~ 0.24\$. These costs make the logging inefficient in high transaction scenarios. An alternative decentralized data structure to overcome the mentioned drawbacks could be the Tangle proposed by the IOTA foundation [Po16]. However, at the time of writing, IOTA does not support smart contracts which makes it impossible to build a trustless system for the proposed use case.

⁴ In Ethereum, transaction costs and block times depend on several factors. All numbers are estimated averages on the current time of writing. Source: The Ethereum blockchain itself and <https://etherscan.io> (accessed on 10/10/2017)

Literaturverzeichnis

- [Ma07] Ma, Dan: The business model of software-as-a-service". In: Services Computing, 2007. SCC 2007. IEEE International Conference on. IEEE, S. 701–702, 2007.
- [Po16] Popov, Serguei: The tangle. IOTA, 2016.
- [Wo14] Wood, Gavin: Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 151, 2014.