API3 — Decentralized APIs for the Decentralized Web

Heikki Vänttinen
Follow
Sep 16, 2020 · 6 min read

To start, I would like to thank all of the API3 partners for their help in realizing API3, as well as our whole team for their effort in the past many months it took to get to this point. From the start, API3 has been, and continues to be, a distributed endeavor to create solutions that serve the entire Web 3.0 ecosystem and everyone who would like to build externally connected decentralized applications within it.

What is API3?

API3 is a collaborative effort to build, manage and monetize decentralized APIs (dAPIs) at scale. dAPIs are blockchain-native, decentralized API services built by combining multiple provider-operated oracle nodes into aggregated data feeds, without the use of third parties. Furthermore, to achieve a system that is end-to-end decentralized, both dAPIs and the API3 project as a whole will

have a completely open and direct governance model from the beginning, powered by the API3 token and the API3 DAO.

What is an API?

To properly understand API3's value proposal, one first needs to understand what APIs are, and how these standardized data and service interfaces have transformed the way applications are built today.

An Application Programming Interface, or an API for short, is a thoroughly defined and documented protocol used by Web and mobile applications alike, that enables them to interact with one-another by way of exchanging information and services. Using an API as the channel, online businesses today are able to offer their data and services as monetizable service modules, which developers can then integrate into their applications. This, in turn, substantially improves the efficiency of building software both in terms of cost and build-time. When contrasted with a past where developers would have to build every function of their application from start to finish, it is clear why the massive increase in efficiency enabled by APIs has made them the foremost building blocks of the digital world.

Why are APIs relevant to smart contracts?

As the use-cases for blockchain technology have evolved throughout the past decade from cryptocurrencies like Bitcoin, to smart contract platforms like Ethereum, to the recent boom in DeFi (Decentralized Finance), the evolutionary line plotted through these developments has provided a relatively clear path for the future use-cases of decentralized technologies. What this path has shown is not only a broadened capacity for blockchain-based applications to execute increasingly complicated tasks in a trust-minimized manner, but also a deepened connection between decentralized applications and the real world. This has occurred through an expansion of smart contracts into services that extend beyond simple ledgers for transferring value from one account to another, and more resemble democratized services and financial products that not long ago used to only be provided by multinational corporations and large financial houses.

Indeed, today we have arrived at a reality where smart contracts and decentralized applications can be used to increase efficiency, reduce middlemen and costs, and provide increased transparency in hundreds of real-world applications through dozens of legacy industries. What's more, while the first step of this evolution towards real-world-connected decentralized applications has been in the realm of finance, we have only just gotten started. Moving beyond DeFi, smart contracts are poised to transform entire industries from insurance to supply chain management to gambling. What all of these industries have in common, besides

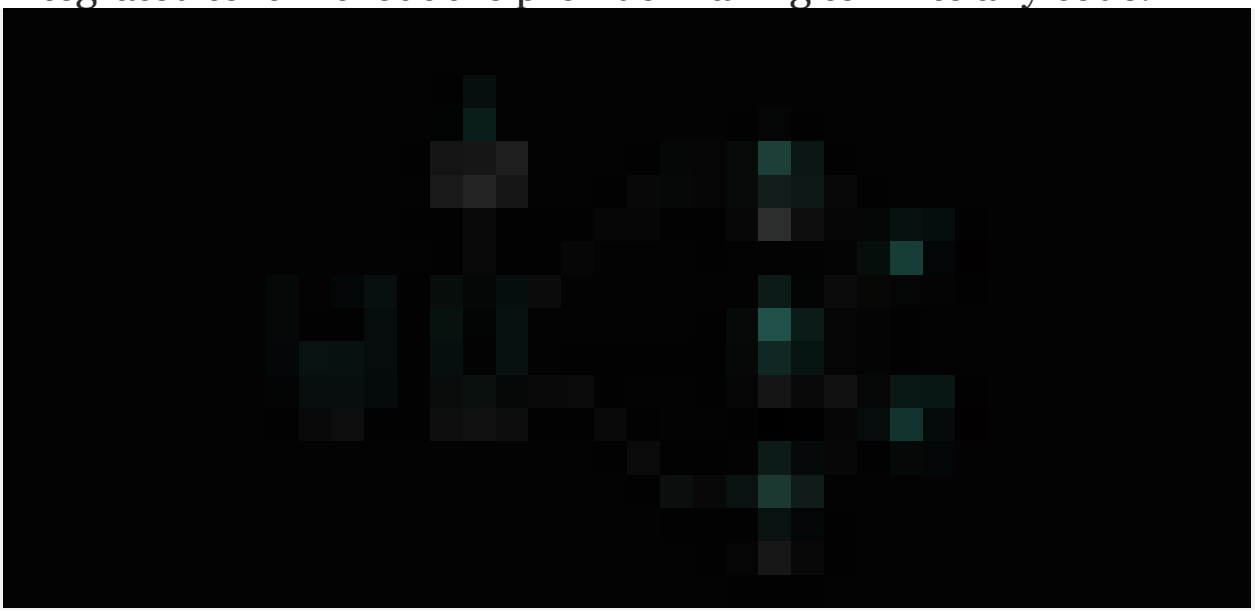
their hospitality to blockchain-derived disruption, is their reliance on timely, reliable real-world data provided by APIs.

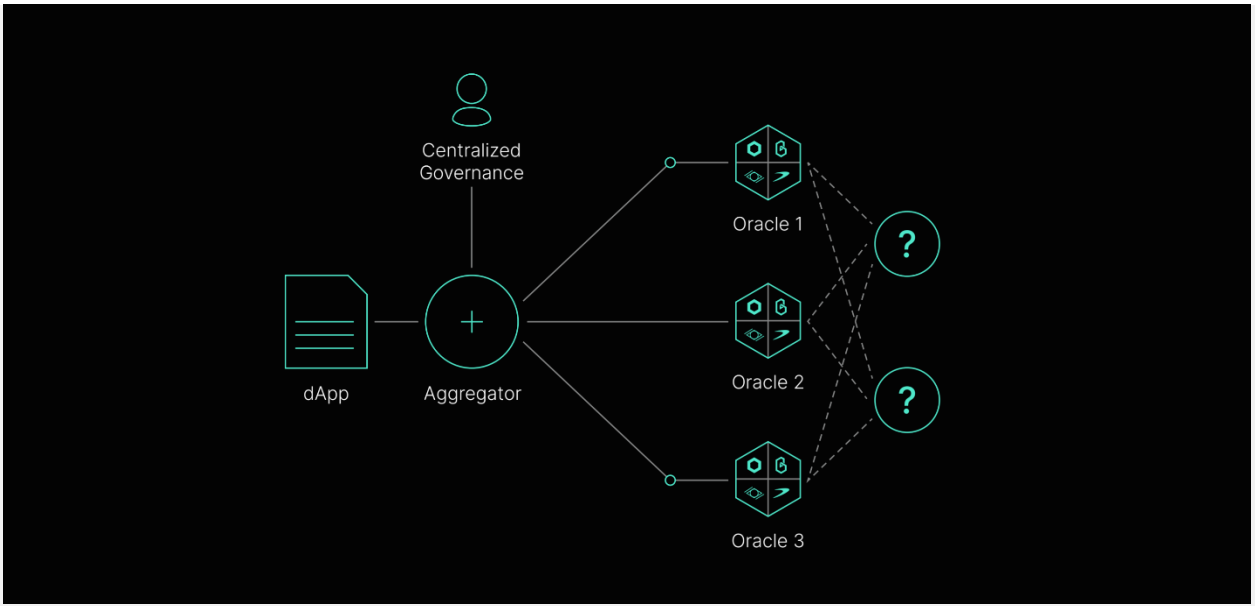
The API Connectivity Problem

The inability of smart contracts to access data not presently found in the blockchain is commonly known as *The Oracle Problem*. In practical terms, what this means is that due to the unique consensus-based security guarantees derived from using a decentralized network of nodes as the application platform, the smart contracts powering these applications cannot directly call APIs from the blockchain that the contract resides in. What is needed, then, is a solution that provides smart contracts with the ability to access API data in a way that maximally upholds the security guarantees of the underlying system, without introducing new attack surfaces to it. As this is ultimately at the core of the oracle problem as it has been defined today, we feel that it is useful to rename the problem as the *API Connectivity Problem*, as this demystified approach through a strategic reduction of scope significantly improves our ability to solve it in the most efficient and secure way. As a solution to the API connectivity problem, API3 proposes a new oracle paradigm, called a **decentralized API**.

How do dAPIs solve the API Connectivity Problem?

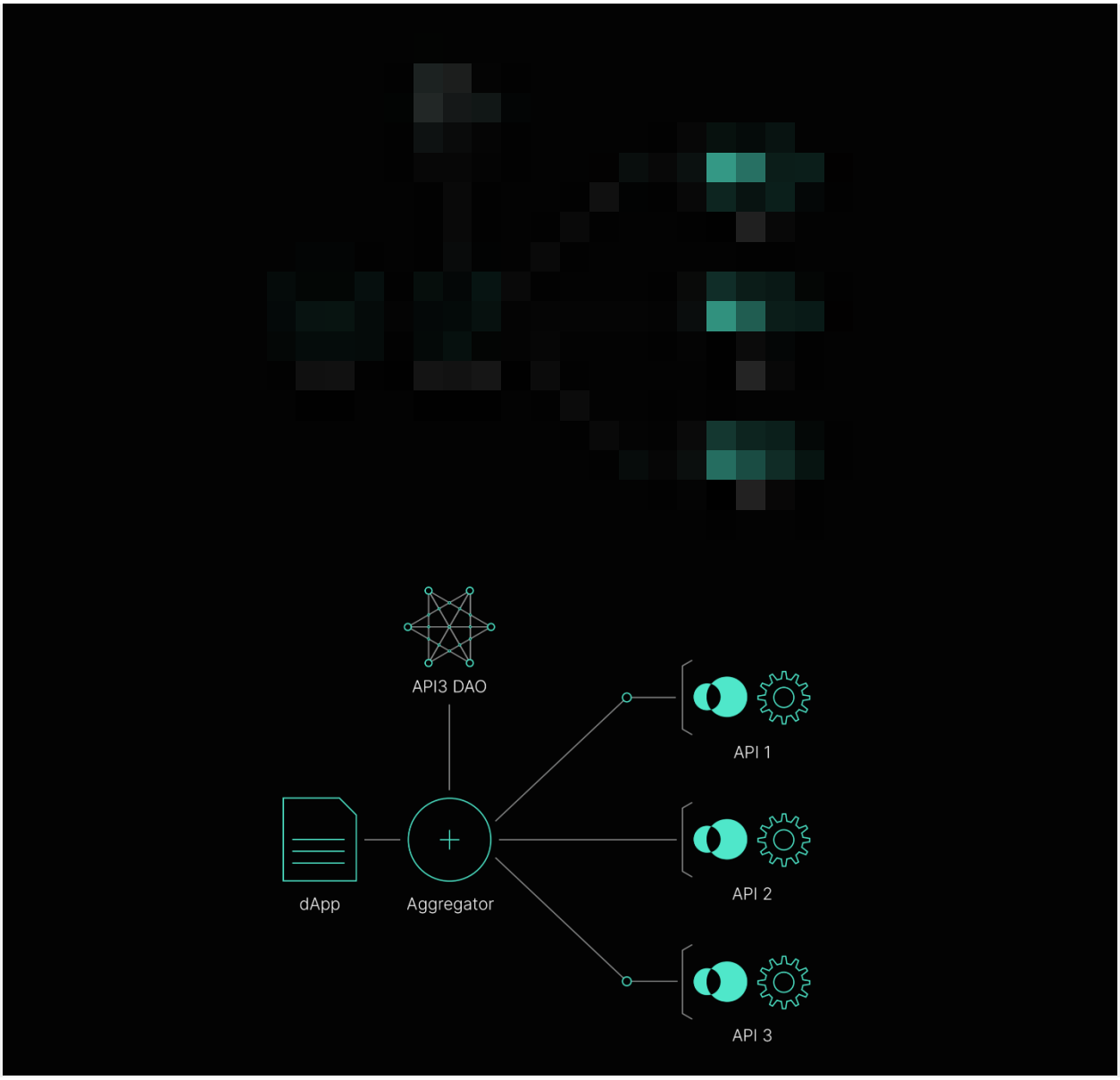
Decentralized APIs (dAPIs for short) are blockchain-native, fully decentralized API services composed of multiple data provider-operated nodes. The way in which dAPIs differ from existing decentralized oracle solutions, is that contrary to the current solutions, dAPIs include the APIs underlying the data feed in the scope of the solution. This enables them to provide superior data transparency all the way to the factual data source level, compared to existing decentralized oracles, which do not consider the data source API to be within the scope of their solution. In the context of dAPIs, this transparency is derived from having the API providers themselves run the middleware necessary to connect their API to the smart contract platform where its data and services are needed. We call this middleware the **Airnode** due to its light but robust serverless build, which can be deployed in minutes, does not require the API provider to manage and attend to the middleware on a day-to-day basis, and can have APIs integrated to it without the provider having to write any code.





Existing decentralized oracles are centrally governed and do not include the source API within their scope.

In addition to the added level of transparency provided by the use of first-party oracles, this approach also significantly reduces the transactional costs of delivering API data onto the blockchain, as it gets rid of middlemen node operators who in alternative schemes are needed to run the middleware as a compensated service to both the source data provider and the end user smart contract. As the over-redundant layer of middleman node operators is removed, transaction fees in the form of gas get lowered, and the node-specific fees flow to the factual source provider of the data.



dAPIs are decentrally governed data feeds, composed of first-party oracle nodes.

Furthermore, to improve upon the current standards of decentralization and trust-minimization employed in the governance of existing oracle solutions, dAPIs composed by API3 will be fully and transparently governed by a Decentralized Autonomous Organization (DAO) of dAPI consumers, service providers, industry experts and partners. We believe that this new

level of transparency, cost efficiency, decentralization and alignment of data provider incentives with those of the solutions they will serve, will lead to a proliferation of new data feeds available to the growing community of smart contract developers the world over.

How to get involved

API3 is formed by people who spontaneously found each other through a common understanding: Despite being largely overlooked, the API Connectivity Problem is the most critical obstacle in front of meaningful decentralized applications being built at scale.

If existing oracle solutions do not fulfill the needs of the decentralized applications you want to build, give you a fair opportunity to monetize your API business, or feel like they are working against the ethos of decentralization, we invite you to be a part of API3 so that we can solve the API Connectivity Problem together. The API Connectivity Problem is in essence an ecosystem building problem, more so than a technical one. This means that API3 can use the skills of all kinds of experts and the decentralized nature of the project will allow you to shape it in a way that makes the best use of your skills.

API3 Links

Twitter — https://twitter.com/API3DAO
Telegram — https://t.me/API3DAO (Community Chat)
Keybase — https://keybase.io/team/api3 (Developer Chat)
Github — https://github.com/api3dao
Reddit — https://www.reddit.com/r/API3/
Medium — https://medium.com/api3
DAOtalk — https://daotalk.org/c/daos/api3-dao/37

First-Party vs Third-Party Oracles

Saša Milić

Follow

Oct 18, 2020 · 6 min read

This is the third post in our series, "Getting APIs on the Blockchain". Previously, we defined and contextualized the importance of modern-day web APIs and introduced The API Connectivity Problem.

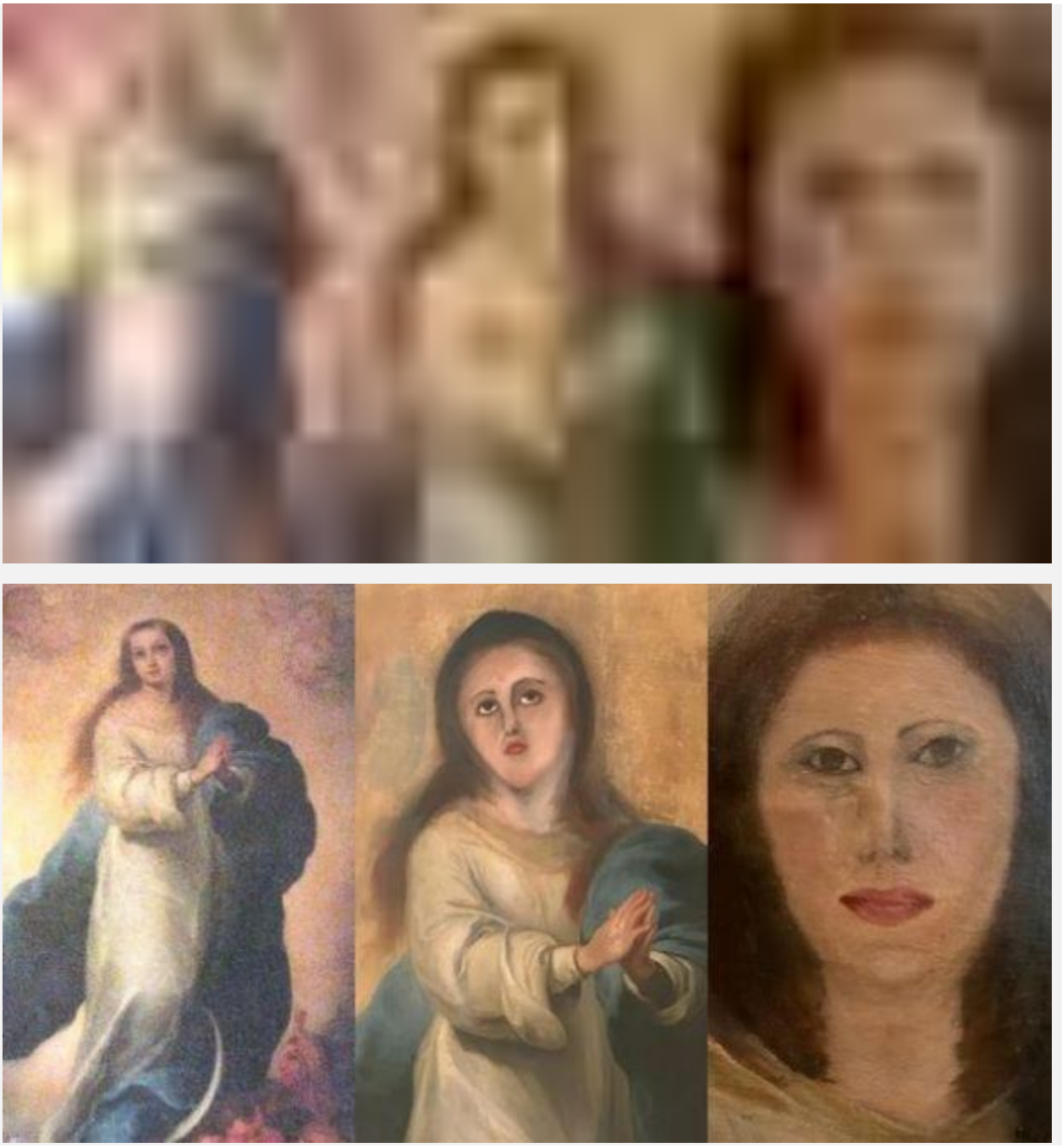
Cocoons of a parasitic wasp on a tomato hornworm © Wordpress / pattyler1

Originating from and (unfortunately) still associated with its historical and mystical connotations — namely, a person with divinatory abilities — a blockchain *oracle* is simply a piece of

software that takes information that lives outside of the blockchain and records it onto the blockchain, effectively acting as a bridge between off-chain and on-chain worlds (a bridge with varying degrees of security, I must note).

Our series *Getting APIs on the Blockchain* won't focus *too much* on oracles — in particular, because we focus our sights on the API Connectivity Problem rather than the Oracle Problem — however, we will dedicate a few articles to oracles, given they are a necessary part of the solution to the API Connectivity Problem. You need an oracle node that records data from off-chain APIs onto the blockchain.

However, importantly, we make a clear distinction between first-party and third-party oracles. **First-party oracles are operated by the API providers themselves. Third-party oracles are not operated by the owners of the data they serve**, acting as middlemen between the data source and the blockchain.



Bartolomé Esteban Murillo’s original Immaculate Conception (left) and two attempts at restoring it (right) Photo: Europa Press 2020/Wikimedia Commons

Problems with third-party oracles as middlemen

The “Oracle Problem” is an abstract and overgeneralized problem that has been somewhat more formalized as: two arbitrary systems needing to interoperate with one another — through their technical interfaces — in the most general sense imaginable. This over-generality necessitates an ever-flexible interface that can only be supported by third-party oracles.

Solutions birthed out of an underspecified problem, however, are often not optimal because the practical scope of the problem is far more constrained. In most cases — especially now, with the current state of the blockchain space— the decentralized interoperability problem is actually, in practice, the problem of receiving services from traditional API providers in a decentralized way.

I will now list several problems that arise when third-parties are used as an interface layer between APIs and the blockchain.

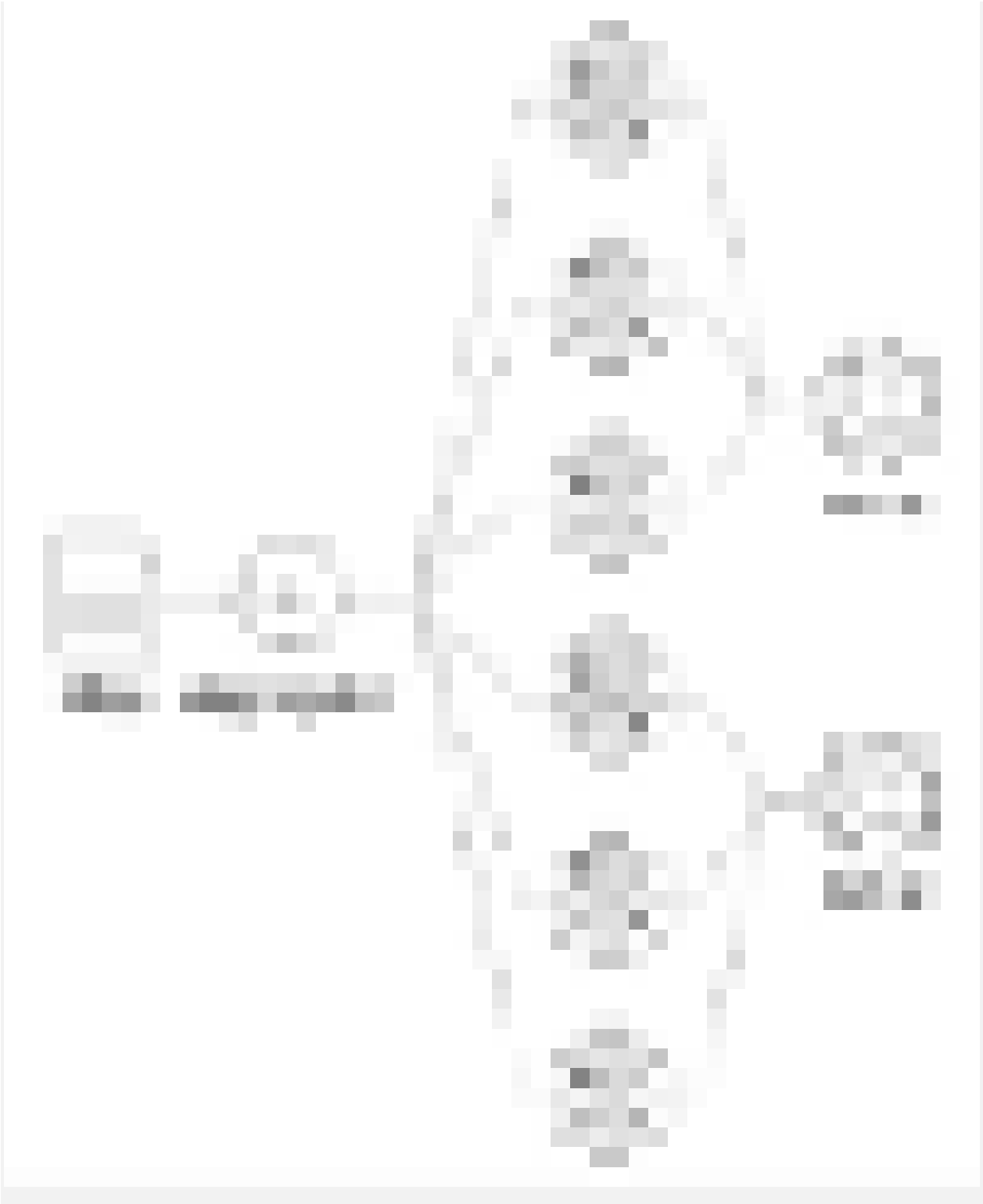
Vulnerability

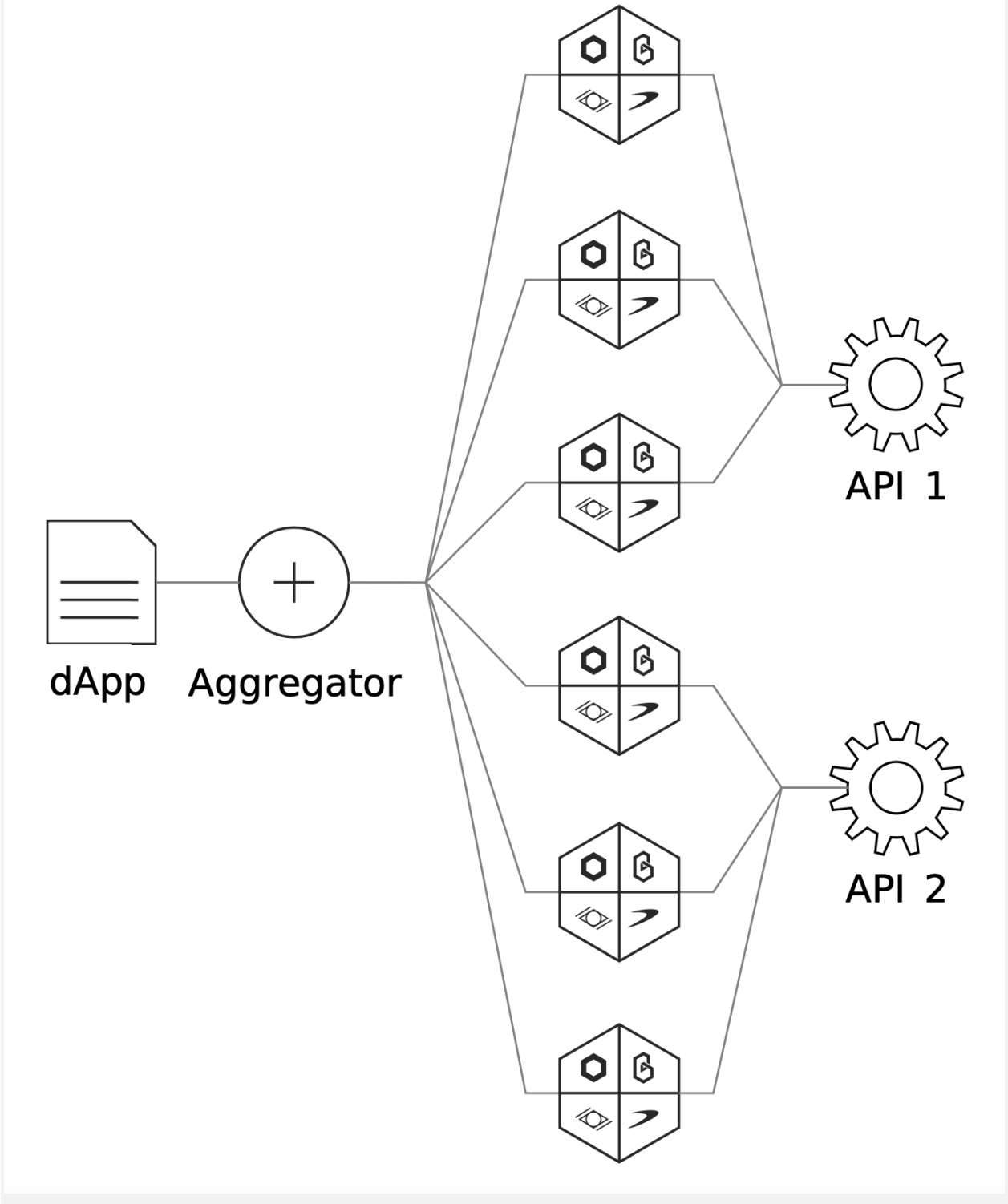
The addition of a “decentralized interface” creates an entirely new attack surface for blockchain applications. Groups of malicious third-party oracles can collude to manipulate outcomes — and, in fact, a single oracle can skew outcomes when performing “consensus” on real-valued, continuous data (a more formal treatise on such issues is forthcoming).

What's more, a single actor can fabricate multiple oracle node operator identities — as well as build a sufficient track record of honest operation — to perform the same types of attacks entirely by themselves. This is widely referred to as a Sybil attack.

Middleman tax

In order to balance and counteract the vulnerability of an entire new attack surface, the potential benefit an oracle will gain from acting honestly *must* exceed (by a lot, ideally) the amount that can be gained from misreporting *at all times*, in order to avoid any malicious behaviour by the oracle. The game theoretic argument is treated with much more detail in Section 3.2 in the API3 whitepaper. This in an additional cost (a tax, if you will) on top of the actual cost of the data that must be paid to these third-party middlemen.




dApp
Aggregator
API 1
API 2

Expensive & ineffective redundancy

Data feeds depending on third-party oracles require over redundancy at the oracle level. This is because third-party oracles are far less trustworthy than API providers (as already discussed: the latter have traditional off-chain businesses in “meatspace” and respective reputations to maintain).

Note that this decentralization does not provide any additional security at the data source level. It only serves in decreasing the additional vulnerability caused by using third-party oracles in the first place. This increases gas costs and other costs associated with operation personnel, at the very least.

Lack of transparency

As discussed in our previous post: a price feed fed by *x* oracles rarely represents *x* unique data points. The number of oracles serving a data feed does not necessarily correspond to higher quality and more robust data — it’s a bit of a game of optics. Oftentimes, the users of decentralized oracle networks overlook this fact and confuse decentralization at the oracle level with the overall decentralization of the system. This is primarily caused by a lack of transparency regarding the data sources used by the oracles, which disguises the fact that decentralization is severely bottlenecked at the data source (API) level.

Importantly, when data feeds are not transparent with where they get their data from, it becomes quite difficult for developers to accurately appraise the quality of the data feed. And, the quality of the data feed should come into question precisely because data sources are obscured. Oracles have an incentive to gather cheap and easily accessible data since little is enforcing or incentivizing them to do otherwise.

This is a particularly salient point for price feeds since some API providers use advanced filtering and aggregation methods (and price accordingly for their services). A particularly inspiring example is CoinMarketCap's volume inflation detection algorithm. Clearly, with this widely applicable use case, **data source matters**.

Leveraging off-chain, real-world reputation

Using *first-party* oracles leverages the off-chain reputation of the API provider— something we often take for granted in the measurable, deterministic, and closed world of the blockchain. That is, although API providers might not be explicitly staking (or using some other quantifiable measure) to prove their trustworthiness, their trustworthiness is evidenced by their reputation and success in the off-chain business world (to varying degrees).


MetLife

New York City skyline © Getty Images / Achim Thomae

Further, the company's revenue acts as a kind of "soft stake". Gambling with one's off-chain, real-world reputation (and thus one's revenue stream) acts as a major deterrent to malicious first-party oracle behaviour. Take for example the Coinbase oracle which offers signed price data for select markets, leveraging their trustworthiness as a business — even more so as a major business in the crypto/blockchain space that benefits from the health of the entire ecosystem and is thereby less likely to act maliciously than a no-name third party.

Further, it is often taken for granted in third-party oracle solutions that the data source is already trustworthy. Indeed, in the Chainlink network, for example, a user can select where third-party oracles gather their data from via job specifications and adapters (although proving that the oracles do in fact gather their data from those sources is another matter). Such a construction de-facto assumes data sources are trustworthy and that it's only a matter of finding trustworthy middlemen to transport that data on-chain.

In contrast, first-party oracles tap into the wellstream of trust and reputation already established and maintained in "meatspace".

Conclusion and next blog post

To conclude, I argue that *if* a first-party oracle exists for a particular data source — that is, an API provider directly servicing their data on the blockchain via an oracle — then there is no benefit to third-party oracles.

(I should make quick note that I am not stating first-party oracles are perfectly trustworthy. Of course not. Issues such as downtime and data quality will be addressed in later posts.)

Now, what about that conditional ("if")? You may be asking: what if there *isn't* a first-party oracle? How can we assume the existence (or future existence) of abundant first-party oracles? In the next blog post I'll cover present hurdles to first-party oracle integration (and why we don't see more of it) as well as how API3 tackles those exact challenges. Stay tuned!