



Covalent Network Whitepaper

V E R I F I A B L E D A T A F O R
W E B 3

Covalent provides the industry-leading **Unified API** bringing visibility to billions of Web3 data points.

Developers use Covalent to build exciting multi-chain applications like crypto wallets, NFT galleries, and investor dashboard tools utilizing data from 32+ blockchains. Covalent is trusted by a community of 27,000+ developers and 1000+ applications including 0x, Zerion, Rainbow Wallet, Bitski and many others.

Motivation

Blockchain technologies are ushering in a revolutionary change to traditional internet computing paradigms. While the internet allowed for the nearly-free and nearly-instant transfer of information, blockchains will allow for the nearly-free and nearly-instant transfer of value, paving the way for native currencies on the world wide web. Similar to how the internet disrupted traditional media companies (television networks, newspapers, and radio), blockchain technology will disrupt traditional financial institutions such as banks, loan providers, accounting firms, and so forth.

We have already seen the seeds of this disruption take root in decentralized finance (DeFi) projects. At the time of this writing, there is more than \$50 billion locked in various DeFi protocols. The rapid adoption of these nascent projects is a testament to how powerful the technology is, and still only a glimpse of what is to come.

But decentralized financial applications are just the beginning. Smart contracts enable a shift in programming paradigms, allowing the automation of almost everything from legal contracts, to the promise of decentralized autonomous organizations (DAOs), to the proliferation of non-fungible tokens (NFTs), and so much more. However, in order for these protocols to function properly for the end-user, access to timely and accurate deep blockchain data is critical.

While blockchain data is purported to be public, deep, granular, and historical blockchain data is prohibitively difficult to access, especially at the speed and detail that modern end-users expect from their user experience. The Covalent Network solves this problem while maintaining the decentralized ethos of blockchains, ensuring that deep, granular, and historical blockchain data is always accessible, ultimately providing true transparency and visibility into the future of digital assets.

Deep, Granular, and Historical Blockchain Data is Inaccessible

Despite the proliferation of digital assets on the blockchain, granular and historical blockchain data is almost impossible to access by traditional institutions. Querying blockchains directly is time-consuming and compute-intensive, while additionally refining and manipulating the data adds another layer of complexity.

In theory, a blockchain node software like Geth already has the blockchain data, with a handy JSON-RPC layer to pull out the data. However, there are four problems:

Expensive

Accessing historical EVM data is cumbersome and expensive, and requires a blockchain node like Geth to be run in a special configuration known as “full archive mode” which currently takes up hundreds of gigabytes of storage space and has other special hardware requirements.

Slow

The JSON-RPC interface is what’s known as a “point-query” interface, i.e., you can only ask for a single object (block, transaction, etc.) at a time. What you need is a way to batch export the data, which means completely rethinking the JSON-RPC layer.

Incomplete data.

The most interesting blockchain data is actually the data structures inside the contract state and not visible outside even on a comprehensive tool like Etherscan. It’s currently too hard or simply impossible to reconstruct these data structures through the JSON-RPC layer.

Too niche

Hundreds of query languages have come and gone over the years, but nothing has stood the test of time like plain old SQL. SQL has been around for 40 years and will be around for the next 40. It’s the Lindy effect in action. A niche query interface is a blocker to mainstream adoption. We need SQL for mainstream adoption.

Every DeFi portfolio tracker or wallet falls short because they run into one of the above problems. It’s relatively simple to get account balances through the JSON-RPC interface without an archive node, but anything to do with historical data is very difficult, and with the rapid adoption of DeFi protocols, the need for easy access to deep blockchain data is more critical than ever.

The Covalent Solution

For the past 3 years, the Covalent team has been indexing the entire Ethereum blockchain, including every contract state, every single transaction, every single storage slot, into the Covalent Database. Early on, we recognized the need for access to deep, granular, and historical blockchain data and have built a robust system that would supply this data accurately and reliably through a unified API. Due to the team’s enterprise background, we pride ourselves in building a product that is enterprise-grade which means reliability and robustness. This attention to quality has allowed us to sign on significant paying partners within the blockchain ecosystem, providing crucial blockchain data to organizations such as Consensys, SKALE Networks, CoinGecko, Ox, Zerion, and Zapper, just to name a few.

Multiple enterprise customers and blockchain networks are paying for our service, a clear signal from the market that this deep, granular, and historical data is highly valuable, despite free services such as Etherscan. Having achieved product-market fit, we are now planning to execute the next phase of Covalent, which is a progressive decentralization that will enable the Covalent Network to be operated by its users and incentivize them appropriately.

As the interoperability and complexity of blockchain transactions increase, current approaches such as Etherscan fall short of providing a unified and fully transparent view of anything that's more than a simple one-to-one transaction. Protocols and products that offer functionality such as custodial lending, staking, variable APRs, and more require Covalent's API to gain insight into their complex transaction activity, and as blockchain transactions invariably become more complex, Covalent is the only off-the-shelf tool that is available for this.

The Covalent Data Model

At the core of Covalent's offering is our data model: a unified, canonical analytical storage representation of all blockchain data. When blockchain data (blocks, transactions, log events, state transitions, trace events, etc.) is imported into our data model, it is normalized into a single representation, regardless of source blockchain.

This representation has a super-set of the expressive capabilities of all the internal data representations of the blockchains Covalent supports. This means that our data model can faithfully store and report on all the interesting data held on each blockchain, including data of types unique to specific blockchains.

Our data model does not manage this expressive capability by storing slightly different data representations per blockchain, as the internal storage schemas of many block-explorers' data-warehouses do. Rather, in our data model, the various blockchains' data representations are brought in line with one another, such that they can share a single polymorphic representation. (As such, each new layer-1 blockchain protocol supported in the Covalent data model involves design decisions on how to best merge new chain-internal types into our existing types—including considerations on lowest-cost schema-migration paths for existing data warehouses.)

This expressive power enables cross-chain **query reuse**: identical queries can be applied to data of very different original shapes (e.g. Proof-of-Work vs. Proof-of-Stake transactions), and the Principle of Least Surprise will apply to the results.

Despite this expressive power, our data model is also pre-tuned for the efficiency of querying. The polymorphic data representations are relationally normalized (i.e. broken into tuples of scalars, with numeric foreign-key cross-references) and are then binary-packed into compressible columnar storage formats (e.g. Parquet; ORC.)

The binary objects resulting from this normalization — which we call **block specimens** — form our canonical archival representation of a blockchain’s historical state.

Each block specimen is, on its own, enough data to represent, for specific snapshots in time (the beginning of execution of each of the range of historical blocks the specimen covers), the subset of active working state of the blockchain that was needed to execute those specific blocks. As such, a block specimen can be used in **place of a synced chain** for the purposes of running trace re-executions of the blocks represented in the specimen. A block specimen does not, however, store a complete representation of the state as of that block, and so cannot be used to answer hypothetical questions from that block (i.e. to run `eth_call`-like operations.)

Block-specimen objects are generated deterministically. Presuming two extract-and-normalize workers, each working against the synced and in-consensus (“ancient”) historical data from an arbitrary blockchain node on the same network, a bitwise-identical set of blockchain specimen objects are guaranteed to be produced. As such, despite not embedding any cryptographic proof of their provenance, block specimens can be independently verified by separate reproduction by interested parties.

These block-specimen objects can then be imported into a data warehouse, where they map cleanly to sets of tables in Domain-Key Normal Form. For scalability, our own data warehouse makes extensive use of table partitioning — to separate blocks by time, and by chain. Any other data warehouse importing full sets of these blocks would likely need to do the same. As such, our data model is also designed to ensure that each data type’s primary keys are also natural partitioning keys.

Finally, our data model embeds hints into the schema included in each block specimen, about which columns and expressions would need to be indexed in a data-warehouse representation for efficient querying. This curated choice of indices is based on Covalent’s hard-won operational experience in serving many types of generalized OLAP queries against our data model and is expected to serve well for almost any analytical use-case, while also minimizing index overhead. (However, it is also specified with partitioned tables in mind; a data warehouse with live-streaming import of block-specimen data that does use our index specifications but does not use table partitioning, may find itself without enough IOPS to keep up with index-build operations.

Progressive Decentralization of the Covalent Database

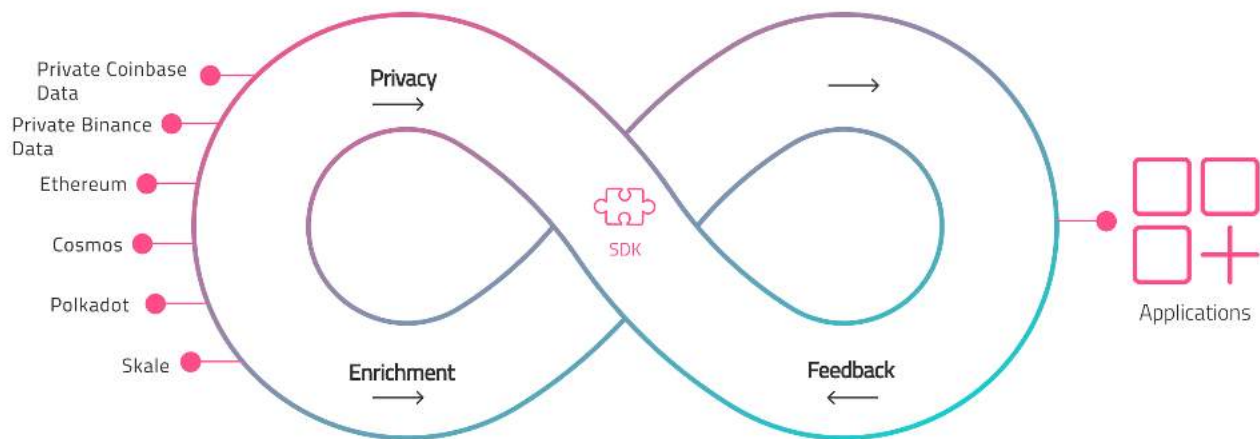
In order to ensure no single point of failure for the centralized Covalent Database, maximum uptime, and the lowest latency possible, we will begin the process of progressively decentralizing the Covalent Database to create the Covalent Network. The Covalent Network will resemble a virtual, global co-operative for blockchain data and will be governed by the Covalent Query Token (CQT) which will allow holders to propose, vote on, and make changes to the parameters of the network.

The Covalent Network, a Virtual Global Co-op for Blockchain Data

The Covalent Network is a global co-op for blockchain data beginning with the provisioning of deep, granular, and historical blockchain data accessed through a unified API. Network Operators across the globe will act as points of presence (PoP) that will be rewarded for serving query requests from data consumers for their end-users. Penalties, or slashing, occurs when there are malicious actors in the network, or when inaccurate data is provided. Once all blockchain data is indexed, Covalent will begin to allow the ability to optionally join enterprise private data with public, blockchain data. Lastly, Covalent's software developer kit (SDK) will be available for all developers to address the long-tail of use-cases with this rich data set.

The Covalent Network Architecture

Base data is refined & repackaged for multiple applications



The same base data is refined and repackaged for multiple applications. The four key components are:

- The base layer is a replica of the source blockchain data (eg: Ethereum and all other L1 chains, 100+ Cosmos Zones, 100+ Polkadot Parachains, sidechains such as Matic and SKALE, etc.)
- The privacy & enrichment layer that refines the base layer and optionally joins private enterprise data
- The DeFi SDK allows developers to engineer the enriched data for different use-cases (eg: taxation, asset balances, fund management, etc.)
- The App Store layer where developers can list their applications for marketing exposure and earn revenue

Covalent Query Token (CQT)

The native digital cryptographically-secured utility token, Covalent Query Token (“CQT”), is the network access token of the Covalent Network, which is designed to play a major role in the functioning of the ecosystem on the Covalent Network and intended to be solely used as the primary utility token on the network.

CQT is a network access token distributed to a set of Network Operators that fulfill data queries for users of the API. CQT is not, and not intended to be, a medium of exchange accepted by the public (or a section of the public) as payment for goods or services or for the discharge of debt; nor is it designed or intended to be used by any person as payment for any goods or services whatsoever that are provided outside of the Covalent Network. CQT does not in any way represent any shareholding, participation, right, title, or interest in the entities developing the Covalent Network and/or CQT, any distributor of CQT or sale platform, sale partner or exchange, their respective affiliates, or any other company, enterprise or undertaking, nor will CQT entitle token holders to any promise of fees, dividends, revenue, profits or investment returns, and are not intended to constitute securities in any relevant jurisdiction. CQT may only be utilized on the Covalent Network, and ownership of CQT carries no rights, express or implied, other than the right to use CQT as a means to enable usage of and interaction within the Covalent Network.

CQT also functions as the economic incentive to encourage users to contribute and maintain the ecosystem on the Covalent Network, thereby creating a win-win system where every participant is fairly incentivized for its efforts. CQT is an integral and indispensable part of the Covalent Network, because, without CQT, there would be no incentive for users to expend resources to participate in activities or provide services for the benefit of the entire ecosystem on the Covalent Network.

The use of a token in Covalent is to help decentralize how we operate. In this regard, the use-cases of the token are not for speculation but rather to influence user behaviour within our network. We lay some of the critical use-cases we see for the token in the list below.

Infrastructure Currency

CQT will be used as the platform currency that powers interactions between users on the Covalent Network. Each time a query is responded to, the Network Operator delivering the data will be **compensated in CQT tokens**. Those making requests for data (as customers) will however never interact with the CQT economy. Instead, their payments will be made in standard dollar figures and charged in stablecoins. This makes it possible for standard agreements and pre-set pricing to be made when a company chooses to use the Covalent Network for running its operations. It also protects them from speculating on the underlying currency.

Rewards for Indexing, Hosting, and Serving Query Requests

Since a community of users can deliver data better than a single firm could, our philosophy is to expand the number of individuals serving data through the network. In many ways, this would be similar to vendors on Amazon. As the amount and types of data on the network increase with more indexers incentivized to provide data, Covalent Network's footing in the industry will be better established. Users will be compensated for providing data-sets that are niche and hard to find. The compensation will be a split of the transaction fee from the buyer and our internal allocations kept aside for early adopters.

Multipliers for Meeting Service Level Agreements

Users with the highest uptick and lowest latency for data on the network may be given a multiplier of their typical CQT rewards in tokens. This is in order to incentivize the use of better hardware and infrastructure to cater to data queries. Naturally, we will see a power law in terms of the leading vendors being able to afford increasingly sophisticated hardware and thereby increasing their share of rewards in the network. Ultimately the benefits of this will pass on to the end-user who will have faster data queries for lower costs when compared to a traditional alternative. In the same vein, the Network may punish curators and indexers that fail to maintain high uptime. This will be done through a staking model where the vendor is expected to lock up a certain amount of CQT tokens to be a validator on the network. Validators that fail to maintain their uptime will see their CQT tokens reduced.

Delegation By the Average User

In order to make it possible for users to participate in the Covalent Network without taking all the risk of acquiring the necessary tokens on their own, we will make it possible for individuals with smaller token holdings to delegate their tokens to Network Operators. These validators have the upside of being able to partake in the data economy without necessarily acquiring the CQT tokens on their own. The token-based incentives they receive would in turn be split with users that delegate on these networks.

Governance

The CQT token will be gradually used to remove the need for the Covalent team to be a key player for the management, storage, and relaying of data on the network, by allowing holders to vote on network features. Instead, we will have an ecosystem of multiple data- providers who are also able to make decisions about fee models, nature of the interaction between buyers and sellers, and the variety of data sold on the network without a single party deciding on it (for the avoidance of doubt, the right to vote is restricted solely to voting on features of the Covalent Network; the right to vote does not entitle CQT holders to vote on the operation and management of any entity developing the Network, its affiliates, or their assets, and does not constitute an equity interest in any of these entities).

This is part of the reason why we give tokens to vendors on the network that provide data during the initial phases of the project. By incentivizing those that are net beneficial to the network, we are curating a community of the world's best data-vendors to manage the marketplace.

Data Analytics App Store

Covalent will be creating a software developer kit (SDK) to allow developers to create applications that address the long-tail of use-cases with the Covalent network's database. CQT will be used as rewards in order to incentivize the creation of these apps.

CQT will be used in the context of creating an economy of data vendors and buyers that are each incentivized to engage with the network in ways that are net beneficial. The ownership of the token itself will also enable them to improve the system overall through partaking in governance

Staking and Network Operators

The Covalent decentralized network would allow accessing blockchain data using a simple API that allows for quick integration in new or existing projects. To have queries processed, users would interact with one of the "query" nodes of the decentralized Covalent Network, providing payment in stablecoins or CQT tokens, which are the internal network currency. CQT tokens would live on Ethereum and would also be available on Moonbeam after successful integration.

- The core on-chain logic would have to happen on Moonbeam for efficiency reasons and for maintaining tight compatibility with the Polkadot ecosystem.
- Users would lock up stablecoins / CQTs on Ethereum for their balances to be reflected on the Moonbeam chain.
- In order to become a Network Operator in our network, CQT tokens would have to be staked. Network Operators are doing useful work (e.g., parsing and the processing of new ETH blocks, responding to user queries) for the Covalent Network, and staking ensures correct behavior — if Network Operator are ever malicious, they can be slashed.
- By staking CQT tokens, Network Operator earn more CQT tokens by providing utility to the network.
- CQT token holders can delegate their CQT stake to Network Operator and earn a yield.

Covalent Network Operators

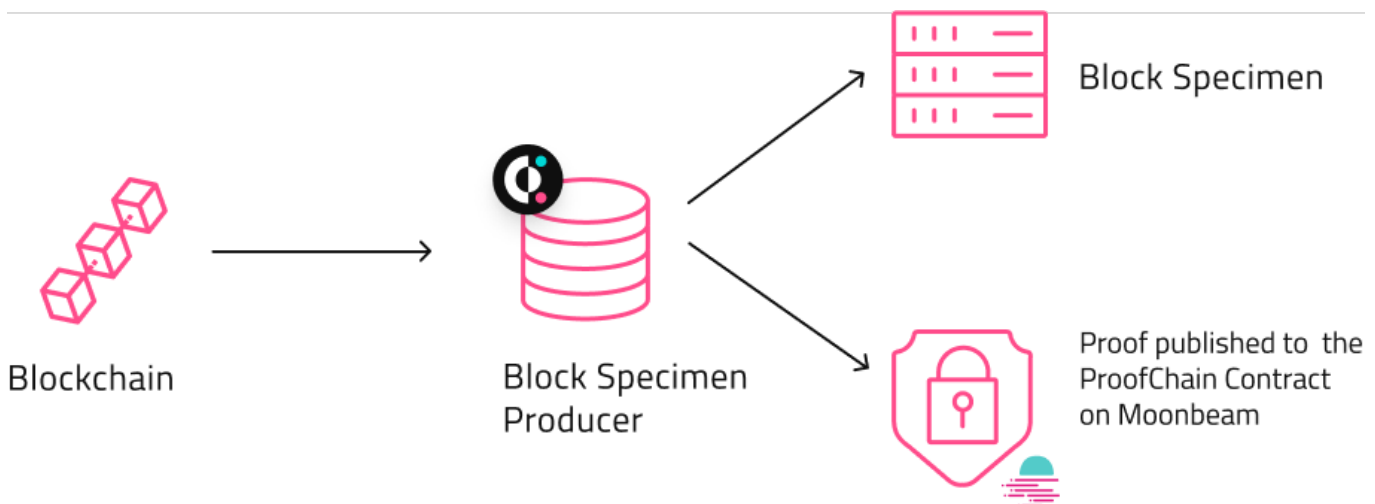
Covalent's decentralized network is expected to consist of 6 roles. Any Network Operator may function in one or all of these roles. The roles have very different operational requirements, however, so it is expected that node-operators will self-select into the subset of roles that best suit their capabilities.

The roles are :

- Proof-of-Stake Validator / Network Block Producer (as per any other PoS network)
- Block-Specimen Producer
- Refiner
- Storage
- Query Operator
- Directory Service Node

Block-Specimen Producers

Block-specimen producers race to consume blocks from external blockchains (however they're able to — likely by the block-producer running internal nodes of these external blockchains for their own use). Producer nodes run an (in future open-sourced) Covalent extract-and-normalize worker against these external chains, to produce block-specimen objects as outputs. They then publish the produced block-specimen objects to a storage node (which can, for efficiency, also be run by the same validator on the same hardware); and then publish a block-specimen production-proof transaction to the ProofChain contract. If valid (i.e. if this is the first time a specimen of these blocks appears on-chain), this tx be rewarded intrinsically with CQT by the consensus algorithm, similar to how blocks are intrinsically rewarded with mining rewards on a PoW chain.



The intrinsic reward is locked for a given period, because these block-specimen production-proofs have a period during which they are open to independent auditing. If another block-specimen producer node can prove that a peer submitted an incorrect block-specimen, they can submit this malfeasance proof as a transaction to the contract, and so slash that peer, destroying the original production-proof tx's locked reward along with some of the block-specimen producer's staked CQT.

Block-specimen producers require direct write access to the contract, and are therefore (along with indexers) already incentivized to be on-chain validators, so that they can validate their own block-production-proof transactions "for free." As such, we think it is not too painful to require that block-specimen producers always be validators (i.e. to stake CQT) in order to be allowed to submit specimen production-proof transactions.

This block-specimen producer role, then, serves as the backbone of the Covalent chain: we have at least as many Network Operators as we have block-specimen producers.

Refiners

Refiner nodes behave similarly to block-specimen producer nodes, in that they race to produce an output, publish it to a storage node, and then publish the content-hash of that output to the chain, for an intrinsic CQT reward.

The difference between the two node types is that, while block-producer nodes must have access to fast nodes of external blockchains, Refiner nodes must merely have access to the storage nodes.

The job of a Refiner is to fetch the block-specimens already produced by a Block Specimen Producer; to run a tracing re-execution of these block-specimens using the latest (in future open-sourced) implementation of Covalent's stateless-tracer worker; and to output the data artifact generated by this process — a trace specimen — back into the storage network.

A trace specimen, unlike a block specimen, does not consist of a single binary object, but rather exists as multiple representations: a single trace-event stream object; a set of abstract contract-state specimen objects; and a set of abstract contract-state metadata manifest files. The abstract contract-state specimen objects + metadata files form something like a Git repository, with each new contract-state object+manifest-file set being a new "commit" to that repository. The "commit hash" of this new commit, along with the content-hash of the trace-event stream object, are both published to the ProofChain contract. All relevant data is published to the storage network.

The requirements of being a Refiner are easier to satisfy than the requirements of being a block producer. This is why the two roles are decoupled: we expect some parties to wish to only run the infrastructure required for refining, not for block-production.

Refiners are, like block-specimen producers, incentivized to collude with a local PoS validator; as such, we may require Refiners to also serve as validators. If we don't, we will at least require them to stake CQT through delegation. Trace specimens, like block specimens, can be proven invalid, triggering slashing, and so we need Refiner nodes to offer a certain CQT balance to be slashed if necessary.

Refiners are also highly incentivized to run local storage nodes themselves, as the high number of individual files/objects they need to register/publish into storage would be high-cost if they relied on external storage-network peers. As such, we may require Refiners to also serve as storage nodes for at least their own outputs. As storage nodes need not store anything other than what their owner publishes to them, this should not pose any undue burden on indexers.

Storage

Rather than every node in the Covalent network being expected to hold its own output artifacts, this responsibility is delegated to specific network-peers who opt into it.

These storage nodes are expected to provide an interface similar to IPFS (and may indeed be implemented purely as a gateway + a private-network IPFS node), but with the important difference that these nodes can charge per request, similar to paid IaaS object-storage services. Different nodes can publish (to directory nodes) their offered content-hashes, and a cost (in CQT) for fetching that object from them.

Other nodes in the network, if they decide they're happy with the storage node's prices, will submit signed fetch requests directly (i.e. not on-chain) to storage nodes. These nodes then 1. respond to the client node with the requested object; and 2. wrap the signed request in a proof of their own response, hash that composite object, and hold onto that hash. At any point, they may take a batch of these hashes and package them into a storage-response proof transaction, and submit that tx to the ProofChain.

Unlike with block-producer or Refiner nodes, there is no race to submit storage-response proofs, as only a storage-node that received+responded to a given request has the information required to build a proof for that response. As such, storage nodes are not incentivized to also be validators, and we will not require storage nodes to be validators. We expect storage nodes to function as light-clients or RPC clients within the Covalent network.

Storage-response proofs can, however, also be slashed (by the original client of the request, by submitting a malfeasance-proof containing the original responder's signed response to them, when this response corresponds to the nonce of a hash in the storage node's submitted storage-response batch proof, but where the submitted hash itself does not match the hash of the evidence.) As such, storage nodes must stake CQT, likely as delegators. It is expected that often a storage node would be colocated with a Refiner node, so storage nodes will likely delegate their stake to the validator of this colocated Refiner node.

Query Operators

Processing user queries works very similarly to processing storage requests. In both cases, a client submits a signed request to the node; the node sends a signed response; and then asynchronously submits a batch of hashed signed-response proofs to the ProofChain, being rewarded for this by the transfer of CQT from the signers' accounts to the node's account, in a locked state.

The difference between storage nodes and query nodes comes down to the payloads of the request and response. While storage requests reference specific existing objects in the storage network, query requests are arbitrary SQL or Primer queries. While storage-response payloads come directly from existing objects in the storage network, query-response payloads are computed by the node itself, from a local data warehouse the node-owner maintains.

Query Operators may populate their associated data warehouse with whatever they wish. However, the expectation of users of the network is that the Query Operator will use (in future open-sourced).

Covalent ETL worker software to preload chosen data from the storage network into their data warehouse, with the storage-node request fees considered an operational expense.

This ETL worker software will load the data into a specific schema (shape) and will add specific indices to this data, according to on-chain-governance-directed steering of the design and development of the ETL software. Query Operators are thus incentivized to participate in on-chain governance, to direct — and hopefully optimize the engineering of — the proposals that affect their own nodes' storage. By extension, Query Operators are incentivized to aid in the production of this ETL worker software, and possibly also the block-production worker and Refiner worker software, as the upstream design choices in these codebases will affect the constraints on the design-space of their own warehoused data.

While both storage nodes and Query Operators can be theoretically queried either by other nodes in the network or by external clients acting directly through HTTP requests, it is expected that storage nodes will mostly handle internal traffic (write traffic from producer/Refiner nodes, and read traffic from Query Operators), while Query Operators will mostly handle external traffic (end-user analytical queries, similar to those submitted to the Unified Covalent API today.)

Directory Service Nodes

In order for clients external to the network to find network nodes with specific resources/capabilities (i.e. storage-network nodes holding specific objects; Query Operators who have indexed specific contracts) or to find network nodes within the client's "price range", external clients need access to directory servers to which the storage+query nodes can publish their capabilities.

Directory nodes will likely act together to form a Kademlia-like DHT, mostly containing small pointers (content-hashes whose values point to the DHT slot of the node), together with node-advertisements that act as both manifests of available data, and price lists/formulas for fetching that data/type of data from the node. Clients can then discover nodes advertising desired resources through this DHT.

Because directory nodes must answer numerous very tiny queries with extreme efficiency/low overhead, there is no practical mechanism by which to implement a reward incentive for running a directory node. As such, it is expected that directory nodes will be run as an ancillary service provided by nodes acting in one of the other roles. We don't expect directory services will be under-resourced in the network, even if made entirely optional/not tied to any other role; after all, many organizations happily run free public DNS servers, with roughly the same economics.

Other Network considerations

All the described logic with validator entities/nodes getting rewarded for work rendered, would not happen on Ethereum and would happen on Moonbeam.

Once a Network Operator or a user wants to withdraw their earned / remaining CQT tokens to the Ethereum chain, they will be able to do so using Nomad.

Given that our validator nodes would have to post on chain proofs for the work rendered, such that they can get paid, they will have to pay Moonbeam "gas" (GLMR).

Phased Plan for Network Decentralization

On initial public launch, the Covalent Network will have full feature-parity with the final network, but will not be fully decentralized in the manner of the final network.

Decentralization of the network will be gradual, occurring over time, with the node roles described above gradually transitioning from being performed by the centralized backend nodes of the Covalent API offering, to being performed by arbitrary third-party network node operators.

Specifically, at network launch, only the roles of network block validator and query-request responder will be fully decentralized.

A piece of blockchain-node software will be offered for Network Operators to run—initially to observe blocks being produced by our centralized block-producer; later to participate in block-production/refining themselves.

A virtual appliance will be offered to interested parties to run query nodes of their own, to serve their own requests, or the requests of clients.

Block production, refining, and directory services will be provided at launch by centralized systems run by Covalent engineers — the same systems currently powering Covalent’s commercial SaaS solution.

From launch, this partially decentralized form of the network will still have full feature-parity with the final network architecture. Facilities available to external clients via requests to the network (querying, staking, governance, slashing) will be available from the initial launch in some semantically-equivalent centralized or partially-decentralized form. The state of these centralized systems will be ported into the final network, as equivalent distributed facilities for these operations become available.

The main limitation preventing immediate full decentralization is the software architecture of Covalent’s current commercial offering. This software (including blockchain-node patches, tracing components, ETL worker logic, API-layer logic, etc.) must be refactored and enhanced to enable it to “slot into” the appropriate places in the final network architecture.

As such, it is expected that network decentralization will increase in phases, wherein each phase, an individual node-role will become enabled for decentralization as the appropriate software components are extracted from Covalent’s internal software stack and polished into reusable network infrastructure components that can be run by node-operators. The sequencing of these phases will be directed by community governance, enabling first the node-roles that our community of node-operators has the most interest in performing.

Conclusion

As the adoption of blockchain technologies accelerates, the need for a data infrastructure layer that reliably offers deep, granular, and historical blockchain data will become ever more important. We have already seen the value of middleware in the traditional markets, with Segment’s acquisition by Twilio for \$3.2 billion, as well as the \$6 billion acquisition of Mulesoft by Salesforce. The recent adoption of cryptocurrencies by PayPal is a signal that every single company is now discussing a cryptocurrency strategy for their firm. It is our vision that every single FinTech company will be using Covalent to access blockchain data. This whitepaper serves as a general outline of the network architecture and technology behind the Covalent Network and technology decisions may change in the future.

About Covalent

Covalent provides the industry-leading Unified API bringing visibility to billions of Web3 data points. Developers use Covalent to build exciting multi-chain applications like crypto wallets, NFT galleries, and investor dashboard tools utilizing data from 32+ blockchains. Covalent is trusted by a community of 27,000+ developers and powers data for 1000+ applications including Ox, Zerion, Rainbow Wallet, Rotki, Bitski and many others.



One unified API. One billion possibilities.



@Covalent_HQ



covalenthq.com/telegram



covalenthq.com/discord



covalenthq.com