Covalent

Covalent Network Whitepaper

Ethereum's Wayback Machine

Last updated: 16th June 2023

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 and granular blockchain data is prohibitively difficult to access across every blockchain network, especially at the speed and detail modern end-users expect from their user experience. Furthermore, the sustained accessibility and availability of historical data on blockchain networks are becoming an increasing concern, especially for Ethereum.

The Covalent Network solves these problems 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 incredibly difficult to access by traditional institutions and applications. This is the result of two critical problems: 1) querying blockchain data, and 2) long-term data availability.

Querying Blockchain Data

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 (go-ethereum) 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". This configuration takes up hundreds of gigabytes of storage space, makes the node redundant until it has fully synced 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. This is extremely time-consuming and does not scale. 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 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 last 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.

Without the much-needed data infrastructure tooling, every application requiring blockchain data will fall short because of 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 (i.e. past transactions) is very difficult.

The Long-term Data Availability Problem

Covalent defines long-term data availability as the sustained accessibility and availability of historical data on a blockchain network over extended periods of time. While Ethereum's historical data has always been available directly through the client, this is changing.

To reduce the requirements of running a node on Ethereum, the network is introducing the history and state expiry (EIP-4444). This feature moves the responsibility of providing historical data outside of the Ethereum core protocol. Currently, clients fulfill requests for historical data by retrieving it from their local databases. However, with the implementation of history expiry, this approach becomes unreliable if the requested data has been pruned or removed from the client's database.

While Ethereum has made commendable progress in scaling by adopting rollups and modular designs, the long-term availability of historical data remains a crucial concern. As the network evolves, ensuring continued access to blockchain data over extended periods becomes imperative for various applications, including taxation, auditing, AI models, and regulatory compliance to name a few. Balancing the need for scalability with the preservation of historical data presents a significant challenge that the Ethereum community must address to maintain the network's utility and usability.

The Covalent Solution

For the past 5 years, the Covalent team has indexed over 100 blockchains, capturing every contract state, every single transaction, and every single storage slot, for each blockchain into the Covalent Database.

Early on, Covalent recognized the need for access to deep, granular, and historical blockchain data and built a robust system that would supply this data accurately and reliably through the Covalent Unified API. Due to the team's enterprise background, Covalent prides itself in building a product that is enterprise-grade which means reliable and robust. 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, Rainbow, Raleon, Rotki, Utopia, and Tally, 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 Unified API to gain insight into their complex transaction activity. 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 blockchain's 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. In other words, users can query and retrieve data from different blockchains in a uniform and predictable manner, without being surprised by unexpected variations in the query results.

The Covalent 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). These questions can be asked only on processed by-products of block specimens.

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 bytewise-identical set of blockchain specimen objects are guaranteed to be produced. As such, despite not embedding any cryptographic proof of their provenance apart from the sha256 sum of their contents committed to a decentralized secure public blockchain network, block specimens can be independently verified by separate reproduction by interested parties generating the same sha256 sum.

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.)

Introducing The Covalent Network

In order to avoid single points of failure, ensure optimal reliability and uptime, community participation and contributions, the Covalent Database and its corresponding infrastructure components are being progressively decentralized and open-sourced, forming the Covalent Network.

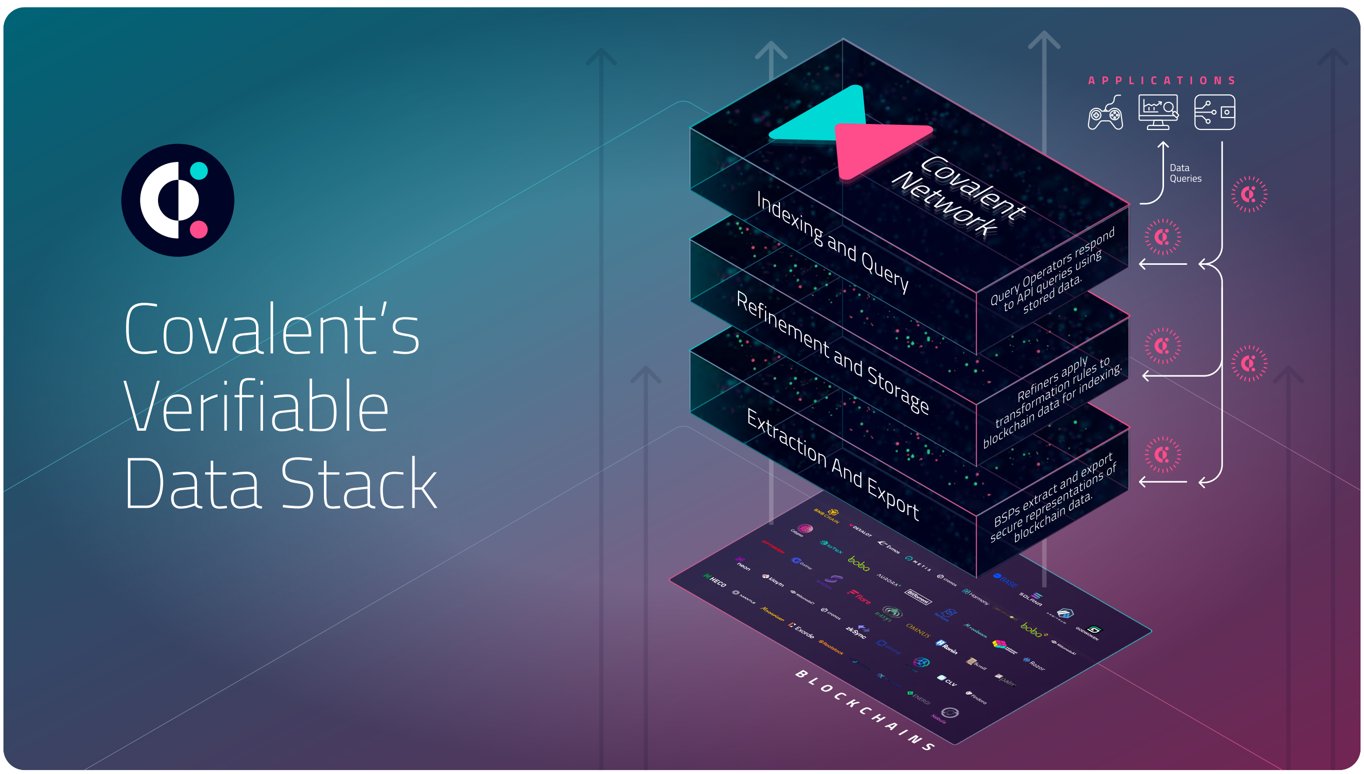
The Covalent Network is a decentralized data network that operates under the governance of the Covalent Query Token (CQT). Its primary function is to provide comprehensive and historical blockchain data through a Unified API. Individuals will also have the ability to optionally join enterprise private data with public, blockchain data.

Simply put, the Covalent Network works by capturing and indexing blockchain data and storing it across multiple points on the network. At the core of the Covalent Network is data verifiability which is achieved by employing cryptographic proofs. For every piece of work completed on the Covalent Network, a respective proof is created ensuring that Network Operators behave honestly and provide data that is accurate.

The Covalent Network is one of the first data middleware protocols to employ a proof-based system to ensure data verifiability. This is not only a novel feature of the Covalent Network but an essential mechanism in rebuilding the new foundations of the web.

Finally, the Covalent Network consists of Network Operators (Validators) who perform roles such as extracting blockchain data, broadcasting it across the network, and serving query requests to data consumers via the Covalent API. Penalties, or slashing, are imposed for malicious activity or inaccurate data provided by Network Operators, which can be identified by comparing and evaluating proofs submitted by them.

Covalent's Verifiable Data Stack



The Covalent Network consists of three key processes:

Extraction and Export › Refinement › Indexing and Query

Extraction and Export

The Covalent Network is built upon a foundation of historically complete and accurate replicas of the source blockchain data, including Ethereum, various Layer 2 chains, Cosmos Zones, Subnets, and more.

Pivotal to this are Block Specimen Producers (BSPs) (see Network Operators), who extract and export block specimens, a 1-to-1 secure representation of a block and its constituent elements. Together these form a canonical representation of a blockchain's full historical state. This allows the Covalent Network and data consumers to efficiently retrieve historical blockchain data with confidence in its accuracy.

Refinement

Taking the base data (block specimens at the time of writing), the refinement and storage layer performs validated data transformation according to blueprints broadcasted across the network.

At the center of this are Refiners (See Network Operators) who host a data processing framework. The Refiner locates a source to apply a transformational rule to and outputs an object generated from applying such a rule. The source as well as the output stored are available through a decentralized storage service such as IPFS. A transformation-proof transaction is emitted confirming that it has done this work along with the output (IPFS) access URL.

Furthermore, Refiners have the capability to perform arbitrary transformations over any binary file concurrently with other transformations. This enables simultaneous data indexing, with any consumer of the data slicing and dicing the data as they see fit. This significantly enhances the Covalent Network as the network benefits from the ability to run parallel re-executions on blocks. While other indexers can perform re-execution on blocks, they are doing so in a centralized system and lack the same concurrency achieved as in a distributed model.

Indexing and Query

Up to this point, all data, either from the base layer or the refinement stage, has been made available via distributed storage and announced on-chain.

Query Operators, the set of Network Operators responsible for responding to API queries over the network, observe these events. Running a local data warehouse, query operators pull data objects from storage that most interest them (based on API user demand), and apply any additional internal database indexing on top. With historical and real-time data available to them, query operators can return API queries so long as they have the appropriate data. For doing so successfully, they will be compensated in CQT.

Furthermore, to fetch data from the network, query operators will have to pay, in CQT, into a "network fund". How much they have to pay is proportionate to the amount of data they've fetched from the network. This fund pays out in turn to the production operators such as Block Specimen Producers and Refiners, as network rewards.

Decentralized Storage

Among each of the three key processes, data is either retrieved or fetched across the network. Network Operators that produce data such as BSPs and refined block objects will push outputs to a decentralized storage instance. They can run this storage instance locally and make it available publicly or make use of external storage options and pinners. A Network Operator who wishes to pull from a public storage repository can observe proofs and fetch them through IPFS using the access URL appended to proofs. Therefore, the storage of various activities is delegated to Network Operators rather than the network itself. This allows for loose coupling between the nodes as long as they upload their work to the decentralized storage layer and others can pull from the same for theirs.

Auditing Proofs

As mentioned, when a Block Specimen is created, a production proof is created and published to the Covalent proof contract. As there will be multiple BSPs, a number of scenarios can arise:

1. Every proof matches and thus every BSP has produced the same Block Specimen.
2. Some proofs mismatch but there is a majority that match.
3. There are no matching proofs.

To determine what scenario has transpired and who should be rewarded per epoch, a check is done for all of the on-chain submissions for a given block specimen (initially by Covalent). Critical to this check is the role of the auditor which is to examine an epoch of proofs, be it historic or present.

Rewards are not calculated or generated until the auditors approve or falsify a given quorum attained by the independent distinct set of operators. To communicate this, the auditor(s) messages the Covalent proof contract calling either a reward or a revoke.

Auditors are selected at random from a base pool of operators in which they play only the role of an auditor for that epoch. For every audit that passes they're awarded the staking block rewards of the operator for which they successfully provide the malfeasance proof. They resubmit the proof for every block and at the end of every epoch, the operators that are found to have invalid proofs are slashed accordingly.

Until this function is developed, Covalent acts as the source of foundational block specimen truth given that Covalent will be producing valid Block Specimens and publishing their respective proofs along with other operators. Thus, proofs that BSPs publish can be compared against Covalent's own. This also mitigates the risk of collusion occurring between Network Operators.

Design Considerations

The design considerations for the technical architecture of the Covalent Network were carefully crafted to address several key factors. Firstly, the Ethereum roadmap, and in particular, the introduction of history expiry. Once this phase is implemented, clients will no longer store historical data older than a year. Hence, alternatives will be needed to access Ethereum's full historical state, especially those that are decentralized to mitigate censorship risks. To facilitate this, the Covalent Network has implemented a robust Extraction and Export layer, where Block Specimen Producers extract and export independent block specimens, creating a secure and accurate representation of a blockchain's historical state off-chain. This ensures that the Covalent Network can provide historically complete and accurate data for Ethereum essentially acting as a Wayback Machine for the network.

Secondly, the increasing trend toward modular blockchain stacks necessitated a flexible and adaptable design. Each of the Covalent Network processes can be performed individually, with Network Operators self-selecting which they would like to perform. Furthermore, each process can, in theory, be adopted for a blockchains modular stack.

For instance, with Block Specimens, a modular blockchain could utilize this technology by adopting the patch for its clients. This way, any blockchain client can bulk export raw blockchain data effortlessly thereby absolving the need to have a data availability layer in-house. A standardized and historically complete representation of the blockchain can then be built off-chain, with no need to store it on-chain.

Another critical consideration was ensuring verifiable data in a permissionless and decentralized environment. Trust is essential in such an environment, and to ensure network participants, whether they be developers or enterprises, can have confidence in the data retrieved, cryptography and consensus mechanisms are utilized. This enables data consumers to have confidence in the data retrieved from the Covalent Network, even in a permissionless and decentralized environment.

Finally, the Covalent Network's design focuses on providing a unified data model for all blockchains, regardless of their source. By normalizing all data into a single polymorphic representation, the Covalent Network can support multiple blockchains seamlessly. This allows data consumers to retrieve and analyze data from different blockchains using a consistent data model, simplifying data integration and analysis across diverse blockchain ecosystems.

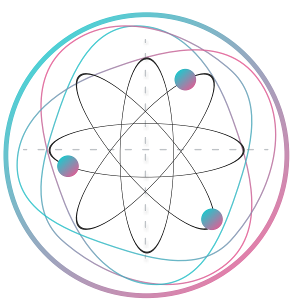
Covalent Network Operators

Covalent's decentralized network is expected to consist of 4 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 Network Operators will self-select into the subset of roles that best suit their capabilities.

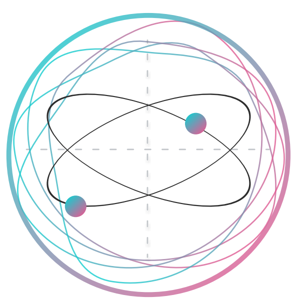
The roles are

- Block-Specimen Producer
- Refiner
- Query Operator
- Delegator

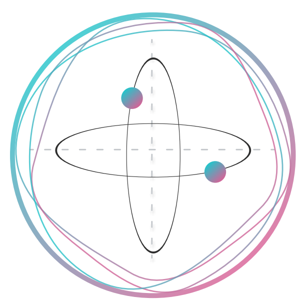
Before outlining the details of each Network Operator, it is worth highlighting that given their crucial roles, each are required to stake a certain amount of CQT in order to become operational on the Covalent Network. This process, similar to how Proof-of-Stake works on Ethereum, provides an incentive to act honestly, as they risk losing their staked CQT via slashing if they act maliciously or fail to perform their duties properly. By requiring Network Operators to stake CQT, the Covalent network is able to maintain a high level of security and integrity while also encouraging participation and decentralization from the community.



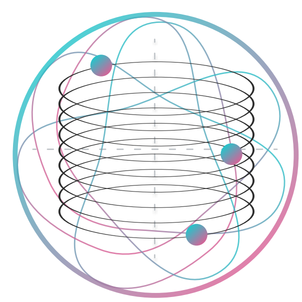
Block Specimen Producer



Refiner



Query Operator

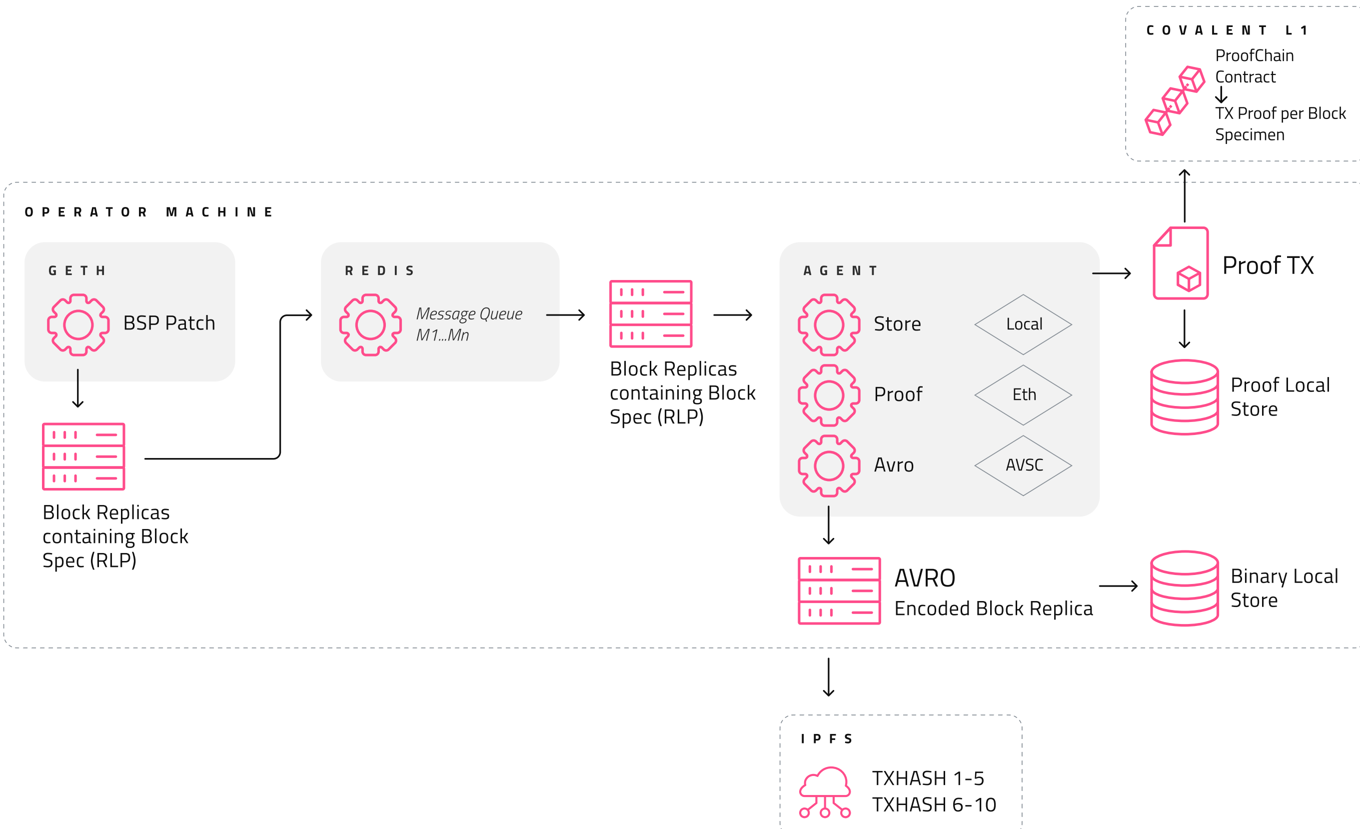


Delegator

Block-Specimen Producers

BSPs consume blocks from external public blockchains using the BSP specification that can be implemented on existing blockchain clients including Geth and Erigon. While running this implementation against external chains, BSPs produce block specimens as outputs all while performing the additional responsibilities of the node (i.e. consensus and/or execution).

They then publish the produced block specimens to a storage layer (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, in which they require write access (in the future to the Covalent Network L1). If valid (i.e. if this is the first time a specimen of these blocks appears on-chain) this proof transaction is rewarded intrinsically with CQT by the consensus algorithm along with other block specimens that are determined by the majority submission hashes, similar to how blocks are intrinsically rewarded with mining rewards on a PoW chain.



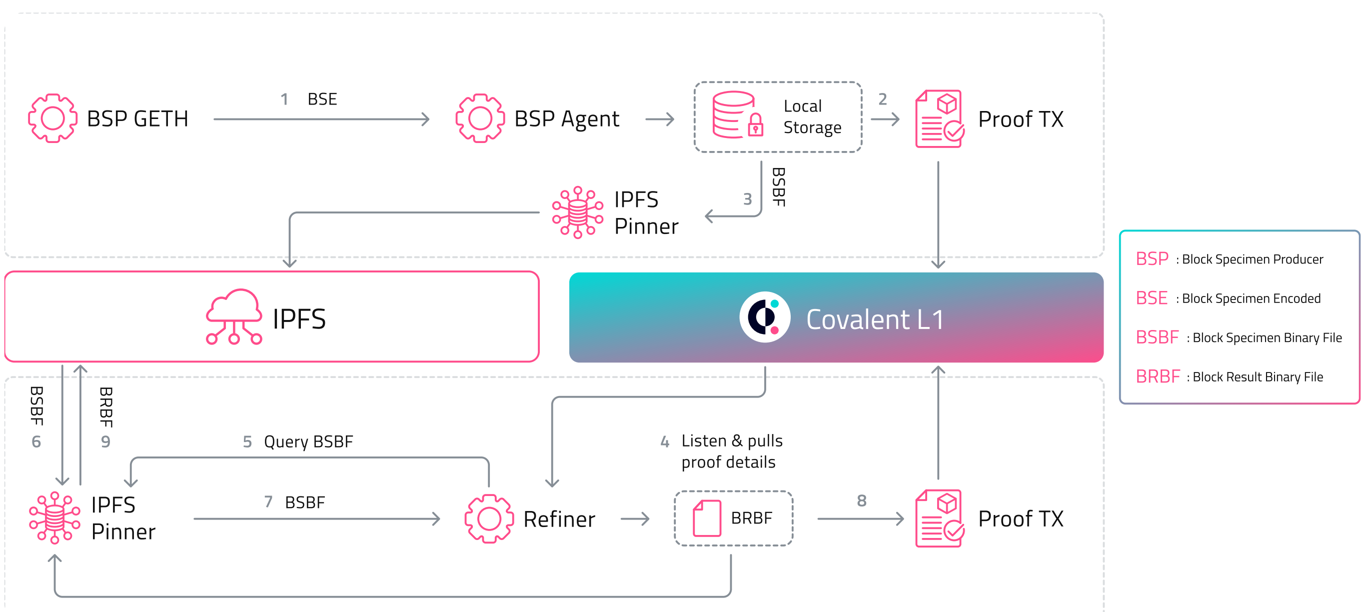
The intrinsic reward is locked for a given period because these block-specimen production proofs have a period during which they are open to independent auditing. If another block-specimen producer node can prove that a peer submitted an incorrect block-specimen, they can submit this malfeasance proof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

BSPs serve as the backbone of the Covalent Network. There will always be as many Network Operators as there are BSPs.

Refiners

As it stands, the block specimens capture a state snapshot. That is, all of the state read and transaction information. However, it doesn't capture the side effects of executing the block, the information you would get from a trace. In theory, the block specimen could contain this information. However, the block specimen should only contain the minimum information required to re-execute a block since this makes the network more efficient.

Refiners are operators that behave similarly to BSPs, in that they produce an output, publish it to the storage layer, and then publish the content-hash of that output to the ProofChain contract or the Covalent L1 chain, for an intrinsic CQT reward. The difference between the two node types is that, while BSPs must have access to fast nodes of external blockchains, Refiners must merely have access to the storage layer and proof layer.



The job of a Refiner is to fetch the block specimens already produced by a BSP; to run a tracing re-execution of these block specimens using the latest open-source implementation of Covalent's stateless-tracer worker; and to output the data artifact generated by this process back into the storage layer.

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 or the Covalent L1. All relevant data files and binaries produced from this activity are published to the storage layer and the respective proofs to the proofing layer.

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

Trace specimens and Block Results, like block specimens, can be proven invalid, triggering slashing, and so Refiner nodes offer a certain CQT balance to be slashed if necessary.

Query Operators

The primary responsibility of Query Operators is to respond to user API queries requested over the Covalent Network above all else. To do so, they will first have to build a local data warehouse(s) that is populated with data made available over the Covalent Network. Doing so involves observing the on-chain announcements made by Network Operators downstream (BSPs, Refiners) which will hold access URLs to their stored data. Query Operators will then pull data objects from storage that most interest them (based on API user demand), and apply any additional internal database indexing on top.

Given that Query Operators will load the data into a specific schema (shape) and add specific indices to this data, Query Operators are incentivized to participate in on-chain governance, to direct — and hopefully optimize the engineering of — the design and development of the Refiner software. By extension, Query Operators are also incentivized to aid in the design of the BSP as the upstream design choices in these codebases will affect the constraints on the design space of their own warehoused data.

To process user API queries, a client will submit 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 query-response payloads are computed by the Query Operator itself, from the local data warehouse the operator maintains.

While Query Operators can be theoretically queried either by other nodes in the network or by external clients acting directly through HTTP requests, Query Operators will mostly handle external traffic (end-user analytical queries, similar to those submitted to the Unified Covalent API today.)

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.

Delegators

In order to make it possible for users to participate in the Covalent Network without taking all the risk of acquiring the necessary tokens to meet the minimum stake per role and maintaining the operations of running as a Network Operator, individuals with smaller token holdings can delegate their tokens to Network Operators. Delegators 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.

Other Network Considerations

All the described logic with Network Operators getting rewarded for work rendered does not happen on Ethereum and will be settled on the Covalent Network Layer 1. At the time of writing, proofs are settled on Moonbeam until the Covalent Network Layer 1 is developed.

Given that Network Operators have to post on chain proofs for the work rendered, such that they can get paid, they have to pay Moonbeam "gas" (GLMR).

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.

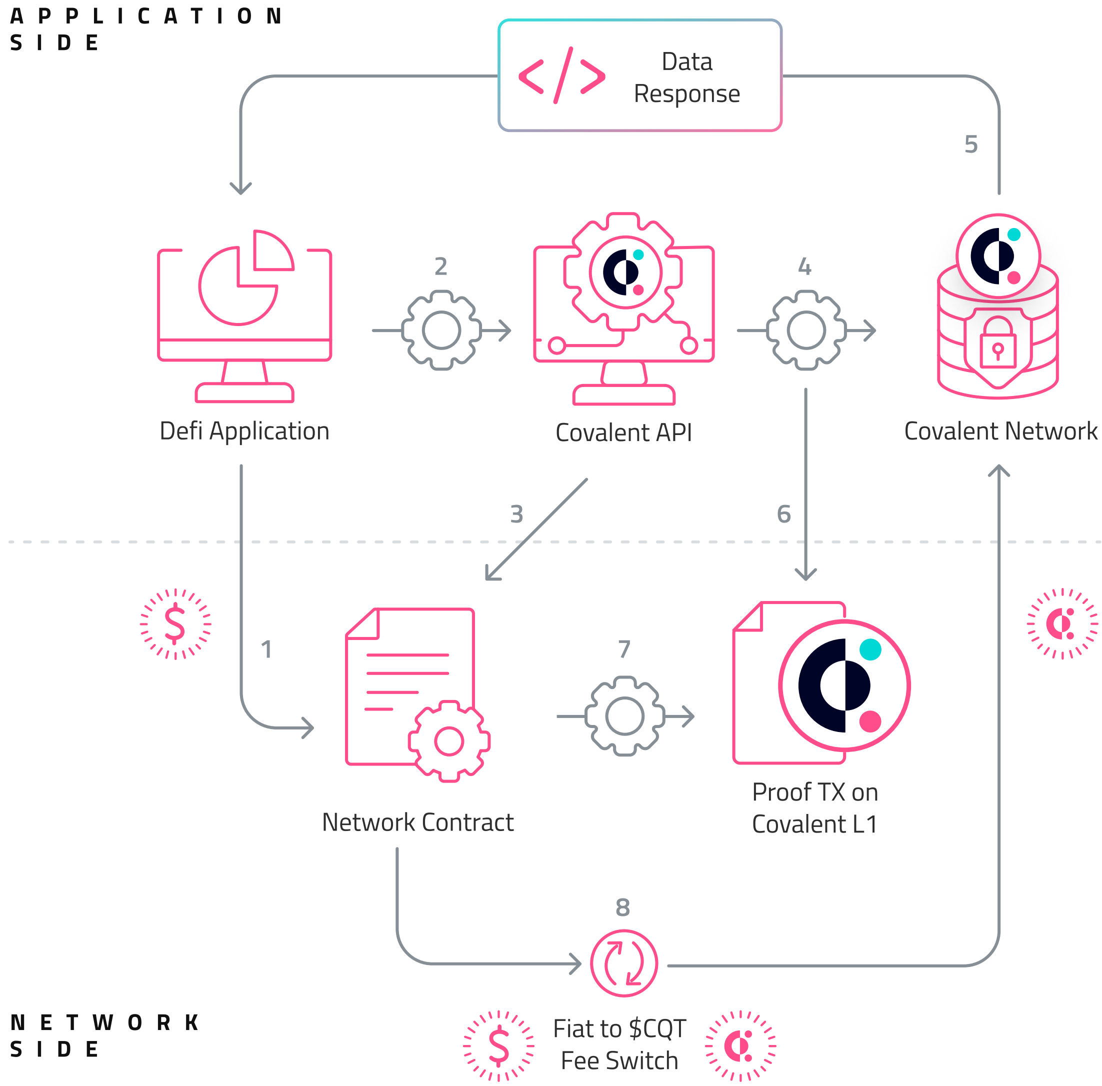
CQT is a network access token distributed to a set of Network Operators that fulfill duties properly, either through broadcasting data over the network or responding to user API queries. CQT is not, and not intended to be, a medium of exchange accepted by the public (or a section of the public) as payment 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. Outlined below are some of the critical use cases for CQT.

Infrastructure Currency

CQT will be used as the platform currency that powers interactions between users on the Covalent Network. Query Operators who deliver verifiable data to API users successfully are compensated in CQT tokens.



This works as follows:

1. The application/developer loads their deposit account with stablecoin assets into the network smart contract.
2. The application queries the Covalent API.
3. A check is made to verify that there are sufficient funds in the deposit account before sending the respective query request to Operators.
4. The query is sent to the Query Operator to fulfill the request.
5. The desired data is sent back to the application.
6. An entry is made on the Moonbeam ledger with the amount of data that is being consumed, and which Operator(s) are fulfilling the request and their cost.
7. The balances between the network contract, CQT and the work performed are reconciled.
8. The USD funds are drawn down from the developer's deposit account and swapped for CQT via a market buy mechanism and settled against an Operator's outstanding balance.

Furthermore, those making requests for data (as customers) will never interact with the CQT economy. Instead, their payments will be made in standard dollar figures and charged in stablecoins. This is for a number of reasons:

- Stablecoins are the most highly adopted and liquid tokens in the Web3 ecosystem.
- Enterprise customers do not want to hold a token that is susceptible to volatility on their balance sheets.
- It 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 parties from speculating on the underlying currency.

Rewards for Indexing, Hosting, and Serving Data Over the Network

Since a community of users can deliver data better than a single firm could, our philosophy is to expand and incentivize 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 over the Covalent Network, especially those 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.

Covalent Network Staking

Staking is an essential feature of the Covalent Network. All Network Operators have to meet the minimum staking requirement. This mechanism is in place to promote and ensure correct behaviour in the Covalent Network. If Network Operators are ever malicious or dishonest, a percentage of their staked amount will be slashed. Furthermore, Network Operators stand to earn more CQT by providing utility to the network.

Beyond Network Operators, Delegators can delegate their tokens to Network Operators who stand to earn yield for doing so.

There are some further points to note with regard to delegating:

- Staking is live on Moonbeam with bridging facilitated by Wormhole.
- Delegating is non-custodial. While CQT is held in the staking contract, only the owner of the respective staked CQT can interact with it.
- Staked CQT is held in escrow on the network. Consequently, staked assets are inaccessible to the token holder while they are being used to secure the network. In order to reverse this, the delegator must un-stake the principal amount of CQT they staked.

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.

Governance

The CQT token will be gradually used to remove the need for the Covalent team to be a key player in 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, the 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.

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 Network Operator 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 the network launch, only the BSP role 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.

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 adoption of cryptocurrencies by many large enterprises such as 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 and analysts use Covalent to build exciting multi-chain applications like crypto wallets, NFT galleries, and investor dashboard tools utilizing data from 110 + blockchains. Covalent is trusted by a community of 40,000+ developers and powers data for 5,000+ applications, including 0x, Zerion, Rainbow Wallet, Rotki, Bitski, and many others.