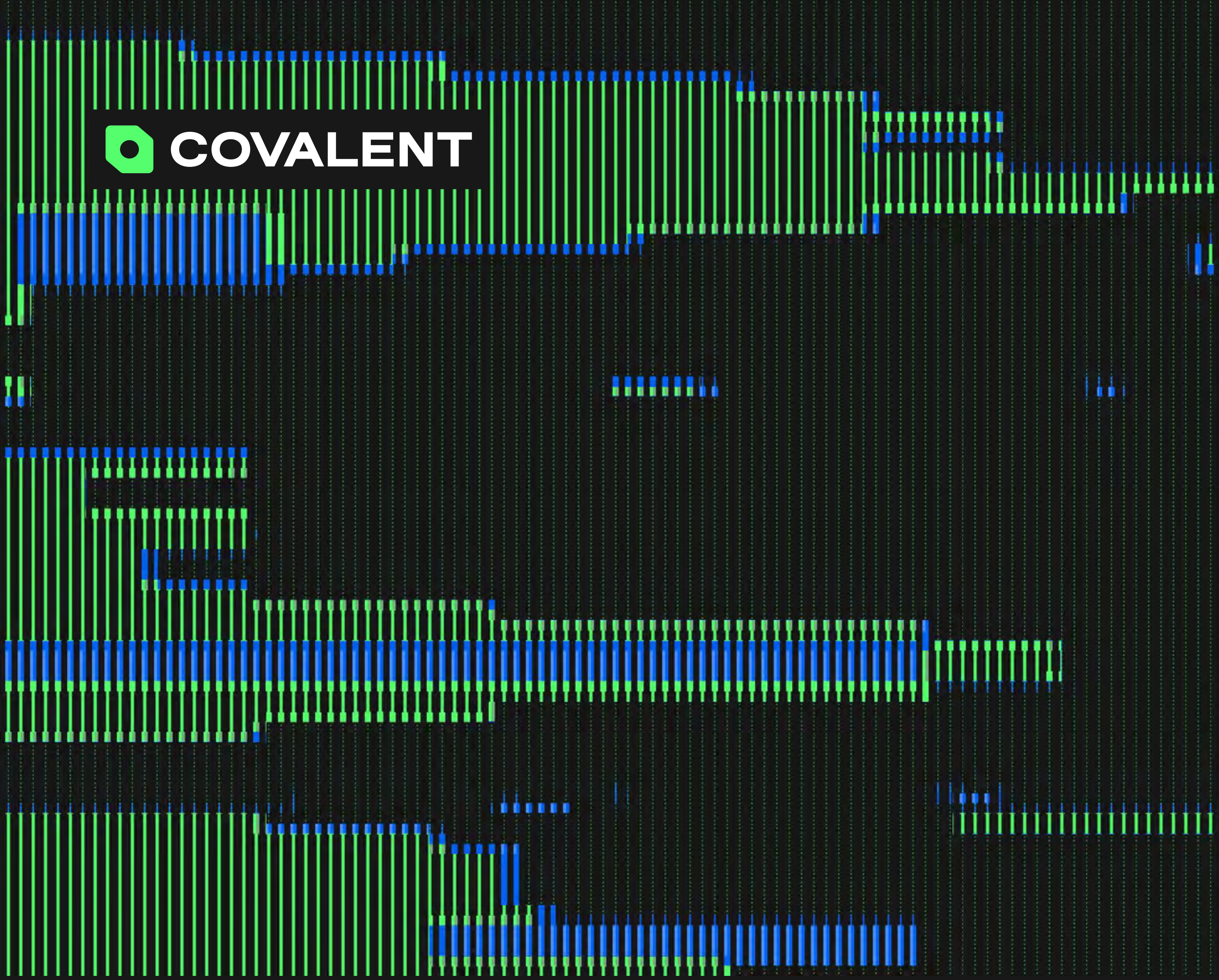


The Block Result

Bringing unparalleled insight to blockchain data through concurrent block re-execution technology.

Last updated: 27th August 2024

Background

Since open-sourcing the Block Specimen specification and reference implementation in 2022, Covalent has remained steadfast in its mission to bring transparency to blockchain data. The Block Specimen brought data standards into Web3 that any client can adopt, while the GoldRush API has powered more than 5,000 Web3 projects.

However, the landscape of blockchain technology has continued to evolve since the release of the Block Specimen and demands further advancements. Although Ethereum has made commendable progress in efficient transaction write scaling by adopting proof-of-stake, rollups, L2s, privacy-preserving EVMs and modular designs (consensus-execution-data availability separation), their focus has been on data write, while the long-term availability of historical data remains an increasingly concerning issue. A decentralized, comprehensive and provably correct data solution will be critical in the coming years for this exact reason.

Fortunately, the Covalent Network, and specifically the Block Specimen, is the answer to Ethereum's long-term data availability problem, with anyone being able to create a canonical representation of a blockchain's full historical state in the form of standalone block data objects. And now, Covalent introduces the next phase of the Covalent Network, the Block Result Producer. The Block Result Producer builds upon the innovation of the Block Specimen, offering enhanced capabilities that cater to the evolving demands of the blockchain community.

Granular Blockchain Data Remains Inaccessible

A lot of progress has been made in accessing blockchain data in recent years. Compared to Covalent’s founding year in 2018, it has become significantly easier to access "high-level" blockchain data like historical transactions and balances. However, the same cannot be said for deep, granular data, with the demand for such only increasing especially from enterprises.

To truly unlock the full potential of blockchains, the industry needs simple access to trace data, which offers unparalleled insights into transaction details, internal transaction calls, and contract interactions, native token balance transfers, error decoding, transaction debugging, and more. Without this, no one should expect institutions and enterprises to be adopting blockchain technology, both of which are increasingly demanding this type of granularity. Significant insights can be drawn for transactions with this data that are especially important for taxation, auditing, enhanced analytics, and investigations. Simply put, reading granular blockchain data at scale will be critical for efficient web business operations and continuity.

Currently, there are a number of methods for extracting granular blockchain data: 1) using a block explorer, 2) trace APIs or 3) running a node. Unfortunately, these methods have notable drawbacks that hinder developers, researchers, and users from accessing the comprehensive data they require.

Block explorers have been a popular means of exploring blockchain transactions and data. They are excellent at examining value transfers, events emitted and so on. But even block explorers have their limitations, especially when it comes to examining what was executed by the EVM during a specific transaction set. Only one transaction can be examined at a time for example.

Trace APIs, meanwhile, offer access to trace data related to transactions and blocks. This data includes details such as virtual machine execution traces, and state differences, among others. However, these APIs utilize the JSON-RPC layer, which results in a slower and less scalable approach.

Finally, one could run a full archive Geth node that allows for tracing and inspecting the Ethereum network's block execution. Running a Geth node certainly provides more extensive data access and control over the tracing process. However, running a node also comes with its own set of challenges:

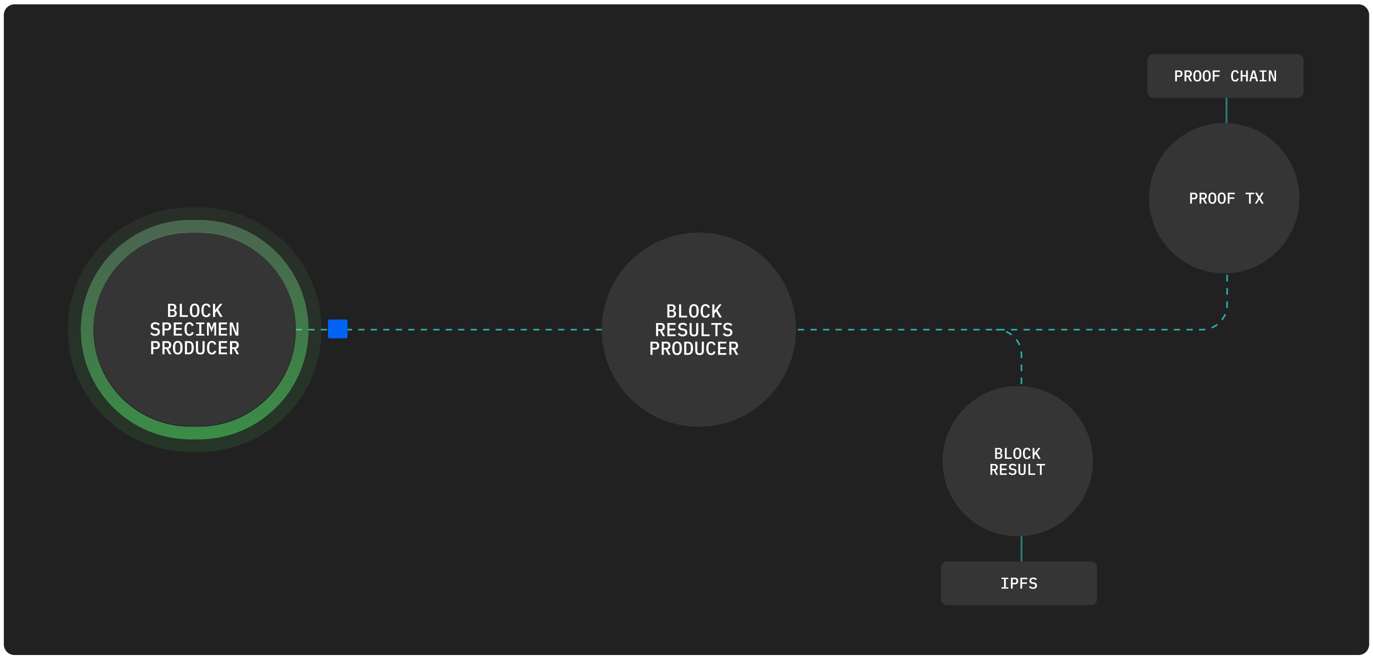
1. **Resource Intensive:** Running a node requires substantial computational resources and storage capacity. Blockchain networks, especially popular ones like Ethereum, demand significant processing power and storage to synchronize and maintain the node. Not to mention, when one wishes to perform a trace, every transaction must be re-executed from Genesis. This takes significant time and resources.
2. **Synchronization Time:** Setting up a new node or catching up with the latest blockchain state can be time-consuming, as it involves downloading and verifying the entire blockchain's history.
3. **Technical Expertise:** Running a node requires a deep understanding of blockchain software and infrastructure.
4. **Capital Intensive:** While a necessary security guarantee, running a node (validator) requires high upfront capital investment in the underlying stake token.

Introducing the Block Result

The Covalent Network's latest innovation, the Block Result, aims to address these challenges and unlock the true potential of granular data access in the Web3 space.

The Block Result Producer employs a data processing framework, the purpose of which is verified data transformation with transformation scalability. The output, one of many possible, is a Block Result; a one-to-one representation of block data returned from an RPC call to a blockchain node with optional and additional informative fields.

And importantly, the Block Result Producer can perform arbitrary transformations over any input concurrently with other transformations. As a result, the Block Result Producer can concurrently re-execute ethereum blocks and transactions completely external to their source node client software. This is something that no blockchain node clients can currently do. This enables customizable low overhead simultaneous data indexing, for any consumer of Block Specimen data, slicing and dicing it as they see fit.

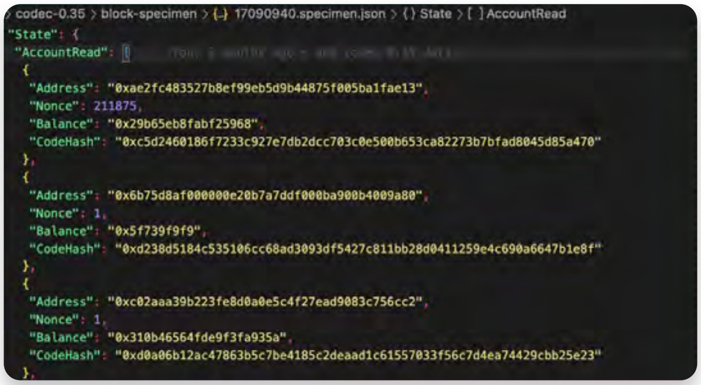


To draw an industrial analogy, an oil Refinery takes in crude oil and manufactures various useful products such as gasoline (petrol), diesel, asphalt base, fuel oils etc in many cases simultaneously in the same refractory location. Similarly, the Block Specimen, like crude oil, can be used to manufacture many data artifacts such as Block Results and trace event streams through the re-execution done within the refinement process.

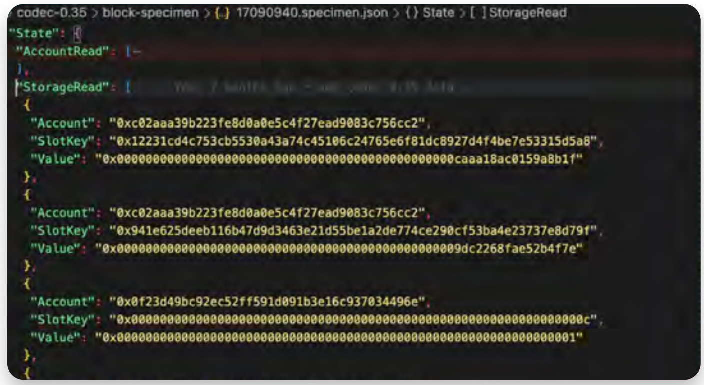
Expected Enrichment

As it stands, a Block Specimen (the output from a Block Specimen Producer) captures a state snapshot. That is, all of the states read and transaction information. However, it doesn't capture the side effects of executing the block and its transactions (like receipts) or 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 Covalent Network more storage and bandwidth efficient. Extracting and storing Block Specimens allows the network to move away from relying on blockchain nodes and executing blocks sequentially to re-execute an independent block and draw detailed insights from underlying captured data. And as mentioned, this can be done concurrently, as verified, secure Block Specimens can be re-executed without having to come through a sequential canonical chain.

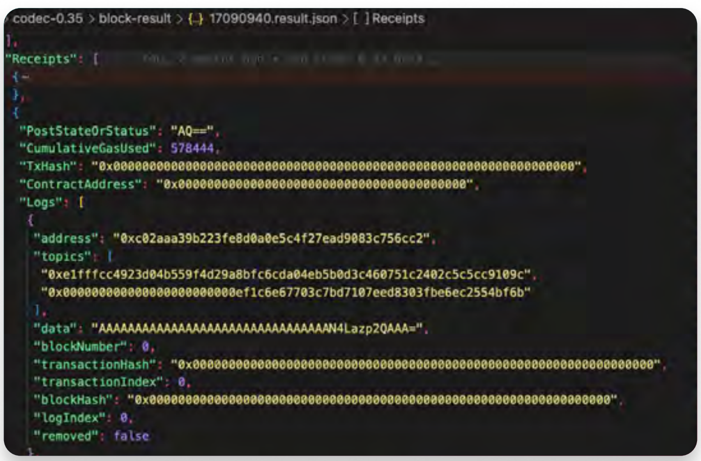


State - Account Read snippet
(Specimen only)

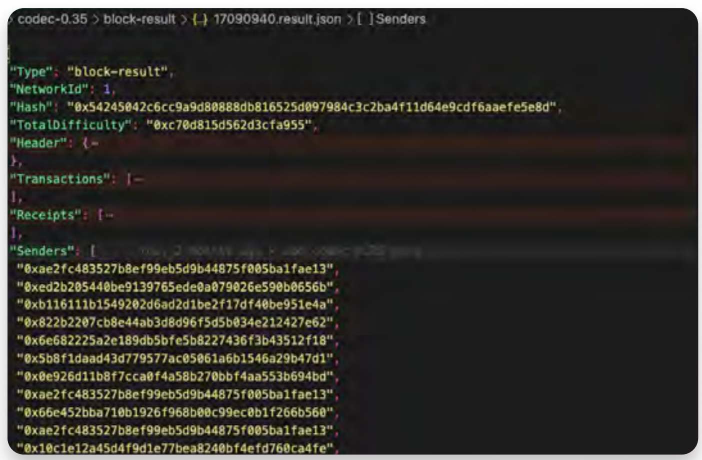


State - Storage Read snippet
(Specimen only)

Block Results are different from Block Specimens in two main fields; 1) Receipts and 2) State Reads. Receipts are generated for Block Results from re-executing all the transactions with a stateless ethereum virtual machine tool and all the pre-state existing in a Block Specimen in the form of state reads (and senders) fields. State reads come in the form of accounts, block hashes and code read from Ethereum storage for that particular Block Specimen export.



Receipts field snippet
(Result only)



Senders field snippet
(Specimen & Result)

Further, the receipts field contains all of the event log information contained in a block including topic hashes while the senders field contains the externally owned accounts or contract accounts that initiated all the transactions in that particular block. Sender recovery is expensive hence it's pre-computed during export from the block producer node itself and available in both Block Specimens and Block Results for their utility. However, Block Results do not contain any of the aforementioned state reads as they are no longer necessary post transaction re-execution and receipts reconstruction.

With the Block Result Producer live, Block Specimens can be enriched and transformed into Block Results by multiple operators on the Covalent Network. These verified Block Results can then be retrieved and indexed by anyone utilizing the Covalent Network.

The net result is that anyone will be able to pull historical and live blocks that are complete with the data one could retrieve by calling the JSON-RPC layer but without the bottlenecks associated with the latter method. Furthermore, anyone wishing to analyze this data or build a Web3 data application on top of the Covalent Network, such as block explorers, can pull this type of granular data without running a node. This is due to the property of Block Results being independent securely proven data objects that can exist in any publicly available cloud storage or decentralized storage network. Anyone can simply specify a parameter to filter by time or block height ranges for example and pull the desired Block Results and its contents easily.

Transformational Benefits to Web3 Data

Distributed Processing: With the Block Result Producer built and deployed in a distributed environment decoupled from storage, execution or consensus nodes, the Covalent Network benefits from the clear separation of performance concerns and further the ability to run concurrent re-executions of blocks via just their Block Specimen replicas.

To explore this point more, let's examine tracing. To trace a blockchain linearly, one must use a blockchain node. Not only could this take a month or more to do depending on the size of the blockchain, but it locks up that node from being used. This issue scales with how many blockchains one wishes to index and also with all historical data that is not made available as Ethereum employs state reduction through pruning.

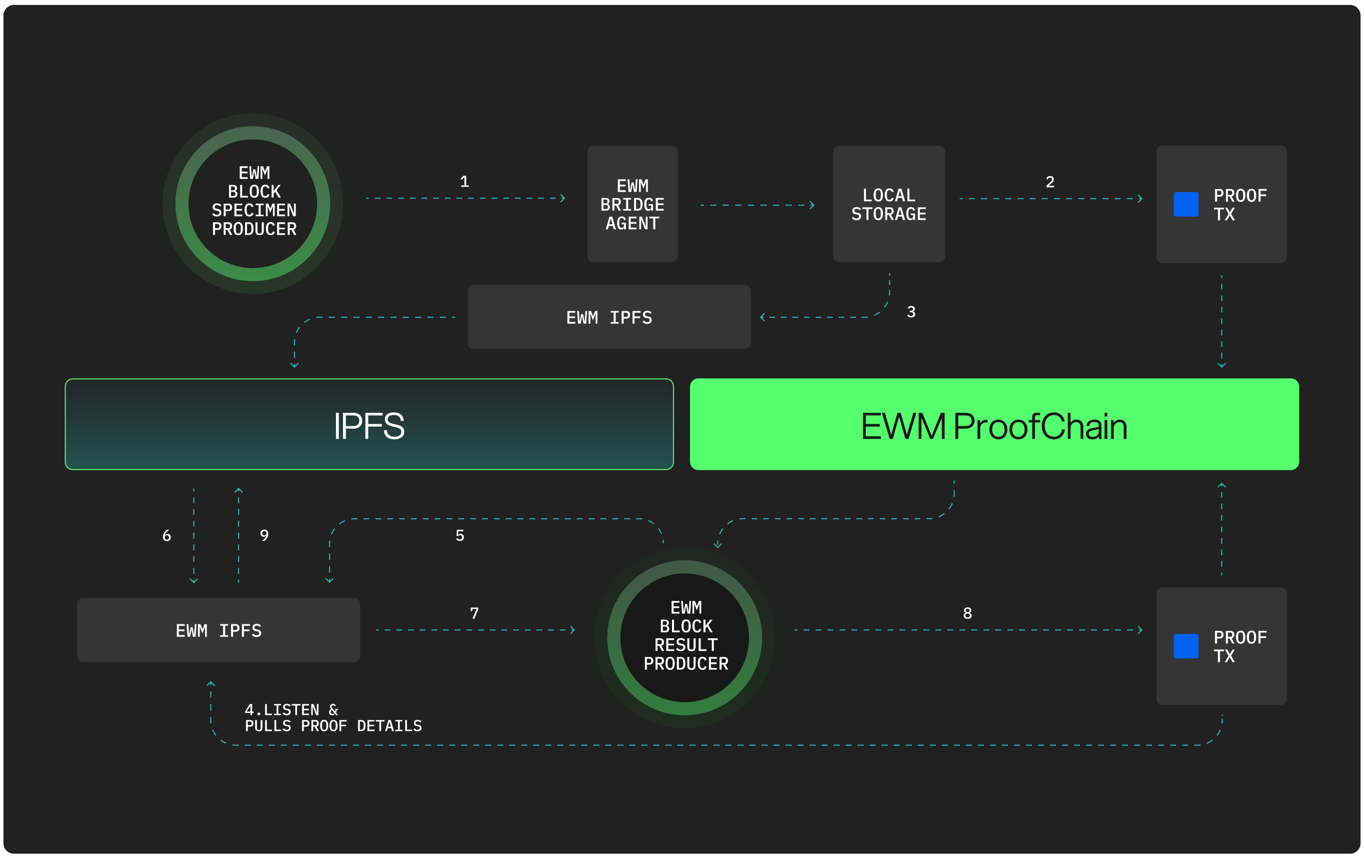
Given the Block Specimen format and the ability to now re-execute them using the Block Result Producer, one can trace or retrace over a small slice of blocks independently of the entire blockchain. If each node does this, the entire blockchain can be re-executed much faster.

In all other approaches, computations for extraction and indexing are executed on sequential blocks starting over from genesis. This is why there are synchronization issues with pulling this type of data by running a node locally. With the Covalent Network, these computations are completed in parallel leading to a) greater efficiency and less overhead on the network and b) faster and more flexible chain data sync-up.

No More JSON-RPC Calls: With Block Results, anyone can retrieve one-to-one representations of blocks without having to call the JSON-RPC layer. This allows Covalent, and any other protocol that builds on top of the Covalent Network, to scale freely without the constraint of running nodes. Furthermore, nodes can focus on what they are actually built for; executing transactions on the blockchain, instead of worrying about read scalability.

Cryptographically Secure: Similar to the production of Block Specimens, all of the work that the Network Operators do with the Block Result is cryptographically secure. For every Block Result created, a corresponding proof is created which undergoes validation, ensuring the data is honest and accurate. Once consensus is achieved, the Network Operator who published the proof is rewarded in CXT.

How the Block Result Producer Works

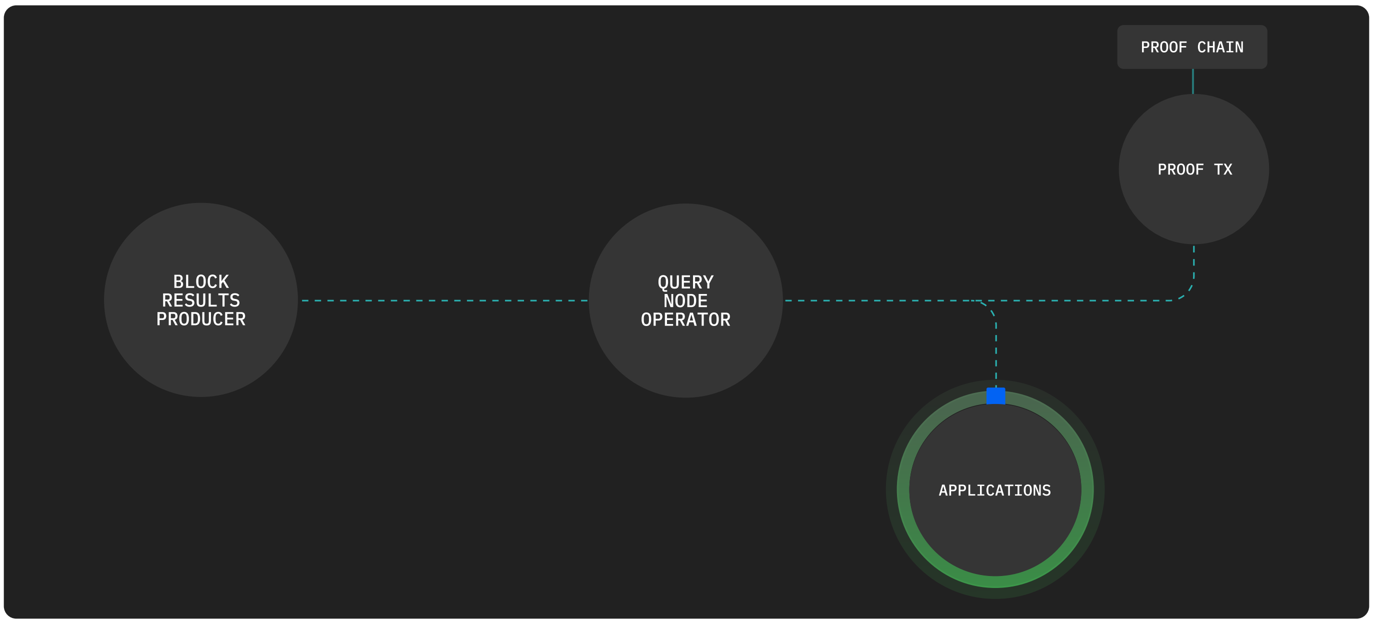


At a very high level, the Block Result Producer locates a source to apply a transformation to and outputs a data object, generated from applying such a rule is the target. In details these three components are as follows:

- **Source:** The Block Specimen that serves as an input to the Block Result Producer.
- **Transformation:** A transformation plugin or binary that can act on the Block Specimen (source). These can be compared to blueprints that have been shown to produce the data objects needed. Furthermore, anyone can create these rules to get a desired data object. Hence, one can generate an output with desired additional or reductive fields, offering enhanced flexibility to the Block Result Producer. Rule versions thus need to exist, tied to the decoded Block Specimen versions they are applied on.
- **Target:** The output generated from running the rule over the object that came from the source.

Currently, the only source the Block Result Producer can take is the Block Specimen, as well as the only output currently produced, is the Block Result. These data objects are eventually made available through a decentralized storage service such as the interplanetary file system (IPFS). A transformation-proof transaction is emitted and executed on a public blockchain confirming that it has done this work along with the output (IPFS) access URL. Thereby Block Result rules become task blueprints registered on-chain. Block Result Producers pick these rules up and generate their tasks to work on. The rules denote a <Source, Transformation, Target> triplet or tuple, each of which represents an input network artifact, a transformation engine (or Block Result plugin), and the output network artifact, respectively.

Block Result Generation Pipeline



Drilling down further into the fine-grained operation of the Block Result Producer, in the current state of the data transformation pipeline, we begin with Block Specimen discovery, identifying Block Specimens to transform and then locating it with the content storage hashes once several other checks have been completed. To identify the Block Specimen, the Block Result Producer listens to log events emitted from the current Proof Chain which is the smart contract logic situated on a public blockchain for receiving and verifying proofs, for finalized Block Specimens that have achieved consensus and Covalent Network rewards. As the Block Result Producer listens to such events, it must consider:

- Whether it has the applicable rule set that turns it into the desired target.
- If so, the Block Result Producer must then determine whether the source has already undergone a transformation (the target has already been created by another network operator). The system should work so that every Block Result instance builds every target.
- To determine this, the Block Result Producer listens to events emitted from the Proof Chain, creating a list of targets it has seen. If the source hasn't been transformed, the applicable rule to transform is available and if another network operator hasn't picked up the work, the idle network operator will perform the work.

Ideally, the Block Result Producers should be a distributed data-build system, where nodes can pick up transformations of data objects that haven't been transformed but can exist as incomplete targets on their local machines. In order to achieve this proofing scalability is necessary and this is being currently researched with plans leading to the deployment of an independent Covalent public L1 blockchain.

To locate a Block Specimen source, network operators can do the following:

1. Discover the source from the Proof Chain log events, these contain the content hashes of the data objects to pull files from distributed storage.
2. Find it locally if they're already performing Block Specimen Producer duties in the network.

Once one such event is found, the finalized Block Specimen is fetched and processed in the pipeline.
This involves:

1. Decoding the Block Specimen for Block Result Producers' underlying Erlang runtime.
2. Block processor takes the decoded Block Specimen, runs it through a stateless EVM re-execution tool and returns a Block Result.
3. Block Results cross-verified by submitting proof for consensus to the Proof Chain contract
4. IPFS interactor uploads the Block Result to a decentralized storage network directly through IPFS.

Codified Example

One way to produce a trace event stream (TES) of format TES1.0, is to find a Block Specimen of format BSP1.0, and feed it into a build plugin that is registered as a "TraceEngine" with version "1.0.0". This rule will then be run against all Block Specimens of format BSP1.0. The stringified version of such a rule is as follows:

```
TraceEventStream(Format=TES1.0) ← BlockSpecimen(Format=BSP1.0): TraceEngine(Version=1.0.0)
```

Let's say later the Covalent Network gets an upgrade to the TraceEngine plugin to version 1.0.1, as a bug fix for how it generates traces, this requires the addition of a build rule to the list of build rules that the Block Result Producer is aware of as follows:

```
TraceEventStream(Format=TES1.0) ← BlockSpecimen(Format=BSP1.0): TraceEngine(Version=1.0.1)
```

For the new rule set in this system, the Block Result Producer restarts the build for all BSP1.0 Block Specimens and produces new Block Result targets. Similarly, we can have the following rule for generation of Block Results:

```
BlockResult(Format=BRP1.0) ← BlockSpecimen(Format=BSP1.0): StatelessEVM(Version=2.0.0)
```

The updated finalized proof submitted by the Block Result Producer then consists of:

- Source version and hash;
- Transformation plugin version;
- Target hash.

This allows the Block Result Producers to make fully informed decisions on the availability of new plugins and sources to be worked on. Codifying the upgrade process further allows the Block Result Producer to decide whether to skip building a particular transformation, in case the proof was already submitted and consensus reached on-chain.

Coordination

Deploying a distributed engine presents a challenge in coordinating workload and ensuring honest computations between Network participants and validators.

To coordinate efforts among nodes, the Covalent Network will use a proactive optimistic model. This model involves discovering work across the network and nodes picking up work according to above explained predetermined rules.

To ensure the accuracy and honesty of transformations, cryptographic proofs will be sent to the upgraded Proofs I Chain. When consensus is achieved among proofs, it is possible to determine if a network operator's work is honest and accurate.

Conclusion

With the Block Specimen and Block Result, the Covalent Network will become a hub of decentralized and comprehensive data solutions. The GoldRush API has already empowered thousands of Web3 developers, and now, with the added capabilities of the Block Result Producer, the possibilities are boundless. Data applications, block explorers, and other protocols can leverage the Covalent Network to access and analyze web3 data without the need to run nodes, democratizing access and fueling innovation.

This exciting future envisions a thriving blockchain community where transparency, scalability, and granular data accessibility are the norm. It paves the way for broader blockchain adoption as institutions and enterprises recognize the potential and benefits of this transformative technology. Covalent's commitment to pushing the boundaries of read scalability and granular data access propels the blockchain ecosystem into new frontiers, enabling groundbreaking applications, informed decision-making, and a more inclusive and transparent digital economy.

In this world, the potential of blockchain technology is fully realized, and the possibilities are limitless. With the Block Specimen and the Block Result leading the way, we can expect an era of unparalleled decentralization, and transparency in data read scalability.

About Covalent

Covalent is the leading modular data infrastructure layer dedicated to solving major challenges in blockchain and AI, including verifiability, decentralized AI inference, and Long-Term Data Availability. Its large reservoir of structured, verifiable data enhances decentralized training and inference, reducing the risk of manipulated or biased AI models. Additionally, the Covalent Network's Ethereum Wayback Machine ensures secure, decentralized access to Ethereum's transaction data. Trusted by over 3,000 leading organizations, Covalent powers AI, DeFi, GameFi, and more with unfettered access to on-chain data from over 200 blockchains.

COVALENT

Modular Data Infrastructure for AI.

@Covalent_HQ

covalenthq.com/telegram

covalenthq.com/discord

covalenthq.com