erigon.tech

BEYOND SOFTWARE FRONTIERS

ERIGON 3

A New **Paradigm** for Ethereum Clients

ABOUT US

#erigon.tech Is global remote development team specializing in efficient blockchain client software

Erigon a combined **CL/EL** Client

- Based on **Turbogeth** it was designed to synchronize a **Full Archive Node** on commodity hardware
- Supports multiple EVM based chains including **Ethereum, Gnosis** and **Polygon**
- **Erigon 3** is the latest version and is in **alpha** for all supported chains, due for **beta** by **perctra**

erigon.tech

BEYOND SOFTWARE FRONTIERS

OVERVIEW

PARADIGM

SHIFT

JOURNEY

ARCHITECTURE

IMPLICATIONS

FUTURE

erigon.tech

BEYOND SOFTWARE FRONTIERS

THE PARADIGM SHIFT

*From **Consensus vs Execution** , to **Dissemination vs Distribution***

Chain Dissemination	Chain Distribution
Operates in real time	Operates after finalization
Interpreted	Compiled
Negotiated Serialized Data	Verifiable Binary Data
Block/Slot/Batch Granularity	Transaction Granularity
Verified by Re-Execution	Verified by Data Hashing

erigon.tech
BEYOND SOFTWARE FRONTIERS

JOURNEY

Erigon 1,2 ...
How we got here

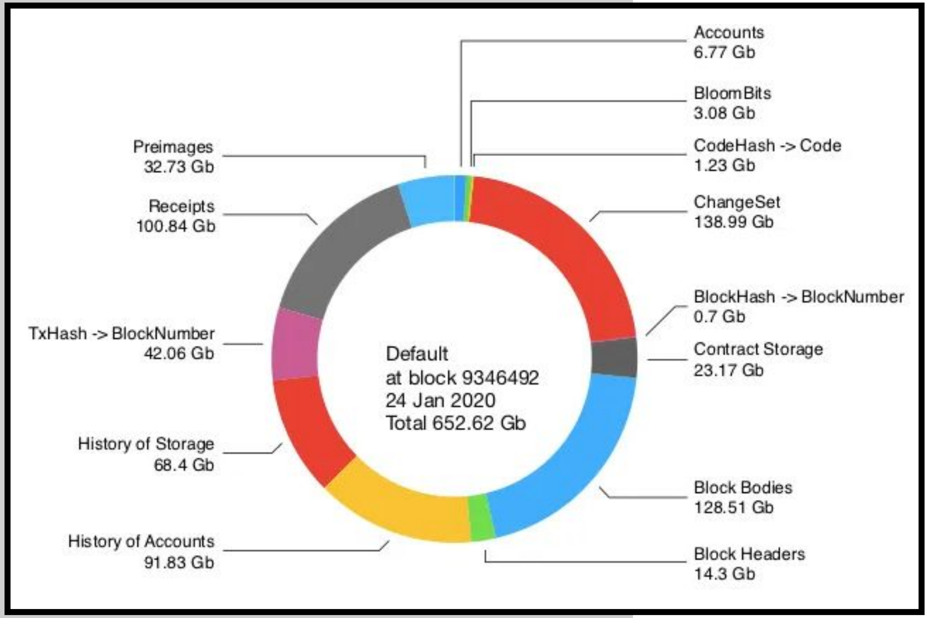
erigon.tech
BEYOND SOFTWARE FRONTIERS

TURBO-GETH

2017-2020: Optimized Ethereum Client

An experiment to challenge the design choices made in major Ethereum clients

- Optimized Ethereum storage
- Used Bucketing to increase data retrieval speed
- Used third party B+- Trees instead of LSM as its underlying database model



Chain Size Block No. 9346492:

Geth:	**3.7 TB**
Parity:	**3.6 TB**
Turbo-Geth:	**0.7 TB**

#**tubo.geth:** DEVCON 4

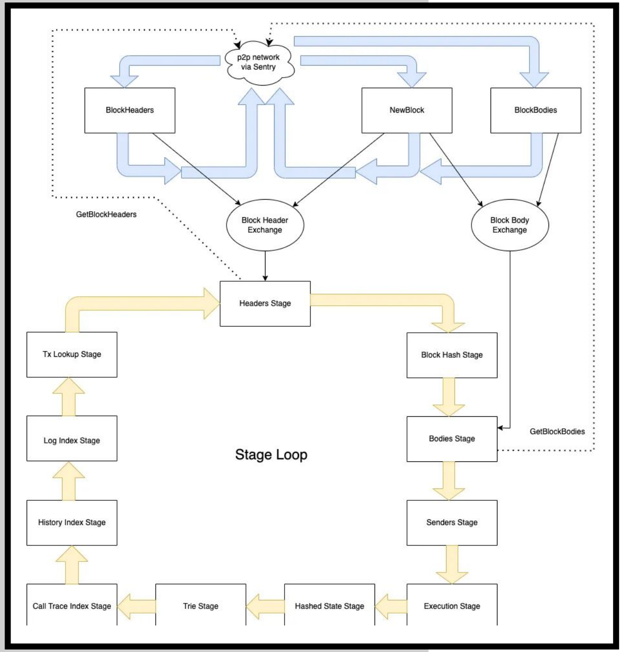
erigon.tech
BEYOND SOFTWARE FRONTIERS

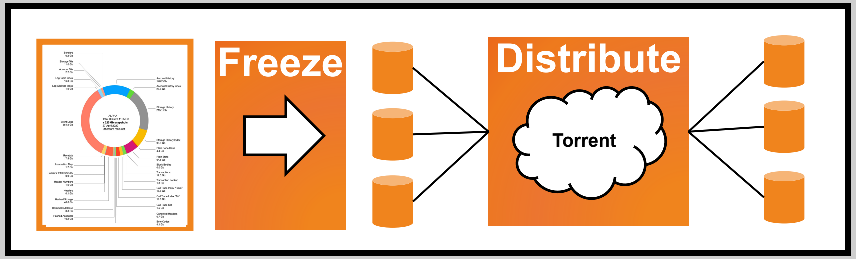
ERIGON 2

2020-2022: Staged Client With Snapshots

A new name and a performance-oriented re-imagining of the turbo-geth data model

- Dynamic export of aged **frozen** transaction data
- Transition to a **page-based** loading strategy
- Torrent distribution of hash protected historic data
- Componentized **stages** with **ACID** guarantees and long running transactions





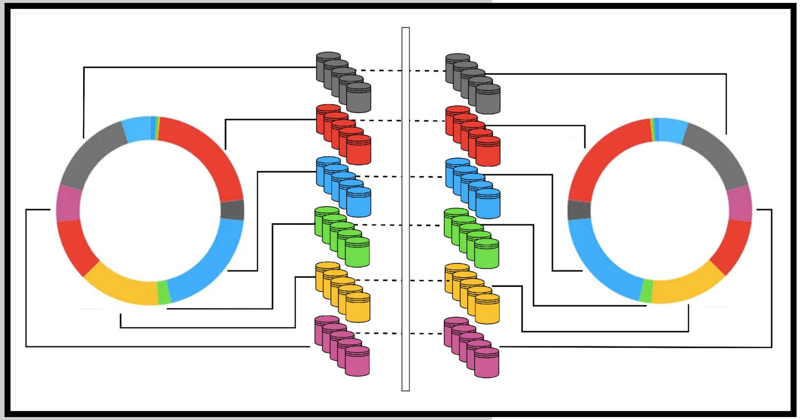
erigon.tech

BEYOND SOFTWARE FRONTIERS

ERIGON 3

2022-2024: Ethereum Client With Native DLT Storage Model

Erigon 2 evolved to complete the extraction of all data types into an aged data store designed from the disk up to handle DLT specific requirements



- Dynamic export of aged **frozen** transaction and state, core database storage minimization
- **Temporal** sharding of all significant data types using a monotonic transaction identifiers
- **Deterministic** persistent data which can be distributed and validated without the need for re-execution.
- **Transaction granularity** history and re-execution when processing queries

#**Erigon 3** is more cold-start-friendly, os-pre-fetch-friendly, cloud-drives-friendly than Erigon 2, avoiding the reliance on expensive NVME disk capacity.

erigon.tech
BEYOND SOFTWARE FRONTIERS

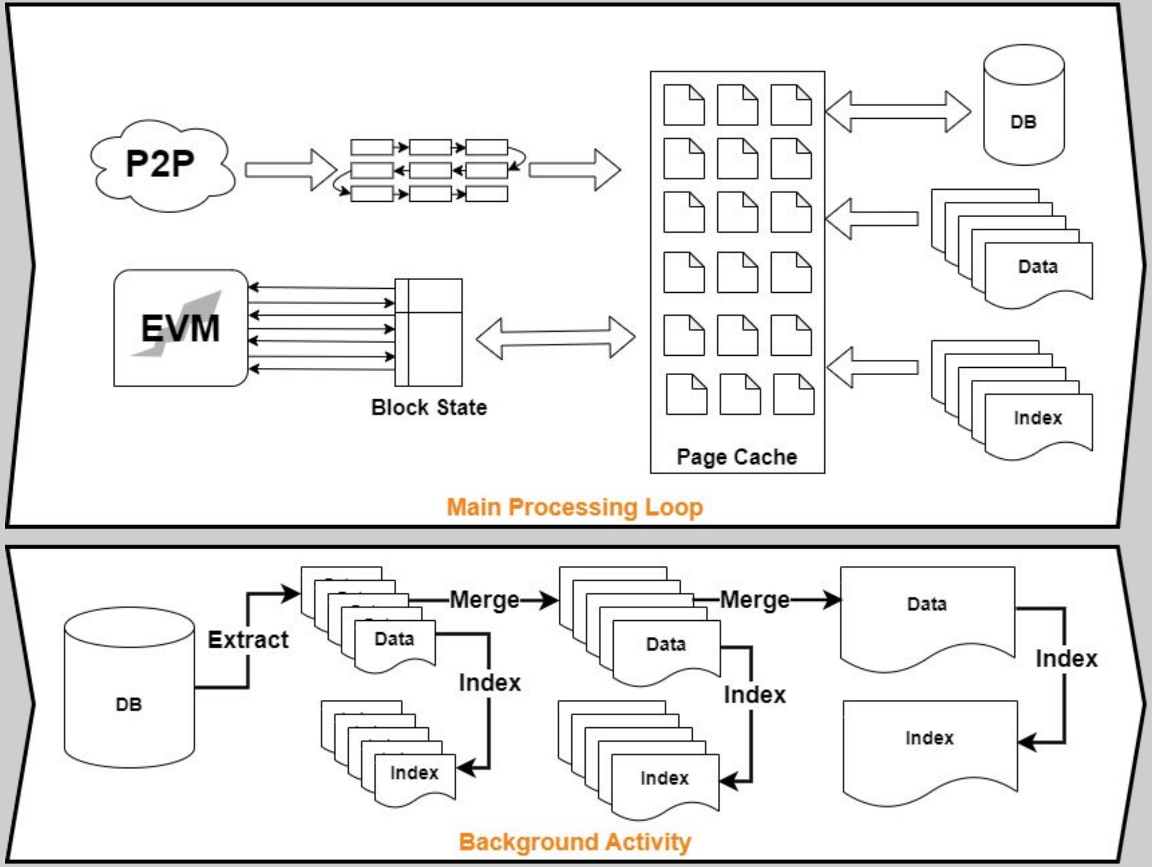
ARCHITECTURE

Where we are now

erigon.tech

BEYOND SOFTWARE FRONTIERS

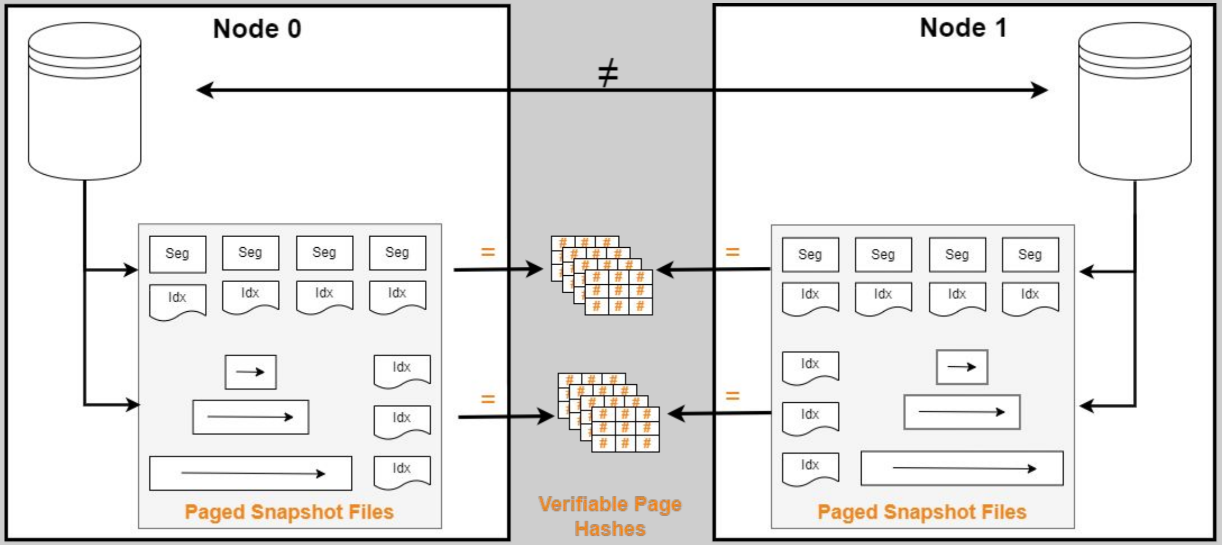
PROCESS FLOW



Erigon 3 has three types of mapped files:

- Appendable Segments
- QLSM State
- Indexes

PAGE FILE DETERMINISM



- **Database** files deal with interspersed read/write operations where page operations and optimisation lead to physical file layouts which **change** between nodes and process runs

- **Snapshot** files have a **consistent** optimized layout which is guaranteed during the snapshot creation process.

IMPLICATIONS

What this means for chain operations

erigon.tech

BEYOND SOFTWARE FRONTIERS

SYNC PERFORMANCE

Database sizes after first sync

Chain	Archive	Full	Minimal
Ethereum (EL+CL)	1.6 TB	838 GB	235 GB
Gnosis (EL+CL)	486 GB	268 GB	91 GB
Polygon	4.2 TB	2 TB	873 GB

Sync times from scratch to chain tip (100 Mbyte/Sec Network)

Chain	Archive	Full	Minimal
Ethereum (EL+CL)	7h 55m	4h 23m	1h 41m
Gnosis (EL+CL)	2h 10m	1h 05m	33m
Polygon	42h 50m	21h 41m	11h 54m

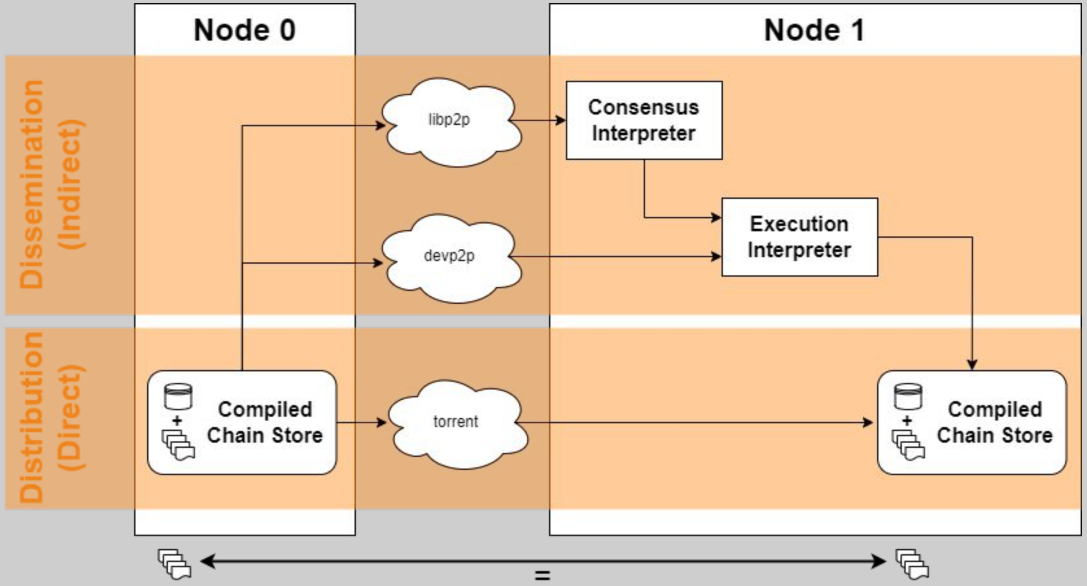
Erigon 3 sync times are proportional to available network bandwidth.

On a **1GByte/Sec** network a **Polygon** archive node syncs in **2h 40m**

erigon.tech

BEYOND SOFTWARE FRONTIERS

CHAIN DISTRIBUTION



- **Distributed** data in binary form is available for processing after its hashes are verified – files can be delivered via any available transfer medium.

- **Disseminated** data is serialized for transport and must be deserialized and interpreted before its available for processing

FUTURE

New opportunities…

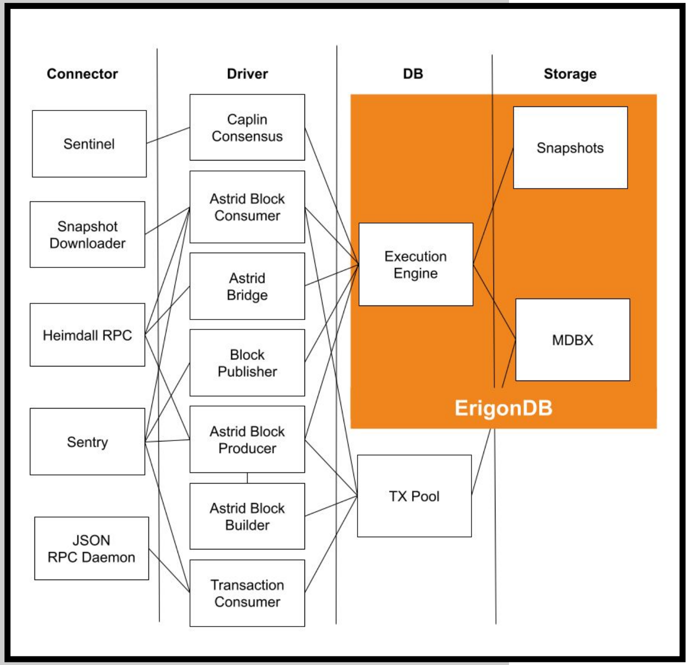
erigon.tech

BEYOND SOFTWARE FRONTIERS

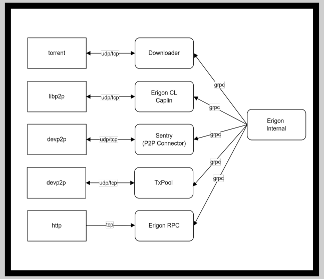
COMPONENTIZATION

Post Erigon 3 we will be adjusting Erigon's component model to achieve the following goals:

- Active vs Passive Components
- Functional Decomposition
- Development Compartmentalization
- Deployment Flexibility



Development

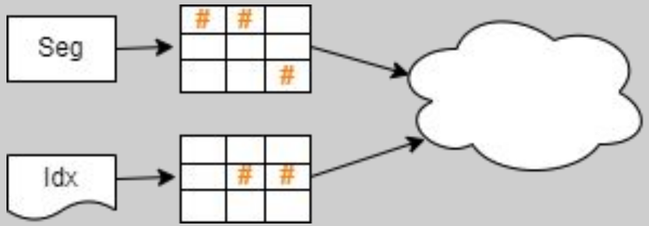


Deployment

Erigon has several incarnations of components with gRPC interfaces, stages and the engine-api based driver/db split. As the DB interface becomes more stable we can concentrate on engineering this for consistency, reliability and extensibility.

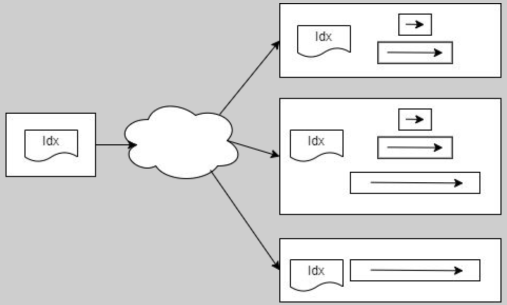
DISTRIBUTION MODEL EXTENSIONS

Sparse Clients



- For clients who inly need a defined subset of chain data
- Additional data can be retrieved via paging at the cost of network latency

Distributed Indexing



- For query operations from clients requiring full index access
- Index segregation strategies require further R&D effort

erigon.tech
BEYOND SOFTWARE FRONTIERS

QUESTIONS

- Hash verification is currently Erigon dependent. Can this, should this be more decentralized
- Hashing provides verification how about availability
- Is the model more widely adoptable. Can it, should it be standardized
- Are sparse clients an alternative to light clients
- Does the addressability of the binary format have uses other than distribution. For example, could it provide a proof target.

erigon.tech

BEYOND SOFTWARE FRONTIERS

THANK YOU

Mark Holt

mark.holt@erigon.tech

github.com/erigontech/erigon

www.erigon.tech