

# Pluggable Consensus

keorn edited this page on 15 Apr · 1 revision

Parity comes bundled with a number of consensus engines. While the most widely used is the `Ethash` proof of work Engine, there are others which can be used for proof of authority or stake chains. The Engine is chosen by placing an appropriate entry in the `"engine"` field of the spec and providing the correct `"seal"` under `"genesis"` field, as described on the [Chain specification](#) page.

## Ethash

Original Ethereum PoW Engine .

```
"engine": {
  "Ethash": {
    "params": {
      "gasLimitBoundDivisor": "0x0400",
      "minimumDifficulty": "0x020000",
      "difficultyBoundDivisor": "0x0800",
      "durationLimit": "0xd",
      "blockReward": "0x4563918244F40000",
      "registrar": "0x81a4b044831c4f12ba601adb9274516939e9b8a2",
      "homesteadTransition": 0,
      "eip150Transition": 0,
      "eip155Transition": 10,
      "eip160Transition": 10,
      "eip161abcTransition": 10,
      "eip161dTransition": 10,
      "maxCodeSize": 24576
    }
  }
}
```

The `"params"` object for `"Ethash"` may contain the following keys (YP refers to the Yellow Paper equation numbers):

- `"gasLimitBoundDivisor"` influences how rapidly is the gas limit allowed to evolve, redefines the value in (YP:45–46), originally  $1024 = 0x400$
- `"minimumDifficulty"` is self-explanatory, redefines the value of `D0` in the latter two lines of (YP:39)
- `"difficultyBoundDivisor"` influences how rapidly is the difficulty evolving, redefines the value in (YP:41), originally  $2048 = 0x800$
- `"durationLimit"` is the equilibrium block interval for the pre-Homestead era difficulty evolution, irrelevant for block numbers above `homesteadTransition`, redefines the value in (YP:42), originally  $13 = 0xd$
- `"difficultyIncrementDivisor"` is related to the equilibrium block intervals for the Homestead era difficulty evolution, redefines the value in (YP:43), originally  $10 = 0xa$
- `"homesteadTransition"` is `NH` of (YP:13, 39) etc., or the block number after which the second line of (YP:39) is superseded by the third line, changing the difficulty evolution function
- `"blockReward"` specifies the reward in wei given for authoring a block

```
"seal": {
  "ethereum": {
    "nonce": "0x0000000000000042",
    "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
  }
}
```

► Pages 87

## Parity

- [Setup](#)
- [Getting Synced](#)
- [Basic Usage](#)
- [FAQ](#)

## Using Parity

- [Parity Wallet](#)
  - [Back-up & Restore](#)
  - [Wallets & Vaults](#)
  - [Ledger Nano S](#)
- [Configuring Parity](#)
  - [Available Chains](#)
  - [For Mining](#)
  - [Wallet Remote Access](#)
  - [Network Config](#)
  - [Docker](#)
- [Community Guides](#)

## Developing

- [Writing DApps](#)
  - [oo7 Examples](#)
  - [oo7-parity.js Reference](#)
  - [Deploying Dapps to Parity Wallet](#)
  - [Parity Dapp Registry](#)
  - [Dapp Discovery](#)
  - [Parity Name Registry](#)
  - [Parity GitHub Hint](#)
- [ERC20 Tokens](#)
  - [Token Deployment](#)
  - [Token Registry](#)
- [Smart Contracts](#)
- [Dapp Tutorial](#)
  - [1: Get Started](#)
  - [2: oo7 Bonds](#)
  - [3: Parity Bonds](#)
  - [4: Call Contracts](#)
  - [5: Post Transactions](#)
  - [6: A New Contract](#)
  - [7: Interaction](#)
  - [8: Events](#)
  - [9: Deploy Contracts](#)
  - [10: Sign Data](#)
- [JSONRPC Guide](#)
  - [web3](#)
  - [net](#)
  - [eth](#)
  - [eth\\_pubsub](#)
  - [personal](#)
  - [parity](#)
  - [parity\\_accounts](#)
  - [parity\\_set](#)

## Instant Seal

Engine which can be used for development, as described in [Private development chain](#).

```
"engine": {
  "instantSeal": { "params": {} }
}

"seal": {
  "generic": "0x0"
}
```

Optional: "registrar" address of a contract containing [registry](#), used for UI

## Validator engines

The following Engine s achieve consensus by referring to a list of "validators" (referred to as authorities, when they are linked to physical entities). Validator set is a group of accounts which are allowed to participate in the consensus, they validate the transactions and blocks to later sign messages about them.

In the simplest case they can be specified at genesis using a simple "list" (as shown in the Authority Round and Tendermint sections):

```
"validators": {
  "list": [
    "0xd577a597b2742b498cb5cf0c26cdcd726d39e6e",
    "0x82a978b3f5962a5b0957d9ee9eef472ee55b42f1"
  ]
}
```

More information can be found on the [Validator Set](#) page.

## Aura

Simple and fast consensus algorithm, each validator gets an assigned time slot in which they can release a block. The time slots are determined by the system clock of each validator.  
[More details](#).

```
"engine": {
  "authorityRound": {
    "params": {
      "gasLimitBoundDivisor": "0x400",
      "stepDuration": "5",
      "validators" : {
        "list": [
          "0x37f93cfe411fa244b87ff257085ee360fca245e8",
          "0x610a3a37b98bf0c91c35442e489c246096739324"
        ]
      }
    }
  }
}
```

"gasLimitBoundDivisor" determines how fast the gas limit should adjust, most of the time 0x400 is fine "stepDuration" determines the lowest interval between blocks in seconds, too low might cause reorgs if the system clocks are not synchronized, too high leads to slow block issuance "validators" is the list of addresses of the entities which will be allowed to issue blocks Optional: "blockReward" determines the reward given to issuing authority "registrar" address of a contract containing [registry](#), used for UI

```
"seal": {
  "authorityRound": {
```

- pubsub
- signer
- trace
- shh

### Parity Chains

- Chain Specification
  - Pluggable Consensus
    - Aura
  - Validator Set
    - Validator contracts
- Proof of Work Chains
- Proof of Authority Chains
  - Demo PoA Tutorial
- Private Chains
  - Private Dev Chain
- Service transactions

### Technologies


- Automatic Updating
- Ethereum-IPFS API
- Light Client
  - Parity Light (PIP)
  - Light Ethereum (LES)
- Secret Store
- Warp Sync
  - Snapshot Format
- Whisper
  - Overview
  - Whisper PoC #2
  - Wire Protocol
- WebAssembly (WASM)

### Hacking on Parity

- Coding Guide
- Labeling
- Release Notes <sup>[+]</sup>
- Gitter <sup>[+]</sup>
- GitHub <sup>[+]</sup>
- Website <sup>[+]</sup>

#### Clone this wiki locally

<https://github.com/parity> 

 Clone in Desktop

\_\_\_\_\_

If malicious authorities are possible then `--force-sealing` is advised, this will ensure that the correct chain is the longest (making it BFT with finality of `authorities_count * step_duration` given no network partitions).

\_\_\_\_\_