Byron era

The Byron era represents the initial phase of the Cardano blockchain, focusing on setting up the network and establishing basic functionalities like transactions. The `byron-genesis.json` file includes parameters specific to this foundational era.

Key parameters in `byron-genesis.json`

```
{
    "avvmDistr": {
        "0cAZzmtB2CUjlsMULb30YcBrocvK7VhQjUk--MyY2_Q=": "8333333000000",
        ...
        "zE56EfVAvv0XekVyBDGh2Iz1QT6X38YxlcBYO20hqa0=": "1773135000000"
    },
    "blockVersionData": {
        "heavyDelThd": "300000000000",
        "maxBlockSize": "2000000",
        "maxHeaderSize": "2000000",
        "maxProposalSize": "700",
        "maxTxSize": "4096",
        "mpcThd": "20000000000000",
        "scriptVersion": 0,
        "slotDuration": "20000",
        "softforkRule": {
            "initThd": "900000000000000",
            "minThd": "600000000000000",
            "thdDecrement": "50000000000000"
        },
        "txFeePolicy": {
            "multiplier": "43946000000",
            "summand": "155381000000000"
        },
        "unlockStakeEpoch": "18446744073709551615",
        "updateImplicit": "10000",
        "updateProposalThd": "100000000000000",
        "updateVoteThd": "1000000000000"
    },
    "bootStakeholders": {
        "0d916567f96b6a65d204966e6aab5fbd242e56c321833f8ba5d607da": 1,
        "4bd1884f5ce2231be8623ecf5778a9112e26514205b39ff53529e735": 1,
        "5f53e01e1366aeda8811c2a630f0e037077a7b651093d2bdc4ef7200": 1,
        "ada3ab5c69b945c33c15ca110b444aa58906bf01fcfe55d8818d9c49": 1
    },
    "ftsSeed": "736b6f766f726f6461204767757264612062 0f726f64612070726f766f646120",
    "nonAvvmBalances": {},
    "protocolConsts": {
        "k": 2160,
        "protocolMagic": 60987900,
        "vssMaxTTL": 6,
        "vssMinTTL": 2
    },
    "heavyDelegation": {
        "c6c7fb227037a1719b9d871ea49b6039325aeb293915da7db9620e3f": {
            "cert":
"0f2ff9d1f071793dbd90fce8d47e96a09b594ee6dd00a4bac9664e4bd6af89830035ec921698b774c779eb1b6a2772d3d6ae37e6
30c06c75fbfecd02a6410a02",
            "delegatePk":
"I07+5HIUW0Lkq0poMMzGziuILwxSyTJ4lMSsoQz4HZunD5Xsx3MfBZWc4l+206lUOFaU+spdJg7MkmFKBoVV0g==",
            "issuerPk":
"AZzlT+pC5M4xG+GuRHJEXS+isikmVBorTqOyjRDGUQ+Lst9fn1zQn5OGKSXK29G2dn7R7JCugfcUebr0Dq7wPw==",
            "omega": 1
        }
    },
    "startTime": 1505621332,
    "vssCerts": {
        "0d916567f96b6a65d204966e6aab5fbd242e56c321833f8ba5d607da": {
            "expiryEpoch": 1,
            "signature":
"396fe505f4287f832fd26c1eba1843a49f3d23b64d664fb3c8a2f25c8de73ce6f2f4cf37ec7fa0fee7750d1d6c55e1b07e1018ce
0c6443bacdb01fb8e15f1a0f",
            "signingKey": "ohsV3RtEFD1jeOzKwNulmMRhBG2RLdFxPbcSGbkmJ+xd/2cOALSDahPlydFRjd15sH0PkPE
/zTvP4iN8wJr/hA==",
            "vssKey": "WCECtpe8B/5XPefEhgg7X5veUIYH/RRcvXbz6w7MIJBwWYU="
        },
        ...
    }
}
```

Section "avvmDistr" contains AVVM addresses with corresponding balances (lovelaces).

Section "blockVersionData" contains fundamental blockchain-related values:

- "heavyDelThd" - heavyweight delegation threshold,
- "maxBlockSize" - maximum size of block, in bytes,
- "maxHeaderSize" - maximum size of block's header, in bytes,
- "maxProposalSize" - maximum size of Cardano SL update proposal, in bytes,
- "maxTxSize" - maximum size of transaction, in bytes,
- "mpcThd" - threshold for participation in SSC (shared seed computation),
- "scriptVersion" - script version,
- "slotDuration" - slot duration, in microseconds,
- "softforkRule" - rules for softfork:
  - "initThd" - initial threshold, right after proposal is confirmed,
  - "minThd" - minimal threshold (i.e. threshold can't become less than this one),
  - "thdDecrement" - theshold will be decreased by this value after each epoch,
- "txFeePolicy" - transaction fee policy's values,
- "unlockStakeEpoch" - unlock stake epoch after which bootstrap era ends,
- "updateImplicit" - update implicit period, in slots,
- "updateProposalThd" - threshold for Cardano SL update proposal,
- "updateVoteThd" - threshold for voting for Cardano SL update proposal.

Section "bootStakeholders" contains bootstrap era stakeholders' identifiers (StakeholderIds) with corresponding weights.

Field "ftsSeed" contains seed value required for Follow-the-Satoshi mechanism (hex-encoded).

Section "protocolConsts" contains basic protocol constants:

- "k" - security parameter from the paper,
- "protocolMagic" - protocol magic section, described fully below:
  - "pm" - protocol magic number,
  - "requiresNetworkMagic" - either "NMMustBeNothing" or "NMMustBeJust",
- "vssMaxTTL" - VSS certificates maximum timeout to live (number of epochs),
- "vssMinTTL" - VSS certificates minimum timeout to live (number of epochs).

Section "protocolMagic" defines the protocol magic number. When the protocol magic is changed, all signatures become invalid. This is used to distinguish different networks.

- "pm" - is the protocol magic number, is included in serialized blocks and headers, and is part of signed data.
- "requiresNetworkMagic" - will be either "NMMustBeNothing" or "NMMustBeJust"

The "protocolMagic" value can either be an object with the two fields described above, or just a plain integer. In the latter case, "requires NetworkMagic" will take the default value of "NMMustBeJust".

The "requiresNetworkMagic" setting forms part of the genesis data, influences both the genesis data and the address format that the node uses. It can be either:

- "NMMustBeNothing" (mainnet setting) — means that the protocol magic value will *not* be included in the address format or transactions.
- "NMMustBeJust" (public testnet setting, the default) — means that the protocol magic value will be included in the address format and hence transactions.

Section "heavyDelegation" contains an information about heavyweight delegation:

- "cert" - delegation certificate,
- "delegatePk" - delegate's public key,
- "issuerPk" - stakeholder's (issuer's) public key,
- "omega" - index of epoch the block PSK is announced in, it is needed for replay attack prevention, can be set to arbitrary value in genesis.

Keys in heavyDelegation dictionary are StakeholderIds of certificates' issuers. They must be consistent with issuerPk values.

Field "startTime" is the timestamp of the 0-th slot. Ideally it should be few seconds later than the cluster actually starts. If it's significantly later, nodes won't be doing anything for a while. If it's slightly before the actual starts, some slots will be missed, but it shouldn't be critical as long as less than k slots are missed.

Section "vssCerts" contains VSS certificates:

- `"expiryEpoch"` - index of epoch until which (inclusive) the certificate is valid;
- `"signature"` - signature of certificate,
- `"signingKey"` - key used for signing,
- `"vssKey"` - VSS public key.

Keys in `vssCerts` dictionary are `StakeholderId`s of certificates' issuers. They must be consistent with `signingKey` values.