

---

## title: FAQ

### Can I connect to multiple relays with MEV-Boost?

Yes. You can add multiple relays comma-separated to the `-relays` flag, like this: `-relays https://relay1,https://relay2`

### How should I think about running `mev-boost` vs regular block construction?

The alternative to running `mev-boost` would be to get blocks from your local execution client, which can only get transactions from the public mempool and is not optimized for MEV extraction, meaning your rewards are likely to be less.

Or to implement your own builder, which is a complicated task and still leaves you with the problem of finding transactions that extract MEV and are not going through the public mempool.

### Can I check relay status when starting MEV-Boost?

Yes. The `-relay-check` flag can be called to check the status of relays, will return an error if none of the configured relays are responsive.

```
```bash
```

### Example `-relay-check` call:

```
./mev-boost -goerli -relays -relay-check  
https://0xaf4c6985aa049fb79dd37010438cfebeb0f2bd42b115b89dd678dab0670c1de38da0c4e9138c9290a398ecd9a0b3110@builder-  
relay-goerli.flashbots.net ```
```

### What is the difference between a beacon node, validator, and validator client?

A "**node**" or "**beacon node**" follows and reads the beacon chain. **validator clients (VC)** are specialized software that stake 32 ETH as collateral within Ethereum's **consensus layer** in order to participate in consensus duties. Validator clients are responsible for executing duties, such as proposing blocks and signing of attestations within Ethereum's proof-of-stake consensus mechanism, and will fully replace proof-of-work miners after [The Merge](#). **validators** most often refers to a validator client instance, but can refer to an individual that physically manages a validator client. This is an optional role in which a user posts ETH as collateral to a validator client in order to verify and attest to blocks, and seek financial returns in exchange for building and securing the protocol. This is similar to proof-of-work networks in which miners provide collateral in the form of hardware/hash-power to seek returns in exchange for building and securing the protocol. [Read more here](#).

### What prevents block proposers from stealing MEV from submitted builders' blocks?

Slashing penalties. A builder provides the proposer with a "blind" execution layer header to incorporate into a block, and a "value" amount which will be transferred to the proposer once they create a block using this header. Once a proposer signs a block with a header, they are bound to this choice (or risk being slashed due to equivocation). Should a validator wish to steal MEV from a builder, they would need to sign a second block including the exploited MEV, which would result in a slashing penalty. This penalty is significant enough to discourage this behavior, allowing the builder to reveal the blinded transactions without the possibility of the proposer tampering with them, or stealing MEV. [Read more about slashing events here](#).

### Does MEV-Boost have knowledge about the Beacon Chain?

No. MEV-boost has no knowledge about the beacon chain, such as which slots were proposed, etc.