This is a brainstorming to collect more specific details of what does it mean for a relay to be trust-worthy. Please reply with your requirements. This topic will be updated to reflect the main findings.

API

- Fully implement the Relay API Specification v1.
- Score blocks as the difference in the balance of the fee recipient account before and after the block execution

Operation

- Operated by a reputable organization well-known in the Ethereum community.
- Open source code.
- Transparent deployment process.
- Data available to verify source code behavior.
- Clear statement of any filtering or censoring.
- Wait for 16 epochs after the merge to propose the first blocks.
- Treats all builders equally.
- Take down the server when a high-severity bug is found, so validators switch to local building and don't miss slots.
- Publishes a post-mortem after every incident.
- In case of missing blocks, does not retroactively pay to the affected validators.

Performance

- Always reply to the requests from the current slot proposer.
- Reply to a header request in less than 1 second.
- Reply to a block request in less than 1 second.
- Always serves valid blocks.

Testing

- More than 90% unit test code coverage.
- Test relay running in Goerli.
- Pass a shadow fork test.

Security

- Supports the relay monitor.
- An indepedent security audit published.
- A bug bounty.
- Aware of the risks of vertical integration, and willing to step-down if their role becomes risky.

With the current design of proposer/builder separation that will go live at the merge, the relay is still a trusted mediator between proposers and builders.

In particular:

- it can steal MEV opportunities from builders

- it can lie on the amount to be paid to the proposer
- it can withhold the block body after it has been signed by the proposer
- it can deliver an invalid block to the proposer
- it can filter out any transactions it dislikes

So a relay has to promise to do its best effort to avoid those situations, it has to promise to be trust-worthy.

For builders, see How do I choose a good block builder?