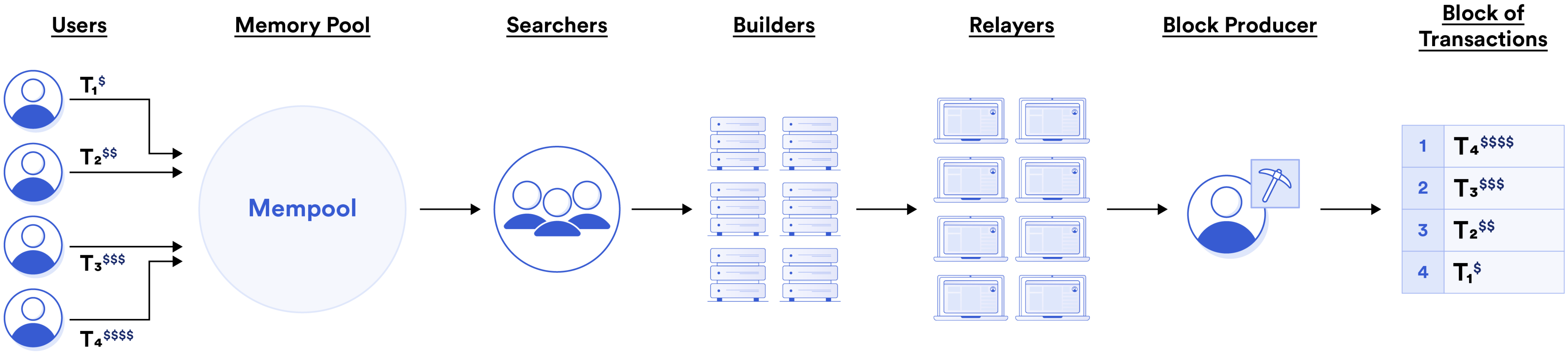
MEV DeepDive

Remind MEV(Maximal Extractable Value)



1	$T_4^{\$\$\$\$}$
2	$T_3^{\$\$\$}$
3	$T_2^{\$\$}$
4	$T_1^{\$}$

Flashbots

Flashbots is a research and development organization formed to mitigate the negative externalities posed by Maximal Extractable Value (MEV) to stateful blockchains, starting with Ethereum.

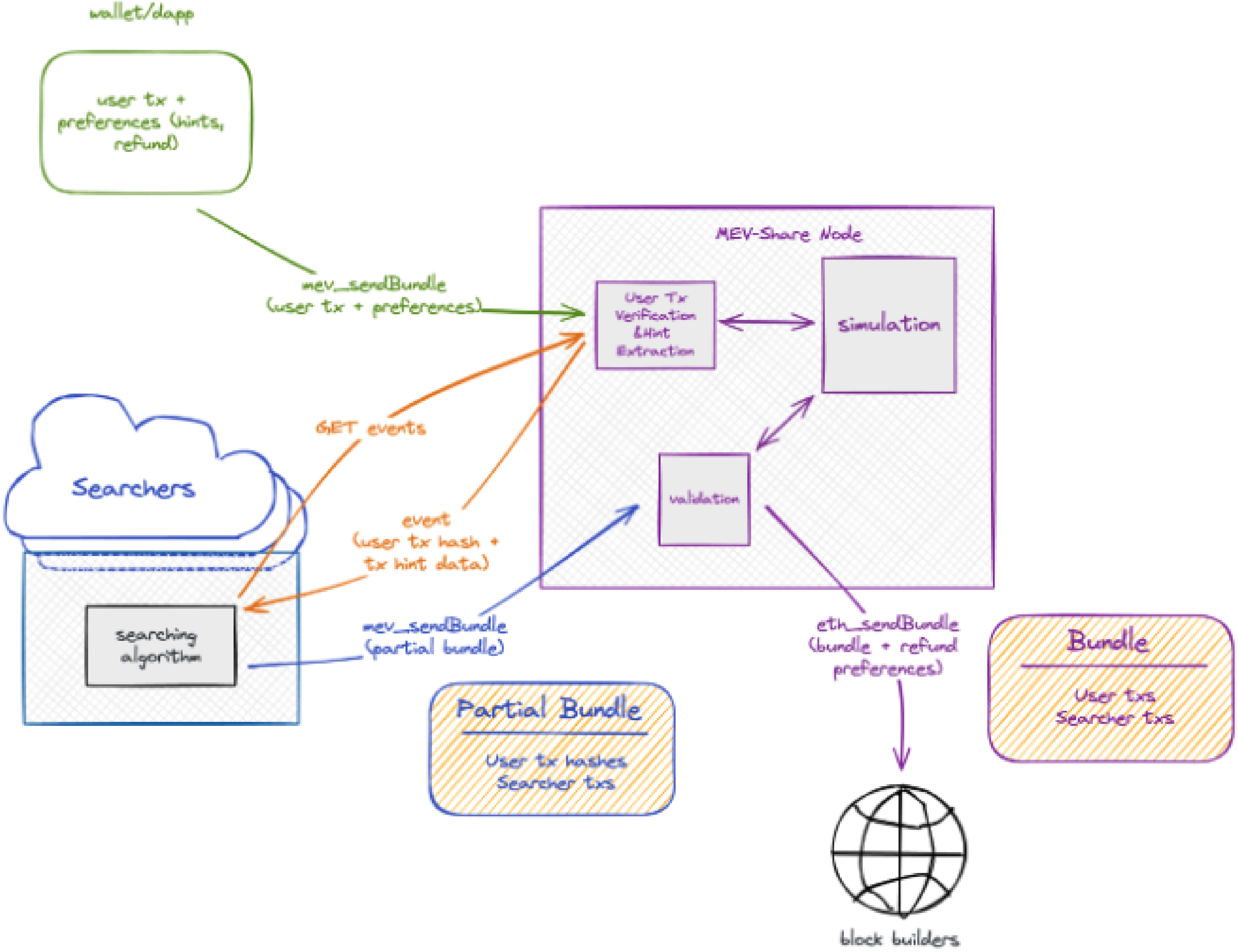
MEV-Share Protocol

Parties

1. Orderflow providers / sources (users, wallets, dapps)
2. Searchers
3. MEV-Share Nodes
4. Blockspace providers / proxies (builders, sequencers, validators, bundlers)

Minimum standardizable interfaces

1. How orderflow providers send orderflow and preferences (eg. privacy, redistribution) to MEV-Share Nodes
2. How MEV-Share Nodes share information about orderflow with searchers
3. How searchers send bids, orderflow, and preferences (eg. validity conditions) to MEV-Share Nodes
4. How MEV-Share Nodes send orderflow and preferences to blockspace providers
5. How value is redistributed to orderflow providers, blockspace providers, and MEV-Share Nodes


wallet/dapp
user tx + preferences (hints, refund)
mev_sendBundle (user tx + preferences)
MEV-Share Node
User Tx Verification &Hint Extraction
simulation
validation
GET events
Searchers
event (user tx hash + tx hint data)
searching algorithm
mev_sendBundle (partial bundle)
Partial Bundle
User tx hashes
Searcher txs
eth_sendBundle (bundle + refund preferences)
Bundle
User txs
Searcher txs
block builders

mev_sendBundle

- https://github.com/flashbots/mev-share/blob/main/specs/bundles/v0.1.md

eth_sendBundle

- https://github.com/flashbots/mev-share/blob/main/specs/bundles/refund-recipient.md

event-stream

- https://github.com/flashbots/mev-share/blob/main/specs/events/v0.1.md

Mev-Share-Node

Dependencies

- Redis : Used for hint streaming and priority queue
- Postgres : Used for storing bundles and historical hints

Supported Methods

`mev_sendBundle`

for submitting bundles to the relay

- in : bundle
- out : bundle hash
  detailed structure of bundles

`mev_simBundle`

has similar arguments to `mev_sendBundle`

Only fully matched bundles can be simulated

out : simulation result (no submit to the relay)

Node processes the bundle in the following manner:

1. **Validates** the structure of the bundle.
2. If the bundle is unmatched, i.e., if a hash element exists in the body, the node searches for a corresponding bundle with the same hash in the database. If a match is found and the target bundle can be matched, the hash is replaced with the bundle body. If not, the bundle is rejected.

3. Adds the bundle to the simulation queue.
4. Simulates the bundle when the block preceding its target block is reached.
5. If the `privacy.hint` of the bundle is set, relevant hints are extracted and added to the Redis channel. A separate service will then stream it over the SSE endpoint.
6. Sends the bundle to the builders specified in the `privacy.builders` field of the bundle. By default, the Flashbots builder is assumed.

Infra Setting

```
git clone https://github.com/flashbots/mev-share-node
cd mev-share-node

docker-compose up # start services: redis and postgres

# apply migration
for file in sql/*.sql; do psql "postgres://postgres:postgres@localhost:5432/postgres?sslmode=disable" -f $file; done

# get flashbots/builder, see /local-builder/devnet/README.md
cd ..
git clone https://github.com/flashbots/builder
cd builder
make
./local-builder/devnet/devnet run

# run node
make && ./build/node
```

builder's configurations

builder geth configuration

local run fail issue

실행을 위해서 설정 변경

from :

```
- --miner.algotype greedy \
```

to :

```
+ --builder.algotype greedy \
```

감사합니다.