MEV Pirates

@KernelKennethG

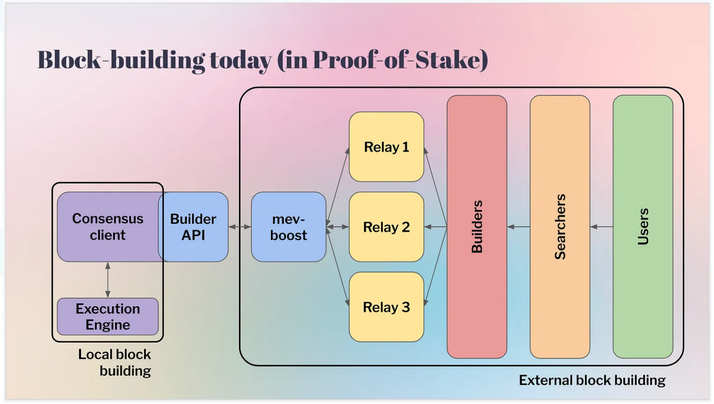
Proposer-Builder Separation (PBS)

What is PBS?

- A mechanism in Ethereum where the task of creating blocks (building) is separated from the task of proposing them to the network.

- Block builders assemble and order transactions

- Block proposers, typically validators, select the most profitable block (usually from relayers) to propose without initially knowing its contents (only knowing it’s header, the stateRoot).



Proposer-Builder Separation (PBS)

Why PBS?

- Makes it harder for builders to exclude transactions unfairly.

- Seen as a foundation for future upgrades like Danksharding, increasing Ethereum’s scalability.

- Validators can participate without needing advanced MEV extraction capabilities.

Maximal Extractable Value (MEV)

MEV represents the profit from reordering, inserting, or censoring transactions.

- **Front-running**: Placing a transaction ahead of a known upcoming transaction to profit from price movements.

- **Backrunning**: Inserting a transaction immediately after another to capitalize on the state change, such as arbitraging price differences post-trade.

- **Sandwich Attacks**: Surrounding a user's transaction with buy and sell orders to manipulate the price, often to the user's detriment.

The MEV Market

Relay	Payloads	Percent
relay.ultrasound.money	2,168	31.92 %
bloxroute.max-profit.blxrbdn.com	1,809	26.64 %
bloxroute.regulated.blxrbdn.com	1,535	22.60 %
titanrelay.xyz	865	12.74 %
boost-relay.flashbots.net	234	3.45 %
agnostic-relay.net	94	1.38 %
aestus.live	65	0.96 %
relay.edennetwork.io	18	0.27 %
mainnet-relay.securerpc.com	3	0.04 %

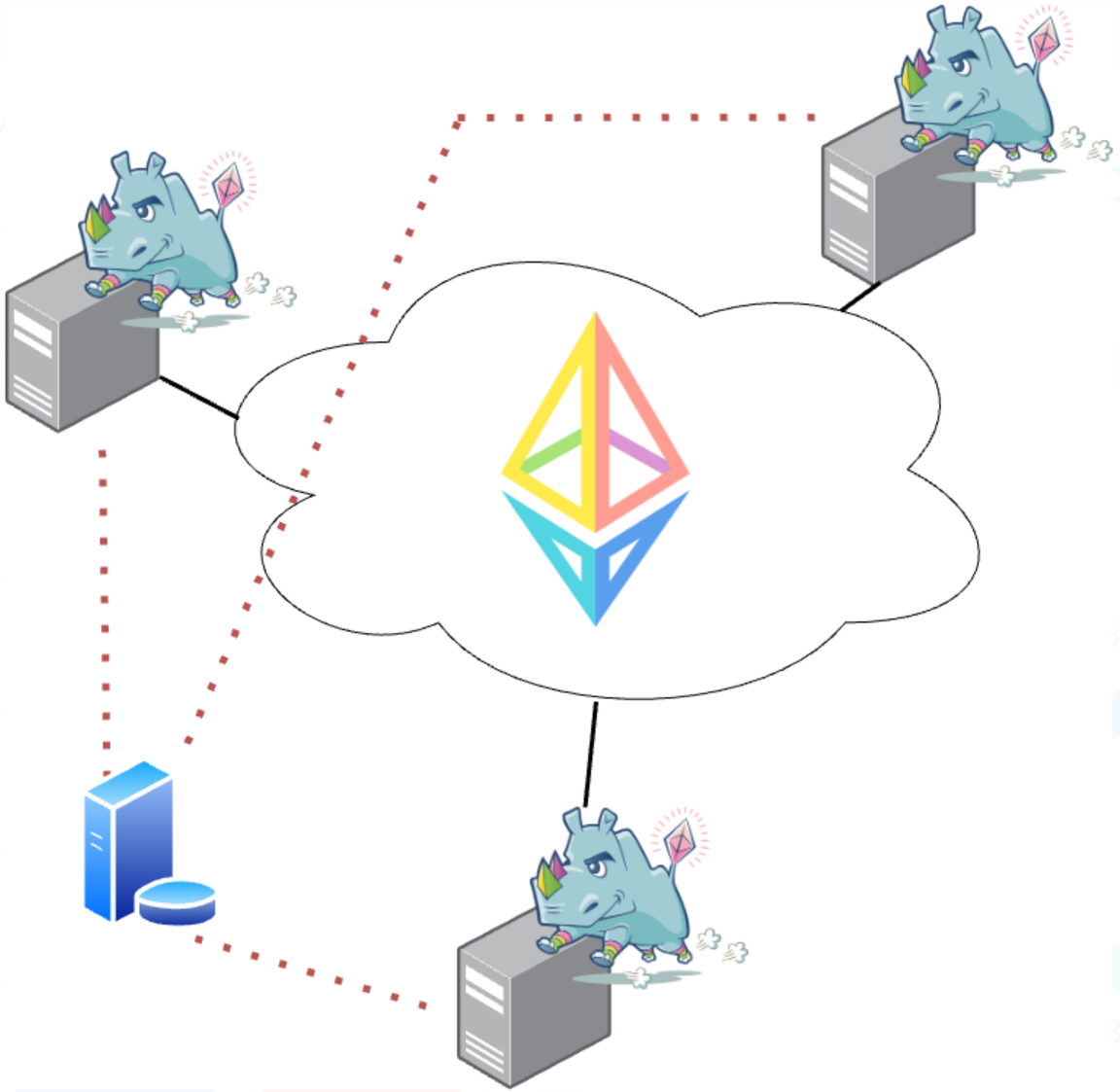
Builder (extra_data)	Blocks	Percent
beaverbuild.org	1,496	45.75 %
Titan (titanbuilder.xyz)	1,461	44.68 %
rsync-builder.xyz ⓘ	157	4.80 %
BuilderNet (Flashbots)	28	0.86 %
Builder+ www.btcs.com/builder	25	0.76 %
BuilderNet (Nethermind)	24	0.73 %
Quasar (quasar.win)	22	0.67 %
iobuilder	17	0.52 %
bobTheBuilder.xyz	14	0.43 %
BuilderNet (Beaver)	10	0.31 %
gigabuilder.io	4	0.12 %
BuildAI (https://buildai.net)	4	0.12 %
Ty For The Block	2	0.06 %
boba-builder.com	2	0.06 %
MevRefund - 1, Bad guys - 0	1	0.03 %
Powered by bloXroute	1	0.03 %
smithbot.xyz	1	0.03 %
GrizzlyBuilder	1	0.03 %

The MEV Market

Relay	Payloads	Percent
relay.ultrasound.money	2,168	31.92 %
bloxroute.max-profit.blxrbdn.com	1,809	26.64 %
bloxroute.regulated.blxrbdn.com	1,535	22.60 %
titanrelay.xyz	865	12.74 %
boost-relay.flashbots.net	234	3.45 %
agnostic-relay.net	94	1.38 %
aestus.live	65	0.96 %
relay.edennetwork.io	18	0.27 %
mainnet-relay.securerpc.com	3	0.04 %

Builder (extra_data)	Blocks	Percent
beaverbuild.org	1,496	45.75 %
Titan (titanbuilder.xyz)	1,461	44.68 %
rsync-builder.xyz ⓘ	157	4.80 %
BuilderNet (Flashbots)	28	0.86 %
Builder+ www.btcs.com/builder	25	0.76 %
BuilderNet (Nethermind)	24	0.73 %
Quasar (quasar.win)	22	0.67 %
iobuilder	17	0.52 %
bobTheBuilder.xyz	14	0.43 %
BuilderNet (Beaver)	10	0.31 %
gigabuilder.io	4	0.12 %
BuildAI (https://buildai.net)	4	0.12 %
Ty For The Block	2	0.06 %
boba-builder.com	2	0.06 %
MevRefund - 1, Bad guys - 0	1	0.03 %
Powered by bloXroute	1	0.03 %
smithbot.xyz	1	0.03 %
GrizzlyBuilder	1	0.03 %

System Architecture



Uniswap

```
const uniswapV4Decoder = (transaction) => {
    let uniswapV4Interface;
    if (ethers.Interface) {
        uniswapV4Interface = new ethers.Interface([{"inputs":[{"components":[{"internalType":"address","name":"permit
    } else {
        throw new Error("Ethers.js Interface not found - check ethers version");
      }
      let rawDecoded = uniswapV4Interface.parseTransaction({ data: transaction.input, value: transaction.value });
        // Add a custom replacer function to convert BigInt to strings
        console.log(JSON.stringify(rawDecoded, (key, value) =>
        typeof value === 'bigint' ? value.toString() : value
        ));

      return rawDecoded
}
```

```
async function getV2Slippage(v2RawDecoded) {
    // v2RawDecoded.fragment.inputs = amountIn|uint256, amountOutMin|uint256, path|address[], to|address, deac
    // v2RawDecoded.fragment.outputs = amounts | unit256[]
    let tradingPath = []
    let amountOutMin = 0n; // Use BigInt literal

    switch (v2RawDecoded.name) {
        case "swapExactTokensForTokens":
            tradingPath = v2RawDecoded.args[2]
            break;

        default:
            break;
    }

    let quote = tradingPath.length > 0 ?
        await getMultiQuote(tradingPath[0], tradingPath[tradingPath.length - 1], 3000, v2RawDecoded.args[0])
        : {};

    let slippageAbsolute = 0n; // Use BigInt literal
    let slippagePercentage = 0;
```

```
async function getMultiQuote(tokenInAddress, tokenOutAddress, fee = 3000, amountInWei) {
    // Common fee tiers used by Uniswap V3
    const feeTiers = [100, 500, 3000, 10000]; // 0.01%, 0.05%, 0.3%, 1%

    // If the specified fee is not in our standard tiers, add it to the beginning of the array
    if (!feeTiers.includes(fee)) {
        feeTiers.unshift(fee);
    }

    // Keep track of all errors for detailed reporting if all attempts fail
    const errors = [];

    // Try each fee tier
    for (const currentFee of feeTiers) {
        try {
            // Get token details
            let tokenInDetails = await getTokenDecimals(tokenInAddress, provider);
            console.log(`Attempting quote with fee tier: ${currentFee / 10000}%`);
            console.log("tokenInDetails:", tokenInDetails);

            // Check if provider is connected
            const blockNumber = await provider.getBlockNumber();
            console.log('Connected to network at block:', blockNumber);

            // Convert to BigInt to ensure proper handling
            const amountIn = BigInt(amountInWei.toString());

            // Format for readable output
            const amountInHuman = ethers.formatUnits(amountIn, tokenInDetails.decimals);
            console.log(`Amount in human-readable: ${amountInHuman} ${tokenInDetails.symbol}`);

            // Encode the path
            const path = ethers.solidityPacked(
                ['address', 'uint24', 'address'],
                [tokenInAddress, currentFee, tokenOutAddress]
            );

            console.log("Path encoded:", path);

            // Call the quoter
            console.log(`Calling quoter contract with fee tier: ${currentFee / 10000}%`);
            const [quotedAmountOut, , , ] = await quoterContract.quoteExactInput.staticCall(
                path,
                amountIn
            );
```

1inch Aggregator

```
const oneInchV6Decoder = (transaction) => {
    let oneInchV6Interface;
    if (ethers.Interface) {
        oneInchV6Interface = new ethers.Interface([{"inputs":[{"internalType":"contract
        IWETH","name":"weth","type":"address"}],"stateMutability":"nonpayable",
        "type":"constructor"},{"inputs":[],"name":"AdvanceEpochFailed","type":"error"},
        {"inputs":[],"name":"ArbitraryStaticCallFailed","type":"error"},{"inputs":[],
        "name":"BadCurveSwapSelector","type":"error"},{"inputs":[],"name":"BadPool",
        "type":"error"},{"inputs":[],"name":"BadSignature","type":"error"},{"inputs":[],
        "name":"BitInvalidatedOrder","type":"error"},{"inputs":[],
        "name":"ETHTransferFailed","type":"error"},{"inputs":[],"name":"ETHTransferFaile
        "type":"error"},{"inputs":[],"name":"EnforcedPause","type":"error"},{"inputs":[
        "name":"EpochManagerAndBitInvalidatorsAreIncompatible","type":"error"},{"inputs"
        [],"name":"EthDepositRejected","type":"error"},{"inputs":[],"name":"ExpectedPaus
        "type":"error"},{"inputs":[],"name":"InsufficientBalance","type":"error"},
        {"inputs":[],"name":"InvalidMsgValue","type":"error"},{"inputs":[],
        "name":"InvalidMsgValue","type":"error"},{"inputs":[],
        "name":"InvalidPermit2Transfer","type":"error"},{"inputs":[],
        "name":"InvalidShortString","type":"error"},{"inputs":[],"name":"InvalidatedOrde
        "type":"error"},{"inputs":[],"name":"MakingAmountTooLow","type":"error"},{"input
        [],"name":"MismatchArraysLengths","type":"error"},{"inputs":[],
        "name":"OrderExpired","type":"error"},{"inputs":[],
        "name":"OrderIsNotSuitableForMassInvalidation","type":"error"},{"inputs":
        [{"internalType":"address","name":"owner","type":"address"}],
        "name":"OwnableInvalidOwner","type":"error"},{"inputs":[{"internalType":"address
        "name":"account","type":"address"}],"name":"OwnableUnauthorizedAccount",
        "type":"error"},{"inputs":[],"name":"PartialFillNotAllowed","type":"error"},
```

```
async function getV6Slippage(v6RawDecoded) {
    // v3RawDecoded.fragment.inputs = executor|address,
    // desc.components|turple(srcToken|address, dstToken|address, srcReceiver|address,
    // data|bytes

    // v3RawDecoded.name = swap
    // v3RawDecoded.args = [executor [desc.components], data]...
    try {
        console.log("Processing 1Inch V6 swap");

        if (!v6RawDecoded || !v6RawDecoded.name || v6RawDecoded.name !== "swap" || !v6R
            console.log("Invalid v6RawDecoded or not a swap:", v6RawDecoded?.name);
            return {
                error: "Invalid transaction data or not a swap"
            };
        }

        // Extract trading path and amounts from desc tuple
        const desc = v6RawDecoded.args[1];

        if (!desc || !Array.isArray(desc) || desc.length < 6) {
            console.log("Invalid desc tuple:", desc?.length);
            return {
                error: "Invalid desc tuple"
            };
        }

        const srcToken = desc[0];  // srcToken
        const dstToken = desc[1];  // dstToken
        const amount = desc[4];    // amount
        const minReturnAmount = desc[5]; // minReturnAmount
```

WorldCoin Unique Personhood

Verify you're human to unlock MEV API access: **Verify with World ID**

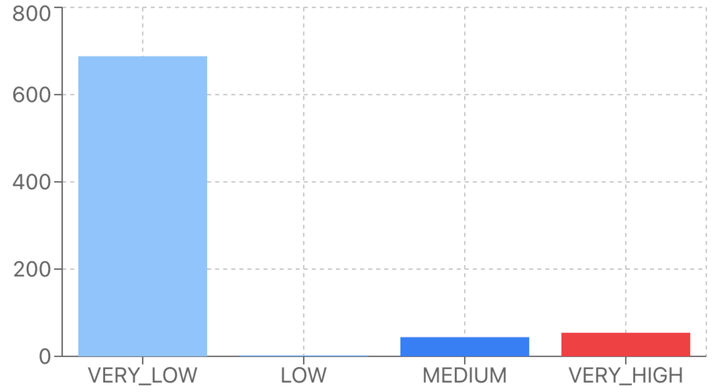
MEV Opportunity Dashboard

Real-time monitoring of MEV opportunities in Ethereum mempool

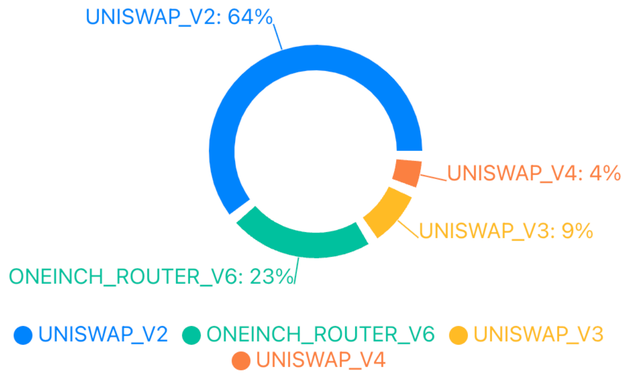
Auto-refresh: 30s Last updated: 04:33:44

Total Transactions	High-Value Opportunities	Total MEV Value	Average Slippage
20	20	1739190344795828736.0000 GWEI	89.68%
Past 24 hours	High or Very High slippage	Estimated extractable value	Across all transactions

Slippage Distribution



DEX Distribution



rollup-boost
by Worldcoin
+ Flashbots

MEV Pirates

Decentralise MEV,
Decentralise Block Building Today !