Protocol Engineer Take Home Test - Beacon Block Root on L2

Background

EIP-4788 introduces a "beacon block root" into the execution-layer block-header and EVM. This block root is an SSZ hash-tree-root of the consensus-layer contents of the previous consensus block. With the adoption of EIP-4399 in the Bedrock upgrade, the OP-Stack includes the `PREVRANDAO` of the L1. As a result, with EIP-4788, the L1 beacon block root is made available. The `parent_beacon_block_root` of the L1 origin is now embedded in the L2 block header.

The "Beacon roots contract" is deployed at Ecotone upgrade-time, or embedded at genesis if activated at genesis. The OP block state-transition process now includes the same special beacon-block-root EVM processing as L1 Ethereum.

Task

Assume the beacon root is not already in the L2 block header. Create a protocol that uses EIP-4788 and the canonical OP stack bridge to enable withdrawal root verification on the L2. Your solution should include:

1. A description of the protocol architecture
2. Code (in Solidity or any language that compiles to EVM bytecode) demonstrating the implementation of the protocol

Requirements

- The protocol should utilize the EIP-4788 beacon block root
- The protocol should use the canonical OP stack bridge for communication between L1 and L2

Bonus - one of the following:

1. **Built-in Beacon Root Verification**: Use the built-in beacon root on the L2 to verify the bridged root. Your solution should include:
    - Implementation demonstrating the built-in beacon root matches the bridged version
2. **Optimized Protocol**: Optimize the protocol to reduce gas costs and improve performance. Your solution should include:
    - A detailed description of the optimizations made

Submission Guidelines

Please submit your solution as a private github repository containing:

- Markdown document describing the protocol architecture
- Documented code (in Solidity or any other relevant programming language) demonstrating the implementation of the protocol

- Any additional context, files, or resources required to run the code

Evaluation Criteria

Your solution will be evaluated based on:

- Correctness and completeness of the protocol
- Quality and readability of the code
- Clarity and thoroughness of the documentation

Deadline

Please submit your solution within 7 days of receiving this take home test. Late submissions will not be accepted.