There are a few use cases for the GAS opcode which I'm going to label as "legitimate" in that they are used to implement functionality that we wish to preserve.

- A: Checking that there is enough gas before executing a sub-call for something like a meta transaction.
- B: Measuring how much gas a subcall used by checking before and after and taking the difference.

I propose a solution for A here: Removing gas observability in the context of EIP-3074 (AUTH & AUTHCALL) - #5 by pipermerriam

To solve B, we need an alternate way for contracts to measure how much gas a subcall takes. For this I propose a new opcode: GASMETER.

We introduce a new value that will be tracked for each frame of execution, CUMULATIVE_SUBCALL_GAS

. This value is initialized to zero at the beginning of each execution frame.

For each CALL style operation (CALL/CALLCODE/DELEGATECALL/STATICCALL), after execution has finished, the amount of gas consumed by the call is added to CUMULATIVE_SUBCALL_GAS.

We also introduce a new opcode GASMETER which pushes the current value of CUMULATIVE_SUBCALL_GAS onto the stack.