Some L1-to-L2 cross-chain operations–that is, operations initiated at L1 that cause things to happen at L2–need the L1 caller to pay a fee to deter malicious overuse of the operation. We use the obvious implementation, having the L1 method to do these operations be payable, and reverting if the correct payment isn't supplied.

But making these methods payable is a problem for people who want to invoke them from a method that needs to be non-payable for other reasons. This comes up in some token bridging use cases, for example.

An alternative would be to impose a cost on the caller by uselessly burning L1 gas. This seems like an unfriendly thing to do, because it destroys a valuable resource that the Ethereum community wants to use, and marginally raises the cost of L1 gas for everyone.

One approach would be to use a useless gas burn, but make it more costly than the ETH payment method so that, say, you could use a payable method and pay X gwei, or use a gas-burning method and uselessly burn 1.5X gwei worth of gas. That way, those who could use a payable method would be incented to use it, but those who just couldn't do payable would have a way to use the API.

To repeat, this is not something we would do for every transaction, or even for every L1-to-L2 functionality. We would just do it in one or two cases.

Does this seem reasonable? Is there a better way?