My idea surrounds the concept of onboarding users. I organise hackathons and give out an original NFT to every competitor. These users mostly have crypto, but some don't. I have to provide a faucet where they can request ETH (or the EVM-compatible blockchain the hackathon is sponsored by) so they have enough crypto to submit their voucher and receive their NFT. The issue I have is that this faucet needs to calculate the correct amount of ETH to dispense and then they need to submit their transaction with the correct fee to make it work.

It would be much easier if I could have some opcode or similar to describe that a certain contract invocation is paid for by the contract.

The way I envision this is that you perform a transaction giving exactly 0 gas. Then the first opcode in the function you call tells the EVM that the contract will pay for this invocation, possibly with a certain gas price and gas limit, like you do with regular transactions. If a transaction is sent with 0 gas and this opcode is not the first instruction, then the transaction is invalid.

If your transaction supplies too little gas, this may por may not supply the transaction with additional gas, but I leave that discussion to you.

I envision this as a modifier in Solidity, like this:

```
function functionName() public paysOwnGas(gasLimit, gasPrice) { }
```

What do you think?