Etherama smart contracts code review #2

By Anton Akentiev (ChainCloud company).

Not for distribution. This is an informal review, for discussion purposes only, with no warranties. I took a careful look at the contract but make no guarantees.

File I evaluated (at commit):
https://github.com/Goldmint/gm-dapp/commit/422f3e4d9728c0b0897a22cbb9beabab65e008f6/contracts/Etherama.sol

I did not evaluate anything else in the repo.

This is review #2. Please see review #1 here -
https://docs.google.com/document/d/1zKIPwRURNzdrYoRyKIyRCRj62_g60go8sPgJoolYABM/edit?usp=sharing

Detailed checklist

- ✓ Bad randomness
- ✓ Denial of Service
- ✓ Incorrect Interface
- ✓ Integer Overflow/Underflow/Bad casts
- ✓ Forced Ether Reception
- ✓ Missing Constructor
- ✓ Race Condition (frontrunning)
- ✓ Reentrancy (don’t use call.value!)
- ✓ Unchecked External Call (don’t use address.send!)
- ✓ Unprotected Function

Bugs/vulnerabilities found

1) [LOW severity]
   transferOwnership does not remove current admin from the list.

2) [LOW severity]
   Controller should be upgraded
   (1) in ONE function only!
   (2) should transfer ALL **rights** and **funds** to new controller.

3) [LOW severity]
   Math.
   Be aware that int/uint and uint/real conversions will **loose precision** and can **overflow**.

   This was double checked by the Goldmint team and they decided not to change the behaviour.

Recommendations

I recommend these steps to improve the code:

1) Migrate to truffle
2) Use standard OpenZeppelin contracts
3) Fix all solc warnings
4) Fix all linters warnings
5) Add command to calculate the code coverage and run it