ZNS - Zer0 Naming Service

Essential items:

- The URL of a repo containing the source code
  - https://github.com/zer0-os/ZNS
- Please invite the GitHub users **@david-oz** and **@tintinweb** to the repo
- The commit hash to be reviewed (OK if this is changing until March 1st)
  - ~~Version 1 (outdated)~~
    - ~~b05e503ea1ee87dbe62b1d58426aaa518068e395 [b05e503]~~
  - ~~Version 2~~
    - ~~Ab7d62a7b8d51b04abea895e241245674a640fc1~~
  - Version 3
    - bc5fea725f84ae4025f5fb1a9f03fb7e9926859a
- A list of the files in scope and out of scope
  - Smart Contracts
    - Registrar.sol
    - BasicController.sol
    - *And any relevant Interfaces*
- Documentation describing the intended functionality of the system
  - https://docs.google.com/document/d/1iDsJHNTSVHrgMt6HAHWRkKy1MQNSsrVsozs5Od2U1Bo/edit?usp=sharing
- A list of the key risks for us to ensure are mitigated
  - Theft of Domains
  - Unauthorized Creation of Domains
  - Attack vectors to destabilize the system
    - Denial of service
    - Prevention of normal/authorized usage
- Names and emails of the key personnel we will work with during the engagement
  - Zachary Nawar (zachary@dorg.tech)
  - Chris Dixon (chris@dorg.tech)

Additionally, the following items are non-essential, but are very helpful to us:

- Clear instructions for setting up the system and run the tests (usually this is in a README file)
  - Documented in the README: https://github.com/zer0-os/zNS/blob/master/README.md
- Any past audits
- Tooling output logs (e.g. MythX, Surya, etc.)
- Output generated by running the test suite
  - Please run `yarn test`
- Test coverage report

    - This is generated by `yarn coverage` and can be found in the `coverage.json` file at root.

Dili Notes:

- We revised issue “zNS - Avoid no-ops” removing the examples for setMetadataUri and setRoyaltyAmount as otherwise this would create problems when registering a subdomain in extended mode without setting a royalty (default=0; would throw because value not changed). The changes for the controller are good and we closed the issue. @zachary We would recommend removing the check that enforces that royalty needs to change.

Some observation and notes about potential simplifications that might be helpful to optimize the gas footprint of the application and validate the design:

- Domain nonce might not be needed at all as a bid can only be fulfilled for a domain hash once, since the domain nft can only be minted once. This should automatically invalidate all bids for a specific domain (unless there will be means to transfer them in the future)
- withdrawRequest might not be needed at all as the bid can only be fulfilled by the requestor and they can just decide to never fulfill instead. The domain owner can still approve another bid for the same id.
- Note: A field in the struct can be saved (gas) by checking request.requestor!=0x0 instead of request.valid.