

```
diff --git a/.github/workflows/pull_ci.yml b/.github/workflows/pull_ci.yml
deleted file mode 100644
index 1ddb847..0000000
--- a/.github/workflows/pull_ci.yml
+++ /dev/null
@@ -1,52 +0,0 @@
-name: PR CI
-on: [pull_request]
-
-jobs:
-  lint:
-    name: Lint
-    runs-on: ubuntu-18.04
-    steps:
-      - uses: actions/checkout@v2
-      - name: golangci-lint
-        uses: golangci/golangci-lint-action@v2
-        with:
-          version: latest
-          working-directory: .
-          args: --timeout 3m
-  test:
-    name: Golang Unit Tests v${{ matrix.go }} (${{ matrix.os }})
-    runs-on: ${{ matrix.os }}
-    strategy:
-      matrix:
-        go: ['1.16']
-        os: [ubuntu-20.04]
-    steps:
-      - uses: actions/checkout@v2
-      - uses: actions/setup-go@v1
-        with:
-          go-version: ${{ matrix.go }}
-      - run: go mod download
-        shell: bash
-      - run: ./scripts/build.sh evm
-        shell: bash
-      - run: ./scripts/build_test.sh
-        shell: bash
-  e2e:
-    name: Golang E2E Tests v${{ matrix.go }} (${{ matrix.os }})
-    runs-on: ${{ matrix.os }}
-    strategy:
-      matrix:
-        go: [ '1.16' ]
-        os: [ ubuntu-20.04 ]
-    steps:
-      - uses: actions/checkout@v2
-      - uses: actions/setup-go@v1
-        with:
-          go-version: ${{ matrix.go }}
-      - run: .github/workflows/run_e2e_tests.sh --parallelism 1 --client-id $KURTOSIS_CLIENT_ID --client-secret $KURTOSIS_CLIENT_SECRET
-        shell: bash
-      - env:
-          DOCKER_USERNAME: ${{ secrets.DOCKER_USERNAME }}
-          DOCKER_PASS: ${{ secrets.DOCKER_PASS }}
-          KURTOSIS_CLIENT_ID: ${{ secrets.KURTOSIS_CLIENT_ID }}
-          KURTOSIS_CLIENT_SECRET: ${{ secrets.KURTOSIS_CLIENT_SECRET }}
diff --git a/.github/workflows/push_ci.yml b/.github/workflows/push_ci.yml
deleted file mode 100644
index 1ac73f1..0000000
--- a/.github/workflows/push_ci.yml
+++ /dev/null
@@ -1,52 +0,0 @@
-name: Branch Push CI
-on: [push]
-
-jobs:
-  lint:
-    name: Lint
-    runs-on: ubuntu-18.04
-    steps:
-      - uses: actions/checkout@v2
-      - name: golangci-lint
-        uses: golangci/golangci-lint-action@v2
-        with:
-          version: latest
-          working-directory: .
-          args: --timeout 3m
-  test:
-    name: Golang Unit Tests v${{ matrix.go }} (${{ matrix.os }})
-    runs-on: ${{ matrix.os }}
-    strategy:
-      matrix:
-        go: ['1.16']
-        os: [macos-11.0, ubuntu-18.04, ubuntu-20.04, windows-latest]
-    steps:
-      - uses: actions/checkout@v2
-      - uses: actions/setup-go@v1
-        with:
-          go-version: ${{ matrix.go }}
-      - run: go mod download
-        shell: bash
-      - run: ./scripts/build.sh evm
-        shell: bash
-      - run: ./scripts/build_test.sh
-        shell: bash
-  e2e:
-    name: Golang E2E Tests v${{ matrix.go }} (${{ matrix.os }})
-    runs-on: ${{ matrix.os }}
-    strategy:
-      matrix:
-        go: [ '1.16' ]
-        os: [ ubuntu-20.04 ]
-    steps:
-      - uses: actions/checkout@v2
-      - uses: actions/setup-go@v1
-        with:
-          go-version: ${{ matrix.go }}
-      - run: .github/workflows/run_e2e_tests.sh --parallelism 1 --client-id $KURTOSIS_CLIENT_ID --client-secret $KURTOSIS_CLIENT_SECRET
-        shell: bash
-      - env:
-          DOCKER_USERNAME: ${{ secrets.DOCKER_USERNAME }}
-          DOCKER_PASS: ${{ secrets.DOCKER_PASS }}
-          KURTOSIS_CLIENT_ID: ${{ secrets.KURTOSIS_CLIENT_ID }}
-          KURTOSIS_CLIENT_SECRET: ${{ secrets.KURTOSIS_CLIENT_SECRET }}
diff --git a/.github/workflows/run_e2e_tests.sh b/.github/workflows/run_e2e_tests.sh
deleted file mode 100755
index 04f297f..0000000
--- a/.github/workflows/run_e2e_tests.sh
+++ /dev/null
@@ -1,86 +0,0 @@
-set -o errexit
-set -o nounset
-set -o pipefail
-
-# If Docker Credentials are not available fail
-if [[ -z ${DOCKER_USERNAME} ]]; then
-  echo "Skipping Tests because Docker Credentials were not present."
-  exit 1
-fi
-
-# Testing specific variables
```

```

avalanche_testing_repo="avaplatform/avalanche-testing"
avalanchego_repo="avaplatform/avalanchego"
-# Define default avalanche testing version to use
-avalanche_testing_image="${avalanche_testing_repo}:master"
-
-# Avalanche root directory
-CORETH_PATH=$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )
-
-# Load the versions
-source "$CORETH_PATH/scripts/versions.sh
-
-# Load the constants
-source "$CORETH_PATH/scripts/constants.sh
-
-# Login to docker
-echo "$DOCKER_PASS" | docker login --username "$DOCKER_USERNAME" --password-stdin
-
-# Checks available docker tags exist
-function docker_tag_exists() {
-    TOKEN=$(curl -s -H "Content-Type: application/json" -X POST -d '{"username": "${DOCKER_USERNAME}', "password": "${DOCKER_PASS}'}' https://hub.docker.com/v2/users/login/ | jq -r .token)
-    curl --silent -H "Authorization: JWT ${TOKEN}" -f --head -LL https://hub.docker.com/v2/repositories/$1/tags/$2/ > /dev/null
-}
-
-# Defines the avalanche-testing tag to use
-# Either uses the same tag as the current branch or uses the default
-if docker_tag_exists $avalanche_testing_repo $current_branch; then
-    echo "avalanche_testing_repo:$current_branch exists; using this image to run e2e tests"
-    avalanche_testing_image="avalanche_testing_repo:$current_branch"
-else
-    echo "avalanche_testing_repo $current_branch does NOT exist; using the default image to run e2e tests"
-fi
-
-echo "Using $avalanche_testing_image for e2e tests"
-
-# Defines the avalanchego tag to use
-# Either uses the same tag as the current branch or uses the default
-# Disable matchup in favor of explicit tag
-# TODO re-enable matchup when our workflow better supports it.
-# if docker_tag_exists $avalanchego_repo $current_branch; then
-#     echo "avalanchego_repo:$current_branch exists; using this avalanchego image to run e2e tests"
-#     AVALANCHE_VERSION=$current_branch
-# else
-#     echo "avalanchego_repo $current_branch does NOT exist; using the default image to run e2e tests"
-# fi
-
-# pulling the avalanche-testing image
-docker pull $avalanche_testing_image
-
-# Setting the build ID
-git_commit_id=$( git rev-list -1 HEAD )
-
-# Build current avalanchego
-source "$CORETH_PATH/scripts/build_image.sh
-
-# Target built version to use in avalanche-testing
-avalanche_image="avaplatform/avalanchego:$build_image_id"
-
-echo "Running Avalanche Image: ${avalanche_image}"
-echo "Running Avalanche Testing Image: ${avalanche_testing_image}"
-echo "Git Commit ID : ${git_commit_id}"
-
-# >>>>>> avalanche-testing custom parameters <<<<<<<<<<<<
-custom_params_json="{
-  \"isKurtosisCoreDevMode\": false,
-  \"avalanchegoImage\": \"${avalanche_image}\",
-  \"testBatch\": \"avalanchego\"
-}"
-# >>>>>> avalanche-testing custom parameters <<<<<<<<<<<<
-
-bash "$CORETH_PATH/.kurtosis/kurtosis.sh" \
-  --tests "C-Chain Bombard WorkFlow" \
-  --custom-params "${custom_params_json}" \
-  "${avalanche_testing_image}" \
-  $@
diff --git a/.gitignore b/.gitignore
index 87ff040..f3d6549 100644
--- a/.gitignore
+++ b/.gitignore
@@ -1,46 +1 @@
-./main
-
-*.log
-*.DS_Store
-
-awscli
-
-# Binaries for programs and plugins
-*.exe
-*.exe~
-*.dll
-*.so
-*.dylib
-*.profile
-
-# Test binary, build with `go test -c`
-*.test
-
-# Output of the go coverage tool, specifically when used with LiteIDE
-*.out
-
-# ignore GoLand metafiles directory
-idea/
-
-*.logs/
-
-*.vscode*
-
-*.pb*
-
-*.cpu[0-9]*
-*.mem[0-9]*
-*.lock[0-9]*
-*.profile
-*.swp
-*.aux
-*.fdb*
-*.fls
-*.gz
-*.pdf
-
-coverage
-
-bin/
-build/
+/build/
\ No newline at end of file
diff --git a/.kurtosis/kurtosis.sh b/.kurtosis/kurtosis.sh
deleted file mode 100755
index a3d1cd8..0000000
--- a/.kurtosis/kurtosis.sh
+++ /dev/null
@@ -1,226 +0,0 @@

```

```
-# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! WARNING !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
-#  
-# Do not modify this file! It will get overwritten when you upgrade Kurtosis!  
-#  
-# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! WARNING !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
-  
-set -euo pipefail  
-  
-  
-# =====  
-# Constants  
-# =====  
-# The directory where Kurtosis will store files it uses in between executions, e.g. access tokens  
-# Can make this configurable if needed  
-KURTOSIS_DIRPATH="${HOME}/.kurtosis"  
-  
-KURTOSIS_CORE_TAG="1.8"  
-KURTOSIS_DOCKERHUB_ORG="kurtosistech"  
-INITIALIZER_IMAGE="${KURTOSIS_DOCKERHUB_ORG}/kurtosis-core_initializer:${KURTOSIS_CORE_TAG}"  
-API_IMAGE="${KURTOSIS_DOCKERHUB_ORG}/kurtosis-core_api:${KURTOSIS_CORE_TAG}"  
-  
-POSITIONAL_ARG_DEFINITION_FRAGMENTS=2  
-  
-  
-# =====  
-# Arg Parsing  
-# =====  
-function print_help_and_exit() {  
-    echo ""  
-    echo "${basename "$0"} [--custom-params custom_params_json] [--client-id client_id] [--client-secret client_secret] [--help] [--kurtosis-log-level kurtosis_log_level] [--list] [--parallelism parallelism] [--test-suite-image test_suite_image]"  
-    echo ""  
-    echo " --custom-params custom_params_json      JSON string containing arbitrary data that will be passed as-is to your testsuite, so it can modify its behaviour based on input (default: {})"  
-    echo " --client-id client_id                    An OAuth client ID which is needed for running Kurtosis in CI, and should be left empty when running Kurtosis on a local machine"  
-    echo " --client-secret client_secret            An OAuth client secret which is needed for running Kurtosis in CI, and should be left empty when running Kurtosis on a local machine"  
-    echo " --help                                  Display this message"  
-    echo " --kurtosis-log-level kurtosis_log_level The log level that all output generated by the Kurtosis framework itself should log at (panic|fatal|error|warning|info|debug|trace) (default: info)"  
-    echo " --list                                  Rather than running the tests, lists the tests available to run"  
-    echo " --parallelism parallelism               The number of texts to execute in parallel (default: 4)"  
-    echo " --tests test_names                      List of test names to run, separated by ',' (default or empty: run all tests)"  
-    echo " --test-suite-log-level test_suite_log_level A string that will be passed as-is to the test suite container to indicate what log level the test suite container should output at; this string must be one of the following values: panic, fatal, error, warning, info, debug, trace"  
-    echo " --test-suite-image test_suite_image     The Docker image containing the testsuite to execute"  
-  
-    echo ""  
-    exit 1 # Exit with an error code, so that if it gets accidentally called in parent scripts/CI it fails loudly  
-}  
-  
-# =====  
-# Arg Parsing  
-# =====  
-client_id=""  
-client_secret=""  
-custom_params_json=""  
-do_list="false"  
-kurtosis_log_level="info"  
-parallelism="4"  
-show_help="false"  
-test_names=""  
-test_suite_image=""  
-test_suite_log_level="info"  
-  
-  
-POSITIONAL=(  
-    while [ $# -gt 0 ]; do  
-        key="${1}"  
-        case "${key}" in  
-  
-            --custom-params)  
-                custom_params_json="${2}"  
-                shift # Shift to clear out the flag  
-                shift # Shift again to clear out the value  
-                ;;  
-  
-            --client-id)  
-                client_id="${2}"  
-                shift # Shift to clear out the flag  
-                shift # Shift again to clear out the value  
-                ;;  
-  
-            --client-secret)  
-                client_secret="${2}"  
-                shift # Shift to clear out the flag  
-                shift # Shift again to clear out the value  
-                ;;  
-  
-            --help)  
-                show_help="true"  
-                shift # Shift to clear out the flag  
-                ;;  
-  
-            --kurtosis-log-level)  
-                kurtosis_log_level="${2}"  
-                shift # Shift to clear out the flag  
-                shift # Shift again to clear out the value  
-                ;;  
-  
-            --list)  
-                do_list="true"  
-                shift # Shift to clear out the flag  
-                ;;  
-  
-            --parallelism)  
-                parallelism="${2}"  
-                shift # Shift to clear out the flag  
-                shift # Shift again to clear out the value  
-                ;;  
-  
-            --tests)  
-                test_names="${2}"  
-                shift # Shift to clear out the flag  
-                shift # Shift again to clear out the value  
-                ;;  
-  
-            --test-suite-log-level)  
-                test_suite_log_level="${2}"  
-                shift # Shift to clear out the flag  
-                shift # Shift again to clear out the value  
-                ;;  
-  
-            *)  
-                echo "ERROR: Unrecognized flag '${key}'" >&2
```

```

exit 1
;;
*)
    POSITIONAL+=("${1}")
    shift
    ;;
esac
done
-
-
-if "${show_help}"; then
-     print_help_and_exit
-fi
-
-# Restore positional parameters and assign them to variables
-set -- "${POSITIONAL[@]}"
-test_suite_image="${1:-}"
-
-
-
-
-# =====
-#                               Arg Validation
-# =====
-if [ "${#}" -ne 1 ]; then
-     echo "ERROR: Expected 1 positional variables but got ${#}" >&2
-     print_help_and_exit
-fi
-
-if [ -z "$test_suite_image" ]; then
-     echo "ERROR: Variable 'test_suite_image' cannot be empty" >&2
-     exit 1
-fi
-
-
-
-# =====
-#                               Main Logic
-# =====# Because Kurtosis X.Y.Z tags are normalized to X.Y so that minor patch updates are transparently
-# used, we need to pull the latest API & initializer images
-echo "Pulling latest versions of API & initializer image..."
-if ! docker pull "${INITIALIZER_IMAGE}"; then
-     echo "WARN: An error occurred pulling the latest version of the initializer image (${INITIALIZER_IMAGE}); you may be running an out-of-date version" >&2
-else
-     echo "Successfully pulled latest version of initializer image"
-fi
-if ! docker pull "${API_IMAGE}"; then
-     echo "WARN: An error occurred pulling the latest version of the API image (${API_IMAGE}); you may be running an out-of-date version" >&2
-else
-     echo "Successfully pulled latest version of API image"
-fi
-
-# Kurtosis needs a Docker volume to store its execution data in
-# To learn more about volumes, see: https://docs.docker.com/storage/volumes/
-sanitized_image=$(echo "${test_suite_image}" | sed 's/[^a-zA-Z0-9_]/_/g')
-suite_execution_volume=$(date +%Y-%m-%dT%H.%M.%S)${sanitized_image}
-if ! docker volume create "${suite_execution_volume}" > /dev/null; then
-     echo "ERROR: Failed to create a Docker volume to store the execution files in" >&2
-     exit 1
-fi
-
-if ! mkdir -p "${KURTOSIS_DIRPATH}"; then
-     echo "ERROR: Failed to create the Kurtosis directory at '${KURTOSIS_DIRPATH}'" >&2
-     exit 1
-fi
-
-docker run \
-    `# The Kurtosis initializer runs inside a Docker container, but needs to access to the Docker engine; this is how to do it` \
-    `# For more info, see the bottom of: http://jpetazzo.github.io/2015/09/03/do-not-use-docker-in-docker-for-ci/` \
-    --mount "type=bind,source=/var/run/docker.sock,target=/var/run/docker.sock" \
-    \
-    `# Because the Kurtosis initializer runs inside Docker but needs to persist & read files on the host filesystem between execution,` \
-    `# the container expects the Kurtosis directory to be bind-mounted at the special "/kurtosis" path` \
-    --mount "type=bind,source=${KURTOSIS_DIRPATH},target=/kurtosis" \
-    \
-    `# The Kurtosis initializer image requires the volume for storing suite execution data to be mounted at the special "/suite-execution" path` \
-    --mount "type=volume,source=${suite_execution_volume},target=/suite-execution" \
-    \
-    `# Keep these sorted alphabetically` \
-    --env CLIENT_ID="${client_id}" \
-    --env CLIENT_SECRET="${client_secret}" \
-    --env CUSTOM_PARAMS_JSON="${custom_params_json}" \
-    --env DO_LIST="${do_list}" \
-    --env KURTOSIS_API_IMAGE="${API_IMAGE}" \
-    --env KURTOSIS_LOG_LEVEL="${kurtosis_log_level}" \
-    --env PARALLELISM="${parallelism}" \
-    --env SUITE_EXECUTION_VOLUME="${suite_execution_volume}" \
-    --env TEST_NAMES="${test_names}" \
-    --env TEST_SUITE_IMAGE="${test_suite_image}" \
-    --env TEST_SUITE_LOG_LEVEL="${test_suite_log_level}" \
-    \
-    "${INITIALIZER_IMAGE}"
diff --git a/accounts/abi/bind/auth.go b/accounts/abi/bind/auth.go
index 859083c..46b0dc2 100644
--- a/accounts/abi/bind/auth.go
+++ b/accounts/abi/bind/auth.go
@@ -34,13 +34,13 @@ import (
    "io/ioutil"
    "math/big"

-    "github.com/ava-labs/coreth/accounts"
-    "github.com/ava-labs/coreth/accounts/external"
-    "github.com/ava-labs/coreth/accounts/keystore"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/crypto"
-    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts"
+    "github.com/flare-foundation/coreth/accounts/external"
+    "github.com/flare-foundation/coreth/accounts/keystore"
+    "github.com/flare-foundation/coreth/core/types"
)

// ErrNoChainID is returned whenever the user failed to specify a chain id.
diff --git a/accounts/abi/bind/backend.go b/accounts/abi/bind/backend.go
index 29a4b3c..393097a 100644
--- a/accounts/abi/bind/backend.go
+++ b/accounts/abi/bind/backend.go
@@ -31,9 +31,9 @@ import (
    "errors"
    "math/big"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
-    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
+
)

var (
diff --git a/accounts/abi/bind/backends/simulated.go b/accounts/abi/bind/backends/simulated.go
index 4860cb3..174083a 100644
--- a/accounts/abi/bind/backends/simulated.go
+++ b/accounts/abi/bind/backends/simulated.go

```

```

@@ -34,27 +34,27 @@ import (
    "sync"
    "time"

-    "github.com/ava-labs/coreth/eth"
-
-    "github.com/ava-labs/coreth/accounts/abi"
-    "github.com/ava-labs/coreth/accounts/abi/bind"
-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/bloombits"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/eth/filters"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/interfaces"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/flare-foundation/coreth/eth"
+
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/hexutil"
+    "github.com/ethereum/go-ethereum/common/math"
+    "github.com/ethereum/go-ethereum/event"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts/abi"
+    "github.com/flare-foundation/coreth/accounts/abi/bind"
+    "github.com/flare-foundation/coreth/consensus/dummy"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/bloombits"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/eth/filters"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/interfaces"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
)

// Verify that SimulatedBackend implements required interfaces
diff --git a/accounts/abi/bind/backends/simulated_test.go b/accounts/abi/bind/backends/simulated_test.go
index d979e79..9acb2bd 100644
--- a/accounts/abi/bind/backends/simulated_test.go
+++ b/accounts/abi/bind/backends/simulated_test.go
@@ -37,14 +37,14 @@ import (
    "testing"
    "time"

-    "github.com/ava-labs/coreth/accounts/abi"
-    "github.com/ava-labs/coreth/accounts/abi/bind"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/accounts/abi"
+    "github.com/flare-foundation/coreth/accounts/abi/bind"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
+    "github.com/flare-foundation/coreth/params"
)

func TestSimulatedBackend(t *testing.T) {
diff --git a/accounts/abi/bind/base.go b/accounts/abi/bind/base.go
index 58ac1ea..ce50ed9 100644
--- a/accounts/abi/bind/base.go
+++ b/accounts/abi/bind/base.go
@@ -34,12 +34,12 @@ import (
    "strings"
    "sync"

-    "github.com/ava-labs/coreth/accounts/abi"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/ethereum/go-ethereum/event"
+    "github.com/flare-foundation/coreth/accounts/abi"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
)

// SignerFn is a signer function callback when a contract requires a method to
diff --git a/accounts/abi/bind/base_test.go b/accounts/abi/bind/base_test.go
index d798851..dfaf441 100644
--- a/accounts/abi/bind/base_test.go
+++ b/accounts/abi/bind/base_test.go
@@ -33,14 +33,14 @@ import (
    "strings"
    "testing"

-    "github.com/ava-labs/coreth/accounts/abi"
-    "github.com/ava-labs/coreth/accounts/abi/bind"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/hexutil"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/accounts/abi"
+    "github.com/flare-foundation/coreth/accounts/abi/bind"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
+    "github.com/stretchr/testify/assert"
)

diff --git a/accounts/abi/bind/bind.go b/accounts/abi/bind/bind.go
index 22e1a8c..27ef476 100644
--- a/accounts/abi/bind/bind.go
+++ b/accounts/abi/bind/bind.go
@@ -40,8 +40,8 @@ import (
    "text/template"
    "unicode"

-    "github.com/ava-labs/coreth/accounts/abi"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts/abi"
)

// Lang is a target programming language selector to generate bindings for.
diff --git a/accounts/abi/bind/template.go b/accounts/abi/bind/template.go
index f66f175..9722fd9 100644
--- a/accounts/abi/bind/template.go
+++ b/accounts/abi/bind/template.go
@@ -26,7 +26,7 @@

```

package bind

```

-import "github.com/ava-labs/coreth/accounts/abi"
+import "github.com/flare-foundation/coreth/accounts/abi"

// tmplData is the data structure required to fill the binding template.
type tmplData struct {
diff --git a/accounts/abi/bind/util.go b/accounts/abi/bind/util.go
index 3fb6b5c..f3e509c 100644
--- a/accounts/abi/bind/util.go
+++ b/accounts/abi/bind/util.go
@@ -31,9 +31,9 @@ import (
    "errors"
    "time"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/core/types"
)

// WaitMined waits for tx to be mined on the blockchain.
diff --git a/accounts/abi/bind/util_test.go b/accounts/abi/bind/util_test.go
index aab4ae8..2b62390 100644
--- a/accounts/abi/bind/util_test.go
+++ b/accounts/abi/bind/util_test.go
@@ -33,12 +33,12 @@ import (
    "testing"
    "time"

-    "github.com/ava-labs/coreth/accounts/abi/bind"
-    "github.com/ava-labs/coreth/accounts/abi/bind/backends"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/accounts/abi/bind"
+    "github.com/flare-foundation/coreth/accounts/abi/bind/backends"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/types"
)

var testKey, _ = crypto.HexToECDSA("b71c71a67e1177ad4e901695e1b4b9ee17ae16c6668d313eac2f96dbcda3f291")
diff --git a/accounts/accounts.go b/accounts/accounts.go
index dd7df0b..f8af837 100644
--- a/accounts/accounts.go
+++ b/accounts/accounts.go
@@ -31,10 +31,10 @@ import (
    "fmt"
    "math/big"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/event"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
+    "golang.org/x/crypto/sha3"
)

diff --git a/accounts/external/backend.go b/accounts/external/backend.go
index 558700f..1c04766 100644
--- a/accounts/external/backend.go
+++ b/accounts/external/backend.go
@@ -31,15 +31,15 @@ import (
    "math/big"
    "sync"

-    "github.com/ava-labs/coreth/accounts"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
-    "github.com/ava-labs/coreth/rpc"
-    "github.com/ava-labs/coreth/signer/core/apitypes"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/common/hexutil"
-    "github.com/ethereum/go-ethereum/event"
-    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
+    "github.com/flare-foundation/coreth/rpc"
+    "github.com/flare-foundation/coreth/signer/core/apitypes"
)

type ExternalBackend struct {
diff --git a/accounts/keystore/account_cache.go b/accounts/keystore/account_cache.go
index 4c35aa7..d35d11f 100644
--- a/accounts/keystore/account_cache.go
+++ b/accounts/keystore/account_cache.go
@@ -37,10 +37,10 @@ import (
    "sync"
    "time"

-    "github.com/ava-labs/coreth/accounts"
-    mapset "github.com/deckarep/golang-set"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts"
)

// Minimum amount of time between cache reloads. This limit applies if the platform does
diff --git a/accounts/keystore/account_cache_test.go b/accounts/keystore/account_cache_test.go
index ad7ab51..01671df 100644
--- a/accounts/keystore/account_cache_test.go
+++ b/accounts/keystore/account_cache_test.go
@@ -36,10 +36,10 @@ import (
    "testing"
    "time"

-    "github.com/ava-labs/coreth/accounts"
-    "github.com/cespare/cp"
-    "github.com/davecgh/go-spew/spew"
-    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/accounts"
)

var (
diff --git a/accounts/keystore/key.go b/accounts/keystore/key.go
index 71402d3..bb7a543 100644
--- a/accounts/keystore/key.go
+++ b/accounts/keystore/key.go
@@ -39,9 +39,9 @@ import (
    "strings"
    "time"

-    "github.com/ava-labs/coreth/accounts"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/accounts"
+    "github.com/google/uuid"
)

diff --git a/accounts/keystore/keystore.go b/accounts/keystore/keystore.go
index ff82ef8..e63e891 100644
--- a/accounts/keystore/keystore.go
+++ b/accounts/keystore/keystore.go

```

```

@@ -42,11 +42,11 @@ import (
    "sync"
    "time"

-    "github.com/ava-labs/coreth/accounts"
-    "github.com/ava-labs/coreth/core/types"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/crypto"
    "github.com/ethereum/go-ethereum/event"
+    "github.com/flare-foundation/coreth/accounts"
+    "github.com/flare-foundation/coreth/core/types"
)

var (
diff --git a/accounts/keystore/keystore_test.go b/accounts/keystore/keystore_test.go
index 651ab70..c04370d 100644
--- a/accounts/keystore/keystore_test.go
+++ b/accounts/keystore/keystore_test.go
@@ -38,10 +38,10 @@ import (
    "testing"
    "time"

-    "github.com/ava-labs/coreth/accounts"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/crypto"
    "github.com/ethereum/go-ethereum/event"
+    "github.com/flare-foundation/coreth/accounts"
)

var testSigData = make([]byte, 32)
diff --git a/accounts/keystore/passphrase.go b/accounts/keystore/passphrase.go
index 5da41f6..abe81df 100644
--- a/accounts/keystore/passphrase.go
+++ b/accounts/keystore/passphrase.go
@@ -48,10 +48,10 @@ import (
    "os"
    "path/filepath"

-    "github.com/ava-labs/coreth/accounts"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/common/math"
    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/accounts"
    "github.com/google/uuid"
    "golang.org/x/crypto/pbkdf2"
    "golang.org/x/crypto/scrypt"
diff --git a/accounts/keystore/presale.go b/accounts/keystore/presale.go
index 1dfbd9c..49cb183 100644
--- a/accounts/keystore/presale.go
+++ b/accounts/keystore/presale.go
@@ -35,8 +35,8 @@ import (
    "errors"
    "fmt"

-    "github.com/ava-labs/coreth/accounts"
    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/accounts"
    "github.com/google/uuid"
    "golang.org/x/crypto/pbkdf2"
)

diff --git a/accounts/keystore/wallet.go b/accounts/keystore/wallet.go
index 7193526..084c519 100644
--- a/accounts/keystore/wallet.go
+++ b/accounts/keystore/wallet.go
@@ -29,10 +29,10 @@ package keystore
import (
    "math/big"

-    "github.com/ava-labs/coreth/accounts"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/accounts"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
)

// keystoreWallet implements the accounts.Wallet interface for the original
diff --git a/accounts/scwallet/hub.go b/accounts/scwallet/hub.go
index 7a630fa..c037ceb 100644
--- a/accounts/scwallet/hub.go
+++ b/accounts/scwallet/hub.go
@@ -51,10 +51,10 @@ import (
    "sync"
    "time"

-    "github.com/ava-labs/coreth/accounts"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/event"
    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts"
+    pcsc "github.com/gballet/go-libpcsclite"
)

diff --git a/accounts/scwallet/wallet.go b/accounts/scwallet/wallet.go
index fcecc10..bdd7b3e 100644
--- a/accounts/scwallet/wallet.go
+++ b/accounts/scwallet/wallet.go
@@ -43,12 +43,12 @@ import (
    "sync"
    "time"

-    "github.com/ava-labs/coreth/accounts"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/crypto"
    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"
    pcsc "github.com/gballet/go-libpcsclite"
    "github.com/status-im/keycard-go/derivationpath"
)

diff --git a/chain/chain_test.go b/chain/chain_test.go
index 680e91b..2c2d09d 100644
--- a/chain/chain_test.go
+++ b/chain/chain_test.go
@@ -9,20 +9,20 @@ import (
    "math/rand"
    "testing"

-    "github.com/ava-labs/coreth/accounts/keystore"
-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/eth"
-    "github.com/ava-labs/coreth/eth/ethconfig"
-    "github.com/ava-labs/coreth/node"
-    "github.com/ava-labs/coreth/params"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/common/hexutil"

```

```

        "github.com/ethereum/go-ethereum/log"
        "github.com/ethereum/go-ethereum/rlp"
+       "github.com/flare-foundation/coreth/accounts/keystore"
+       "github.com/flare-foundation/coreth/consensus/dummy"
+       "github.com/flare-foundation/coreth/core"
+       "github.com/flare-foundation/coreth/core/rawdb"
+       "github.com/flare-foundation/coreth/core/state"
+       "github.com/flare-foundation/coreth/core/types"
+       "github.com/flare-foundation/coreth/eth"
+       "github.com/flare-foundation/coreth/eth/ethconfig"
+       "github.com/flare-foundation/coreth/node"
+       "github.com/flare-foundation/coreth/params"
    )

    type testChain struct {
diff --git a/chain/coreth.go b/chain/coreth.go
index 34543a9..2402ad3 100644
--- a/chain/coreth.go
+++ b/chain/coreth.go
@@ -7,15 +7,15 @@ import (
        "fmt"
        "time"

-       "github.com/ava-labs/coreth/consensus/dummy"
-       "github.com/ava-labs/coreth/core"
-       "github.com/ava-labs/coreth/core/state"
-       "github.com/ava-labs/coreth/core/types"
-       "github.com/ava-labs/coreth/eth"
-       "github.com/ava-labs/coreth/ethdb"
-       "github.com/ava-labs/coreth/node"
-       "github.com/ava-labs/coreth/rpc"
+       "github.com/ethereum/go-ethereum/common"
+       "github.com/flare-foundation/coreth/consensus/dummy"
+       "github.com/flare-foundation/coreth/core"
+       "github.com/flare-foundation/coreth/core/state"
+       "github.com/flare-foundation/coreth/core/types"
+       "github.com/flare-foundation/coreth/eth"
+       "github.com/flare-foundation/coreth/ethdb"
+       "github.com/flare-foundation/coreth/node"
+       "github.com/flare-foundation/coreth/rpc"
    )

    var (
diff --git a/chain/counter_test.go b/chain/counter_test.go
index 7209ac5..5d1037b 100644
--- a/chain/counter_test.go
+++ b/chain/counter_test.go
@@ -15,8 +15,8 @@ import (
        "testing"

-       "github.com/ava-labs/coreth/core/types"
        "github.com/ethereum/go-ethereum/common"
+       "github.com/flare-foundation/coreth/core/types"
    )

diff --git a/chain/multicoin_test.go b/chain/multicoin_test.go
index a2ce4cb..e5c46a3 100644
--- a/chain/multicoin_test.go
+++ b/chain/multicoin_test.go
@@ -28,19 +28,19 @@ import (
        "strings"
        "testing"

-       "github.com/ava-labs/coreth/accounts/keystore"
-       "github.com/ava-labs/coreth/consensus/dummy"
-       "github.com/ava-labs/coreth/core"
-       "github.com/ava-labs/coreth/core/rawdb"
-       "github.com/ava-labs/coreth/core/types"
-       "github.com/ava-labs/coreth/core/vm"
-       "github.com/ava-labs/coreth/eth"
-       "github.com/ava-labs/coreth/eth/ethconfig"
-       "github.com/ava-labs/coreth/node"
+       "github.com/ethereum/go-ethereum/accounts/abi"
+       "github.com/ethereum/go-ethereum/common"
+       "github.com/ethereum/go-ethereum/crypto"
+       "github.com/ethereum/go-ethereum/log"
+       "github.com/flare-foundation/coreth/accounts/keystore"
+       "github.com/flare-foundation/coreth/consensus/dummy"
+       "github.com/flare-foundation/coreth/core"
+       "github.com/flare-foundation/coreth/core/rawdb"
+       "github.com/flare-foundation/coreth/core/types"
+       "github.com/flare-foundation/coreth/core/vm"
+       "github.com/flare-foundation/coreth/eth"
+       "github.com/flare-foundation/coreth/eth/ethconfig"
+       "github.com/flare-foundation/coreth/node"
    )

    // TestMulticoin tests multicoin low-level state management and regular
diff --git a/chain/payment_test.go b/chain/payment_test.go
index 5d9da4d..7ae81fc 100644
--- a/chain/payment_test.go
+++ b/chain/payment_test.go
@@ -7,8 +7,8 @@ import (
        "math/big"
        "testing"

-       "github.com/ava-labs/coreth/core/types"
        "github.com/ethereum/go-ethereum/log"
+       "github.com/flare-foundation/coreth/core/types"
    )

    // TestPayment tests basic payment (balance, not multi-coin)
diff --git a/chain/subscribe_accepted_heads_test.go b/chain/subscribe_accepted_heads_test.go
index 0a94bfd..db69c8f 100644
--- a/chain/subscribe_accepted_heads_test.go
+++ b/chain/subscribe_accepted_heads_test.go
@@ -4,10 +4,10 @@ import (
        "math/big"
        "testing"

-       "github.com/ava-labs/coreth/core"
-       "github.com/ava-labs/coreth/core/types"
        "github.com/ethereum/go-ethereum/common"
        "github.com/ethereum/go-ethereum/log"
+       "github.com/flare-foundation/coreth/core"
+       "github.com/flare-foundation/coreth/core/types"
    )

    func TestAcceptedHeadSubscriptions(t *testing.T) {
diff --git a/chain/subscribe_block_logs_test.go b/chain/subscribe_block_logs_test.go
index 26661da..186e6b6 100644
--- a/chain/subscribe_block_logs_test.go
+++ b/chain/subscribe_block_logs_test.go
@@ -6,10 +6,10 @@ import (
        "testing"
        "time"

-       "github.com/ava-labs/coreth/eth/filters"
+       "github.com/flare-foundation/coreth/eth/filters"

-       "github.com/ava-labs/coreth/core/types"
        "github.com/ethereum/go-ethereum/common"
    }

```



```

+         "github.com/flare-foundation/coreth/core/types"
+     )

    func TestBlockLogsAllowUnfinalized(t *testing.T) {
diff --git a/chain/subscribe_transactions_test.go b/chain/subscribe_transactions_test.go
index aac6db4..9ed4393 100644
--- a/chain/subscribe_transactions_test.go
+++ b/chain/subscribe_transactions_test.go
@@ -4,10 +4,10 @@ import (
    "math/big"
    "testing"

-    "github.com/ava-labs/coreth/eth/filters"
+    "github.com/flare-foundation/coreth/eth/filters"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/types"
)

    func TestSubscribeTransactions(t *testing.T) {
diff --git a/chain/test_chain.go b/chain/test_chain.go
index e4420de..0060a52 100644
--- a/chain/test_chain.go
+++ b/chain/test_chain.go
@@ -8,17 +8,17 @@ import (
    "math/big"
    "testing"

-    "github.com/ava-labs/coreth/accounts/keystore"
-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/eth"
-    "github.com/ava-labs/coreth/eth/ethconfig"
-    "github.com/ava-labs/coreth/node"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/hexutil"
+    "github.com/flare-foundation/coreth/accounts/keystore"
+    "github.com/flare-foundation/coreth/consensus/dummy"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/eth"
+    "github.com/flare-foundation/coreth/eth/ethconfig"
+    "github.com/flare-foundation/coreth/node"
+    "github.com/flare-foundation/coreth/params"
)

    var (
diff --git a/changes.sh b/changes.sh
new file mode 100755
index 0000000..b2b8d30
--- /dev/null
+++ b/changes.sh
@@ -0,0 +1,4 @@
+#!/bin/bash
+
+## Requires wkhtmltopdf and aha tools
+## Install using: sudo apt install wkhtmltopdf aha
+git diff --color upstream-v0.7.4-rc.1 | aha > CHANGES.html && wkhtmltopdf CHANGES.html CHANGES.pdf && rm CHANGES.html
+
+ \ No newline at end of file
diff --git a/cmd/abigen/main.go b/cmd/abigen/main.go
index ebb5b59..a7f2837 100644
--- a/cmd/abigen/main.go
+++ b/cmd/abigen/main.go
@@ -35,13 +35,13 @@ import (
    "regexp"
    "strings"

-    "github.com/ava-labs/coreth/accounts/abi"
-    "github.com/ava-labs/coreth/accounts/abi/bind"
-    "github.com/ava-labs/coreth/internal/flags"
+    "github.com/ethereum/go-ethereum/cmd/utlils"
+    "github.com/ethereum/go-ethereum/common/compiler"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/accounts/abi"
+    "github.com/flare-foundation/coreth/accounts/abi/bind"
+    "github.com/flare-foundation/coreth/internal/flags"
+    "gopkg.in/urfave/cli.v1"
)

diff --git a/consensus/consensus.go b/consensus/consensus.go
index 675be10..35a2106 100644
--- a/consensus/consensus.go
+++ b/consensus/consensus.go
@@ -30,11 +30,11 @@ package consensus
import (
    "math/big"

-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
)

// ChainHeaderReader defines a small collection of methods needed to access the local
diff --git a/consensus/dummy/consensus.go b/consensus/dummy/consensus.go
index 08cc229..4e97b72 100644
--- a/consensus/dummy/consensus.go
+++ b/consensus/dummy/consensus.go
@@ -10,13 +10,13 @@ package dummy
import (
    "math/big"
    "time"

-    "github.com/ava-labs/coreth/consensus"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/rpc"
-    "github.com/ava-labs/coreth/trie"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/consensus"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
+    "github.com/flare-foundation/coreth/trie"
)

    var (
diff --git a/consensus/dummy/consensus_test.go b/consensus/dummy/consensus_test.go
index 64a8439..5aa7a66 100644
--- a/consensus/dummy/consensus_test.go
+++ b/consensus/dummy/consensus_test.go
@@ -8,8 +8,8 @@ import (
    "math/big"
)


```

```

"testing"

- "github.com/ava-labs/coreth/core/types"
- "github.com/ethereum/go-ethereum/common"
+ "github.com/flare-foundation/coreth/core/types"
)

func TestVerifyBlockFee(t *testing.T) {
diff --git a/consensus/dummy/dynamic_fees.go b/consensus/dummy/dynamic_fees.go
index e944248..e8aee6c 100644
--- a/consensus/dummy/dynamic_fees.go
+++ b/consensus/dummy/dynamic_fees.go
@@ -8,11 +8,11 @@ import (
    "fmt"
    "math/big"

- "github.com/ava-labs/avalanchego/utils/wrappers"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/params"
+ "github.com/ethereum/go-ethereum/common"
+ "github.com/ethereum/go-ethereum/common/math"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/flare-foundation/flare/utils/wrappers"
)

var (
diff --git a/consensus/dummy/dynamic_fees_test.go b/consensus/dummy/dynamic_fees_test.go
index 9bbfcc1..a41d9de 100644
--- a/consensus/dummy/dynamic_fees_test.go
+++ b/consensus/dummy/dynamic_fees_test.go
@@ -8,10 +8,10 @@ import (
    "math/big"
    "testing"

- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/params"
+ "github.com/ethereum/go-ethereum/common/math"
+ "github.com/ethereum/go-ethereum/log"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/stretchr/testify/assert"
)

diff --git a/consensus/misc/dao.go b/consensus/misc/dao.go
index a0ab402..be7c8df 100644
--- a/consensus/misc/dao.go
+++ b/consensus/misc/dao.go
@@ -31,9 +31,9 @@ import (
    "errors"
    "math/big"

- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/params"
+ "github.com/flare-foundation/coreth/core/state"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/params"
)

var (
diff --git a/core/block_validator.go b/core/block_validator.go
index 287a42f..5954543 100644
--- a/core/block_validator.go
+++ b/core/block_validator.go
@@ -29,11 +29,11 @@ package core
import (
    "fmt"

- "github.com/ava-labs/coreth/consensus"
- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/params"
- "github.com/ava-labs/coreth/trie"
+ "github.com/flare-foundation/coreth/consensus"
+ "github.com/flare-foundation/coreth/core/state"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/flare-foundation/coreth/trie"
)

// BlockValidator is responsible for validating block headers, uncles and
diff --git a/core/blockchain.go b/core/blockchain.go
index 9e46bcd..04fc819 100644
--- a/core/blockchain.go
+++ b/core/blockchain.go
@@ -37,18 +37,18 @@ import (
    "sync/atomic"
    "time"

- "github.com/ava-labs/coreth/consensus"
- "github.com/ava-labs/coreth/core/rawdb"
- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/core/state/snapshot"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/core/vm"
- "github.com/ava-labs/coreth/ethdb"
- "github.com/ava-labs/coreth/params"
- "github.com/ava-labs/coreth/trie"
+ "github.com/ethereum/go-ethereum/common"
+ "github.com/ethereum/go-ethereum/event"
+ "github.com/ethereum/go-ethereum/log"
+ "github.com/flare-foundation/coreth/consensus"
+ "github.com/flare-foundation/coreth/core/rawdb"
+ "github.com/flare-foundation/coreth/core/state"
+ "github.com/flare-foundation/coreth/core/state/snapshot"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/core/vm"
+ "github.com/flare-foundation/coreth/ethdb"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/flare-foundation/coreth/trie"
+ lru "github.com/hashicorp/golang-lru"
)

diff --git a/core/blockchain_reader.go b/core/blockchain_reader.go
index 33adaea..b0a4531 100644
--- a/core/blockchain_reader.go
+++ b/core/blockchain_reader.go
@@ -29,15 +29,15 @@ package core
import (
    "math/big"

- "github.com/ava-labs/coreth/consensus"
- "github.com/ava-labs/coreth/core/rawdb"
- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/core/state/snapshot"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/core/vm"
- "github.com/ava-labs/coreth/params"
+ "github.com/ethereum/go-ethereum/common"
+ "github.com/ethereum/go-ethereum/event"
+ "github.com/flare-foundation/coreth/consensus"
+ "github.com/flare-foundation/coreth/core/rawdb"
+ "github.com/flare-foundation/coreth/core/state"

```

```

+         "github.com/flare-foundation/coreth/core/state/snapshot"
+         "github.com/flare-foundation/coreth/core/types"
+         "github.com/flare-foundation/coreth/core/vm"
+         "github.com/flare-foundation/coreth/params"
+     )
+
+ // CurrentHeader retrieves the current head header of the canonical chain. The
diff --git a/core/blockchain_test.go b/core/blockchain_test.go
index 1685e04..7c25e27 100644
--- a/core/blockchain_test.go
+++ b/core/blockchain_test.go
@@ -7,14 +7,14 @@ import (
     "math/big"
     "testing"

-     "github.com/ava-labs/coreth/consensus/dummy"
-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/core/state"
-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/core/vm"
-     "github.com/ava-labs/coreth/ethdb"
-     "github.com/ava-labs/coreth/params"
+     "github.com/flare-foundation/coreth/consensus/dummy"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/core/state"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/core/vm"
+     "github.com/flare-foundation/coreth/ethdb"
+     "github.com/flare-foundation/coreth/params"
+ )

func TestArchiveBlockChain(t *testing.T) {
diff --git a/core/bloom_indexer.go b/core/bloom_indexer.go
index 4d8f58f..9404a81 100644
--- a/core/bloom_indexer.go
+++ b/core/bloom_indexer.go
@@ -20,12 +20,12 @@ import (
     "context"
     "time"

-     "github.com/ava-labs/coreth/core/bloombits"
-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/ethdb"
+     "github.com/flare-foundation/coreth/core/bloombits"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/ethdb"
+ )

const (
diff --git a/core/bloombits/generator.go b/core/bloombits/generator.go
index c0422ca..32e011a 100644
--- a/core/bloombits/generator.go
+++ b/core/bloombits/generator.go
@@ -29,7 +29,7 @@ package bloombits
import (
    "errors"

-     "github.com/ava-labs/coreth/core/types"
+     "github.com/flare-foundation/coreth/core/types"
+ )

var (
diff --git a/core/bloombits/generator_test.go b/core/bloombits/generator_test.go
index 067c1db..345ec4e 100644
--- a/core/bloombits/generator_test.go
+++ b/core/bloombits/generator_test.go
@@ -31,7 +31,7 @@ import (
    "math/rand"
    "testing"

-     "github.com/ava-labs/coreth/core/types"
+     "github.com/flare-foundation/coreth/core/types"
+ )

// Tests that batched bloom bits are correctly rotated from the input bloom
diff --git a/core/chain_indexer.go b/core/chain_indexer.go
index 975f82b..7163929 100644
--- a/core/chain_indexer.go
+++ b/core/chain_indexer.go
@@ -34,12 +34,12 @@ import (
    "sync/atomic"
    "time"

-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/ethdb"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/ethdb"
+ )

// ChainIndexerBackend defines the methods needed to process chain segments in
diff --git a/core/chain_indexer_test.go b/core/chain_indexer_test.go
index 3edf175..481f4e1 100644
--- a/core/chain_indexer_test.go
+++ b/core/chain_indexer_test.go
@@ -35,9 +35,9 @@ import (
    "testing"
    "time"

-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/core/types"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/core/types"
+ )

// ChainIndexerBackend defines the methods needed to process chain segments in
diff --git a/core/chain_makers.go b/core/chain_makers.go
index 9830ff7..3575fc0 100644
--- a/core/chain_makers.go
+++ b/core/chain_makers.go
@@ -30,15 +30,15 @@ import (
    "fmt"
    "math/big"

-     "github.com/ava-labs/coreth/consensus"
-     "github.com/ava-labs/coreth/consensus/dummy"
-     "github.com/ava-labs/coreth/consensus/misc"
-     "github.com/ava-labs/coreth/core/state"
-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/core/vm"
-     "github.com/ava-labs/coreth/ethdb"
-     "github.com/ava-labs/coreth/params"
+     "github.com/flare-foundation/coreth/consensus"
+     "github.com/flare-foundation/coreth/consensus/dummy"
+     "github.com/flare-foundation/coreth/consensus/misc"
+     "github.com/flare-foundation/coreth/core/state"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/core/vm"
+     "github.com/flare-foundation/coreth/ethdb"
+     "github.com/flare-foundation/coreth/params"
+     "github.com/ethereum/go-ethereum/common"
+ )

```

```

+      "github.com/flare-foundation/coreth/consensus/dummy"
+      "github.com/flare-foundation/coreth/consensus/misc"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/ethdb"
+      "github.com/flare-foundation/coreth/params"
+    )

    // BlockGen creates blocks for testing.
diff --git a/core/chain_makers_test.go b/core/chain_makers_test.go
index cd00393..84a4b7e 100644
--- a/core/chain_makers_test.go
+++ b/core/chain_makers_test.go
@@ -30,13 +30,13 @@ import (
     "fmt"
     "math/big"

-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/consensus/dummy"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/params"
 )

func ExampleGenerateChain() {
diff --git a/core/error.go b/core/error.go
index 0265448..ba6153e 100644
--- a/core/error.go
+++ b/core/error.go
@@ -29,7 +29,7 @@ package core
import (
    "errors"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/types"
)

var (
diff --git a/core/events.go b/core/events.go
index 4898dbc..f71c2d4 100644
--- a/core/events.go
+++ b/core/events.go
@@ -27,8 +27,8 @@ package core
import (
-    "github.com/ava-labs/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/types"
    "github.com/ethereum/go-ethereum/common"
)

// NewTxEvent is posted when a batch of transactions enter the transaction pool.
diff --git a/core/evm.go b/core/evm.go
index d45f241..378fb87 100644
--- a/core/evm.go
+++ b/core/evm.go
@@ -29,10 +29,10 @@ package core
import (
    "math/big"

-    "github.com/ava-labs/coreth/consensus"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
+    "github.com/flare-foundation/coreth/consensus"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/log"
)

diff --git a/core/gen_genesis.go b/core/gen_genesis.go
index a4ec8f5..2ef02e0 100644
--- a/core/gen_genesis.go
+++ b/core/gen_genesis.go
@@ -7,10 +7,10 @@ import (
    "errors"
    "math/big"

-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/params"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/common/hexutil"
    "github.com/ethereum/go-ethereum/common/math"
)

var _ = (*genesisSpecMarshaling)(nil)
diff --git a/core/genesis.go b/core/genesis.go
index 7e10249..fd66643 100644
--- a/core/genesis.go
+++ b/core/genesis.go
@@ -34,16 +34,16 @@ import (
    "fmt"
    "math/big"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/trie"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/trie"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/common/hexutil"
    "github.com/ethereum/go-ethereum/common/math"
    "github.com/ethereum/go-ethereum/log"
)

//go:generate gencodec -type Genesis -field-override genesisSpecMarshaling -out gen_genesis.go
diff --git a/core/headerchain.go b/core/headerchain.go
index 092350a..3c7396a 100644
--- a/core/headerchain.go
+++ b/core/headerchain.go
@@ -33,12 +33,12 @@ import (
    "math/rand"
    "sync/atomic"

-    "github.com/ava-labs/coreth/consensus"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/flare-foundation/coreth/consensus"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
    "github.com/flare-foundation/coreth/params"
)

```

```

+       "github.com/ethereum/go-ethereum/common"
+       "github.com/flare-foundation/coreth/consensus"
+       "github.com/flare-foundation/coreth/core/rawdb"
+       "github.com/flare-foundation/coreth/core/types"
+       "github.com/flare-foundation/coreth/ethdb"
+       "github.com/flare-foundation/coreth/params"
+       lru "github.com/hashicorp/golang-lru"
+
+   )
+
+   diff --git a/core/keeper.go b/core/keeper.go
+   new file mode 100644
+   index 0000000..a011f45
+   --- /dev/null
+   +++ b/core/keeper.go
+   @@ -0,0 +1,149 @@
+   +// (c) 2021, Flare Networks Limited. All rights reserved.
+   +// Please see the file LICENSE for licensing terms.
+   +
+   +package core
+   +
+   +import (
+   +    "fmt"
+   +    "math/big"
+   +
+   +    "github.com/ethereum/go-ethereum/common"
+   +    "github.com/ethereum/go-ethereum/log"
+   +
+   +    "github.com/flare-foundation/coreth/core/vm"
+   +)
+   +
+   +// Define errors
+   +type ErrInvalidKeeperData struct{}
+   +
+   +func (e *ErrInvalidKeeperData) Error() string { return "invalid return data from keeper trigger" }
+   +
+   +type ErrKeeperDataEmpty struct{}
+   +
+   +func (e *ErrKeeperDataEmpty) Error() string { return "return data from keeper trigger empty" }
+   +
+   +type ErrMaxMintExceeded struct {
+   +    mintMax      *big.Int
+   +    mintRequest  *big.Int
+   +}
+   +
+   +func (e *ErrMaxMintExceeded) Error() string {
+   +    return fmt.Sprintf("mint request of %s exceeded max of %s", e.mintRequest.Text(10), e.mintMax.Text(10))
+   +}
+   +
+   +type ErrMintNegative struct{}
+   +
+   +func (e *ErrMintNegative) Error() string { return "mint request cannot be negative" }
+   +
+   +// Define interface for dependencies
+   +type EVMCaller interface {
+   +    Call(caller vm.ContractRef, addr common.Address, input []byte, gas uint64, value *big.Int) (ret []byte, leftOverGas uint64, err error)
+   +    GetBlockNumber() *big.Int
+   +    GetGasLimit() uint64
+   +    AddBalance(addr common.Address, amount *big.Int)
+   +}
+   +
+   +// Define maximums that can change by block height
+   +func GetKeeperGasMultiplier(blockNumber *big.Int) uint64 {
+   +    switch {
+   +    default:
+   +        return 100
+   +    }
+   +}
+   +
+   +func GetSystemTriggerContractAddr(blockNumber *big.Int) string {
+   +    switch {
+   +    default:
+   +        return "0x1000000000000000000000000000000000000000000000000000000000000002"
+   +    }
+   +}
+   +
+   +func GetSystemTriggerSelector(blockNumber *big.Int) []byte {
+   +    switch {
+   +    default:
+   +        return []byte{0x7f, 0xec, 0x8d, 0x38}
+   +    }
+   +}
+   +
+   +func GetPrioritisedFTS0Contract(blockTime *big.Int) string {
+   +    switch {
+   +    default:
+   +        return "0x1000000000000000000000000000000000000000000000000000000000000003"
+   +    }
+   +}
+   +
+   +func GetMaximumMintRequest(blockNumber *big.Int) *big.Int {
+   +    switch {
+   +    default:
+   +        maxRequest, _ := new(big.Int).SetString("50000000000000000000000000000000", 10)
+   +        return maxRequest
+   +    }
+   +}
+   +
+   +func triggerKeeper(evm EVMCaller) (*big.Int, error) {
+   +    bigZero := big.NewInt(0)
+   +    // Get the contract to call
+   +    systemTriggerContract := common.HexToAddress(GetSystemTriggerContractAddr(evm.GetBlockNumber()))
+   +    // Call the method
+   +    triggerRet, _, triggerErr := evm.Call(
+   +        vm.AccountRef(systemTriggerContract),
+   +        systemTriggerContract,
+   +        GetSystemTriggerSelector(evm.GetBlockNumber()),
+   +        GetKeeperGasMultiplier(evm.GetBlockNumber())*evm.GetGasLimit(),
+   +        bigZero)
+   +    // If no error and a value came back...
+   +    if triggerErr == nil && triggerRet != nil {
+   +        // Did we get one big int?
+   +        if len(triggerRet) == 32 {
+   +            // Convert to big int
+   +            // Mint request cannot be less than 0 as SetBytes treats value as unsigned
+   +            mintRequest := new(big.Int).SetBytes(triggerRet)
+   +            // return the mint request
+   +            return mintRequest, nil
+   +        } else {
+   +            // Returned length was not 32 bytes
+   +            return bigZero, &ErrInvalidKeeperData{}
+   +        }
+   +    } else {
+   +        if triggerErr != nil {
+   +            return bigZero, triggerErr
+   +        } else {
+   +            return bigZero, &ErrKeeperDataEmpty{}
+   +        }
+   +    }
+   +}
+   +
+   +func mint(evm EVMCaller, mintRequest *big.Int) error {
+   +    // If the mint request is greater than zero and less than max
+   +    max := GetMaximumMintRequest(evm.GetBlockNumber())
+   +    if mintRequest.Cmp(big.NewInt(0)) > 0 &&

```

[illegible]

[illegible]

[illegible]


```

+         t.Errorf("AddBalance not called as expected")
+     }
+     if defaultEVMMock.mockEVMCallerData.lastAddBalanceAddr.String() != GetSystemTriggerContractAddr(big.NewInt(0)) {
+         t.Errorf("wanted addr %s; got addr %s", GetSystemTriggerContractAddr(big.NewInt(0)), defaultEVMMock.mockEVMCallerData.lastAddBalanceAddr)
+     }
+     if defaultEVMMock.mockEVMCallerData.lastAddBalanceAmount.Cmp(mintRequest) != 0 {
+         t.Errorf("wanted amount %s; got amount %s", mintRequest.Text(10), defaultEVMMock.mockEVMCallerData.lastAddBalanceAmount.Text(10))
+     }
+ } else {
+     t.Errorf("unexpected error returned; was = %s", err.Error())
+ }
+ }
+
+func TestKeeperTriggerShouldNotErrorMintingZero(t *testing.T) {
+    // Assemble
+    mintRequest := big.NewInt(0)
+    mockEVMCallerData := &MockEVMCallerData{
+        blockNumber:    *big.NewInt(0),
+        gasLimit:        0,
+        mintRequestReturn: *big.NewInt(0),
+    }
+    defaultEVMMock := &DefaultEVMMock{
+        mockEVMCallerData: *mockEVMCallerData,
+    }
+
+    // Act
+    err := mint(defaultEVMMock, mintRequest)
+
+    // Assert
+    if err == nil {
+        if defaultEVMMock.mockEVMCallerData.addBalanceCalls != 0 {
+            t.Errorf("AddBalance called unexpectedly")
+        }
+    } else {
+        t.Errorf("unexpected error returned; was %s", err.Error())
+    }
+ }
+
+func TestKeeperTriggerFiredAndMinted(t *testing.T) {
+    mintRequestReturn, _ := new(big.Int).SetString("5000000000000000000000", 10)
+    mockEVMCallerData := &MockEVMCallerData{
+        blockNumber:    *big.NewInt(0),
+        gasLimit:        0,
+        mintRequestReturn: *mintRequestReturn,
+    }
+    defaultEVMMock := &DefaultEVMMock{
+        mockEVMCallerData: *mockEVMCallerData,
+    }
+
+    log := log.New()
+    triggerKeeperAndMint(defaultEVMMock, log)
+
+    // EVM Call function calling the keeper should have been called
+    if defaultEVMMock.mockEVMCallerData.callCalls != 1 {
+        t.Errorf("EVM Call count not as expected. got %d want 1", defaultEVMMock.mockEVMCallerData.callCalls)
+    }
+    // AddBalance should have been called on the state database, minting the request asked for
+    if defaultEVMMock.mockEVMCallerData.addBalanceCalls != 1 {
+        t.Errorf("Add balance call count not as expected. got %d want 1", defaultEVMMock.mockEVMCallerData.addBalanceCalls)
+    }
+ }
+
+func TestKeeperTriggerShouldNotMintMoreThanLimit(t *testing.T) {
+    mintRequestReturn, _ := new(big.Int).SetString("500000000000000000000001", 10)
+    mockEVMCallerData := &MockEVMCallerData{
+        blockNumber:    *big.NewInt(0),
+        gasLimit:        0,
+        mintRequestReturn: *mintRequestReturn,
+    }
+    defaultEVMMock := &DefaultEVMMock{
+        mockEVMCallerData: *mockEVMCallerData,
+    }
+
+    log := log.New()
+    triggerKeeperAndMint(defaultEVMMock, log)
+
+    // EVM Call function calling the keeper should have been called
+    if defaultEVMMock.mockEVMCallerData.callCalls != 1 {
+        t.Errorf("EVM Call count not as expected. got %d want 1", defaultEVMMock.mockEVMCallerData.callCalls)
+    }
+    // AddBalance should not have been called on the state database, as the mint request was over the limit
+    if defaultEVMMock.mockEVMCallerData.addBalanceCalls != 0 {
+        t.Errorf("Add balance call count not as expected. got %d want 1", defaultEVMMock.mockEVMCallerData.addBalanceCalls)
+    }
+ }
+
+diff --git a/core/mkalloc.go b/core/mkalloc.go
+index 76978a5..95fcafb 100644
+--- a/core/mkalloc.go
++++ b/core/mkalloc.go
+@@ -45,8 +45,8 @@ import (
+     "sort"
+     "strconv"
+
+ -     "github.com/ava-labs/coreth/core"
+     "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/core"
+ )
+
+ type allocItem struct{ Addr, Balance *big.Int }
+
+diff --git a/core/rawdb/accessors_chain.go b/core/rawdb/accessors_chain.go
+index 8b134e2..0a7bf9f 100644
+--- a/core/rawdb/accessors_chain.go
++++ b/core/rawdb/accessors_chain.go
+@@ -32,12 +32,12 @@ import (
+     "errors"
+     "math/big"
+
+ -     "github.com/ava-labs/coreth/core/types"
+ -     "github.com/ava-labs/coreth/ethdb"
+ -     "github.com/ava-labs/coreth/params"
+     "github.com/ethereum/go-ethereum/common"
+     "github.com/ethereum/go-ethereum/log"
+     "github.com/ethereum/go-ethereum/rlp"
+ +     "github.com/flare-foundation/coreth/core/types"
+ +     "github.com/flare-foundation/coreth/ethdb"
+ +     "github.com/flare-foundation/coreth/params"
+ )
+
+ // ReadCanonicalHash retrieves the hash assigned to a canonical block number.
+diff --git a/core/rawdb/accessors_chain_test.go b/core/rawdb/accessors_chain_test.go
+index 9d35e7b..7c9b27e 100644
+--- a/core/rawdb/accessors_chain_test.go
++++ b/core/rawdb/accessors_chain_test.go
+@@ -25,10 +25,10 @@ import (
+     "reflect"
+     "testing"
+
+ -     "github.com/ava-labs/coreth/core/types"
+ -     "github.com/ava-labs/coreth/params"
+     "github.com/ethereum/go-ethereum/common"
+     "github.com/ethereum/go-ethereum/rlp"
+ +     "github.com/flare-foundation/coreth/core/types"
+ +     "github.com/flare-foundation/coreth/params"
+     "qolang.org/x/crypto/sha3"

```

```

)

diff --git a/core/rawdb/accessors_indexes.go b/core/rawdb/accessors_indexes.go
index 0ac33211..a75dc38 100644
--- a/core/rawdb/accessors_indexes.go
+++ b/core/rawdb/accessors_indexes.go
@@ -30,12 +30,12 @@ import (
    "bytes"
    "math/big"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/params"
)

// ReadTxLookupEntry retrieves the positional metadata associated with a transaction
diff --git a/core/rawdb/accessors_indexes_test.go b/core/rawdb/accessors_indexes_test.go
index e64818b1..0d2aaf7 100644
--- a/core/rawdb/accessors_indexes_test.go
+++ b/core/rawdb/accessors_indexes_test.go
@@ -22,10 +22,10 @@ import (
    "math/big"
    "testing"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
+    "golang.org/x/crypto/sha3"
)

diff --git a/core/rawdb/accessors_metadata.go b/core/rawdb/accessors_metadata.go
index 6b1a0f11..e8a9e61 100644
--- a/core/rawdb/accessors_metadata.go
+++ b/core/rawdb/accessors_metadata.go
@@ -29,11 +29,11 @@ package rawdb
import (
    "encoding/json"

-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/params"
)

// ReadDatabaseVersion retrieves the version number of the database.
diff --git a/core/rawdb/accessors_snapshot.go b/core/rawdb/accessors_snapshot.go
index 2c7c6c71..e89a353 100644
--- a/core/rawdb/accessors_snapshot.go
+++ b/core/rawdb/accessors_snapshot.go
@@ -29,9 +29,9 @@ package rawdb
import (
    "encoding/binary"

-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/ethdb"
)

// ReadSnapshotRoot retrieves the root of the block whose state is contained in
diff --git a/core/rawdb/accessors_state.go b/core/rawdb/accessors_state.go
index 38f663c1..f17ac27 100644
--- a/core/rawdb/accessors_state.go
+++ b/core/rawdb/accessors_state.go
@@ -27,9 +27,9 @@ package rawdb
import (
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/ethdb"
)

// ReadPreimage retrieves a single preimage of the provided hash.
diff --git a/core/rawdb/database.go b/core/rawdb/database.go
index 55a42d11..2049cc9 100644
--- a/core/rawdb/database.go
+++ b/core/rawdb/database.go
@@ -32,10 +32,10 @@ import (
    "os"
    "time"

-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
+    "github.com/olekukonko/tablewriter"
)

diff --git a/core/rawdb/table.go b/core/rawdb/table.go
index 179ebf31..3e4f4bb 100644
--- a/core/rawdb/table.go
+++ b/core/rawdb/table.go
@@ -27,9 +27,9 @@ package rawdb
import (
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb"
)

// table is a wrapper around a database that prefixes each key access with a pre-
diff --git a/core/rawdb/table_test.go b/core/rawdb/table_test.go
index c7cac981..2f3bd97 100644
--- a/core/rawdb/table_test.go
+++ b/core/rawdb/table_test.go
@@ -30,7 +30,7 @@ import (
    "bytes"
    "testing"

-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb"
)

func TestTableDatabase(t *testing.T) {
    { testTableDatabase(t, "prefix") }
}
diff --git a/core/state/database.go b/core/state/database.go
index bef0af11..eb98ad6 100644
--- a/core/state/database.go
+++ b/core/state/database.go

```

```

@@ -31,11 +31,11 @@ import (
    "fmt"

    "github.com/VictoriaMetrics/fastcache"
-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/core/types"
-   "github.com/ava-labs/coreth/ethdb"
-   "github.com/ava-labs/coreth/trie"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/core/types"
+   "github.com/flare-foundation/coreth/ethdb"
+   "github.com/flare-foundation/coreth/trie"
    lru "github.com/hashicorp/golang-lru"
)

diff --git a/core/state/dump.go b/core/state/dump.go
index 8f30d48..e0cd9f8 100644
--- a/core/state/dump.go
+++ b/core/state/dump.go
@@ -31,12 +31,12 @@ import (
    "fmt"
    "time"

-   "github.com/ava-labs/coreth/core/types"
-   "github.com/ava-labs/coreth/trie"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/common/hexutil"
+   "github.com/ethereum/go-ethereum/log"
+   "github.com/ethereum/go-ethereum/rlp"
+   "github.com/flare-foundation/coreth/core/types"
+   "github.com/flare-foundation/coreth/trie"
)

// DumpConfig is a set of options to control what portions of the state will be
diff --git a/core/state/iterator.go b/core/state/iterator.go
index 2ad4ed9..01ab5d5 100644
--- a/core/state/iterator.go
+++ b/core/state/iterator.go
@@ -30,10 +30,10 @@ import (
    "bytes"
    "fmt"

-   "github.com/ava-labs/coreth/core/types"
-   "github.com/ava-labs/coreth/trie"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/rlp"
+   "github.com/flare-foundation/coreth/core/types"
+   "github.com/flare-foundation/coreth/trie"
)

// NodeIterator is an iterator to traverse the entire state trie post-order,
diff --git a/core/state/snapshot/conversion.go b/core/state/snapshot/conversion.go
index bfb157a..01b3d23 100644
--- a/core/state/snapshot/conversion.go
+++ b/core/state/snapshot/conversion.go
@@ -36,12 +36,12 @@ import (
    "sync"
    "time"

-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/ethdb"
-   "github.com/ava-labs/coreth/trie"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/log"
+   "github.com/ethereum/go-ethereum/rlp"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/ethdb"
+   "github.com/flare-foundation/coreth/trie"
)

// trieKV represents a trie key-value pair
diff --git a/core/state/snapshot/difflayer_test.go b/core/state/snapshot/difflayer_test.go
index 245acc8..562929f 100644
--- a/core/state/snapshot/difflayer_test.go
+++ b/core/state/snapshot/difflayer_test.go
@@ -32,9 +32,9 @@ import (
    "testing"

    "github.com/VictoriaMetrics/fastcache"
-   "github.com/ava-labs/coreth/ethdb/memorydb"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/crypto"
+   "github.com/flare-foundation/coreth/ethdb/memorydb"
)

func copyDestructs(destructs map[common.Hash]struct{}) map[common.Hash]struct{} {
diff --git a/core/state/snapshot/disklayer.go b/core/state/snapshot/disklayer.go
index 07add6b..932722d 100644
--- a/core/state/snapshot/disklayer.go
+++ b/core/state/snapshot/disklayer.go
@@ -32,11 +32,11 @@ import (
    "time"

    "github.com/VictoriaMetrics/fastcache"
-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/ethdb"
-   "github.com/ava-labs/coreth/trie"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/rlp"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/ethdb"
+   "github.com/flare-foundation/coreth/trie"
)

// diskLayer is a low level persistent snapshot built on top of a key-value store.
diff --git a/core/state/snapshot/disklayer_test.go b/core/state/snapshot/disklayer_test.go
index adf3778..310a730 100644
--- a/core/state/snapshot/disklayer_test.go
+++ b/core/state/snapshot/disklayer_test.go
@@ -30,11 +30,11 @@ import (
    "bytes"
    "testing"

-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/ethdb"
-   "github.com/ava-labs/coreth/ethdb/memorydb"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/rlp"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/ethdb"
+   "github.com/flare-foundation/coreth/ethdb/memorydb"
)

// reverse reverses the contents of a byte slice. It's used to update random accs
diff --git a/core/state/snapshot/generate.go b/core/state/snapshot/generate.go
index ae655ae..b1031b6 100644
--- a/core/state/snapshot/generate.go
+++ b/core/state/snapshot/generate.go
@@ -34,14 +34,14 @@ import (
    "time"

    "github.com/VictoriaMetrics/fastcache"
-   "github.com/ava-labs/coreth/core/rawdb"

```

```
-      "github.com/ava-labs/coreth/ethdb"
-      "github.com/ava-labs/coreth/trie"
      "github.com/ethereum/go-ethereum/common"
      "github.com/ethereum/go-ethereum/common/math"
      "github.com/ethereum/go-ethereum/crypto"
      "github.com/ethereum/go-ethereum/log"
      "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/ethdb"
+     "github.com/flare-foundation/coreth/trie"
  )

  var (
diff --git a/core/state/snapshot/generate_test.go b/core/state/snapshot/generate_test.go
index 8c480e1..ee77b9c 100644
--- a/core/state/snapshot/generate_test.go
+++ b/core/state/snapshot/generate_test.go
@@ -33,13 +33,13 @@ import (
      "testing"
      "time"

-     "github.com/ava-labs/coreth/ethdb"
-     "github.com/ava-labs/coreth/ethdb/memorydb"
-     "github.com/ava-labs/coreth/trie"
      "github.com/ethereum/go-ethereum/common"
      "github.com/ethereum/go-ethereum/core/rawdb"
      "github.com/ethereum/go-ethereum/log"
      "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/ethdb"
+     "github.com/flare-foundation/coreth/ethdb/memorydb"
+     "github.com/flare-foundation/coreth/trie"
+     "golang.org/x/crypto/sha3"
  )

diff --git a/core/state/snapshot/iterator.go b/core/state/snapshot/iterator.go
index b07c293..3bb36a5 100644
--- a/core/state/snapshot/iterator.go
+++ b/core/state/snapshot/iterator.go
@@ -31,9 +31,9 @@ import (
      "fmt"
      "sort"

-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/ethdb"
      "github.com/ethereum/go-ethereum/common"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/ethdb"
  )

  // Iterator is an iterator to step over all the accounts or the specific
diff --git a/core/state/snapshot/iterator_test.go b/core/state/snapshot/iterator_test.go
index 9d1b744..fb56870 100644
--- a/core/state/snapshot/iterator_test.go
+++ b/core/state/snapshot/iterator_test.go
@@ -33,8 +33,8 @@ import (
      "math/rand"
      "testing"

-     "github.com/ava-labs/coreth/core/rawdb"
      "github.com/ethereum/go-ethereum/common"
+     "github.com/flare-foundation/coreth/core/rawdb"
  )

  // TestAccountIteratorBasics tests some simple single-layer(diff and disk) iteration
diff --git a/core/state/snapshot/journal.go b/core/state/snapshot/journal.go
index 7feba07..43d2a66 100644
--- a/core/state/snapshot/journal.go
+++ b/core/state/snapshot/journal.go
@@ -33,12 +33,12 @@ import (
      "time"

      "github.com/VictoriaMetrics/fastcache"
-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/ethdb"
-     "github.com/ava-labs/coreth/trie"
      "github.com/ethereum/go-ethereum/common"
      "github.com/ethereum/go-ethereum/log"
      "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/ethdb"
+     "github.com/flare-foundation/coreth/trie"
  )

  // journalGenerator is a disk layer entry containing the generator progress marker.
diff --git a/core/state/snapshot/snapshot.go b/core/state/snapshot/snapshot.go
index 6c65f93..eff4e1c 100644
--- a/core/state/snapshot/snapshot.go
+++ b/core/state/snapshot/snapshot.go
@@ -36,12 +36,12 @@ import (
      "time"

      "github.com/VictoriaMetrics/fastcache"
-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/ethdb"
-     "github.com/ava-labs/coreth/trie"
      "github.com/ethereum/go-ethereum/common"
      "github.com/ethereum/go-ethereum/log"
      "github.com/ethereum/go-ethereum/metrics"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/ethdb"
+     "github.com/flare-foundation/coreth/trie"
  )

  const (
diff --git a/core/state/snapshot/snapshot_test.go b/core/state/snapshot/snapshot_test.go
index 6ccee14..4f48399 100644
--- a/core/state/snapshot/snapshot_test.go
+++ b/core/state/snapshot/snapshot_test.go
@@ -33,9 +33,9 @@ import (
      "testing"
      "time"

-     "github.com/ava-labs/coreth/core/rawdb"
      "github.com/ethereum/go-ethereum/common"
+     "github.com/flare-foundation/coreth/core/rawdb"
  )

  // randomHash generates a random blob of data and returns it as a hash.
diff --git a/core/state/snapshot/wipe.go b/core/state/snapshot/wipe.go
index 298a77a..01f3cec 100644
--- a/core/state/snapshot/wipe.go
+++ b/core/state/snapshot/wipe.go
@@ -30,10 +30,10 @@ import (
      "bytes"
      "time"

-     "github.com/ava-labs/coreth/core/rawdb"
-     "github.com/ava-labs/coreth/ethdb"
      "github.com/ethereum/go-ethereum/common"
      "github.com/ethereum/go-ethereum/log"
+     "github.com/flare-foundation/coreth/core/rawdb"
+     "github.com/flare-foundation/coreth/ethdb"
  )
```

```

// wipeSnapshot starts a goroutine to iterate over the entire key-value database
diff --git a/core/state/snapshot/wipe_test.go b/core/state/snapshot/wipe_test.go
index dbf009e..d2486d0 100644
--- a/core/state/snapshot/wipe_test.go
+++ b/core/state/snapshot/wipe_test.go
@@ -30,9 +30,9 @@ import (
    "math/rand"
    "testing"

-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/ethdb/memorydb"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/ethdb/memorydb"
)

// Tests that given a database with random data content, all parts of a snapshot
diff --git a/core/state/state_object.go b/core/state/state_object.go
index 0e5be40..8d09e5f 100644
--- a/core/state/state_object.go
+++ b/core/state/state_object.go
@@ -34,11 +34,11 @@ import (
    "sync"
    "time"

-   "github.com/ava-labs/coreth/core/types"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/crypto"
+   "github.com/ethereum/go-ethereum/metrics"
+   "github.com/ethereum/go-ethereum/rlp"
+   "github.com/flare-foundation/coreth/core/types"
)

var emptyCodeHash = crypto.Keccak256(nil)
diff --git a/core/state/state_test.go b/core/state/state_test.go
index e7a05ef..89c1de8 100644
--- a/core/state/state_test.go
+++ b/core/state/state_test.go
@@ -27,9 +27,9 @@ package state

import (
-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/ethdb"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/ethdb"
)

type stateTest struct {
diff --git a/core/state/statedb.go b/core/state/statedb.go
index 44a9ee4..2b7f0fe 100644
--- a/core/state/statedb.go
+++ b/core/state/statedb.go
@@ -34,15 +34,15 @@ import (
    "sort"
    "time"

-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/core/state/snapshot"
-   "github.com/ava-labs/coreth/core/types"
-   "github.com/ava-labs/coreth/trie"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/crypto"
+   "github.com/ethereum/go-ethereum/log"
+   "github.com/ethereum/go-ethereum/metrics"
+   "github.com/ethereum/go-ethereum/rlp"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/core/state/snapshot"
+   "github.com/flare-foundation/coreth/core/types"
+   "github.com/flare-foundation/coreth/trie"
)

type revision struct {
diff --git a/core/state/statedb_test.go b/core/state/statedb_test.go
index 9c295f9..a4a9887 100644
--- a/core/state/statedb_test.go
+++ b/core/state/statedb_test.go
@@ -39,11 +39,11 @@ import (
    "testing"
    "testing/quick"

-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/core/state/snapshot"
-   "github.com/ava-labs/coreth/core/types"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/crypto"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/core/state/snapshot"
+   "github.com/flare-foundation/coreth/core/types"
)

// Tests that updating a state trie does not leak any database writes prior to
diff --git a/core/state/trie_prefetcher_test.go b/core/state/trie_prefetcher_test.go
index 563a773..183baca 100644
--- a/core/state/trie_prefetcher_test.go
+++ b/core/state/trie_prefetcher_test.go
@@ -31,8 +31,8 @@ import (
    "testing"
    "time"

-   "github.com/ava-labs/coreth/core/rawdb"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/flare-foundation/coreth/core/rawdb"
)

func filledStateDB() *StateDB {
diff --git a/core/state/connector.go b/core/state/connector.go
new file mode 100644
index 0000000..1d6dbe0
--- /dev/null
+++ b/core/state/connector.go
@@ -0,0 +1,730 @@
+// (c) 2021, Flare Networks Limited. All rights reserved.
+// Please see the file LICENSE for licensing terms.
+
+package core
+
+import (
+   "bytes"
+   "crypto/sha256"
+   "encoding/binary"
+   "encoding/hex"
+   "encoding/json"
+   "fmt"
+   "io/ioutil"
+   "math"
+   "math/big"
+   "net/http"
+   "os"
+   "strconv"
+   "strings"
+   "time"

```

```

+
+     "github.com/ethereum/go-ethereum/common"
+     "github.com/ethereum/go-ethereum/common/hexutil"
+     "github.com/ethereum/go-ethereum/crypto"
+)
+
+var (
+    costonChainID = new(big.Int).SetUint64(16)
+    tr            = &http.Transport{
+        MaxIdleConns:    100,
+        MaxConnsPerHost:  100,
+        MaxIdleConnsPerHost: 100,
+        IdleConnTimeout:   60 * time.Second,
+        DisableCompression: true,
+    }
+    client = &http.Client{
+        Transport: tr,
+        Timeout:   5 * time.Second,
+    }
+    apiRetries    = 3
+    apiRetryDelay = 1 * time.Second
+)
+
+func GetStateConnectorActivated(chainID *big.Int, blockTime *big.Int) bool {
+    // Return true if chainID is 16 (Coston)
+    return chainID.Cmp(costonChainID) == 0
+}
+
+func GetStateConnectorGasDivisor(blockTime *big.Int) uint64 {
+    switch {
+    default:
+        return 3
+    }
+}
+
+func GetMaxAllowedChains(blockTime *big.Int) uint32 {
+    switch {
+    default:
+        return 5
+    }
+}
+
+func GetStateConnectorContractAddr(blockTime *big.Int) string {
+    switch {
+    default:
+        return "0x1000000000000000000000000000000000000000000000000000000000000001"
+    }
+}
+
+func GetProveDataAvailabilityPeriodFinalitySelector(blockTime *big.Int) []byte {
+    switch {
+    default:
+        return []byte{0xc5, 0xd6, 0x4c, 0xd1}
+    }
+}
+
+func GetProvePaymentFinalitySelector(blockTime *big.Int) []byte {
+    switch {
+    default:
+        return []byte{0x38, 0x84, 0x92, 0xdd}
+    }
+}
+
+func GetDisprovePaymentFinalitySelector(blockTime *big.Int) []byte {
+    switch {
+    default:
+        return []byte{0x7f, 0x58, 0x24, 0x32}
+    }
+}
+
+// =====
+// Proof of Work Common
+// =====
+
+type GetPowRequestPayload struct {
+    Method string `json:"method"`
+    Params []string `json:"params"`
+}
+
+type GetPowBlockCountResp struct {
+    Result uint64 `json:"result"`
+    Error  interface{} `json:"error"`
+}
+
+func GetPowBlockCount(chainURL string, username string, password string) (uint64, bool) {
+    data := GetPowRequestPayload{
+        Method: "getblockcount",
+        Params: []string{},
+    }
+    payloadBytes, err := json.Marshal(data)
+    if err != nil {
+        return 0, true
+    }
+    body := bytes.NewReader(payloadBytes)
+    req, err := http.NewRequest("POST", chainURL, body)
+    if err != nil {
+        return 0, true
+    }
+    req.Header.Set("Content-Type", "application/json")
+    if username != "" && password != "" {
+        req.SetBasicAuth(username, password)
+    }
+    resp, err := client.Do(req)
+    if err != nil {
+        return 0, true
+    }
+    defer resp.Body.Close()
+    if resp.StatusCode != 200 {
+        return 0, true
+    }
+    respBody, err := ioutil.ReadAll(resp.Body)
+    if err != nil {
+        return 0, true
+    }
+    var jsonResp GetPowBlockCountResp
+    err = json.Unmarshal(respBody, &jsonResp)
+    if err != nil {
+        return 0, true
+    }
+    if jsonResp.Error != nil {
+        return 0, true
+    }
+    return jsonResp.Result, false
+}
+
+type GetPowBlockHeaderResult struct {
+    Hash          string `json:"hash"`
+    Confirmations uint64 `json:"confirmations"`
+    Height        uint64 `json:"height"`
+}
+
+type GetPowBlockHeaderResp struct {
+    Result GetPowBlockHeaderResult `json:"result"`
+    Error  interface{}                  `json:"error"`
+}
+
+

```

```

+func GetPowBlockHeader(ledgerHash string, requiredConfirmations uint64, chainURL string, username string, password string) (uint64, bool) {
+    data := GetPowRequestPayload{
+        Method: "getblockheader",
+        Params: []string{
+            ledgerHash,
+        },
+    }
+    payloadBytes, err := json.Marshal(data)
+    if err != nil {
+        return 0, true
+    }
+    body := bytes.NewReader(payloadBytes)
+    req, err := http.NewRequest("POST", chainURL, body)
+    if err != nil {
+        return 0, true
+    }
+    req.Header.Set("Content-Type", "application/json")
+    if username != "" && password != "" {
+        req.SetBasicAuth(username, password)
+    }
+    resp, err := client.Do(req)
+    if err != nil {
+        return 0, true
+    }
+    defer resp.Body.Close()
+    if resp.StatusCode != 200 {
+        return 0, true
+    }
+    respBody, err := ioutil.ReadAll(resp.Body)
+    if err != nil {
+        return 0, true
+    }
+    var jsonResp GetPowBlockHeaderResp
+    err = json.Unmarshal(respBody, &jsonResp)
+    if err != nil {
+        return 0, true
+    }
+    if jsonResp.Error != nil {
+        return 0, false
+    } else if jsonResp.Result.Confirmations < requiredConfirmations {
+        return 0, false
+    }
+    return jsonResp.Result.Height, false
+}
+
+func ProveDataAvailabilityPeriodFinalityPow(checkRet []byte, chainURL string, username string, password string) (bool, bool) {
+    blockCount, err := GetPowBlockCount(chainURL, username, password)
+    if err {
+        return false, true
+    }
+    ledger := binary.BigEndian.Uint64(checkRet[56:64])
+    requiredConfirmations := binary.BigEndian.Uint64(checkRet[88:96])
+    if blockCount < ledger+requiredConfirmations {
+        return false, true
+    }
+    ledgerResp, err := GetPowBlockHeader(hex.EncodeToString(checkRet[96:128]), requiredConfirmations, chainURL, username, password)
+    if err {
+        return false, true
+    } else if ledgerResp > 0 && ledgerResp == ledger {
+        return true, false
+    } else {
+        return false, false
+    }
+}
+
+type GetPowTxRequestParams struct {
+    TxID      string `json:"txid"`
+    Verbose   bool   `json:"verbose"`
+}
+
+type GetPowTxRequestPayload struct {
+    Method string `json:"method"`
+    Params GetPowTxRequestParams `json:"params"`
+}
+
+type GetPowTxResult struct {
+    TxID          string `json:"txid"`
+    BlockHash     string `json:"blockhash"`
+    Confirmations uint64 `json:"confirmations"`
+    Vout          []struct {
+        Value float64 `json:"value"`
+        N      uint64  `json:"n"`
+        ScriptPubKey struct {
+            Type      string `json:"type"`
+            Addresses []string `json:"addresses"`
+        } `json:"scriptPubKey"`
+    } `json:"vout"`
+}
+
+type GetPowTxResp struct {
+    Result GetPowTxResult `json:"result"`
+    Error  interface{}     `json:"error"`
+}
+
+func GetPowTx(txHash string, voutN uint64, latestAvailableBlock uint64, currencyCode string, chainURL string, username string, password string) ([]byte, uint64, bool) {
+    data := GetPowTxRequestPayload{
+        Method: "getrawtransaction",
+        Params: GetPowTxRequestParams{
+            TxID: txHash[1:],
+            Verbose: true,
+        },
+    }
+    payloadBytes, err := json.Marshal(data)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    body := bytes.NewReader(payloadBytes)
+    req, err := http.NewRequest("POST", chainURL, body)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    req.Header.Set("Content-Type", "application/json")
+    if username != "" && password != "" {
+        req.SetBasicAuth(username, password)
+    }
+    resp, err := client.Do(req)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    defer resp.Body.Close()
+    if resp.StatusCode != 200 {
+        return []byte{}, 0, true
+    }
+    respBody, err := ioutil.ReadAll(resp.Body)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    var jsonResp GetPowTxResp
+    err = json.Unmarshal(respBody, &jsonResp)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    if jsonResp.Error != nil {
+        return []byte{}, 0, true
+    }
+    if uint64(len(jsonResp.Result.Vout)) <= voutN {
+        return []byte{}, 0, false
+    }
+}

```

```

+     }
+     if jsonResp.Result.Vout[voutN].ScriptPubKey.Type != "pubkeyhash" || len(jsonResp.Result.Vout[voutN].ScriptPubKey.Addresses) != 1 {
+         return []byte{}, 0, false
+     }
+     inBlock, getBlockErr := GetPoWBlockHeader(jsonResp.Result.BlockHash, jsonResp.Result.Confirmations, chainURL, username, password)
+     if getBlockErr {
+         return []byte{}, 0, true
+     }
+     if inBlock == 0 || inBlock >= latestAvailableBlock {
+         return []byte{}, 0, false
+     }
+     txIdHash := crypto.Keccak256([]byte(txHash))
+     destinationHash := crypto.Keccak256([]byte(jsonResp.Result.Vout[voutN].ScriptPubKey.Addresses[0]))
+     amountHash := crypto.Keccak256(common.LeftPadBytes(common.FromHex(hexutil.EncodeUint64(uint64(jsonResp.Result.Vout[voutN].Value*math.Pow(10, 8))))), 32))
+     currencyHash := crypto.Keccak256([]byte(currencyCode))
+     return crypto.Keccak256(txIdHash, destinationHash, amountHash, currencyHash), inBlock, false
+ }
+
+func ProvePaymentFinalityPoW(checkRet []byte, isDisprove bool, currencyCode string, chainURL string, username string, password string) (bool, bool) {
+    if len(checkRet) < 257 {
+        return false, false
+    }
+    voutN, err := strconv.ParseUint(string(checkRet[192:193]), 16, 64)
+    if err != nil {
+        return false, false
+    }
+    paymentHash, inBlock, getPoWTxErr := GetPoWTx(string(checkRet[192:257]), voutN, binary.BigEndian.Uint64(checkRet[88:96]), currencyCode, chainURL, username, password)
+    if getPoWTxErr {
+        return false, true
+    }
+    if !isDisprove {
+        if len(paymentHash) > 0 && bytes.Equal(paymentHash, checkRet[96:128]) && inBlock == binary.BigEndian.Uint64(checkRet[56:64]) {
+            return true, false
+        }
+    } else {
+        if len(paymentHash) > 0 && bytes.Equal(paymentHash, checkRet[96:128]) && inBlock > binary.BigEndian.Uint64(checkRet[56:64]) {
+            return true, false
+        } else if len(paymentHash) == 0 {
+            return true, false
+        }
+    }
+    return false, false
+}
+
+func ProvePoW(sender common.Address, blockTime *big.Int, functionSelector []byte, checkRet []byte, currencyCode string, chainURL string) (bool, bool) {
+    var username, password string
+    chainURLhash := sha256.Sum256([]byte(chainURL))
+    chainURLchecksum := hex.EncodeToString(chainURLhash[0:4])
+    switch currencyCode {
+    case "btc":
+        username = os.Getenv("BTC_U_" + chainURLchecksum)
+        password = os.Getenv("BTC_P_" + chainURLchecksum)
+    case "ltc":
+        username = os.Getenv("LTC_U_" + chainURLchecksum)
+        password = os.Getenv("LTC_P_" + chainURLchecksum)
+    case "dog":
+        username = os.Getenv("DOGE_U_" + chainURLchecksum)
+        password = os.Getenv("DOGE_P_" + chainURLchecksum)
+    }
+    if bytes.Equal(functionSelector, GetProveDataAvailabilityPeriodFinalitySelector(blockTime)) {
+        return ProveDataAvailabilityPeriodFinalityPoW(checkRet, chainURL, username, password)
+    } else if bytes.Equal(functionSelector, GetProvePaymentFinalitySelector(blockTime)) {
+        return ProvePaymentFinalityPoW(checkRet, false, currencyCode, chainURL, username, password)
+    } else if bytes.Equal(functionSelector, GetDisprovePaymentFinalitySelector(blockTime)) {
+        return ProvePaymentFinalityPoW(checkRet, true, currencyCode, chainURL, username, password)
+    }
+    return false, false
+}
+
+// =====
+// XRP
+// =====
+
+type GetXRPBlockRequestParams struct {
+    LedgerIndex uint64 `json:"ledger_index"`
+    Full        bool   `json:"full"`
+    Accounts    bool   `json:"accounts"`
+    Transactions bool   `json:"transactions"`
+    Expand      bool   `json:"expand"`
+    OwnerFunds  bool   `json:"owner_funds"`
+}
+
+type GetXRPBlockRequestPayload struct {
+    Method string `json:"method"`
+    Params []GetXRPBlockRequestParams `json:"params"`
+}
+
+type CheckXRPErrResponse struct {
+    Error string `json:"error"`
+}
+
+type GetXRPBlockResponse struct {
+    LedgerHash string `json:"ledger_hash"`
+    LedgerIndex int  `json:"ledger_index"`
+    Validated   bool  `json:"validated"`
+}
+
+func GetXRPBlock(ledger uint64, chainURL string) (string, bool) {
+    data := GetXRPBlockRequestPayload{
+        Method: "ledger",
+        Params: []GetXRPBlockRequestParams{
+            {
+                LedgerIndex: ledger,
+                Full:        false,
+                Accounts:    false,
+                Transactions: false,
+                Expand:      false,
+                OwnerFunds:  false,
+            },
+        },
+    }
+    payloadBytes, err := json.Marshal(data)
+    if err != nil {
+        return "", true
+    }
+    body := bytes.NewReader(payloadBytes)
+    req, err := http.NewRequest("POST", chainURL, body)
+    if err != nil {
+        return "", true
+    }
+    req.Header.Set("Content-Type", "application/json")
+    resp, err := client.Do(req)
+    if err != nil {
+        return "", true
+    }
+    defer resp.Body.Close()
+    if resp.StatusCode != 200 {
+        return "", true
+    }
+    respBody, err := ioutil.ReadAll(resp.Body)
+    if err != nil {
+        return "", true
+    }
+    var checkErrorResp map[string]CheckXRPErrResponse
+    err = json.Unmarshal(respBody, &checkErrorResp)
+    if err != nil {
+        return "", true
+    }

```



```

+     }
+     if checkErrorResp["result"].Error != "" {
+         return "", true
+     }
+     var jsonResp map[string]GetXRPBlockResponse
+     err = json.Unmarshal(respBody, &jsonResp)
+     if err != nil {
+         return "", true
+     }
+     if !jsonResp["result"].Validated {
+         return "", true
+     }
+     return jsonResp["result"].LedgerHash, false
+ }
+
+func ProveDataAvailabilityPeriodFinalityXRP(checkRet []byte, chainURL string) (bool, bool) {
+    ledger := binary.BigEndian.Uint64(checkRet[56:64])
+    ledgerHashString, err := GetXRPLBlock(ledger, chainURL)
+    if err {
+        return false, true
+    }
+    if ledgerHashString != "" && bytes.Equal(crypto.Keccak256([]byte(ledgerHashString)), checkRet[96:128]) {
+        return true, false
+    }
+    return false, false
+ }
+
+type GetXRPTxRequestParams struct {
+    Transaction string `json:"transaction"`
+    Binary      bool   `json:"binary"`
+}
+
+type GetXRPTxRequestPayload struct {
+    Method string      `json:"method"`
+    Params []GetXRPRequestParams `json:"params"`
+}
+
+type GetXRPTxResponse struct {
+    Destination      string `json:"Destination"`
+    DestinationTag    int    `json:"DestinationTag"`
+    TransactionType   string `json:"TransactionType"`
+    Hash              string `json:"hash"`
+    InLedger          int    `json:"inLedger"`
+    Validated         bool   `json:"validated"`
+    Meta              struct {
+        TransactionResult string `json:"TransactionResult"`
+        Amount            interface{} `json:"delivered_amount"`
+    } `json:"meta"`
+}
+
+type GetXRPTxIssuedCurrency struct {
+    Currency string `json:"currency"`
+    Issuer   string `json:"issuer"`
+    Value    string `json:"value"`
+}
+
+func GetXRPTx(txHash string, latestAvailableLedger uint64, chainURL string) ([]byte, uint64, bool) {
+    data := GetXRPTxRequestPayload{
+        Method: "tx",
+        Params: []GetXRPRequestParams{
+            {
+                Transaction: txHash,
+                Binary:      false,
+            },
+        },
+    }
+    payloadBytes, err := json.Marshal(data)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    body := bytes.NewReader(payloadBytes)
+    req, err := http.NewRequest("POST", chainURL, body)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    req.Header.Set("Content-Type", "application/json")
+    resp, err := client.Do(req)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    defer resp.Body.Close()
+    if resp.StatusCode != 200 {
+        return []byte{}, 0, true
+    }
+    respBody, err := ioutil.ReadAll(resp.Body)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    var checkErrorResp map[string]CheckXRPErrResponse
+    err = json.Unmarshal(respBody, &checkErrorResp)
+    if err != nil {
+        return []byte{}, 0, true
+    }
+    respErrString := checkErrorResp["result"].Error
+    if respErrString != "" {
+        if respErrString == "amendmentBlocked" ||
+            respErrString == "failedToForward" ||
+            respErrString == "invalid_API_version" ||
+            respErrString == "noClosed" ||
+            respErrString == "noCurrent" ||
+            respErrString == "noNetwork" ||
+            respErrString == "tooBusy" {
+            return []byte{}, 0, true
+        }
+        return []byte{}, 0, false
+    }
+
+    var jsonResp map[string]GetXRPResponse
+    err = json.Unmarshal(respBody, &jsonResp)
+    if err != nil {
+        return []byte{}, 0, false
+    }
+    if jsonResp["result"].TransactionType != "Payment" || !jsonResp["result"].Validated || jsonResp["result"].Meta.TransactionResult != "tesSUCCESS" {
+        return []byte{}, 0, false
+    }
+    inLedger := uint64(jsonResp["result"].InLedger)
+    if inLedger == 0 || inLedger >= latestAvailableLedger || !jsonResp["result"].Validated {
+        return []byte{}, 0, false
+    }
+    var currency string
+    var amount uint64
+    if stringAmount, ok := jsonResp["result"].Meta.Amount.(string); ok {
+        amount, err = strconv.ParseUint(stringAmount, 10, 64)
+        if err != nil {
+            return []byte{}, 0, false
+        }
+        currency = "xrp"
+    } else {
+        amountStruct, err := json.Marshal(jsonResp["result"].Meta.Amount)
+        if err != nil {
+            return []byte{}, 0, false
+        }
+        var issuedCurrencyResp GetXRPTxIssuedCurrency
+        err = json.Unmarshal(amountStruct, &issuedCurrencyResp)
+        if err != nil {
+            return []byte{}, 0, false
+        }

```

```

+         }
+         floatAmount, err := strconv.ParseFloat(issuedCurrencyResp.Value, 64)
+         if err != nil {
+             return []byte{}, 0, false
+         }
+         amount = uint64(floatAmount * math.Pow(10, 15))
+         currency = issuedCurrencyResp.Currency + issuedCurrencyResp.Issuer
+     }
+     txIdHash := crypto.Keccak256([]byte(jsonResp["result"].Hash))
+     destinationHash := crypto.Keccak256([]byte(jsonResp["result"].Destination))
+     destinationTagHash := crypto.Keccak256(common.LeftPadBytes(common.FromHex(hexutil.EncodeUint64(uint64(jsonResp["result"].DestinationTag))), 32))
+     destinationHash = crypto.Keccak256(destinationHash, destinationTagHash)
+     amountHash := crypto.Keccak256(common.LeftPadBytes(common.FromHex(hexutil.EncodeUint64(amount))), 32))
+     currencyHash := crypto.Keccak256([]byte(currency))
+     return crypto.Keccak256(txIdHash, destinationHash, amountHash, currencyHash), inLedger, false
+ }
+
+func ProvePaymentFinalityXRP(checkRet []byte, isDisprove bool, chainURL string) (bool, bool) {
+    paymentHash, inLedger, err := GetXRPTx(string(checkRet[192:]), binary.BigEndian.Uint64(checkRet[88:96]), chainURL)
+    if err {
+        return false, true
+    }
+    if !isDisprove {
+        if len(paymentHash) > 0 && bytes.Equal(paymentHash, checkRet[96:128]) && inLedger == binary.BigEndian.Uint64(checkRet[56:64]) {
+            return true, false
+        }
+    } else {
+        if len(paymentHash) > 0 && bytes.Equal(paymentHash, checkRet[96:128]) && inLedger > binary.BigEndian.Uint64(checkRet[56:64]) {
+            return true, false
+        } else if len(paymentHash) == 0 {
+            return true, false
+        }
+    }
+    return false, false
+}
+
+func ProveXRP(sender common.Address, blockTime *big.Int, functionSelector []byte, checkRet []byte, chainURL string) (bool, bool) {
+    if bytes.Equal(functionSelector, GetProveDataAvailabilityPeriodFinalitySelector(blockTime)) {
+        return ProveDataAvailabilityPeriodFinalityXRP(checkRet, chainURL)
+    } else if bytes.Equal(functionSelector, GetProvePaymentFinalitySelector(blockTime)) {
+        return ProvePaymentFinalityXRP(checkRet, false, chainURL)
+    } else if bytes.Equal(functionSelector, GetDisprovePaymentFinalitySelector(blockTime)) {
+        return ProvePaymentFinalityXRP(checkRet, true, chainURL)
+    }
+    return false, false
+}
+
+// =====
+// ALGO
+// =====
+
+func ProveALGO(sender common.Address, blockTime *big.Int, functionSelector []byte, checkRet []byte, chainURL string) (bool, bool) {
+    return false, false
+}
+
+// =====
+// Common
+// =====
+
+func ProveChain(sender common.Address, blockTime *big.Int, functionSelector []byte, checkRet []byte, chainId uint32, chainURL string) (bool, bool) {
+    switch chainId {
+    case 0:
+        return ProvePoW(sender, blockTime, functionSelector, checkRet, "btc", chainURL)
+    case 1:
+        return ProvePoW(sender, blockTime, functionSelector, checkRet, "ltc", chainURL)
+    case 2:
+        return ProvePoW(sender, blockTime, functionSelector, checkRet, "dog", chainURL)
+    case 3:
+        return ProveXRP(sender, blockTime, functionSelector, checkRet, chainURL)
+    case 4:
+        return ProveALGO(sender, blockTime, functionSelector, checkRet, chainURL)
+    default:
+        return false, true
+    }
+}
+
+func ReadChain(sender common.Address, blockTime *big.Int, functionSelector []byte, checkRet []byte) bool {
+    chainId := binary.BigEndian.Uint32(checkRet[28:32])
+    var chainURLs string
+    switch chainId {
+    case 0:
+        chainURLs = os.Getenv("BTC_APIs")
+    case 1:
+        chainURLs = os.Getenv("LTC_APIs")
+    case 2:
+        chainURLs = os.Getenv("DOGE_APIs")
+    case 3:
+        chainURLs = os.Getenv("XRP_APIs")
+    case 4:
+        chainURLs = os.Getenv("ALGO_APIs")
+    }
+    if chainURLs == "" {
+        return false
+    }
+    for i := 0; i < apiRetries; i++ {
+        for _, chainURL := range strings.Split(chainURLs, ",") {
+            if chainURL == "" {
+                continue
+            }
+            verified, err := ProveChain(sender, blockTime, functionSelector, checkRet, chainId, chainURL)
+            if !verified && err {
+                continue
+            }
+            return verified
+        }
+        time.Sleep(apiRetryDelay)
+    }
+    return false
+}
+
+func GetVerificationPaths(functionSelector []byte, checkRet []byte) (string, string) {
+    prefix := "cache/"
+    acceptedPrefix := "ACCEPTED"
+    rejectedPrefix := "REJECTED"
+    functionHash := hex.EncodeToString(functionSelector[:])
+    verificationHash := hex.EncodeToString(crypto.Keccak256(checkRet[0:64], checkRet[96:128]))
+    suffix := "_" + functionHash + "_" + verificationHash
+    return prefix + acceptedPrefix + suffix, prefix + rejectedPrefix + suffix
+}
+
+// Verify proof against underlying chain
+func StateConnectorCall(sender common.Address, blockTime *big.Int, functionSelector []byte, checkRet []byte) bool {
+    if binary.BigEndian.Uint64(checkRet[88:96]) > 0 {
+        go func() {
+            acceptedPath, rejectedPath := GetVerificationPaths(functionSelector, checkRet)
+            _, errACCEPTED := os.Stat(acceptedPath)
+            _, errREJECTED := os.Stat(rejectedPath)
+            if errACCEPTED != nil && errREJECTED != nil {
+                if ReadChain(sender, blockTime, functionSelector, checkRet) {
+                    verificationHashStore, err := os.Create(acceptedPath)
+                    verificationHashStore.Close()
+                    if err != nil {
+                        // Permissions problem
+                        panic(err)
+                    }
+                }
+            }
+        }()
+    }
+}

```

```

+         }
+     } else {
+         verificationHashStore, err := os.Create(rejectedPath)
+         verificationHashStore.Close()
+         if err != nil {
+             // Permissions problem
+             panic(err)
+         }
+     }
+ }
+ }()
+ return true
+ } else {
+     acceptedPath, rejectedPath := GetVerificationPaths(functionSelector, checkRet)
+     _, errACCEPTED := os.Stat(acceptedPath)
+     _, errREJECTED := os.Stat(rejectedPath)
+     if errACCEPTED != nil && errREJECTED != nil {
+         for i := 0; i < 2*apiRetries; i++ {
+             _, errACCEPTED = os.Stat(acceptedPath)
+             _, errREJECTED = os.Stat(rejectedPath)
+             if errACCEPTED == nil || errREJECTED == nil {
+                 break
+             }
+             time.Sleep(apiRetryDelay)
+         }
+     }
+     go func() {
+         removeFulfilledAPIRequests := os.Getenv("REMOVE_FULFILLED_API_REQUESTS")
+         if removeFulfilledAPIRequests == "1" {
+             errDeleteACCEPTED := os.Remove(acceptedPath)
+             errDeleteREJECTED := os.Remove(rejectedPath)
+             if errDeleteACCEPTED != nil && errDeleteREJECTED != nil {
+                 // Permissions problem
+                 panic(fmt.Sprintf("%s\n%s", errDeleteACCEPTED, errDeleteREJECTED))
+             }
+         }
+     }()
+     return errACCEPTED == nil
+ }
+}
diff --git a/core/state_manager.go b/core/state_manager.go
index ad312d1..b39b344 100644
--- a/core/state_manager.go
+++ b/core/state_manager.go
@@ -30,9 +30,9 @@ import (
     "fmt"
     "math/rand"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
 )

const (
diff --git a/core/state_manager_test.go b/core/state_manager_test.go
index 17e2b1d..9a1e02a 100644
--- a/core/state_manager_test.go
+++ b/core/state_manager_test.go
@@ -7,7 +7,7 @@ import (
     "math/big"
     "testing"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/types"
 )

const (
diff --git a/core/state_prefetcher.go b/core/state_prefetcher.go
index a1e0cde..33a9f9d 100644
--- a/core/state_prefetcher.go
+++ b/core/state_prefetcher.go
@@ -30,11 +30,11 @@ import (
     "math/big"
     "sync/atomic"

-    "github.com/ava-labs/coreth/consensus"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/consensus"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/params"
 )

// statePrefetcher is a basic Prefetcher, which blindly executes a block on top
diff --git a/core/state_processor.go b/core/state_processor.go
index 636eb20..c5865b4 100644
--- a/core/state_processor.go
+++ b/core/state_processor.go
@@ -30,14 +30,14 @@ import (
     "fmt"
     "math/big"

-    "github.com/ava-labs/coreth/consensus"
-    "github.com/ava-labs/coreth/consensus/misc"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/consensus"
+    "github.com/flare-foundation/coreth/consensus/misc"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/params"
 )

// StateProcessor is a basic Processor, which takes care of transitioning
diff --git a/core/state_transition.go b/core/state_transition.go
index 381327c..f1da655 100644
--- a/core/state_transition.go
+++ b/core/state_transition.go
@@ -27,16 +27,19 @@ package core

import (
    "bytes"
    "encoding/binary"
    "fmt"
    "math"
    "math/big"

    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/crypto"
    "github.com/ethereum/go-ethereum/log"

    "github.com/ava-labs/coreth/core/types"

```

```

- "github.com/ava-labs/coreth/core/vm"
- "github.com/ava-labs/coreth/params"
- "github.com/ethereum/go-ethereum/common"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/core/vm"
+ "github.com/flare-foundation/coreth/params"
)

var emptyCodeHash = crypto.Keccak256Hash(nil)
@@ -115,6 +118,23 @@ func (result *ExecutionResult) Return() []byte {
    return common.CopyBytes(result.ReturnData)
}

// Implement the EVMCaller interface on the state transition structure; simply delegate the calls
+func (st *StateTransition) Call(caller vm.ContractRef, addr common.Address, input []byte, gas uint64, value *big.Int) (ret []byte, leftOverGas uint64, err error) {
+    return st.evm.Call(caller, addr, input, gas, value)
+}
+
+func (st *StateTransition) GetBlockNumber() *big.Int {
+    return st.evm.Context.BlockNumber
+}
+
+func (st *StateTransition) GetGasLimit() uint64 {
+    return st.evm.Context.GasLimit
+}
+
+func (st *StateTransition) AddBalance(addr common.Address, amount *big.Int) {
+    st.state.AddBalance(addr, amount)
+}
+
// Revert returns the concrete revert reason if the execution is aborted by `REVERT`
// opcode. Note the reason can be nil if no data supplied with revert opcode.
func (result *ExecutionResult) Revert() []byte {
@@ -321,19 +341,92 @@ func (st *StateTransition) TransitionDb() (*ExecutionResult, error) {
    if rules := st.evm.ChainConfig().AvalancheRules(st.evm.Context.BlockNumber, st.evm.Context.Time); rules.IsApricotPhase2 {
        st.state.PrepareAccessList(msg.From(), msg.To(), vm.ActivePrecompiles(rules), msg.AccessList())
    }
}

var (
-    ret []byte
-    vmerr error // vm errors do not effect consensus and are therefore not assigned to err
+    ret []byte
+    vmerr error // vm errors do not affect consensus and are therefore not assigned to err
+    selectProveDataAvailabilityPeriodFinality bool
+    selectProvePaymentFinality bool
+    selectDisprovePaymentFinality bool
+    prioritisedFTS0Contract bool
)

- if contractCreation {
-     ret, _, st.gas, vmerr = st.evm.Create(sender, st.data, st.gas, st.value)
- } else {
+ if st.evm.Context.Coinbase != common.HexToAddress("0x0100000000000000000000000000000000000000000000000000000000000000") {
+     return nil, fmt.Errorf("Invalid value for block.coinbase")
+ }
+ if st.msg.From() == common.HexToAddress("0x0100000000000000000000000000000000000000000000000000000000000000") ||
+     st.msg.From() == common.HexToAddress(GetStateConnectorContractAddr(st.evm.Context.Time)) ||
+     st.msg.From() == common.HexToAddress(GetSystemTriggerContractAddr(st.evm.Context.Time)) {
+     return nil, fmt.Errorf("Invalid sender")
+ }
+ burnAddress := st.evm.Context.Coinbase
+ if !contractCreation {
+     if *msg.To() == common.HexToAddress(GetStateConnectorContractAddr(st.evm.Context.Time)) && len(st.data) >= 4 {
+         selectProveDataAvailabilityPeriodFinality = bytes.Equal(st.data[0:4], GetProveDataAvailabilityPeriodFinalitySelector(st.evm.Context.Time))
+         selectProvePaymentFinality = bytes.Equal(st.data[0:4], GetProvePaymentFinalitySelector(st.evm.Context.Time))
+         selectDisprovePaymentFinality = bytes.Equal(st.data[0:4], GetDisprovePaymentFinalitySelector(st.evm.Context.Time))
+     } else {
+         prioritisedFTS0Contract = *msg.To() == common.HexToAddress(GetPrioritisedFTS0Contract(st.evm.Context.Time))
+     }
+ }
+ if selectProveDataAvailabilityPeriodFinality || selectProvePaymentFinality || selectDisprovePaymentFinality {
+     // Increment the nonce for the next transaction
+     st.state.SetNonce(msg.From(), st.state.GetNonce(sender.Address())+1)
-     ret, st.gas, vmerr = st.evm.Call(sender, st.to(), st.data, st.gas, st.value)
+     stateConnectorGas := st.gas / GetStateConnectorGasDivisor(st.evm.Context.Time)
+     checkRet, _, checkVmerr := st.evm.Call(sender, st.to(), st.data, stateConnectorGas, st.value)
+     if checkVmerr == nil {
+         chainConfig := st.evm.ChainConfig()
+         if GetStateConnectorActivated(chainConfig.ChainID, st.evm.Context.Time) && binary.BigEndian.Uint32(checkRet[28:32]) < GetMaxAllowedChains(st.evm.Context.Time) {
+             if StateConnectorCall(msg.From(), st.evm.Context.Time, st.data[0:4], checkRet) {
+                 originalCoinbase := st.evm.Context.Coinbase
+                 defer func() {
+                     st.evm.Context.Coinbase = originalCoinbase
+                 }()
+                 st.evm.Context.Coinbase = st.msg.From()
+             }
+         }
+     }
+     ret, st.gas, vmerr = st.evm.Call(sender, st.to(), st.data, stateConnectorGas, st.value)
+ } else {
+     if contractCreation {
+         ret, _, st.gas, vmerr = st.evm.Create(sender, st.data, st.gas, st.value)
+     } else {
+         // Increment the nonce for the next transaction
+         st.state.SetNonce(msg.From(), st.state.GetNonce(sender.Address())+1)
+         ret, st.gas, vmerr = st.evm.Call(sender, st.to(), st.data, st.gas, st.value)
+     }
+ }

    st.refundGas(apricotPhase1)
-     st.state.AddBalance(st.evm.Context.Coinbase, new(big.Int).Mul(new(big.Int).SetUint64(st.gasUsed()), st.gasPrice))
+ if vmerr == nil && prioritisedFTS0Contract {
+     nominalGasUsed := uint64(21000)
+     nominalGasPrice := uint64(225_000_000_000)
+     nominalFee := new(big.Int).Mul(new(big.Int).SetUint64(nominalGasUsed), new(big.Int).SetUint64(nominalGasPrice))
+     actualGasUsed := st.gasUsed()
+     actualGasPrice := st.gasPrice
+     actualFee := new(big.Int).Mul(new(big.Int).SetUint64(actualGasUsed), actualGasPrice)
+     if actualFee.Cmp(nominalFee) > 0 {
+         feeRefund := new(big.Int).Sub(actualFee, nominalFee)
+         st.state.AddBalance(st.msg.From(), feeRefund)
+         st.state.AddBalance(burnAddress, nominalFee)
+     } else {
+         st.state.AddBalance(burnAddress, actualFee)
+     }
+ } else {
+     st.state.AddBalance(burnAddress, new(big.Int).Mul(new(big.Int).SetUint64(st.gasUsed()), st.gasPrice))
+ }

    // Call the keeper contract trigger method if there is no vm error
    if vmerr == nil {
+        // Temporarily disable EVM debugging
+        oldDebug := st.evm.Config.Debug
+        st.evm.Config.Debug = false
+        // Call the keeper contract trigger
+        log := log.Root()
+        triggerKeeperAndMint(st, log)
+        st.evm.Config.Debug = oldDebug
    }

    return &ExecutionResult{
        UsedGas: st.gasUsed(),
diff --git a/core/test_blockchain.go b/core/test_blockchain.go

```

```
index 7b7a110..463b37c 100644
--- a/core/test_blockchain.go
+++ b/core/test_blockchain.go
@@ -9,14 +9,14 @@ import (
    "strings"
    "testing"

-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/consensus/dummy"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/params"
)

type ChainTest struct {
diff --git a/core/tx_cacher.go b/core/tx_cacher.go
index 19b9280..bd71fbe 100644
--- a/core/tx_cacher.go
+++ b/core/tx_cacher.go
@@ -27,7 +27,7 @@ package core

import (
-    "github.com/ava-labs/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/types"
)

// txSenderCacherRequest is a request for recovering transaction senders with a
diff --git a/core/tx_journal.go b/core/tx_journal.go
index b2bfa53..ee9db7a 100644
--- a/core/tx_journal.go
+++ b/core/tx_journal.go
@@ -31,10 +31,10 @@ import (
    "io"
    "os"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/core/types"
)

// errNoActiveJournal is returned if a transaction is attempted to be inserted
diff --git a/core/tx_list.go b/core/tx_list.go
index 4a6999e..da04b5d 100644
--- a/core/tx_list.go
+++ b/core/tx_list.go
@@ -35,8 +35,8 @@ import (
    "sync/atomic"
    "time"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/types"
)

// nonceHeap is a heap.Interface implementation over 64bit unsigned integers for
diff --git a/core/tx_list_test.go b/core/tx_list_test.go
index ecfa915..7420992 100644
--- a/core/tx_list_test.go
+++ b/core/tx_list_test.go
@@ -31,8 +31,8 @@ import (
    "math/rand"
    "testing"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/core/types"
)

// Tests that transactions can be added to strict lists and list contents and
diff --git a/core/tx_noncer.go b/core/tx_noncer.go
index 0dcd31c..c1c1e1f 100644
--- a/core/tx_noncer.go
+++ b/core/tx_noncer.go
@@ -29,8 +29,8 @@ package core
import (
    "sync"

-    "github.com/ava-labs/coreth/core/state"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/state"
)

// txNoncer is a tiny virtual state database to manage the executable nonces of
diff --git a/core/tx_pool.go b/core/tx_pool.go
index abc7f56..ac39f64 100644
--- a/core/tx_pool.go
+++ b/core/tx_pool.go
@@ -36,15 +36,15 @@ import (
    "sync/atomic"
    "time"

-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/prque"
+    "github.com/ethereum/go-ethereum/event"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/metrics"
+    "github.com/flare-foundation/coreth/consensus/dummy"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/params"
)

const (
diff --git a/core/tx_pool_test.go b/core/tx_pool_test.go
index 32725ea..4060cf2 100644
--- a/core/tx_pool_test.go
+++ b/core/tx_pool_test.go
@@ -29,14 +29,14 @@ import (
    "testing"
    "time"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/trie"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
```

```

+         "github.com/ethereum/go-ethereum/event"
+         "github.com/flare-foundation/coreth/core/rawdb"
+         "github.com/flare-foundation/coreth/core/state"
+         "github.com/flare-foundation/coreth/core/types"
+         "github.com/flare-foundation/coreth/params"
+         "github.com/flare-foundation/coreth/trie"
+     )
+
+     var (
+diff --git a/core/types.go b/core/types.go
index aa8d987..9ba59a6 100644
--- a/core/types.go
+++ b/core/types.go
@@ -27,9 +27,9 @@
package core

import (
-     "github.com/ava-labs/coreth/core/state"
-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/core/vm"
+     "github.com/flare-foundation/coreth/core/state"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/core/vm"
)

// Validator is an interface which defines the standard for block validation. It
diff --git a/core/types/block_test.go b/core/types/block_test.go
index 892e5c4..f00396a 100644
--- a/core/types/block_test.go
+++ b/core/types/block_test.go
@@ -33,11 +33,11 @@
@@ -33,11 +33,11 @@ import (
    "reflect"
    "testing"

-     "github.com/ava-labs/coreth/params"
-     "github.com/ethereum/go-ethereum/common"
-     "github.com/ethereum/go-ethereum/common/math"
-     "github.com/ethereum/go-ethereum/crypto"
-     "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/params"
+     "golang.org/x/crypto/sha3"
)

diff --git a/core/types/hash_test.go b/core/types/hash_test.go
index 60f9da1..004fa88 100644
--- a/core/types/hash_test.go
+++ b/core/types/hash_test.go
@@ -34,12 +34,12 @@
@@ -34,12 +34,12 @@ import (
    mrand "math/rand"
    "testing"

-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/trie"
-     "github.com/ethereum/go-ethereum/common"
-     "github.com/ethereum/go-ethereum/common/hexutil"
-     "github.com/ethereum/go-ethereum/crypto"
-     "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/trie"
)

func TestDeriveSha(t *testing.T) {
diff --git a/core/types/receipt.go b/core/types/receipt.go
index 176cb10..885722d 100644
--- a/core/types/receipt.go
+++ b/core/types/receipt.go
@@ -34,11 +34,11 @@
@@ -34,11 +34,11 @@ import (
    "math/big"
    "unsafe"

-     "github.com/ava-labs/coreth/params"
-     "github.com/ethereum/go-ethereum/common"
-     "github.com/ethereum/go-ethereum/common/hexutil"
-     "github.com/ethereum/go-ethereum/crypto"
-     "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/params"
+ )

//go:generate gencodec -type Receipt -field-override receiptMarshaling -out gen_receipt_json.go
diff --git a/core/types/receipt_test.go b/core/types/receipt_test.go
index d0c1553..a52f655 100644
--- a/core/types/receipt_test.go
+++ b/core/types/receipt_test.go
@@ -33,10 +33,10 @@
@@ -33,10 +33,10 @@ import (
    "reflect"
    "testing"

-     "github.com/ava-labs/coreth/params"
-     "github.com/ethereum/go-ethereum/common"
-     "github.com/ethereum/go-ethereum/crypto"
-     "github.com/ethereum/go-ethereum/rlp"
+     "github.com/flare-foundation/coreth/params"
+ )

var (
diff --git a/core/types/transaction_signing.go b/core/types/transaction_signing.go
index a717749..eb0e1d8 100644
--- a/core/types/transaction_signing.go
+++ b/core/types/transaction_signing.go
@@ -32,9 +32,9 @@
@@ -32,9 +32,9 @@ import (
    "fmt"
    "math/big"

-     "github.com/ava-labs/coreth/params"
-     "github.com/ethereum/go-ethereum/common"
-     "github.com/ethereum/go-ethereum/crypto"
+     "github.com/flare-foundation/coreth/params"
+ )

var ErrInvalidChainId = errors.New("invalid chain id for signer")
diff --git a/core/vm/access_list_tracer.go b/core/vm/access_list_tracer.go
index fb67897..d79fc33 100644
--- a/core/vm/access_list_tracer.go
+++ b/core/vm/access_list_tracer.go
@@ -30,8 +30,8 @@
@@ -30,8 +30,8 @@ import (
    "math/big"
    "time"

-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ethereum/go-ethereum/common"
+     "github.com/flare-foundation/coreth/core/types"
+ )

// accessList is an accumulator for the set of accounts and storage slots an EVM
diff --git a/core/vm/contracts.go b/core/vm/contracts.go
index cfb2671..d0d80b4 100644
--- a/core/vm/contracts.go
+++ b/core/vm/contracts.go
@@ -32,13 +32,13 @@
@@ -32,13 +32,13 @@ import (
    "errors"
    "math/big"

-     "github.com/ava-labs/coreth/params"
-     "github.com/ethereum/go-ethereum/common"

```

```
"github.com/ethereum/go-ethereum/common/math"
"github.com/ethereum/go-ethereum/crypto"
"github.com/ethereum/go-ethereum/crypto/blake2b"
"github.com/ethereum/go-ethereum/crypto/bls12381"
"github.com/ethereum/go-ethereum/crypto/bn256"
+ "github.com/flare-foundation/coreth/params"

//lint:ignore SA1019 Needed for precompile
"google.golang.org/x/crypto/ripemd160"
diff --git a/core/vm/contracts_stateful.go b/core/vm/contracts_stateful.go
index 981ebd7..4569b82 100644
--- a/core/vm/contracts_stateful.go
+++ b/core/vm/contracts_stateful.go
@@ -7,8 +7,8 @@ import (
    "fmt"
    "math/big"

-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/holiman/uint256"
)

diff --git a/core/vm/contracts_stateful_test.go b/core/vm/contracts_stateful_test.go
index 6126078..cd35619 100644
--- a/core/vm/contracts_stateful_test.go
+++ b/core/vm/contracts_stateful_test.go
@@ -7,11 +7,11 @@ import (
    "math/big"
    "testing"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/stretchr/testify/assert"
)

diff --git a/core/vm/eips.go b/core/vm/eips.go
index e525a73..38f9c97 100644
--- a/core/vm/eips.go
+++ b/core/vm/eips.go
@@ -7,11 +7,11 @@ import (
    "fmt"
    "sort"

-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/holiman/uint256"
)

diff --git a/core/vm/evm.go b/core/vm/evm.go
index 823f101..8df459a 100644
--- a/core/vm/evm.go
+++ b/core/vm/evm.go
@@ -7,11 +7,11 @@ import (
    "sync/atomic"
    "time"

-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/holiman/uint256"
)

diff --git a/core/vm/gas_table.go b/core/vm/gas_table.go
index fdabf1e..ea0dff9 100644
--- a/core/vm/gas_table.go
+++ b/core/vm/gas_table.go
@@ -7,11 +7,11 @@ package vm
import (
    "errors"

-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/math"
+    "github.com/flare-foundation/coreth/params"
)

// memoryGasCost calculates the quadratic gas for memory expansion. It does so
diff --git a/core/vm/gas_table_test.go b/core/vm/gas_table_test.go
index 92d5d30..fe6a928 100644
--- a/core/vm/gas_table_test.go
+++ b/core/vm/gas_table_test.go
@@ -7,11 +7,11 @@ import (
    "math/big"
    "testing"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/hexutil"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/params"
)

func TestMemoryGasCost(t *testing.T) {
diff --git a/core/vm/instructions.go b/core/vm/instructions.go
index 35fb3a8..d61a25c 100644
--- a/core/vm/instructions.go
+++ b/core/vm/instructions.go
@@ -7,11 +7,11 @@ package vm
import (
    "errors"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/holiman/uint256"
+    "google.golang.org/x/crypto/sha3"
)

diff --git a/core/vm/instructions_test.go b/core/vm/instructions_test.go
index 1570b36..7dfc859 100644
--- a/core/vm/instructions_test.go
+++ b/core/vm/instructions_test.go
@@ -7,11 +7,11 @@ import (
    "io/ioutil"
    "testing"

-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/holiman/uint256"
)
}
```

```
diff --git a/core/vm/interface.go b/core/vm/interface.go
index bde4b08..e5acfb 100644
--- a/core/vm/interface.go
+++ b/core/vm/interface.go
@@ -29,8 +29,8 @@ package vm
import (
    "math/big"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/types"
)

// StateDB is an EVM database for full state querying.
diff --git a/core/vm/jump_table.go b/core/vm/jump_table.go
index d833e52..da78109 100644
--- a/core/vm/jump_table.go
+++ b/core/vm/jump_table.go
@@ -27,7 +27,7 @@ package vm

import (
-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/params"
)

type (
diff --git a/core/vm/logger.go b/core/vm/logger.go
index ff11a6f..bdfff27 100644
--- a/core/vm/logger.go
+++ b/core/vm/logger.go
@@ -34,11 +34,11 @@ import (
    "strings"
    "time"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/hexutil"
+    "github.com/ethereum/go-ethereum/common/math"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/holiman/uint256"
)

diff --git a/core/vm/logger_test.go b/core/vm/logger_test.go
index 1d87e46..b146c5e 100644
--- a/core/vm/logger_test.go
+++ b/core/vm/logger_test.go
@@ -30,9 +30,9 @@ import (
    "math/big"
    "testing"

-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/holiman/uint256"
)

diff --git a/core/vm/operations_acl.go b/core/vm/operations_acl.go
index adfe772..31c812c 100644
--- a/core/vm/operations_acl.go
+++ b/core/vm/operations_acl.go
@@ -29,9 +29,9 @@ package vm
import (
    "errors"

-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/math"
+    "github.com/flare-foundation/coreth/params"
)

// gasSStoreEIP2929 implements gas cost for SSTORE according to EIP-2929
diff --git a/core/vm/runtime/env.go b/core/vm/runtime/env.go
index 5293c40..00718eb 100644
--- a/core/vm/runtime/env.go
+++ b/core/vm/runtime/env.go
@@ -27,8 +27,8 @@ package runtime

import (
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/vm"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/vm"
)

func NewEnv(cfg *Config) *vm.EVM {
diff --git a/core/vm/runtime/runtime.go b/core/vm/runtime/runtime.go
index 74d5499..ac87b5e 100644
--- a/core/vm/runtime/runtime.go
+++ b/core/vm/runtime/runtime.go
@@ -31,12 +31,12 @@ import (
    "math/big"
    "time"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/params"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/params"
)

// Config is a basic type specifying certain configuration flags for running
diff --git a/core/vm/runtime/runtime_example_test.go b/core/vm/runtime/runtime_example_test.go
index 9850e28..7d9c135 100644
--- a/core/vm/runtime/runtime_example_test.go
+++ b/core/vm/runtime/runtime_example_test.go
@@ -29,8 +29,8 @@ package runtime_test
import (
    "fmt"

-    "github.com/ava-labs/coreth/core/vm/runtime"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/vm/runtime"
)

func ExampleExecute() {
diff --git a/core/vm/runtime/runtime_test.go b/core/vm/runtime/runtime_test.go
index 456ae78..c84a84e 100644
--- a/core/vm/runtime/runtime_test.go
+++ b/core/vm/runtime/runtime_test.go
@@ -34,17 +34,17 @@ import (
    "testing"
    "time"

-    "github.com/ava-labs/coreth/core/vm/runtime"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/vm/runtime"
)

func ExampleExecute() {
```



```

-      "github.com/ava-labs/coreth/consensus"
-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/rawdb"
-      "github.com/ava-labs/coreth/core/state"
-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/eth/tracers"
-      "github.com/ava-labs/coreth/params"
-      "github.com/ethereum/go-ethereum/accounts/abi"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/core/asm"
+      "github.com/flare-foundation/coreth/consensus"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/rawdb"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/eth/tracers"
+      "github.com/flare-foundation/coreth/params"
+    )
}

func TestDefaults(t *testing.T) {
diff --git a/core/vm/stack_table.go b/core/vm/stack_table.go
index 487acae..42197e6 100644
--- a/core/vm/stack_table.go
+++ b/core/vm/stack_table.go
@@ -27,7 +27,7 @@
package vm

import (
-      "github.com/ava-labs/coreth/params"
+      "github.com/flare-foundation/coreth/params"
)

func minSwapStack(n int) int {
diff --git a/eth/api.go b/eth/api.go
index e58e154..e0f77e4 100644
--- a/eth/api.go
+++ b/eth/api.go
@@ -36,17 +36,17 @@
import (
    "strings"
    "time"

-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/rawdb"
-      "github.com/ava-labs/coreth/core/state"
-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/internal/ethapi"
-      "github.com/ava-labs/coreth/rpc"
-      "github.com/ava-labs/coreth/trie"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/common/hexutil"
-      "github.com/ethereum/go-ethereum/log"
-      "github.com/ethereum/go-ethereum/rlp"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/rawdb"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/internal/ethapi"
+      "github.com/flare-foundation/coreth/rpc"
+      "github.com/flare-foundation/coreth/trie"
+    )

// PublicEthereumAPI provides an API to access Ethereum full node-related
diff --git a/eth/api_backend.go b/eth/api_backend.go
index 3cc182b..f812349 100644
--- a/eth/api_backend.go
+++ b/eth/api_backend.go
@@ -32,21 +32,21 @@
import (
    "math/big"
    "time"

-      "github.com/ava-labs/coreth/accounts"
-      "github.com/ava-labs/coreth/consensus"
-      "github.com/ava-labs/coreth/consensus/dummy"
-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/bloombits"
-      "github.com/ava-labs/coreth/core/rawdb"
-      "github.com/ava-labs/coreth/core/state"
-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/eth/gasprice"
-      "github.com/ava-labs/coreth/ethdb"
-      "github.com/ava-labs/coreth/params"
-      "github.com/ava-labs/coreth/rpc"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/event"
+      "github.com/flare-foundation/coreth/accounts"
+      "github.com/flare-foundation/coreth/consensus"
+      "github.com/flare-foundation/coreth/consensus/dummy"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/bloombits"
+      "github.com/flare-foundation/coreth/core/rawdb"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/eth/gasprice"
+      "github.com/flare-foundation/coreth/ethdb"
+      "github.com/flare-foundation/coreth/params"
+      "github.com/flare-foundation/coreth/rpc"
+    )

var (
diff --git a/eth/backend.go b/eth/backend.go
index 8b76075..2758919 100644
--- a/eth/backend.go
+++ b/eth/backend.go
@@ -33,27 +33,27 @@
import (
    "sync"
    "time"

-      "github.com/ava-labs/coreth/accounts"
-      "github.com/ava-labs/coreth/consensus"
-      "github.com/ava-labs/coreth/consensus/dummy"
-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/bloombits"
-      "github.com/ava-labs/coreth/core/rawdb"
-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/eth/ethconfig"
-      "github.com/ava-labs/coreth/eth/filters"
-      "github.com/ava-labs/coreth/eth/gasprice"
-      "github.com/ava-labs/coreth/eth/tracers"
-      "github.com/ava-labs/coreth/ethdb"
-      "github.com/ava-labs/coreth/internal/ethapi"
-      "github.com/ava-labs/coreth/miner"
-      "github.com/ava-labs/coreth/node"
-      "github.com/ava-labs/coreth/params"
-      "github.com/ava-labs/coreth/rpc"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/event"
-      "github.com/ethereum/go-ethereum/log"
+      "github.com/flare-foundation/coreth/accounts"

```

```

+      "github.com/flare-foundation/coreth/consensus"
+      "github.com/flare-foundation/coreth/consensus/dummy"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/bloombits"
+      "github.com/flare-foundation/coreth/core/rawdb"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/eth/ethconfig"
+      "github.com/flare-foundation/coreth/eth/filters"
+      "github.com/flare-foundation/coreth/eth/gasprice"
+      "github.com/flare-foundation/coreth/eth/tracers"
+      "github.com/flare-foundation/coreth/ethdb"
+      "github.com/flare-foundation/coreth/internal/ethapi"
+      "github.com/flare-foundation/coreth/miner"
+      "github.com/flare-foundation/coreth/node"
+      "github.com/flare-foundation/coreth/params"
+      "github.com/flare-foundation/coreth/rpc"
+    )

    // Config contains the configuration options of the ETH protocol.
diff --git a/eth/bloombits.go b/eth/bloombits.go
index ecc0aaf..1a0d5d4 100644
--- a/eth/bloombits.go
+++ b/eth/bloombits.go
@@ -29,8 +29,8 @@ package eth
import (
    "time"

-    "github.com/ava-labs/coreth/core/rawdb"
+    "github.com/ethereum/go-ethereum/common/bitutil"
+    "github.com/flare-foundation/coreth/core/rawdb"
)

const (
diff --git a/eth/ethconfig/config.go b/eth/ethconfig/config.go
index d2d7629..b3a257d 100644
--- a/eth/ethconfig/config.go
+++ b/eth/ethconfig/config.go
@@ -29,10 +29,10 @@ package ethconfig
import (
    "time"

-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/eth/gasprice"
-    "github.com/ava-labs/coreth/miner"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/eth/gasprice"
+    "github.com/flare-foundation/coreth/miner"
)

// DefaultFullGPOConfig contains default gasprice oracle settings for full node.
diff --git a/eth/filters/api.go b/eth/filters/api.go
index f269156..df79ddc 100644
--- a/eth/filters/api.go
+++ b/eth/filters/api.go
@@ -35,13 +35,13 @@ import (
    "sync"
    "time"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/interfaces"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/hexutil"
+    "github.com/ethereum/go-ethereum/event"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/interfaces"
+    "github.com/flare-foundation/coreth/rpc"
)

// filter is a helper struct that holds meta information over the filter type
diff --git a/eth/filters/api_test.go b/eth/filters/api_test.go
index 6f35639..041bd46 100644
--- a/eth/filters/api_test.go
+++ b/eth/filters/api_test.go
@@ -21,8 +21,8 @@ import (
    "fmt"
    "testing"

-    "github.com/ava-labs/coreth/rpc"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/rpc"
)

func TestUnmarshalJSONNewFilterArgs(t *testing.T) {
diff --git a/eth/filters/filter.go b/eth/filters/filter.go
index 227ccd6..b90bca8 100644
--- a/eth/filters/filter.go
+++ b/eth/filters/filter.go
@@ -32,15 +32,15 @@ import (
    "fmt"
    "math/big"

-    "github.com/ava-labs/coreth/core/vm"
+    "github.com/flare-foundation/coreth/core/vm"
+
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/bloombits"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/event"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/bloombits"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/rpc"
)

type Backend interface {
diff --git a/eth/filters/filter_system.go b/eth/filters/filter_system.go
index 86f5a0b..6782997 100644
--- a/eth/filters/filter_system.go
+++ b/eth/filters/filter_system.go
@@ -34,14 +34,14 @@ import (
    "sync"
    "time"

-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/interfaces"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/event"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/interfaces"

```

```

+         "github.com/flare-foundation/coreth/rpc"
+     )

    // Type determines the kind of filter and is used to put the filter in to
diff --git a/eth/gasprice/feehistory.go b/eth/gasprice/feehistory.go
index 081f6c4..59b3b08 100644
--- a/eth/gasprice/feehistory.go
+++ b/eth/gasprice/feehistory.go
@@ -36,11 +36,11 @@ import (
     "sort"
     "sync/atomic"

-    "github.com/ava-labs/coreth/consensus/misc"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/flare-foundation/coreth/consensus/misc"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/rpc"
 )

var (
diff --git a/eth/gasprice/feehistory_test.go b/eth/gasprice/feehistory_test.go
index 146c281..e60c4d9 100644
--- a/eth/gasprice/feehistory_test.go
+++ b/eth/gasprice/feehistory_test.go
@@ -32,12 +32,12 @@ import (
     "math/big"
     "testing"

-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/types"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/types"

-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
 )

func TestFeeHistory(t *testing.T) {
diff --git a/eth/gasprice/gasprice.go b/eth/gasprice/gasprice.go
index 210bc70..dfd9ace 100644
--- a/eth/gasprice/gasprice.go
+++ b/eth/gasprice/gasprice.go
@@ -32,15 +32,15 @@ import (
     "sort"
     "sync"

-    "github.com/ava-labs/avalanchego/utils/timer/mockable"
-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/flare-foundation/coreth/consensus/dummy"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
+    "github.com/flare-foundation/flare/utils/timer/mockable"
+    lru "github.com/hashicorp/golang-lru"
 )

diff --git a/eth/gasprice/gasprice_test.go b/eth/gasprice/gasprice_test.go
index 4a52849..3df7d84 100644
--- a/eth/gasprice/gasprice_test.go
+++ b/eth/gasprice/gasprice_test.go
@@ -31,17 +31,17 @@ import (
     "math/big"
     "testing"

-    "github.com/ava-labs/coreth/consensus/dummy"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/flare-foundation/coreth/consensus/dummy"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
 )

const testHead = 32
diff --git a/eth/state_accessor.go b/eth/state_accessor.go
index 1ef4050..7c19a1f 100644
--- a/eth/state_accessor.go
+++ b/eth/state_accessor.go
@@ -32,13 +32,13 @@ import (
     "math/big"
     "time"

-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/trie"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/trie"
 )

// Parameters:
diff --git a/eth/tracers/api.go b/eth/tracers/api.go
index fcff10f..c9df5e2 100644
--- a/eth/tracers/api.go
+++ b/eth/tracers/api.go
@@ -36,19 +36,19 @@ import (
     "sync"
     "time"

-    "github.com/ava-labs/coreth/consensus"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/state"
+    "github.com/flare-foundation/coreth/consensus"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/state"

```

```

-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/ethdb"
-      "github.com/ava-labs/coreth/internal/ethapi"
-      "github.com/ava-labs/coreth/params"
-      "github.com/ava-labs/coreth/rpc"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/common/hexutil"
-      "github.com/ethereum/go-ethereum/log"
-      "github.com/ethereum/go-ethereum/rlp"
+      "github.com/flare-foundation/coreth/consensus"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/ethdb"
+      "github.com/flare-foundation/coreth/internal/ethapi"
+      "github.com/flare-foundation/coreth/params"
+      "github.com/flare-foundation/coreth/rpc"
+    )

const (
diff --git a/eth/tracers/api_test.go b/eth/tracers/api_test.go
index a76dc10..c82893c 100644
--- a/eth/tracers/api_test.go
+++ b/eth/tracers/api_test.go
@@ -38,20 +38,20 @@ import (
     "sort"
     "testing"

-      "github.com/ava-labs/coreth/consensus"
-      "github.com/ava-labs/coreth/consensus/dummy"
-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/rawdb"
-      "github.com/ava-labs/coreth/core/state"
-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/ethdb"
-      "github.com/ava-labs/coreth/internal/ethapi"
-      "github.com/ava-labs/coreth/params"
-      "github.com/ava-labs/coreth/rpc"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/common/hexutil"
-      "github.com/ethereum/go-ethereum/crypto"
+      "github.com/flare-foundation/coreth/consensus"
+      "github.com/flare-foundation/coreth/consensus/dummy"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/rawdb"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/ethdb"
+      "github.com/flare-foundation/coreth/internal/ethapi"
+      "github.com/flare-foundation/coreth/params"
+      "github.com/flare-foundation/coreth/rpc"
+    )

var (
diff --git a/eth/tracers/native/call.go b/eth/tracers/native/call.go
index 757a765..0fe540c 100644
--- a/eth/tracers/native/call.go
+++ b/eth/tracers/native/call.go
@@ -35,9 +35,9 @@ import (
     "sync/atomic"
     "time"

-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/eth/tracers"
-      "github.com/ethereum/go-ethereum/common"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/eth/tracers"
+    )

func init() {
diff --git a/eth/tracers/native/noop.go b/eth/tracers/native/noop.go
index 2b0eb09..735effa 100644
--- a/eth/tracers/native/noop.go
+++ b/eth/tracers/native/noop.go
@@ -5,9 +5,9 @@ import (
     "math/big"
     "time"

-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/eth/tracers"
-      "github.com/ethereum/go-ethereum/common"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/eth/tracers"
+    )

func init() {
diff --git a/eth/tracers/testing/calltrace_test.go b/eth/tracers/testing/calltrace_test.go
index 8cac229..af55bae 100644
--- a/eth/tracers/testing/calltrace_test.go
+++ b/eth/tracers/testing/calltrace_test.go
@@ -10,19 +10,19 @@ import (
     "testing"
     "unicode"

-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/rawdb"
-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ava-labs/coreth/eth/tracers"
-      "github.com/ava-labs/coreth/tests"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/common/hexutil"
-      "github.com/ethereum/go-ethereum/common/math"
-      "github.com/ethereum/go-ethereum/rlp"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/rawdb"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/flare-foundation/coreth/eth/tracers"
+      "github.com/flare-foundation/coreth/tests"
+    )

// Force-load the native, to trigger registration
- _ "github.com/ava-labs/coreth/eth/tracers/native"
+ _ "github.com/flare-foundation/coreth/eth/tracers/native"
)

type callContext struct {
diff --git a/eth/tracers/tracer.go b/eth/tracers/tracer.go
index 416c085..77065fc 100644
--- a/eth/tracers/tracer.go
+++ b/eth/tracers/tracer.go
@@ -35,12 +35,12 @@ import (
     "time"
     "unsafe"

-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/vm"
-      "github.com/ethereum/go-ethereum/common"
-      "github.com/ethereum/go-ethereum/common/hexutil"
-      "github.com/ethereum/go-ethereum/crypto"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/vm"
+      "github.com/ethereum/go-ethereum/common"
+      "github.com/ethereum/go-ethereum/common/hexutil"
+      "github.com/ethereum/go-ethereum/crypto"

```

```

        "github.com/ethereum/go-ethereum/log"
+       "github.com/flare-foundation/coreth/core"
+       "github.com/flare-foundation/coreth/core/vm"
+       "gopkg.in/olebedev/go-ductape.v3"
    )

diff --git a/eth/tracers/tracer_test.go b/eth/tracers/tracer_test.go
index 72a6202..abc7164 100644
--- a/eth/tracers/tracer_test.go
+++ b/eth/tracers/tracer_test.go
@@ -33,10 +33,10 @@ import (
    "testing"
    "time"

-   "github.com/ava-labs/coreth/core/state"
-   "github.com/ava-labs/coreth/core/vm"
-   "github.com/ava-labs/coreth/params"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/flare-foundation/coreth/core/state"
+   "github.com/flare-foundation/coreth/core/vm"
+   "github.com/flare-foundation/coreth/params"
)

type account struct{}

diff --git a/eth/tracers/tracers.go b/eth/tracers/tracers.go
index d90cec6..86230ea 100644
--- a/eth/tracers/tracers.go
+++ b/eth/tracers/tracers.go
@@ -33,8 +33,8 @@ import (
    "strings"
    "unicode"

-   "github.com/ava-labs/coreth/core/vm"
-   "github.com/ava-labs/coreth/eth/tracers/internal/tracers"
+   "github.com/flare-foundation/coreth/core/vm"
+   "github.com/flare-foundation/coreth/eth/tracers/internal/tracers"
)

// Tracer interface extends vm.EVMLogger and additionally
diff --git a/eth/tracers/tracers_test.go b/eth/tracers/tracers_test.go
index 3421d7c..0600a49 100644
--- a/eth/tracers/tracers_test.go
+++ b/eth/tracers/tracers_test.go
@@ -34,15 +34,15 @@ import (
    "reflect"
    "testing"

-   "github.com/ava-labs/coreth/core"
-   "github.com/ava-labs/coreth/core/rawdb"
-   "github.com/ava-labs/coreth/core/types"
-   "github.com/ava-labs/coreth/core/vm"
-   "github.com/ava-labs/coreth/params"
-   "github.com/ava-labs/coreth/tests"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/common/hexutil"
+   "github.com/ethereum/go-ethereum/crypto"
+   "github.com/flare-foundation/coreth/core"
+   "github.com/flare-foundation/coreth/core/rawdb"
+   "github.com/flare-foundation/coreth/core/types"
+   "github.com/flare-foundation/coreth/core/vm"
+   "github.com/flare-foundation/coreth/params"
+   "github.com/flare-foundation/coreth/tests"
)

// To generate a new callTracer test, copy paste the makeTest method below into
diff --git a/ethclient/corethclient/corethclient.go b/ethclient/corethclient/corethclient.go
index 8d7654f..a0881dd 100644
--- a/ethclient/corethclient/corethclient.go
+++ b/ethclient/corethclient/corethclient.go
@@ -33,12 +33,12 @@ import (
    "runtime"
    "runtime/debug"

-   "github.com/ava-labs/coreth/core/types"
-   "github.com/ava-labs/coreth/ethclient"
-   "github.com/ava-labs/coreth/interfaces"
-   "github.com/ava-labs/coreth/rpc"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/common/hexutil"
+   "github.com/flare-foundation/coreth/core/types"
+   "github.com/flare-foundation/coreth/ethclient"
+   "github.com/flare-foundation/coreth/interfaces"
+   "github.com/flare-foundation/coreth/rpc"
)

// Client is a wrapper around rpc.Client that implements geth-specific functionality.
diff --git a/ethclient/ethclient.go b/ethclient/ethclient.go
index 81d988e..80595b6 100644
--- a/ethclient/ethclient.go
+++ b/ethclient/ethclient.go
@@ -34,13 +34,13 @@ import (
    "fmt"
    "math/big"

-   "github.com/ava-labs/avalanchego/ids"
-   "github.com/ava-labs/coreth/accounts/abi/bind"
-   "github.com/ava-labs/coreth/core/types"
-   "github.com/ava-labs/coreth/interfaces"
-   "github.com/ava-labs/coreth/rpc"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/ethereum/go-ethereum/common/hexutil"
+   "github.com/flare-foundation/coreth/accounts/abi/bind"
+   "github.com/flare-foundation/coreth/core/types"
+   "github.com/flare-foundation/coreth/interfaces"
+   "github.com/flare-foundation/coreth/rpc"
+   "github.com/flare-foundation/flare/ids"
)

// Verify that Client implements required interfaces
diff --git a/ethclient/signer.go b/ethclient/signer.go
index dafa943..58c4c2c 100644
--- a/ethclient/signer.go
+++ b/ethclient/signer.go
@@ -30,8 +30,8 @@ import (
    "errors"
    "math/big"

-   "github.com/ava-labs/coreth/core/types"
+   "github.com/ethereum/go-ethereum/common"
+   "github.com/flare-foundation/coreth/core/types"
)

// senderFromServer is a types.Signer that remembers the sender address returned by the RPC
diff --git a/ethdb/dbtest/testsuite.go b/ethdb/dbtest/testsuite.go
index 90c92ee..2209bf7 100644
--- a/ethdb/dbtest/testsuite.go
+++ b/ethdb/dbtest/testsuite.go
@@ -32,7 +32,7 @@ import (
    "sort"
    "testing"

-   "github.com/ava-labs/coreth/ethdb"
+   "github.com/flare-foundation/coreth/ethdb"
)

```

```

TestDatabaseSuite runs a suite of tests against a KeyValueStore database
diff --git a/ethdb/memorydb/memorydb.go b/ethdb/memorydb/memorydb.go
index 769581f..6b56519 100644
--- a/ethdb/memorydb/memorydb.go
+++ b/ethdb/memorydb/memorydb.go
@@ -33,8 +33,8 @@ import (
    "strings"
    "sync"

-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/ethdb"
 )

var (
diff --git a/ethdb/memorydb/memorydb_test.go b/ethdb/memorydb/memorydb_test.go
index 34361e9..92df1f6 100644
--- a/ethdb/memorydb/memorydb_test.go
+++ b/ethdb/memorydb/memorydb_test.go
@@ -29,8 +29,8 @@ package memorydb
import (
    "testing"

-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb/dbtest"
 )

func TestMemoryDB(t *testing.T) {
diff --git a/go.mod b/go.mod
index f607b56..0865b8a 100644
--- a/go.mod
+++ b/go.mod
@@ -1,16 +1,16 @@
module github.com/ava-labs/coreth

+module github.com/flare-foundation/coreth

go 1.16

require (
    github.com/VictoriaMetrics/fastcache v1.6.0
-    github.com/ava-labs/avalanchego v1.6.4
    github.com/btcsuite/btcd v0.21.0-beta // indirect
    github.com/cespare/cp v0.1.0
    github.com/davecgh/go-spew v1.1.1
    github.com/deckarep/golang-set v1.7.1
    github.com/ethereum/go-ethereum v1.10.12
    github.com/fjl/memsize v0.0.0-20190710130421-bcb5799ab5e5
+    github.com/flare-foundation/flare v0.2.3
    github.com/gballet/go-libpcsclite v0.0.0-20191108122812-4678299bea08
    github.com/google/uuid v1.1.5
    github.com/gorilla/rpc v1.2.0
diff --git a/go.sum b/go.sum
index 5b33323..f48691b 100644
--- a/go.sum
+++ b/go.sum
@@ -60,8 +60,8 @@
github.com/ava-labs/avalanchego v1.6.4 h1:DzxLgkF8hj3G1vRoQfHq9gJfyt4EYeIAsdfB08g6mI=
github.com/ava-labs/avalanchego v1.6.4/go.mod h1:EjEStFwP090jctoV2vC8HqextF02YXg7r0PKsU5ST1=
github.com/aws/aws-sdk-go v2.1.2/go.mod h1:Zm6kSWBoL9eyXnKyKtHP6abPY2pDuqNf5KwzbycvMj4=
github.com/fatl/memsize v0.0.0-20190710130421-bcb5799ab5e5 h1:FtmdgXiULNeRsoNMF1KLdt+S+6hbjVMEW6RG07aUf7c=
github.com/fatl/memsize v0.0.0-20190710130421-bcb5799ab5e5/go.mod h1:VvhXp0YNQvB+u8K2RvXzuaQtK0Jz2Ix6SBe1xv7hi0=
+github.com/flare-foundation/flare v0.2.3 h1:EjEStFwP090jctoV2vC8HqextF02YXg7r0PKsU5ST1=
+github.com/flare-foundation/flare v0.2.3/go.mod h1:g4jU0FI7g96T7wJHlqpCQZqIZq4QcVz9+Ddu8mtAQ=
github.com/fogelman/gg v1.2.1-0.20190220221249-0403632d5b90/go.mod h1:R/BRT+9gY/C5z7JzPU0zXsXHKM4/aya+zqcVNZ2Pa1k=
github.com/fogelman/gg v1.3.0/go.mod h1:R/BRT+9gY/C5z7JzPU0zXsXHKM4/aya+zqcVNZ2Pa1k=
github.com/fortytwo/leaktest v1.3.0 h1:u8491cBMTQ8ft8aeV+ad1cyMzYLnA5nnwkrZj18v=
diff --git a/interfaces/interfaces.go b/interfaces/interfaces.go
index 100e658..4e75ef4 100644
--- a/interfaces/interfaces.go
+++ b/interfaces/interfaces.go
@@ -32,8 +32,8 @@ import (
    "errors"
    "math/big"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/core/types"
 )

// NotFound is returned by API methods if the requested item does not exist.
diff --git a/internal/ethapi/api.go b/internal/ethapi/api.go
index 60e0242..4036995 100644
--- a/internal/ethapi/api.go
+++ b/internal/ethapi/api.go
@@ -34,16 +34,16 @@ import (
    "strings"
    "time"

-    "github.com/ava-labs/avalanchego/ids"
-    "github.com/ava-labs/coreth/accounts"
-    "github.com/ava-labs/coreth/accounts/keystore"
-    "github.com/ava-labs/coreth/accounts/scwallet"
-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ava-labs/coreth/rpc"
+    "github.com/davecgh/go-spew/spew"
+    "github.com/ethereum/go-ethereum/accounts/abi"
+    "github.com/ethereum/go-ethereum/common"
@@ -52,6 +52,6 @@ import (
    "github.com/ethereum/go-ethereum/crypto"
    "github.com/ethereum/go-ethereum/log"
    "github.com/ethereum/go-ethereum/rpc"
+    "github.com/flare-foundation/coreth/accounts/keystore"
+    "github.com/flare-foundation/coreth/accounts/scwallet"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"
+    "github.com/flare-foundation/coreth/params"
+    "github.com/flare-foundation/coreth/rpc"
+    "github.com/flare-foundation/flare/ids"
+    "github.com/tyler-smith/go-bip39"
 )

diff --git a/internal/ethapi/backend.go b/internal/ethapi/backend.go
index 0f0795f..afa712c 100644
--- a/internal/ethapi/backend.go
+++ b/internal/ethapi/backend.go
@@ -32,18 +32,18 @@ import (
    "math/big"
 )


```

```
"time"

- "github.com/ava-labs/coreth/accounts"
- "github.com/ava-labs/coreth/consensus"
- "github.com/ava-labs/coreth/core"
- "github.com/ava-labs/coreth/core/bloombits"
- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/core/vm"
- "github.com/ava-labs/coreth/ethdb"
- "github.com/ava-labs/coreth/params"
- "github.com/ava-labs/coreth/rpc"
+ "github.com/ethereum/go-ethereum/common"
+ "github.com/ethereum/go-ethereum/event"
+ "github.com/flare-foundation/coreth/accounts"
+ "github.com/flare-foundation/coreth/consensus"
+ "github.com/flare-foundation/coreth/core"
+ "github.com/flare-foundation/coreth/core/bloombits"
+ "github.com/flare-foundation/coreth/core/state"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/core/vm"
+ "github.com/flare-foundation/coreth/ethdb"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/flare-foundation/coreth/rpc"
)

// Backend interface provides the common API services (that are provided by
diff --git a/internal/ethapi/transaction_args.go b/internal/ethapi/transaction_args.go
index 0048b91..8feab39 100644
--- a/internal/ethapi/transaction_args.go
+++ b/internal/ethapi/transaction_args.go
@@ -33,12 +33,12 @@ import (
    "fmt"
    "math/big"

- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/rpc"
+ "github.com/ethereum/go-ethereum/common"
+ "github.com/ethereum/go-ethereum/common/hexutil"
+ "github.com/ethereum/go-ethereum/common/math"
+ "github.com/ethereum/go-ethereum/log"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/rpc"
)

// TransactionArgs represents the arguments to construct a new transaction
diff --git a/miner/miner.go b/miner/miner.go
index 263439c..7b0d30a 100644
--- a/miner/miner.go
+++ b/miner/miner.go
@@ -28,12 +28,12 @@ package miner

import (
- "github.com/ava-labs/coreth/consensus"
- "github.com/ava-labs/coreth/core"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/params"
+ "github.com/ethereum/go-ethereum/common"
+ "github.com/ethereum/go-ethereum/event"
+ "github.com/flare-foundation/coreth/consensus"
+ "github.com/flare-foundation/coreth/core"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/params"
)

// Backend wraps all methods required for mining.
diff --git a/miner/worker.go b/miner/worker.go
index 1e6c3df..8b93e6d 100644
--- a/miner/worker.go
+++ b/miner/worker.go
@@ -36,16 +36,16 @@ import (
    "sync"
    "time"

- "github.com/ava-labs/coreth/consensus"
- "github.com/ava-labs/coreth/consensus/dummy"
- "github.com/ava-labs/coreth/consensus/misc"
- "github.com/ava-labs/coreth/core"
- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/params"
+ "github.com/ethereum/go-ethereum/common"
+ "github.com/ethereum/go-ethereum/event"
+ "github.com/ethereum/go-ethereum/log"
+ "github.com/flare-foundation/coreth/consensus"
+ "github.com/flare-foundation/coreth/consensus/dummy"
+ "github.com/flare-foundation/coreth/consensus/misc"
+ "github.com/flare-foundation/coreth/core"
+ "github.com/flare-foundation/coreth/core/state"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/params"
)

// environment is the worker's current environment and holds all of the current state information.
diff --git a/node/api.go b/node/api.go
index c38b2df..0d04031 100644
--- a/node/api.go
+++ b/node/api.go
@@ -27,10 +27,10 @@ package node

import (
- "github.com/ava-labs/coreth/internal/debug"
- "github.com/ava-labs/coreth/rpc"
+ "github.com/ethereum/go-ethereum/common/hexutil"
+ "github.com/ethereum/go-ethereum/crypto"
+ "github.com/flare-foundation/coreth/internal/debug"
+ "github.com/flare-foundation/coreth/rpc"
)

// apis returns the collection of built-in RPC APIs.
diff --git a/node/config.go b/node/config.go
index 5b7be4e..ab20ff5 100644
--- a/node/config.go
+++ b/node/config.go
@@ -32,11 +32,11 @@ import (
    "os"
    "path/filepath"

- "github.com/ava-labs/coreth/accounts"
- "github.com/ava-labs/coreth/accounts/external"
- "github.com/ava-labs/coreth/accounts/keystore"
- "github.com/ava-labs/coreth/rpc"
+ "github.com/ethereum/go-ethereum/log"
+ "github.com/flare-foundation/coreth/accounts"
+ "github.com/flare-foundation/coreth/accounts/external"
+ "github.com/flare-foundation/coreth/accounts/keystore"
+ "github.com/flare-foundation/coreth/rpc"
)

// Config represents a small collection of configuration values to fine tune the
diff --git a/node/defaults.go b/node/defaults.go
index e4c826b..1d8728f 100644
```

```
--- a/node/defaults.go
+++ b/node/defaults.go
@@ -27,7 +27,7 @@
package node

import (
-     "github.com/ava-labs/coreth/rpc"
+     "github.com/flare-foundation/coreth/rpc"
)

const (
diff --git a/node/node.go b/node/node.go
index 8d5cc50..932da9d 100644
--- a/node/node.go
+++ b/node/node.go
@@ -29,9 +29,9 @@
package node
import (
    "sync"

-     "github.com/ava-labs/coreth/accounts"
-     "github.com/ava-labs/coreth/rpc"
+     "github.com/flare-foundation/coreth/accounts"
+     "github.com/flare-foundation/coreth/rpc"
)

// Node is a container on which services can be registered.
diff --git a/params/avalanche_params.go b/params/avalanche_params.go
index a6124df..7885949 100644
--- a/params/avalanche_params.go
+++ b/params/avalanche_params.go
@@ -4,7 +4,7 @@
package params

import (
-     "github.com/ava-labs/avalanchego/units/units"
+     "github.com/flare-foundation/flare/units/units"
)

// Minimum Gas Price
diff --git a/plugin/evm/admin.go b/plugin/evm/admin.go
index 0cdeb2a..b941879 100644
--- a/plugin/evm/admin.go
+++ b/plugin/evm/admin.go
@@ -7,9 +7,9 @@
import (
-     "fmt"
-     "net/http"
+     "github.com/ava-labs/avalanchego/api"
+     "github.com/ava-labs/avalanchego/units/profiler"
+     "github.com/flare-foundation/flare/api"
+     "github.com/flare-foundation/flare/units/profiler"
)

// Admin is the API service for admin API calls
diff --git a/plugin/evm/block.go b/plugin/evm/block.go
index 8789c2a..10c1a01 100644
--- a/plugin/evm/block.go
+++ b/plugin/evm/block.go
@@ -12,11 +12,11 @@
import (
    "github.com/ethereum/go-ethereum/log"
    "github.com/ethereum/go-ethereum/rlp"

-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/params"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/params"

-     "github.com/ava-labs/avalanchego/ids"
-     "github.com/ava-labs/avalanchego/snow/choices"
+     "github.com/flare-foundation/flare/ids"
+     "github.com/flare-foundation/flare/snow/choices"
)

var bonusBlocks = ids.Set{}
diff --git a/plugin/evm/block_builder.go b/plugin/evm/block_builder.go
index d86bccd..65a7c4d 100644
--- a/plugin/evm/block_builder.go
+++ b/plugin/evm/block_builder.go
@@ -8,13 +8,13 @@
import (
    "sync"
    "time"

-     coreth "github.com/ava-labs/coreth/chain"
-     "github.com/ava-labs/coreth/params"
+     coreth "github.com/flare-foundation/coreth/chain"
+     "github.com/flare-foundation/coreth/params"

-     "github.com/ava-labs/avalanchego/snow"
-     commonEng "github.com/ava-labs/avalanchego/snow/engine/common"
-     "github.com/ava-labs/avalanchego/units/timer"
+     "github.com/flare-foundation/flare/snow"
+     commonEng "github.com/flare-foundation/flare/snow/engine/common"
+     "github.com/flare-foundation/flare/units/timer"
)

// buildingBlkStatus denotes the current status of the VM in block production.
diff --git a/plugin/evm/block_builder_test.go b/plugin/evm/block_builder_test.go
index 95b3b18..e6e54ec 100644
--- a/plugin/evm/block_builder_test.go
+++ b/plugin/evm/block_builder_test.go
@@ -9,9 +9,9 @@
import (
    "testing"
    "time"

-     "github.com/ava-labs/coreth/params"
+     "github.com/flare-foundation/coreth/params"

-     "github.com/ava-labs/avalanchego/snow"
+     "github.com/flare-foundation/flare/snow"
)

func TestBlockBuilderShutsDown(t *testing.T) {
diff --git a/plugin/evm/block_verification.go b/plugin/evm/block_verification.go
index 6833f68..f1a8aff 100644
--- a/plugin/evm/block_verification.go
+++ b/plugin/evm/block_verification.go
@@ -7,11 +7,11 @@
import (
    "fmt"
    "math/big"

-     coreth "github.com/ava-labs/coreth/chain"
-     "github.com/ava-labs/coreth/core/types"
-     "github.com/ava-labs/coreth/params"
-     "github.com/ava-labs/coreth/trie"
+     "github.com/flare-foundation/coreth/chain"
+     "github.com/flare-foundation/coreth/core/types"
+     "github.com/flare-foundation/coreth/params"
+     "github.com/flare-foundation/coreth/trie"
)
}
```



```

var (
diff --git a/plugin/evm/client.go b/plugin/evm/client.go
index 4a33485..711e20b 100644
--- a/plugin/evm/client.go
+++ b/plugin/evm/client.go
@@ -7,12 +7,12 @@ import (
    "fmt"
    "time"

-    "github.com/ava-labs/avalanchego/api"
-    "github.com/ava-labs/avalanchego/ids"
-    "github.com/ava-labs/avalanchego/utls/formatting"
-    cjson "github.com/ava-labs/avalanchego/utls/json"
-    "github.com/ava-labs/avalanchego/utls/rpc"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/flare/api"
+    "github.com/flare-foundation/flare/ids"
+    "github.com/flare-foundation/flare/utls/formatting"
+    cjson "github.com/flare-foundation/flare/utls/json"
+    "github.com/flare-foundation/flare/utls/rpc"
)

// Client ...
diff --git a/plugin/evm/codec.go b/plugin/evm/codec.go
index 3a49eb8..3b6d83b 100644
--- a/plugin/evm/codec.go
+++ b/plugin/evm/codec.go
@@ -4,10 +4,10 @@ package evm
import (
-    "github.com/ava-labs/avalanchego/codec"
-    "github.com/ava-labs/avalanchego/codec/linearcodec"
-    "github.com/ava-labs/avalanchego/utls/wrappers"
-    "github.com/ava-labs/avalanchego/vms/secp256k1fx"
+    "github.com/flare-foundation/flare/codec"
+    "github.com/flare-foundation/flare/codec/linearcodec"
+    "github.com/flare-foundation/flare/utls/wrappers"
+    "github.com/flare-foundation/flare/vms/secp256k1fx"
)

// Codec does serialization and deserialization
diff --git a/plugin/evm/config.go b/plugin/evm/config.go
index 9e3ee4..bf474de 100644
--- a/plugin/evm/config.go
+++ b/plugin/evm/config.go
@@ -7,7 +7,7 @@ import (
    "encoding/json"
    "time"

-    "github.com/ava-labs/coreth/eth"
+    "github.com/flare-foundation/coreth/eth"
+    "github.com/spf13/cast"
)

diff --git a/plugin/evm/database.go b/plugin/evm/database.go
index c86ae5e..53441e0 100644
--- a/plugin/evm/database.go
+++ b/plugin/evm/database.go
@@ -4,9 +4,9 @@ package evm
import (
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb"

-    "github.com/ava-labs/avalanchego/database"
+    "github.com/flare-foundation/flare/database"
)

// Database implements ethdb.Database
diff --git a/plugin/evm/export_tx.go b/plugin/evm/export_tx.go
index fcc8c38..d7dc508 100644
--- a/plugin/evm/export_tx.go
+++ b/plugin/evm/export_tx.go
@@ -7,20 +7,20 @@ import (
    "fmt"
    "math/big"

-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/params"
-
-    "github.com/ava-labs/avalanchego/chains/atomic"
-    "github.com/ava-labs/avalanchego/database"
-    "github.com/ava-labs/avalanchego/ids"
-    "github.com/ava-labs/avalanchego/snow"
-    "github.com/ava-labs/avalanchego/utls/crypto"
-    "github.com/ava-labs/avalanchego/utls/math"
-    "github.com/ava-labs/avalanchego/utls/wrappers"
-    "github.com/ava-labs/avalanchego/vms/components/avax"
-    "github.com/ava-labs/avalanchego/vms/secp256k1fx"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/log"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/params"
+
+    "github.com/flare-foundation/flare/database"
+    "github.com/flare-foundation/flare/ids"
+    "github.com/flare-foundation/flare/snow"
+    "github.com/flare-foundation/flare/utls/crypto"
+    "github.com/flare-foundation/flare/vms/components/avax"
)

// UnsignedExportTx is an unsigned ExportTx
@@ -40,17 +40,17 @@ type UnsignedExportTx struct {

// InputUTXOs returns a set of all the hash(address:nonce) exporting funds.
func (tx *UnsignedExportTx) InputUTXOs() ids.Set {
-    set := ids.NewSet(len(tx.Ins))
-    for _, in := range tx.Ins {
-        // Total populated bytes is 20 (Address) + 8 (Nonce), however, we allocate
-        // 32 bytes to make ids.ID casting easier.
-        var rawID [32]byte
-        packer := wrappers.Packer{Bytes: rawID[:]}
-        packer.PackLong(in.Nonce)
-        packer.PackBytes(in.Address.Bytes())
-        set.Add(ids.ID(rawID))
-    }
-    return set
+    return ids.Set{}
}

// Verify this transaction is well-formed
@@ -59,75 +59,75 @@ func (tx *UnsignedExportTx) Verify(
    ctx *snow.Context,
    rules params.Rules,
) error {
-    switch {
-    case tx == nil:
-        return errNilTx
-    case tx.DestinationChain != xChainID:
-        return errWrongChainID
-    case len(tx.ExportedOutputs) == 0:
-        return errNoExportOutputs
-    }
-    return nil
}

```

```

-     case tx.NetworkID != ctx.NetworkID:
-         return errWrongNetworkID
-     case ctx.ChainID != tx.BlockchainID:
-         return errWrongBlockchainID
-     }
-
-     for _, in := range tx.Ins {
-         if err := in.Verify(); err != nil {
-             return err
-         }
-     }
-
-     for _, out := range tx.ExportedOutputs {
-         if err := out.Verify(); err != nil {
-             return err
-         }
-     }
-
-     if !avax.IsSortedTransferableOutputs(tx.ExportedOutputs, Codec) {
-         return errOutputsNotSorted
-     }
-
-     if rules.IsApricotPhase1 && !IsSortedAndUniqueEVMInputs(tx.Ins) {
-         return errInputsNotSortedUnique
-     }
-
-     return nil
+     return errWrongChainID
+ }
+
+ func (tx *UnsignedExportTx) GasUsed() (uint64, error) {
-     byteCost := calcBytesCost(len(tx.UnsignedBytes()))
-     numSigs := uint64(len(tx.Ins))
-     sigCost, err := math.Mul64(numSigs, secp256k1fx.CostPerSignature)
-     if err != nil {
-         return 0, err
-     }
-     return math.Add64(byteCost, sigCost)
+     return 0, fmt.Errorf("exportTx transactions disabled")
+ }
+
+ // Amount of [assetID] burned by this transaction
+ func (tx *UnsignedExportTx) Burned(assetID ids.ID) (uint64, error) {
-     var (
-         spent uint64
-         input uint64
-         err    error
-     )
-     for _, out := range tx.ExportedOutputs {
-         if out.AssetId() == assetID {
-             spent, err = math.Add64(spent, out.Output().Amount())
-             if err != nil {
-                 return 0, err
-             }
-         }
-     }
-     for _, in := range tx.Ins {
-         if in.AssetId == assetID {
-             input, err = math.Add64(input, in.Amount)
-             if err != nil {
-                 return 0, err
-             }
-         }
-     }
-     return math.Sub64(input, spent)
+     return 0, fmt.Errorf("exportTx transactions disabled")
+ }
+
+ // SemanticVerify this transaction is valid.
+ @@ -138,105 +63,12 @@ func (tx *UnsignedExportTx) SemanticVerify(
-     baseFee *big.Int,
-     rules params.Rules,
- ) error {
-     if err := tx.Verify(vm.ctx.XChainID, vm.ctx, rules); err != nil {
-         return err
-     }
-
-     // Check the transaction consumes and produces the right amounts
-     fc := avax.NewFlowChecker()
-     switch {
-     // Apply dynamic fees to export transactions as of Apricot Phase 3
-     case rules.IsApricotPhase3:
-         gasUsed, err := stx.GasUsed()
-         if err != nil {
-             return err
-         }
-         txFee, err := calculateDynamicFee(gasUsed, baseFee)
-         if err != nil {
-             return err
-         }
-         fc.Produce(vm.ctx.AVAXAssetID, txFee)
-
-     // Apply fees to export transactions before Apricot Phase 3
-     default:
-         fc.Produce(vm.ctx.AVAXAssetID, params.AvalancheAtomicTxFee)
-     }
-     for _, out := range tx.ExportedOutputs {
-         fc.Produce(out.AssetId(), out.Output().Amount())
-     }
-     for _, in := range tx.Ins {
-         fc.Consume(in.AssetId, in.Amount)
-     }
-
-     if err := fc.Verify(); err != nil {
-         return fmt.Errorf("export tx flow check failed due to: %w", err)
-     }
-
-     if len(tx.Ins) != len(stx.Creds) {
-         return fmt.Errorf("export tx contained mismatched number of inputs/credentials (%d vs. %d)", len(tx.Ins), len(stx.Creds))
-     }
-
-     for i, input := range tx.Ins {
-         cred, ok := stx.Creds[i].(*secp256k1fx.Credential)
-         if !ok {
-             return fmt.Errorf("expected *secp256k1fx.Credential but got %T", cred)
-         }
-         if err := cred.Verify(); err != nil {
-             return err
-         }
-
-         if len(cred.Sigs) != 1 {
-             return fmt.Errorf("expected one signature for EVM Input Credential, but found: %d", len(cred.Sigs))
-         }
-         pubKeyIntf, err := vm.secpFactory.RecoverPublicKey(tx.UnsignedBytes(), cred.Sigs[0][:])
-         if err != nil {
-             return err
-         }
-         pubKey, ok := pubKeyIntf.(*crypto.PublicKeySECP256K1R)
-         if !ok {
-             // This should never happen
-             return fmt.Errorf("expected *crypto.PublicKeySECP256K1R but got %T", pubKeyIntf)
-         }
-         if input.Address != PublicKeyToEthAddress(pubKey) {
-             return errPublicKeySignatureMismatch
-         }
-     }

```

```

-     }
-
-     return nil
+     return fmt.Errorf("exportTx transactions disabled")
- }
-
- // Accept this transaction.
- func (tx *UnsignedExportTx) Accept(ctx *snow.Context, batch database.Batch) error {
-     txID := tx.ID()
-
-     elems := make([]*atomic.Element, len(tx.ExportedOutputs))
-     for i, out := range tx.ExportedOutputs {
-         utxo := &avax.UTXO{
-             UTXOID: avax.UTXOID{
-                 TxID:      txID,
-                 OutputIndex: uint32(i),
-             },
-             Asset: avax.Asset{ID: out.AssetID()},
-             Out:    out.Out,
-         }
-
-         utxoBytes, err := Codec.Marshal(codecVersion, utxo)
-         if err != nil {
-             return err
-         }
-         utxoID := utxo.InputID()
-         elem := &atomic.Element{
-             Key:   utxoID[:],
-             Value: utxoBytes,
-         }
-         if out, ok := utxo.Out.(avax.Addressable); ok {
-             elem.Traits = out.Addresses()
-         }
-
-         elems[i] = elem
-     }
-
-     return ctx.SharedMemory.Apply(map[ids.ID]*atomic.Requests{tx.DestinationChain: {PutRequests: elems}}, batch)
+     return fmt.Errorf("exportTx transactions disabled")
- }
-
- // newExportTx returns a new ExportTx
@@ -248,122 +80,10 @@ func (vm *VM) newExportTx(
    baseFee *big.Int, // fee to use post-AP3
    keys []*crypto.PrivateKeySECP256K1R, // Pay the fee and provide the tokens
  ) (*Tx, error) {
-     if vm.ctx.XChainID != chainID {
-         return nil, errWrongChainID
-     }
-
-     outs := []*avax.TransferableOutput{{ // Exported to X-Chain
-         Asset: avax.Asset{ID: assetID},
-         Out: &secp256k1fx.TransferOutput{
-             Amt: amount,
-             OutputOwners: secp256k1fx.OutputOwners{
-                 Locktime: 0,
-                 Threshold: 1,
-                 Addrs:    []ids.ShortID{to},
-             },
-         },
-     }}
-
-     var (
-         avaxNeeded      uint64 = 0
-         ins, avaxIns    []EVMInput
-         signers, avaxSigners [][]*crypto.PrivateKeySECP256K1R
-         err              error
-     )
-
-     // consume non-AVAX
-     if assetID != vm.ctx.AVAXAssetID {
-         ins, signers, err = vm.GetSpendableFunds(keys, assetID, amount)
-         if err != nil {
-             return nil, fmt.Errorf("couldn't generate tx inputs/signers: %w", err)
-         }
-     } else {
-         avaxNeeded = amount
-     }
-
-     rules := vm.currentRules()
-     switch {
-     case rules.IsApricotPhase3:
-         utx := &UnsignedExportTx{
-             NetworkID:      vm.ctx.NetworkID,
-             BlockchainID:    vm.ctx.ChainID,
-             DestinationChain: chainID,
-             Ins:             ins,
-             ExportedOutputs: outs,
-         }
-
-         tx := &Tx{UnsignedAtomicTx: utx}
-         if err := tx.Sign(vm.codec, nil); err != nil {
-             return nil, err
-         }
-
-         var cost uint64
-         cost, err = tx.GasUsed()
-         if err != nil {
-             return nil, err
-         }
-
-         avaxIns, avaxSigners, err = vm.GetSpendableAVAXWithFee(keys, avaxNeeded, cost, baseFee)
-     default:
-         var newAvaxNeeded uint64
-         newAvaxNeeded, err = math.Add64(avaxNeeded, params.AvalancheAtomicTxFee)
-         if err != nil {
-             return nil, errOverflowExport
-         }
-         avaxIns, avaxSigners, err = vm.GetSpendableFunds(keys, vm.ctx.AVAXAssetID, newAvaxNeeded)
-     }
-     if err != nil {
-         return nil, fmt.Errorf("couldn't generate tx inputs/signers: %w", err)
-     }
-     ins = append(ins, avaxIns...)
-     signers = append(signers, avaxSigners...)
-
-     avax.SortTransferableOutputs(outs, vm.codec)
-     SortEVMInputsAndSigners(ins, signers)
-
-     // Create the transaction
-     utx := &UnsignedExportTx{
-         NetworkID:      vm.ctx.NetworkID,
-         BlockchainID:    vm.ctx.ChainID,
-         DestinationChain: chainID,
-         Ins:             ins,
-         ExportedOutputs: outs,
-     }
-
-     tx := &Tx{UnsignedAtomicTx: utx}
-     if err := tx.Sign(vm.codec, signers); err != nil {
-         return nil, err
-     }
-
-     return tx, tx.Verify(vm.ctx.XChainID, vm.ctx, vm.currentRules())
+     return nil, errWrongChainID
  }

```

```

// EVMStateTransfer executes the state update from the atomic export transaction
func (tx *UnsignedExportTx) EVMStateTransfer(ctx *snow.Context, state *state.StateDB) error {
-   addrs := map[20]byte{uint64{}}
-   for _, from := range tx.Ins {
-       if from.AssetId == ctx.AVAXAssetID {
-           log.Debug("crosschain C->X", "addr", from.Address, "amount", from.Amount, "assetID", "AVAX")
-           // We multiply the input amount by x2cRate to convert AVAX back to the appropriate
-           // denomination before export.
-           amount := new(big.Int).Mul(
-               new(big.Int).SetUint64(from.Amount), x2cRate)
-           if state.GetBalance(from.Address).Cmp(amount) < 0 {
-               return errInsufficientFunds
-           }
-           state.SubBalance(from.Address, amount)
-       } else {
-           log.Debug("crosschain C->X", "addr", from.Address, "amount", from.Amount, "assetID", from.AssetId)
-           amount := new(big.Int).SetUint64(from.Amount)
-           if state.GetBalanceMultiCoin(from.Address, common.Hash(from.AssetId)).Cmp(amount) < 0 {
-               return errInsufficientFunds
-           }
-           state.SubBalanceMultiCoin(from.Address, common.Hash(from.AssetId), amount)
-       }
-       if state.GetNonce(from.Address) != from.Nonce {
-           return errInvalidNonce
-       }
-       addrs[from.Address] = from.Nonce
-   }
-   for addr, nonce := range addrs {
-       state.SetNonce(addr, nonce+1)
-   }
-   return nil
+   return errInsufficientFunds
}

diff --git a/plugin/evm/export_tx_test.go b/plugin/evm/export_tx_test.go
index 2861af0..2a27b18 100644
--- a/plugin/evm/export_tx_test.go
+++ b/plugin/evm/export_tx_test.go
@@ -8,15 +8,15 @@ import (
    "math/big"
    "testing"

-   "github.com/ava-labs/avalanchego/chains/atomic"
-   "github.com/ava-labs/avalanchego/ids"
-   engCommon "github.com/ava-labs/avalanchego/snow/engine/common"
-   "github.com/ava-labs/avalanchego/utls/crypto"
-   "github.com/ava-labs/avalanchego/utls/units"
-   "github.com/ava-labs/avalanchego/vms/components/avax"
-   "github.com/ava-labs/avalanchego/vms/secp256k1fx"
-   "github.com/ava-labs/coreth/params"
-   "github.com/ethereum/go-ethereum/common"
+   "github.com/flare-foundation/coreth/params"
+   "github.com/flare-foundation/flare/chains/atomic"
+   "github.com/flare-foundation/flare/ids"
+   engCommon "github.com/flare-foundation/flare/snow/engine/common"
+   "github.com/flare-foundation/flare/utls/crypto"
+   "github.com/flare-foundation/flare/utls/units"
+   "github.com/flare-foundation/flare/vms/components/avax"
+   "github.com/flare-foundation/flare/vms/secp256k1fx"
)

// createExportTxOptions adds funds to shared memory, imports them, and returns a list of export transactions
diff --git a/plugin/evm/factory.go b/plugin/evm/factory.go
index 34ecc89..79388fc 100644
--- a/plugin/evm/factory.go
+++ b/plugin/evm/factory.go
@@ -4,9 +4,9 @@ package evm

import (
-   "github.com/ava-labs/avalanchego/ids"
-   "github.com/ava-labs/avalanchego/snow"
-   "github.com/ava-labs/avalanchego/vms"
+   "github.com/flare-foundation/flare/ids"
+   "github.com/flare-foundation/flare/snow"
+   "github.com/flare-foundation/flare/vms"
)

var (
diff --git a/plugin/evm/formatting.go b/plugin/evm/formatting.go
index d2194ea..cf6c668 100644
--- a/plugin/evm/formatting.go
+++ b/plugin/evm/formatting.go
@@ -6,12 +6,12 @@ package evm

import (
    "fmt"

-   "github.com/ava-labs/avalanchego/ids"
-   "github.com/ava-labs/avalanchego/utls/constants"
-   "github.com/ava-labs/avalanchego/utls/crypto"
-   "github.com/ava-labs/avalanchego/utls/formatting"
-   "github.com/ethereum/go-ethereum/common"
-   ethcrypto "github.com/ethereum/go-ethereum/crypto"
+   "github.com/flare-foundation/flare/ids"
+   "github.com/flare-foundation/flare/utls/constants"
+   "github.com/flare-foundation/flare/utls/crypto"
+   "github.com/flare-foundation/flare/utls/formatting"
)

// ParseLocalAddress takes in an address for this chain and produces the ID
diff --git a/plugin/evm/gasprice_update.go b/plugin/evm/gasprice_update.go
index 71a4ea1..177be08 100644
--- a/plugin/evm/gasprice_update.go
+++ b/plugin/evm/gasprice_update.go
@@ -8,7 +8,7 @@ import (
    "sync"
    "time"

-   "github.com/ava-labs/coreth/params"
+   "github.com/flare-foundation/coreth/params"
)

type gasPriceUpdater struct {
diff --git a/plugin/evm/gasprice_update_test.go b/plugin/evm/gasprice_update_test.go
index 24d8337..5a720fb 100644
--- a/plugin/evm/gasprice_update_test.go
+++ b/plugin/evm/gasprice_update_test.go
@@ -9,7 +9,7 @@ import (
    "testing"
    "time"

-   "github.com/ava-labs/coreth/params"
+   "github.com/flare-foundation/coreth/params"
)

type mockGasPriceSetter struct {
diff --git a/plugin/evm/import_tx.go b/plugin/evm/import_tx.go
index 6ae6352..acf5a34 100644
--- a/plugin/evm/import_tx.go
+++ b/plugin/evm/import_tx.go
@@ -7,19 +7,19 @@ import (
    "fmt"
    "math/big"

-   "github.com/ava-labs/coreth/state"

```

```

-     "github.com/ava-labs/coreth/params"
+     "github.com/flare-foundation/coreth/core/state"
+     "github.com/flare-foundation/coreth/params"

-     "github.com/ava-labs/avalanchego/chains/atomic"
-     "github.com/ava-labs/avalanchego/database"
-     "github.com/ava-labs/avalanchego/ids"
-     "github.com/ava-labs/avalanchego/snow"
-     "github.com/ava-labs/avalanchego/utls/crypto"
-     "github.com/ava-labs/avalanchego/utls/math"
-     "github.com/ava-labs/avalanchego/vms/components/avax"
-     "github.com/ava-labs/avalanchego/vms/secp256k1fx"
-     "github.com/ethereum/go-ethereum/common"
-     "github.com/ethereum/go-ethereum/log"
+     "github.com/flare-foundation/flare/database"
+     "github.com/flare-foundation/flare/ids"
+     "github.com/flare-foundation/flare/snow"
+     "github.com/flare-foundation/flare/utls/crypto"
+     "github.com/flare-foundation/flare/vms/components/avax"
+ )

// UnsignedImportTx is an unsigned ImportTx
@@ -39,11 +35,7 @@ type UnsignedImportTx struct {

// InputUTXOs returns the UTXOIDs of the imported funds
func (tx *UnsignedImportTx) InputUTXOs() ids.Set {
-     set := ids.NewSet(len(tx.ImportedInputs))
-     for _, in := range tx.ImportedInputs {
-         set.Add(in.InputID())
-     }
-     return set
+     return ids.Set{}
}

// Verify this transaction is well-formed
@@ -52,89 +44,16 @@ func (tx *UnsignedImportTx) Verify(
    ctx *snow.Context,
    rules params.Rules,
) error {
-     switch {
-     case tx == nil:
-         return errNilTx
-     case tx.SourceChain != xChainID:
-         return errWrongChainID
-     case len(tx.ImportedInputs) == 0:
-         return errNoImportInputs
-     case tx.NetworkID != ctx.NetworkID:
-         return errWrongNetworkID
-     case ctx.ChainID != tx.BlockchainID:
-         return errWrongBlockchainID
-     case rules.IsApricotPhase3 && len(tx.Outs) == 0:
-         return errNoEVMOutputs
-     }

-     for _, out := range tx.Outs {
-         if err := out.Verify(); err != nil {
-             return fmt.Errorf("EVM Output failed verification: %w", err)
-         }
-     }

-     for _, in := range tx.ImportedInputs {
-         if err := in.Verify(); err != nil {
-             return fmt.Errorf("atomic input failed verification: %w", err)
-         }
-     }

-     if !avax.IsSortedAndUniqueTransferableInputs(tx.ImportedInputs) {
-         return errInputsNotSortedUnique
-     }

-     if rules.IsApricotPhase2 {
-         if !IsSortedAndUniqueEVMOutputs(tx.Outs) {
-             return errOutputsNotSortedUnique
-         }
-     } else if rules.IsApricotPhase1 {
-         if !IsSortedEVMOutputs(tx.Outs) {
-             return errOutputsNotSorted
-         }
-     }

-     return nil
+     return errWrongChainID
}

func (tx *UnsignedImportTx) GasUsed() (uint64, error) {
-     cost := calcBytesCost(len(tx.UnsignedBytes()))
-     for _, in := range tx.ImportedInputs {
-         inCost, err := in.In.Cost()
-         if err != nil {
-             return 0, err
-         }
-         cost, err = math.Add64(cost, inCost)
-         if err != nil {
-             return 0, err
-         }
-     }
-     return cost, nil
+     return 0, fmt.Errorf("exportTx transactions disabled")
}

// Amount of [assetID] burned by this transaction
func (tx *UnsignedImportTx) Burned(assetID ids.ID) (uint64, error) {
-     var (
-         spent uint64
-         input uint64
-         err error
-     )

-     for _, out := range tx.Outs {
-         if out.AssetID == assetID {
-             spent, err = math.Add64(spent, out.Amount)
-             if err != nil {
-                 return 0, err
-             }
-         }
-     }

-     for _, in := range tx.ImportedInputs {
-         if in.AssetID() == assetID {
-             input, err = math.Add64(input, in.Input().Amount())
-             if err != nil {
-                 return 0, err
-             }
-         }
-     }

-     return math.Sub64(input, spent)
+     return 0, fmt.Errorf("exportTx transactions disabled")
}

// SemanticVerify this transaction is valid.
@@ -145,82 +64,7 @@ func (tx *UnsignedImportTx) SemanticVerify(
    baseFee *big.Int,
    rules params.Rules,
) error {
-     if err := tx.Verify(vm.ctx.XChainID, vm.ctx, rules); err != nil {
-         return err
-     }

```

```

-     }
-
-     // Check the transaction consumes and produces the right amounts
-     fc := avax.NewFlowChecker()
-     switch {
-     // Apply dynamic fees to import transactions as of Apricot Phase 3
-     case rules.IsApricotPhase3:
-         gasUsed, err := stx.GasUsed()
-         if err != nil {
-             return err
-         }
-         txFee, err := calculateDynamicFee(gasUsed, baseFee)
-         if err != nil {
-             return err
-         }
-         fc.Produce(vm.ctx.AVAXAssetID, txFee)
-
-     // Apply fees to import transactions as of Apricot Phase 2
-     case rules.IsApricotPhase2:
-         fc.Produce(vm.ctx.AVAXAssetID, params.AvalancheAtomicTxFee)
-     }
-     for _, out := range tx.Outs {
-         fc.Produce(out.AssetIdID, out.Amount)
-     }
-     for _, in := range tx.ImportedInputs {
-         fc.Consume(in.AssetIdID, in.Input().Amount())
-     }
-
-     if err := fc.Verify(); err != nil {
-         return fmt.Errorf("import tx flow check failed due to: %w", err)
-     }
-
-     if len(stx.Creds) != len(tx.ImportedInputs) {
-         return fmt.Errorf("import tx contained mismatched number of inputs/credentials (%d vs. %d)", len(tx.ImportedInputs), len(stx.Creds))
-     }
-
-     if !vm.ctx.IsBootstrapped() {
-         // Allow for force committing during bootstrapping
-         return nil
-     }
-
-     utxoIDs := make([]byte, len(tx.ImportedInputs))
-     for i, in := range tx.ImportedInputs {
-         inputID := in.UTXOID.InputID()
-         utxoIDs[i] = inputID[:]
-     }
-     // allUTXOBytes is guaranteed to be the same length as utxoIDs
-     allUTXOBytes, err := vm.ctx.SharedMemory.Get(tx.SourceChain, utxoIDs)
-     if err != nil {
-         return fmt.Errorf("failed to fetch import UTXOs from %s due to: %w", tx.SourceChain, err)
-     }
-
-     for i, in := range tx.ImportedInputs {
-         utxoBytes := allUTXOBytes[i]
-
-         utxo := &avax.UTXO{}
-         if _, err := vm.codec.Unmarshal(utxoBytes, utxo); err != nil {
-             return fmt.Errorf("failed to unmarshal UTXO: %w", err)
-         }
-
-         cred := stx.Creds[i]
-
-         utxoAssetID := utxo.AssetIdID()
-         inAssetID := in.AssetIdID()
-         if utxoAssetID != inAssetID {
-             return errAssetIDMismatch
-         }
-
-         if err := vm.fx.VerifyTransfer(tx, in.In, cred, utxo.Out); err != nil {
-             return fmt.Errorf("import tx transfer failed verification: %w", err)
-         }
-     }
-
-     return vm.conflicts(tx.InputUTXOs(), parent)
+     return fmt.Errorf("exportTx transactions disabled")
}

// Accept this transaction and spend imported inputs
@@ -229,12 +73,7 @@ func (tx *UnsignedImportTx) SemanticVerify(
// only to have the transaction not be Accepted. This would be inconsistent.
// Recall that imported UTXOs are not kept in a versionDB.
func (tx *UnsignedImportTx) Accept(ctx *snow.Context, batch database.Batch) error {
-     utxoIDs := make([]byte, len(tx.ImportedInputs))
-     for i, in := range tx.ImportedInputs {
-         inputID := in.InputID()
-         utxoIDs[i] = inputID[:]
-     }
-     return ctx.SharedMemory.Apply(map[ids.ID]*atomic.Requests{tx.SourceChain: {RemoveRequests: utxoIDs}}, batch)
+     return fmt.Errorf("exportTx transactions disabled")
}

// newImportTx returns a new ImportTx
@@ -244,160 +83,11 @@ func (vm *VM) newImportTx(
baseFee *big.Int, // fee to use post-AP3
keys []*crypto.PrivateKeySECP256K1R, // Keys to import the funds
)(*Tx, error) {
-     if vm.ctx.XChainID != chainID {
-         return nil, errWrongChainID
-     }
-
-     kc := secp256k1fx.NewKeychain()
-     for _, key := range keys {
-         kc.Add(key)
-     }
-
-     atomicUTXOs, _, err := vm.GetAtomicUTXOs(chainID, kc.Addresses(), ids.ShortEmpty, ids.Empty, -1)
-     if err != nil {
-         return nil, fmt.Errorf("problem retrieving atomic UTXOs: %w", err)
-     }
-
-     importedInputs := []*avax.TransferableInput{}
-     signers := []*crypto.PrivateKeySECP256K1R{}
-
-     importedAmount := make(map[ids.ID]uint64)
-     now := vm.clock.Unix()
-     for _, utxo := range atomicUTXOs {
-         inputIntf, utxoSigners, err := kc.Spend(utxo.Out, now)
-         if err != nil {
-             continue
-         }
-         input, ok := inputIntf.(avax.TransferableIn)
-         if !ok {
-             continue
-         }
-         aid := utxo.AssetIdID()
-         importedAmount[aid], err = math.Add64(importedAmount[aid], input.Amount())
-         if err != nil {
-             return nil, err
-         }
-         importedInputs = append(importedInputs, &avax.TransferableInput{
-             UTXOID: utxo.UTXOID,
-             Asset:  utxo.AssetId,
-             In:     input,
-         })

```

```

-         signers = append(signers, utxoSigners)
-     }
-    avax.SortTransferableInputsWithSigners(importedInputs, signers)
-     importedAVAXAmount := importedAmount[vm.ctx.AVAXAssetID]
-
-     outs := make([]EVMOutput, 0, len(importedAmount))
-     // This will create unique outputs (in the context of sorting)
-     // since each output will have a unique assetID
-     for assetID, amount := range importedAmount {
-         // Skip the AVAX amount since it is included separately to account for
-         // the fee
-         if assetID == vm.ctx.AVAXAssetID || amount == 0 {
-             continue
-         }
-         outs = append(outs, EVMOutput{
-             Address: to,
-             Amount:  amount,
-             AssetID: assetID,
-         })
-     }
-
-     rules := vm.currentRules()
-
-     var (
-         txFeeWithoutChange uint64
-         txFeeWithChange    uint64
-     )
-     switch {
-     case rules.IsApricotPhase3:
-         if baseFee == nil {
-             return nil, errNilBaseFeeApricotPhase3
-         }
-         utx := &UnsignedImportTx{
-             NetworkID:    vm.ctx.NetworkID,
-             BlockchainID:  vm.ctx.ChainID,
-             Outs:          outs,
-             ImportedInputs: importedInputs,
-             SourceChain:   chainID,
-         }
-         tx := &Tx{UnsignedAtomicTx: utx}
-         if err := tx.Sign(vm.codec, nil); err != nil {
-             return nil, err
-         }
-
-         gasUsedWithoutChange, err := tx.GasUsed()
-         if err != nil {
-             return nil, err
-         }
-         gasUsedWithChange := gasUsedWithoutChange + EVMOutputGas
-
-         txFeeWithoutChange, err = calculateDynamicFee(gasUsedWithoutChange, baseFee)
-         if err != nil {
-             return nil, err
-         }
-         txFeeWithChange, err = calculateDynamicFee(gasUsedWithChange, baseFee)
-         if err != nil {
-             return nil, err
-         }
-     case rules.IsApricotPhase2:
-         txFeeWithoutChange = params.AvalancheAtomicTxFee
-         txFeeWithChange = params.AvalancheAtomicTxFee
-     }
-
-     // AVAX output
-     if importedAVAXAmount < txFeeWithoutChange { // imported amount goes toward paying tx fee
-         return nil, errInsufficientFundsForFee
-     }
-
-     if importedAVAXAmount > txFeeWithChange {
-         outs = append(outs, EVMOutput{
-             Address: to,
-             Amount:  importedAVAXAmount - txFeeWithChange,
-             AssetID: vm.ctx.AVAXAssetID,
-         })
-     }
-
-     // If no outputs are produced, return an error.
-     // Note: this can happen if there is exactly enough AVAX to pay the
-     // transaction fee, but no other funds to be imported.
-     if len(outs) == 0 {
-         return nil, errNoEVMOutputs
-     }
-
-     SortEVMOutputs(outs)
-
-     // Create the transaction
-     utx := &UnsignedImportTx{
-         NetworkID:    vm.ctx.NetworkID,
-         BlockchainID:  vm.ctx.ChainID,
-         Outs:          outs,
-         ImportedInputs: importedInputs,
-         SourceChain:   chainID,
-     }
-     tx := &Tx{UnsignedAtomicTx: utx}
-     if err := tx.Sign(vm.codec, signers); err != nil {
-         return nil, err
-     }
-
-     return tx, utx.Verify(vm.ctx.XChainID, vm.ctx, vm.currentRules())
-     return nil, errWrongChainID
+ }
+
+ // EVMStateTransfer performs the state transfer to increase the balances of
+ // accounts accordingly with the imported EVMOutputs
+ func (tx *UnsignedImportTx) EVMStateTransfer(ctx *snow.Context, state *state.StateDB) error {
+     for _, to := range tx.Outs {
+         if to.AssetId == ctx.AVAXAssetID {
+             log.Debug("crosschain X->C", "addr", to.Address, "amount", to.Amount, "assetID", "AVAX")
+             // If the asset is AVAX, convert the input amount in nAVAX to gWei by
+             // multiplying by the x2c rate.
+             amount := new(big.Int).Mul(
+                 new(big.Int).SetUint64(to.Amount), x2cRate)
+             state.AddBalance(to.Address, amount)
+         } else {
+             log.Debug("crosschain X->C", "addr", to.Address, "amount", to.Amount, "assetID", to.AssetId)
+             amount := new(big.Int).SetUint64(to.Amount)
+             state.AddBalanceMultiCoin(to.Address, common.Hash(to.AssetId), amount)
+         }
+     }
+     return nil
+     return errInsufficientFunds
+ }
+
+ diff --git a/plugin/evm/import_tx_test.go b/plugin/evm/import_tx_test.go
+ index b3604e1..4545213 100644
+ --- a/plugin/evm/import_tx_test.go
+ +++ b/plugin/evm/import_tx_test.go
+ @@ -7,14 +7,14 @@ import (
+     "math/big"
+     "testing"
+
+     "github.com/ava-labs/coreth/params"
+     "github.com/ethereum/go-ethereum/common"
+     "github.com/flare-foundation/coreth/params"
+
+     "github.com/ava-labs/avalanchego/chains/atomic"

```

```

-      "github.com/ava-labs/avalanchego/ids"
-      "github.com/ava-labs/avalanchego/utls/crypto"
-      "github.com/ava-labs/avalanchego/vms/components/avax"
-      "github.com/ava-labs/avalanchego/vms/secp256k1fx"
+      "github.com/flare-foundation/flare/chains/atomic"
+      "github.com/flare-foundation/flare/ids"
+      "github.com/flare-foundation/flare/utls/crypto"
+      "github.com/flare-foundation/flare/vms/components/avax"
+      "github.com/flare-foundation/flare/vms/secp256k1fx"
)

// createImportTxOptions adds a UTXO to shared memory and generates a list of import transactions sending this UTXO
diff --git a/plugin/evm/mempool.go b/plugin/evm/mempool.go
index 5385881..6b7ab75 100644
--- a/plugin/evm/mempool.go
+++ b/plugin/evm/mempool.go
@@ -8,9 +8,9 @@ import (
     "fmt"
     "sync"

-      "github.com/ava-labs/avalanchego/cache"
-      "github.com/ava-labs/avalanchego/ids"
+      "github.com/ethereum/go-ethereum/log"
+      "github.com/flare-foundation/flare/cache"
+      "github.com/flare-foundation/flare/ids"
)

const (
diff --git a/plugin/evm/mempool_atomic_gossiping_test.go b/plugin/evm/mempool_atomic_gossiping_test.go
index 43c0f9b..1441852 100644
--- a/plugin/evm/mempool_atomic_gossiping_test.go
+++ b/plugin/evm/mempool_atomic_gossiping_test.go
@@ -6,13 +6,13 @@ package evm
import (
    "testing"

-      "github.com/ava-labs/coreth/params"
+      "github.com/flare-foundation/coreth/params"

-      "github.com/ava-labs/avalanchego/ids"
-      "github.com/ava-labs/avalanchego/utls/crypto"
-      "github.com/ava-labs/avalanchego/vms/components/avax"
-      "github.com/ava-labs/avalanchego/vms/components/chain"
-      "github.com/ava-labs/avalanchego/vms/secp256k1fx"
+      "github.com/flare-foundation/flare/ids"
+      "github.com/flare-foundation/flare/utls/crypto"
+      "github.com/flare-foundation/flare/vms/components/avax"
+      "github.com/flare-foundation/flare/vms/components/chain"
+      "github.com/flare-foundation/flare/vms/secp256k1fx"

    "github.com/stretchr/testify/assert"
)

diff --git a/plugin/evm/message/codec.go b/plugin/evm/message/codec.go
index 8c17274..92fd256 100644
--- a/plugin/evm/message/codec.go
+++ b/plugin/evm/message/codec.go
@@ -4,11 +4,11 @@ package message
import (
-      "github.com/ava-labs/avalanchego/codec"
-      "github.com/ava-labs/avalanchego/codec/linearcodec"
-      "github.com/ava-labs/avalanchego/codec/reflectcodec"
-      "github.com/ava-labs/avalanchego/utls/units"
-      "github.com/ava-labs/avalanchego/utls/wrappers"
+      "github.com/flare-foundation/flare/codec"
+      "github.com/flare-foundation/flare/codec/linearcodec"
+      "github.com/flare-foundation/flare/codec/reflectcodec"
+      "github.com/flare-foundation/flare/utls/units"
+      "github.com/flare-foundation/flare/utls/wrappers"
)

const (
diff --git a/plugin/evm/message/handler.go b/plugin/evm/message/handler.go
index bc88320..9ca264f 100644
--- a/plugin/evm/message/handler.go
+++ b/plugin/evm/message/handler.go
@@ -6,7 +6,7 @@ package message
import (
    "github.com/ethereum/go-ethereum/log"

-      "github.com/ava-labs/avalanchego/ids"
+      "github.com/flare-foundation/flare/ids"
)

var _ Handler = NoopHandler{}
diff --git a/plugin/evm/message/handler_test.go b/plugin/evm/message/handler_test.go
index 179ac56..e53600b 100644
--- a/plugin/evm/message/handler_test.go
+++ b/plugin/evm/message/handler_test.go
@@ -6,7 +6,7 @@ package message
import (
    "testing"

-      "github.com/ava-labs/avalanchego/ids"
+      "github.com/flare-foundation/flare/ids"

    "github.com/stretchr/testify/assert"
)

diff --git a/plugin/evm/message/message.go b/plugin/evm/message/message.go
index 082adbe..a9ab2fe 100644
--- a/plugin/evm/message/message.go
+++ b/plugin/evm/message/message.go
@@ -8,8 +8,8 @@ import (
    "github.com/ethereum/go-ethereum/common"

-      "github.com/ava-labs/avalanchego/ids"
-      "github.com/ava-labs/avalanchego/utls/units"
+      "github.com/flare-foundation/flare/ids"
+      "github.com/flare-foundation/flare/utls/units"
)

const (
diff --git a/plugin/evm/message/message_test.go b/plugin/evm/message/message_test.go
index 5575e25..b0fe13a 100644
--- a/plugin/evm/message/message_test.go
+++ b/plugin/evm/message/message_test.go
@@ -6,8 +6,8 @@ package message
import (
    "testing"

-      "github.com/ava-labs/avalanchego/utls"
-      "github.com/ava-labs/avalanchego/utls/units"
+      "github.com/flare-foundation/flare/utls"
+      "github.com/flare-foundation/flare/utls/units"

    "github.com/stretchr/testify/assert"
)

diff --git a/plugin/evm/network.go b/plugin/evm/network.go
index e3690d6..5d49e7a 100644
--- a/plugin/evm/network.go
+++ b/plugin/evm/network.go
@@ -9,23 +9,23 @@ import (
)

```



```

"sync"
"time"

- "github.com/ava-labs/avalanchego/cache"
- "github.com/ava-labs/avalanchego/ids"
- "github.com/ava-labs/avalanchego/snow"
- "github.com/ava-labs/avalanchego/utls/wrappers"
+ "github.com/flare-foundation/flare/cache"
+ "github.com/flare-foundation/flare/ids"
+ "github.com/flare-foundation/flare/snow"
+ "github.com/flare-foundation/flare/utls/wrappers"

- commonEng "github.com/ava-labs/avalanchego/snow/engine/common"
+ commonEng "github.com/flare-foundation/flare/snow/engine/common"

"github.com/ethereum/go-ethereum/common"
"github.com/ethereum/go-ethereum/log"
"github.com/ethereum/go-ethereum/rlp"

- "github.com/ava-labs/coreth/core"
- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/plugin/evm/message"
+ "github.com/flare-foundation/coreth/core"
+ "github.com/flare-foundation/coreth/core/state"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/plugin/evm/message"

- coreth "github.com/ava-labs/coreth/chain"
+ coreth "github.com/flare-foundation/coreth/chain"
)

const (
diff --git a/plugin/evm/network_atomic_gossiping_test.go b/plugin/evm/network_atomic_gossiping_test.go
index 501bc5c..c8a8c6d 100644
--- a/plugin/evm/network_atomic_gossiping_test.go
+++ b/plugin/evm/network_atomic_gossiping_test.go
@@ -8,11 +8,11 @@ import (
"testing"
"time"

- "github.com/ava-labs/avalanchego/ids"
+ "github.com/flare-foundation/flare/ids"

"github.com/stretchr/testify/assert"

- "github.com/ava-labs/coreth/plugin/evm/message"
+ "github.com/flare-foundation/coreth/plugin/evm/message"
)

// locally issued txs should be gossiped
diff --git a/plugin/evm/network_eth_gossiping_test.go b/plugin/evm/network_eth_gossiping_test.go
index 54a2011..b041842 100644
--- a/plugin/evm/network_eth_gossiping_test.go
+++ b/plugin/evm/network_eth_gossiping_test.go
@@ -12,7 +12,7 @@ import (
"testing"
"time"

- "github.com/ava-labs/avalanchego/ids"
+ "github.com/flare-foundation/flare/ids"

"github.com/ethereum/go-ethereum/common"
"github.com/ethereum/go-ethereum/crypto"
@@ -20,10 +20,10 @@ import (

"github.com/stretchr/testify/assert"

- "github.com/ava-labs/coreth/core"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/params"
- "github.com/ava-labs/coreth/plugin/evm/message"
+ "github.com/flare-foundation/coreth/core"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/flare-foundation/coreth/plugin/evm/message"
)

func fundAddressByGenesis(addr []common.Address) (string, error) {
diff --git a/plugin/evm/service.go b/plugin/evm/service.go
index 0dccc733..f15774f 100644
--- a/plugin/evm/service.go
+++ b/plugin/evm/service.go
@@ -11,17 +11,17 @@ import (
"net/http"
"strings"

- "github.com/ava-labs/avalanchego/api"
- "github.com/ava-labs/avalanchego/ids"
- "github.com/ava-labs/avalanchego/utls/constants"
- "github.com/ava-labs/avalanchego/utls/crypto"
- "github.com/ava-labs/avalanchego/utls/formatting"
- "github.com/ava-labs/avalanchego/utls/json"
- "github.com/ava-labs/coreth/params"
- "github.com/ethereum/go-ethereum/common"
- "github.com/ethereum/go-ethereum/common/hexutil"
- ethcrypto "github.com/ethereum/go-ethereum/crypto"
- "github.com/ethereum/go-ethereum/log"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/flare-foundation/flare/api"
+ "github.com/flare-foundation/flare/ids"
+ "github.com/flare-foundation/flare/utls/constants"
+ "github.com/flare-foundation/flare/utls/crypto"
+ "github.com/flare-foundation/flare/utls/formatting"
+ "github.com/flare-foundation/flare/utls/json"
)

// test constants
diff --git a/plugin/evm/static_service.go b/plugin/evm/static_service.go
index 9c59225..e1db747 100644
--- a/plugin/evm/static_service.go
+++ b/plugin/evm/static_service.go
@@ -7,8 +7,8 @@ import (
"context"
"encoding/json"

- "github.com/ava-labs/avalanchego/utls/formatting"
- "github.com/ava-labs/coreth/core"
+ "github.com/flare-foundation/coreth/core"
+ "github.com/flare-foundation/flare/utls/formatting"
)

// StaticService defines the static API services exposed by the evm
diff --git a/plugin/evm/tx.go b/plugin/evm/tx.go
index 3f0ad92..01bd7c4 100644
--- a/plugin/evm/tx.go
+++ b/plugin/evm/tx.go
@@ -12,19 +12,19 @@ import (

"github.com/ethereum/go-ethereum/common"

- "github.com/ava-labs/coreth/core/state"
- "github.com/ava-labs/coreth/params"
-

```

```

-      "github.com/ava-labs/avalanchego/codec"
-      "github.com/ava-labs/avalanchego/database"
-      "github.com/ava-labs/avalanchego/ids"
-      "github.com/ava-labs/avalanchego/snow"
-      "github.com/ava-labs/avalanchego/utills"
-      "github.com/ava-labs/avalanchego/utills/crypto"
-      "github.com/ava-labs/avalanchego/utills/hashing"
-      "github.com/ava-labs/avalanchego/utills/wrappers"
-      "github.com/ava-labs/avalanchego/vms/components/verify"
-      "github.com/ava-labs/avalanchego/vms/secp256k1fx"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/params"
+
+      "github.com/flare-foundation/flare/codec"
+      "github.com/flare-foundation/flare/database"
+      "github.com/flare-foundation/flare/ids"
+      "github.com/flare-foundation/flare/snow"
+      "github.com/flare-foundation/flare/utills"
+      "github.com/flare-foundation/flare/utills/crypto"
+      "github.com/flare-foundation/flare/utills/hashing"
+      "github.com/flare-foundation/flare/utills/wrappers"
+      "github.com/flare-foundation/flare/vms/components/verify"
+      "github.com/flare-foundation/flare/vms/secp256k1fx"
)

var (
diff --git a/plugin/evm/tx_heap.go b/plugin/evm/tx_heap.go
index 11c1319..2d79d0f 100644
--- a/plugin/evm/tx_heap.go
+++ b/plugin/evm/tx_heap.go
@@ -3,7 +3,7 @@ package evm
import (
    "container/heap"

-      "github.com/ava-labs/avalanchego/ids"
+      "github.com/flare-foundation/flare/ids"
)

// txEntry is used to track the [gasPrice] transactions pay to be included in
diff --git a/plugin/evm/tx_test.go b/plugin/evm/tx_test.go
index e0482e2..c0b0984 100644
--- a/plugin/evm/tx_test.go
+++ b/plugin/evm/tx_test.go
@@ -8,9 +8,9 @@ import (
    "strings"
    "testing"

-      "github.com/ava-labs/avalanchego/chains/atomic"
-      "github.com/ava-labs/avalanchego/snow"
-      "github.com/ava-labs/coreth/params"
+      "github.com/flare-foundation/coreth/params"
+      "github.com/flare-foundation/flare/chains/atomic"
+      "github.com/flare-foundation/flare/snow"
)

func TestCalculateDynamicFee(t *testing.T) {
diff --git a/plugin/evm/user.go b/plugin/evm/user.go
index e190259..fea1be4 100644
--- a/plugin/evm/user.go
+++ b/plugin/evm/user.go
@@ -7,10 +7,10 @@ import (
    "errors"
    "fmt"

-      "github.com/ava-labs/avalanchego/database/encdb"
-      "github.com/ava-labs/avalanchego/ids"
-      "github.com/ava-labs/avalanchego/utills/crypto"
+      "github.com/ethereum/go-ethereum/common"
+      "github.com/flare-foundation/flare/database/encdb"
+      "github.com/flare-foundation/flare/ids"
+      "github.com/flare-foundation/flare/utills/crypto"
)

// Key in the database whose corresponding value is the list of
diff --git a/plugin/evm/vm.go b/plugin/evm/vm.go
index 66b98eb..d5a130a 100644
--- a/plugin/evm/vm.go
+++ b/plugin/evm/vm.go
@@ -16,49 +16,49 @@ import (
    "sync"
    "time"

-      "github.com/ava-labs/avalanchego/database/versiondb"
-      coreth "github.com/ava-labs/coreth/chain"
-      "github.com/ava-labs/coreth/consensus/dummy"
-      "github.com/ava-labs/coreth/core"
-      "github.com/ava-labs/coreth/core/state"
-      "github.com/ava-labs/coreth/core/types"
-      "github.com/ava-labs/coreth/eth/ethconfig"
-      "github.com/ava-labs/coreth/node"
-      "github.com/ava-labs/coreth/params"
-
-      "github.com/ava-labs/coreth/rpc"
+      coreth "github.com/flare-foundation/coreth/chain"
+      "github.com/flare-foundation/coreth/consensus/dummy"
+      "github.com/flare-foundation/coreth/core"
+      "github.com/flare-foundation/coreth/core/state"
+      "github.com/flare-foundation/coreth/core/types"
+      "github.com/flare-foundation/coreth/eth/ethconfig"
+      "github.com/flare-foundation/coreth/node"
+      "github.com/flare-foundation/coreth/params"
+      "github.com/flare-foundation/flare/database/versiondb"
+
+      "github.com/ethereum/go-ethereum/common"
+      "github.com/ethereum/go-ethereum/log"
+      "github.com/ethereum/go-ethereum/rlp"
+      "github.com/flare-foundation/coreth/rpc"
)

avalancheRPC "github.com/gorilla/rpc/v2"

-      "github.com/ava-labs/avalanchego/cache"
-      "github.com/ava-labs/avalanchego/codec"
-      "github.com/ava-labs/avalanchego/codec/linearcodec"
-      "github.com/ava-labs/avalanchego/database"
-      "github.com/ava-labs/avalanchego/database/manager"
-      "github.com/ava-labs/avalanchego/database/prefixdb"
-      "github.com/ava-labs/avalanchego/ids"
-      "github.com/ava-labs/avalanchego/snow"
-      "github.com/ava-labs/avalanchego/snow/choices"
-      "github.com/ava-labs/avalanchego/snow/consensus/snowman"
-      "github.com/ava-labs/avalanchego/snow/engine/snowman/block"
-      "github.com/ava-labs/avalanchego/utills/constants"
-      "github.com/ava-labs/avalanchego/utills/crypto"
-      "github.com/ava-labs/avalanchego/utills/formatting"
-      "github.com/ava-labs/avalanchego/utills/logging"
-      "github.com/ava-labs/avalanchego/utills/math"
-      "github.com/ava-labs/avalanchego/utills/profiler"
-      "github.com/ava-labs/avalanchego/utills/timer/mockable"
-      "github.com/ava-labs/avalanchego/utills/wrappers"
-      "github.com/ava-labs/avalanchego/vms/components/avax"
-      "github.com/ava-labs/avalanchego/vms/components/chain"
-      "github.com/ava-labs/avalanchego/vms/secp256k1fx"
-
-      commonEng "github.com/ava-labs/avalanchego/snow/engine/common"

```

```

- avalancheJSON "github.com/ava-labs/avalanchego/utls/json"
+ "github.com/flare-foundation/flare/cache"
+ "github.com/flare-foundation/flare/codecs"
+ "github.com/flare-foundation/flare/codecs/linearcodec"
+ "github.com/flare-foundation/flare/database"
+ "github.com/flare-foundation/flare/database/manager"
+ "github.com/flare-foundation/flare/database/prefixdb"
+ "github.com/flare-foundation/flare/ids"
+ "github.com/flare-foundation/flare/snow"
+ "github.com/flare-foundation/flare/snow/choices"
+ "github.com/flare-foundation/flare/snow/consensus/snowman"
+ "github.com/flare-foundation/flare/snow/engine/snowman/block"
+ "github.com/flare-foundation/flare/utls/constants"
+ "github.com/flare-foundation/flare/utls/crypto"
+ "github.com/flare-foundation/flare/utls/formatting"
+ "github.com/flare-foundation/flare/utls/logging"
+ "github.com/flare-foundation/flare/utls/math"
+ "github.com/flare-foundation/flare/utls/profiler"
+ "github.com/flare-foundation/flare/utls/timer/mockable"
+ "github.com/flare-foundation/flare/utls/wrappers"
+ "github.com/flare-foundation/flare/vms/components/avax"
+ "github.com/flare-foundation/flare/vms/components/chain"
+ "github.com/flare-foundation/flare/vms/secp256k1fx"
+
+ commonEng "github.com/flare-foundation/flare/snow/engine/common"
+
+ avalancheJSON "github.com/flare-foundation/flare/utls/json"
)

const (
diff --git a/plugin/evm/vm_test.go b/plugin/evm/vm_test.go
index 2050e18..006b1cc 100644
--- a/plugin/evm/vm_test.go
+++ b/plugin/evm/vm_test.go
@@ -16,39 +16,39 @@ import (
    "testing"
    "time"

- "github.com/ava-labs/coreth/trie"
- "github.com/ethereum/go-ethereum/common"
- "github.com/ethereum/go-ethereum/log"
+ "github.com/flare-foundation/coreth/trie"
+
+ "github.com/stretchr/testify/assert"

- "github.com/ava-labs/avalanchego/api/keystore"
- "github.com/ava-labs/avalanchego/chains/atomic"
- "github.com/ava-labs/avalanchego/database/manager"
- "github.com/ava-labs/avalanchego/database/prefixdb"
- "github.com/ava-labs/avalanchego/ids"
- "github.com/ava-labs/avalanchego/snow"
- "github.com/ava-labs/avalanchego/snow/choices"
- "github.com/ava-labs/avalanchego/utls/crypto"
- "github.com/ava-labs/avalanchego/utls/formatting"
- "github.com/ava-labs/avalanchego/utls/hashing"
- "github.com/ava-labs/avalanchego/utls/logging"
- "github.com/ava-labs/avalanchego/utls/units"
- "github.com/ava-labs/avalanchego/version"
- "github.com/ava-labs/avalanchego/vms/components/avax"
- "github.com/ava-labs/avalanchego/vms/components/chain"
- "github.com/ava-labs/avalanchego/vms/secp256k1fx"
-
- engCommon "github.com/ava-labs/avalanchego/snow/engine/common"
-
- "github.com/ava-labs/coreth/consensus/dummy"
- "github.com/ava-labs/coreth/core"
- "github.com/ava-labs/coreth/core/types"
- "github.com/ava-labs/coreth/eth"
- "github.com/ava-labs/coreth/params"
- "github.com/ava-labs/coreth/rpc"
-
- accountKeystore "github.com/ava-labs/coreth/accounts/keystore"
+ "github.com/flare-foundation/flare/api/keystore"
+ "github.com/flare-foundation/flare/chains/atomic"
+ "github.com/flare-foundation/flare/database/manager"
+ "github.com/flare-foundation/flare/database/prefixdb"
+ "github.com/flare-foundation/flare/ids"
+ "github.com/flare-foundation/flare/snow"
+ "github.com/flare-foundation/flare/snow/choices"
+ "github.com/flare-foundation/flare/utls/crypto"
+ "github.com/flare-foundation/flare/utls/formatting"
+ "github.com/flare-foundation/flare/utls/hashing"
+ "github.com/flare-foundation/flare/utls/logging"
+ "github.com/flare-foundation/flare/utls/units"
+ "github.com/flare-foundation/flare/version"
+ "github.com/flare-foundation/flare/vms/components/avax"
+ "github.com/flare-foundation/flare/vms/components/chain"
+ "github.com/flare-foundation/flare/vms/secp256k1fx"
+
+ engCommon "github.com/flare-foundation/flare/snow/engine/common"
+
+ "github.com/flare-foundation/coreth/consensus/dummy"
+ "github.com/flare-foundation/coreth/core"
+ "github.com/flare-foundation/coreth/core/types"
+ "github.com/flare-foundation/coreth/eth"
+ "github.com/flare-foundation/coreth/params"
+ "github.com/flare-foundation/coreth/rpc"
+
+ accountKeystore "github.com/flare-foundation/coreth/accounts/keystore"
)

var (
diff --git a/plugin/main.go b/plugin/main.go
index 96cd428..ced9384 100644
--- a/plugin/main.go
+++ b/plugin/main.go
@@ -9,9 +9,9 @@ import (
    "github.com/hashicorp/go-plugin"

- "github.com/ava-labs/avalanchego/vms/rpcchainvm"
+ "github.com/flare-foundation/flare/vms/rpcchainvm"

- "github.com/ava-labs/coreth/plugin/evm"
+ "github.com/flare-foundation/coreth/plugin/evm"
)

func main() {
diff --git a/scripts/build.sh b/scripts/build.sh
index 98c1967..76d8ce8 100755
--- a/scripts/build.sh
+++ b/scripts/build.sh
@@ -4,7 +4,7 @@ set -o errexit
set -o nounset
set -o pipefail

-# Avalanche root directory
+# Coreth root directory
CORETH_PATH=$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )

# Load the versions
@@ -27,4 +27,4 @@ coreth_commit=${CORETH_COMMIT:-$( git rev-list -1 HEAD )}

```

```

# Build Coreth, which is run as a subprocess
echo "Building Coreth Version: $coreth_version; GitCommit: $coreth_commit"
-go build -ldflags "-X github.com/ava-labs/coreth/plugin/evm.GitCommit=$coreth_commit -X github.com/ava-labs/coreth/plugin/evm.Version=$coreth_version" -o "$binary_path" "plugin/**/*.go"
+go build -ldflags "-X github.com/flare-foundation/coreth/plugin/evm.GitCommit=$coreth_commit -X github.com/flare-foundation/coreth/plugin/evm.Version=$coreth_version" -o "$binary_path" "plugin/**/*.go"
diff --git a/scripts/build_image.sh b/scripts/build_image.sh
deleted file mode 100755
index 6de7673..0000000
--- a/scripts/build_image.sh
+++ /dev/null
@@ -1,20 +0,0 @@
-#!/usr/bin/env bash
-
-set -o errexit
-set -o nounset
-set -o pipefail
-
-# Avalanche root directory
-CORETH_PATH=$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )
-
-# Load the versions
-source "$CORETH_PATH"/scripts/versions.sh
-
-# Load the constants
-source "$CORETH_PATH"/scripts/constants.sh
-
-echo "Building Docker Image: $dockerhub_repo:$build_image_id based of $avalanche_version"
-docker build -t "$dockerhub_repo:$build_image_id" "$CORETH_PATH" -f "$CORETH_PATH/Dockerfile" \
- --build-arg AVALANCHE_VERSION="$avalanche_version" \
- --build-arg CORETH_COMMIT="$coreth_commit" \
- --build-arg CURRENT_BRANCH="$current_branch"
diff --git a/scripts/constants.sh b/scripts/constants.sh
index 262f1a4..086dee9 100644
--- a/scripts/constants.sh
+++ b/scripts/constants.sh
@@ -4,20 +4,12 @@
GOPATH=$(go env GOPATH)"

# Set binary location
-binary_path=${CORETH_BINARY_PATH:-"$GOPATH/src/github.com/ava-labs/avalanchego/build/plugins/evm"}
-
-# Avalabs docker hub
-dockerhub_repo="avaplatform/avalanchego"
+binary_path=${CORETH_BINARY_PATH:-"$GOPATH/src/github.com/flare-foundation/coreth/build/plugins/evm"}

# Current branch
current_branch=${CURRENT_BRANCH:-$(git describe --tags --exact-match 2> /dev/null || git symbolic-ref -q --short HEAD || git rev-parse --short HEAD)}
echo "Using branch: ${current_branch}"

-# Image build id
-# Use an abbreviated version of the full commit to tag the image.
-
-# WARNING: this will use the most recent commit even if there are un-committed changes present
-coreth_commit=$(git --git-dir="$CORETH_PATH/.git" rev-parse HEAD)
-coreth_commit_id=${coreth_commit:0:8}
-
-build_image_id=${BUILD_IMAGE_ID:-"$avalanche_version-$coreth_commit_id"}
diff --git a/scripts/versions.sh b/scripts/versions.sh
index 6f2cc89..48b1df6 100644
--- a/scripts/versions.sh
+++ b/scripts/versions.sh
@@ -1,6 +1,6 @@
-#!/usr/bin/env bash
-
-# Set up the versions to be used
-coreth_version=${CORETH_VERSION:-'v0.7.4'}
+coreth_version=${CORETH_VERSION:-'v0.1.5'}
+coreth_version=${CORETH_VERSION:-'v0.1.5'}
# Don't export them as they're used in the context of other calls
-avalanche_version=${AVALANCHE_VERSION:-'v1.6.4'}
+avalanche_version=${AVALANCHE_VERSION:-'v1.6.4'}
+flare_version=${FLARE_VERSION:-'v0.2.2'}
diff --git a/signer/core/apitypes/types.go b/signer/core/apitypes/types.go
index cab7f9c..60afd52 100644
--- a/signer/core/apitypes/types.go
+++ b/signer/core/apitypes/types.go
@@ -32,9 +32,9 @@ import (
    "math/big"
    "strings"

-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/common/hexutil"
+    "github.com/flare-foundation/coreth/core/types"
)

type ValidationInfo struct {
diff --git a/tests/init.go b/tests/init.go
index d08bca7..8701a5b 100644
--- a/tests/init.go
+++ b/tests/init.go
@@ -31,7 +31,7 @@ import (
    "math/big"
    "sort"

-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/params"
)

// Forks table defines supported forks and their chain config.
diff --git a/tests/init_test.go b/tests/init_test.go
index 1ddbfc..5888be0 100644
--- a/tests/init_test.go
+++ b/tests/init_test.go
@@ -40,7 +40,7 @@ import (
    "strings"
    "testing"

-    "github.com/ava-labs/coreth/params"
+    "github.com/flare-foundation/coreth/params"
)

func readJSON(reader io.Reader, value interface{}) error {
diff --git a/tests/state_test_util.go b/tests/state_test_util.go
index 6e33efa..73ae6c1 100644
--- a/tests/state_test_util.go
+++ b/tests/state_test_util.go
@@ -34,17 +34,17 @@ import (
    "strconv"
    "strings"

-    "github.com/ava-labs/coreth/core"
-    "github.com/ava-labs/coreth/core/state"
-    "github.com/ava-labs/coreth/core/state/snapshot"
-    "github.com/ava-labs/coreth/core/types"
-    "github.com/ava-labs/coreth/core/vm"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/params"
-    "github.com/ethereum/go-ethereum/common"
-    "github.com/ethereum/go-ethereum/common/hexutil"
-    "github.com/ethereum/go-ethereum/common/math"
-    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/core"
+    "github.com/flare-foundation/coreth/core/state"
+    "github.com/flare-foundation/coreth/core/state/snapshot"
+    "github.com/flare-foundation/coreth/core/types"
+    "github.com/flare-foundation/coreth/core/vm"

```

```

+         "github.com/flare-foundation/coreth/ethdb"
+         "github.com/flare-foundation/coreth/params"
+     )

    // StateTest checks transaction processing without block context.
diff --git a/trie/database.go b/trie/database.go
index 4f59c68..c5e59f4 100644
--- a/trie/database.go
+++ b/trie/database.go
@@ -36,12 +36,12 @@ import (
     "time"

-    "github.com/VictoriaMetrics/fastcache"
-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/metrics"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/ethdb"
 )

var (
diff --git a/trie/database_test.go b/trie/database_test.go
index bc3d3e9..626ddfe 100644
--- a/trie/database_test.go
+++ b/trie/database_test.go
@@ -29,8 +29,8 @@ package trie
import (
    "testing"

-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
 )

// Tests that the trie database returns a missing trie node error if attempting
diff --git a/trie/iterator.go b/trie/iterator.go
index 4ab0863..7863a00 100644
--- a/trie/iterator.go
+++ b/trie/iterator.go
@@ -31,9 +31,9 @@ import (
    "container/heap"
    "errors"

-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/ethdb"
 )

// Iterator is a key-value trie iterator that traverses a Trie.
diff --git a/trie/iterator_test.go b/trie/iterator_test.go
index 7ef711c..fe13af8 100644
--- a/trie/iterator_test.go
+++ b/trie/iterator_test.go
@@ -33,10 +33,10 @@ import (
    "math/rand"
    "testing"

-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
 )

func TestIterator(t *testing.T) {
diff --git a/trie/proof.go b/trie/proof.go
index 37ce9ed..5900e9e 100644
--- a/trie/proof.go
+++ b/trie/proof.go
@@ -31,11 +31,11 @@ import (
    "errors"
    "fmt"

-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
 )

// Prove constructs a merkle proof for key. The result contains all encoded nodes
diff --git a/trie/proof_test.go b/trie/proof_test.go
index f37c206..895babf 100644
--- a/trie/proof_test.go
+++ b/trie/proof_test.go
@@ -35,9 +35,9 @@ import (
    "testing"
    "time"

-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
 )

func init() {
diff --git a/trie/secure_trie.go b/trie/secure_trie.go
index 4f6a556..0d324e1 100644
--- a/trie/secure_trie.go
+++ b/trie/secure_trie.go
@@ -29,10 +29,10 @@ package trie
import (
    "fmt"

-    "github.com/ava-labs/coreth/core/types"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/core/types"
 )

// SecureTrie wraps a trie with key hashing. In a secure trie, all
diff --git a/trie/secure_trie_test.go b/trie/secure_trie_test.go
index c9c29d7..642e7d2 100644
--- a/trie/secure_trie_test.go
+++ b/trie/secure_trie_test.go
@@ -32,9 +32,9 @@ import (
    "sync"
    "testing"

-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
 )

func newEmptySecure() *SecureTrie {

```

```

diff --git a/trie/stacktrie.go b/trie/stacktrie.go
index 248b4e4..95fc7da 100644
--- a/trie/stacktrie.go
+++ b/trie/stacktrie.go
@@ -35,10 +35,10 @@ import (
    "io"
    "sync"

-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/ethdb"
)

var ErrCommitDisabled = errors.New("no database for committing")
diff --git a/trie/stacktrie_test.go b/trie/stacktrie_test.go
index 189e9d4..b963d6b 100644
--- a/trie/stacktrie_test.go
+++ b/trie/stacktrie_test.go
@@ -31,9 +31,9 @@ import (
    "math/big"
    "testing"

-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
)

func TestSizeBug(t *testing.T) {
diff --git a/trie/sync.go b/trie/sync.go
index 85b1f5a..2469d9c 100644
--- a/trie/sync.go
+++ b/trie/sync.go
@@ -30,10 +30,10 @@ import (
    "errors"
    "fmt"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/common/prque"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/ethdb"
)

// ErrNotRequested is returned by the trie sync when it's requested to process a
diff --git a/trie/sync_bloom.go b/trie/sync_bloom.go
index 51c318c..233f488 100644
--- a/trie/sync_bloom.go
+++ b/trie/sync_bloom.go
@@ -33,11 +33,11 @@ import (
    "sync/atomic"
    "time"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/ethdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/metrics"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/ethdb"
+    bloomfilter "github.com/holiman/bloomfilter/v2"
)

diff --git a/trie/sync_test.go b/trie/sync_test.go
index d3bbdbd..105a7c1 100644
--- a/trie/sync_test.go
+++ b/trie/sync_test.go
@@ -30,9 +30,9 @@ import (
    "bytes"
    "testing"

-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
)

// makeTestTrie create a sample test trie to test node-wise reconstruction.
diff --git a/trie/trie.go b/trie/trie.go
index ac68272..200d79e 100644
--- a/trie/trie.go
+++ b/trie/trie.go
@@ -33,11 +33,11 @@ import (
    "fmt"
    "sync"

-    "github.com/ava-labs/coreth/core/rawdb"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/ethereum/go-ethereum/log"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/core/types"
)

var (
diff --git a/trie/trie_test.go b/trie/trie_test.go
index 8f2447f..40f64b9 100644
--- a/trie/trie_test.go
+++ b/trie/trie_test.go
@@ -38,13 +38,13 @@ import (
    "testing"
    "testing/quick"

-    "github.com/ava-labs/coreth/core/rawdb"
-    "github.com/ava-labs/coreth/ethdb"
-    "github.com/ava-labs/coreth/ethdb/memorydb"
+    "github.com/davecgh/go-spew/spew"
+    "github.com/ethereum/go-ethereum/common"
+    "github.com/ethereum/go-ethereum/crypto"
+    "github.com/ethereum/go-ethereum/rlp"
+    "github.com/flare-foundation/coreth/core/rawdb"
+    "github.com/flare-foundation/coreth/ethdb"
+    "github.com/flare-foundation/coreth/ethdb/memorydb"
+    "golang.org/x/crypto/sha3"
)

```