Optimism Bridge Box¶

Truffle Optimism Bridge Box provides you with setup necessary to start building applications between multiple Ethereum network layers.

This box contains contracts that interact with the Optimism bridge on L1 and L2, along with a set of migrations for deploying, calling functions, and passing messages and value between both layers.

- Requirements
- Installation
- Setup
- Installing dependencies
- Using the .env File
- Bridging
- Messaging Demo
- Known Issues
- Developing for Optimism
- Support

Table of contents generated with markdown-toc

Requirements¶

We recommend following these instructionshere . The Optimism Box has the following requirements:

- Node.js
- v12 - 16 or later
- NPM
- version 5.2 or later
- Windows, Linux or MacOS
- AnInfura
- account and Project ID
- AMetaMask
- account

Installation¶

truffle unbox optimism-bridge

Setup¶

Installing dependencies¶

truffle unbox should runnpm install as part of the unboxing process.

Using the env File¶

You will need the goerli mnemonic to use with the network. The.dotenv npm package has been installed for you, and you will need to create a.env file for storing your mnemonic and any other needed private information.

The.env file is ignored by git in this project to help protect your private data. It is good security practice to avoid committing information about your private keys to github. Thetruffle-config.ovm.js file expects aGOERLI_MNEMONIC value to exist in.env for running commands on the Goerli and Optimism Goerli testnets and anINFURA_KEY to connect to the network.

If you are unfamiliar with using.env for managing your mnemonics and other keys, the basic steps for doing so are below:

1. Runcp .env.example .env
2. in the command line to copy some important variables into a private.env
3. file.
4. Open the.env
5. file in your preferred IDE
6. Fill in your mnemonic for the networks you intend to use, as well as youInfura key
7. .
8. As you develop your project, you can put any other sensitive information in the.env
9. file. You can access it from other files withrequire('dotenv').config()
10. and refer to the variable you need withprocess.env['']
11. .

Bridging¶

This box includes:

- AnL1 contract
- that sends a message over the Optimism bridge.
- AMigration
- that sends a message from Ethereum to Optimism.
- AnL2 contract
- that sends a message over the Optimism bridge.
- AMigration
- that sends a message from Optimism to Ethereum.
- Ascript
- to automate the process of compiling contracts, running migrations, and sending messages across each side of the bridge.
- Ascript
- to automate the process of sending ETH and DAI across each side of the bridge.

Once you have installed dependencies and set up your.env file, you're ready to start bridging!

Messaging Demo¶

Included is a helperscript that facilitates the full compilation, migration, and bridging of messages between Goerli and Optimism Goerli. To use it, you will need testnet ETH on those networks. Usea faucet to receive some. Additionally, you will need toadd the Optimism addon to your Infura account.

Once youre ready, run:

npm run deploy This script automates the following steps:

Migration 1 + 2¶

The first two migrations are simple contract deploys to each network. These are necessary for the following migrations.

Migration 3¶

Upon completion of the migration, you will be prompted with a link to confirm the bridged message via Etherscan:

Expected output:

Updating the L2 Greetings contract from L1! Greeter txn confirmed on L1! 0xabc... Bridging message to L2 Greeter contract... In about 1

minute, check the Greeter contract "read"

function : https://goerli-optimistic.etherscan.io/address/0xD4c204223d6F1Dfad0b7a0b05BB0bCaB6665e0c9#readContract Click the link and open thegreet function to see your greeting!

Migration 4¶

Upon completion of the migration, you will be prompted with a link to confirm the bridged message via Etherscan:

Expected output:

Updating the L1 Greetings contract from L2! Greeter txn confirmed on L2! 0x93d390d84e99a0e229ef813afe4b42d2cfed8ac1f8f0711e721cce4eab30046c
Bridging message to L1 Greeter contract. This will take at least 1 -5 min... Message not yet received on L1. Retrying in

10

seconds... Message received! Finalizing... Message finalized. Check the L1 Greeter contract "read"

function : https://goerli.etherscan.io/address/0x11fB328D5Bd8E27917535b6d40b881d35BC39Be0#readContract Click the link and open thegreet function to see your greeting!

Known Issues¶

There is known issue that occurs under certain network conditions resulting in the failure of migration 4 with the following error:

Error: Could not find block @trufflesuite/web3-provider-engine/index.js:163 This is due to an issue with a dependency and we are working on a fix. In the meantime, if you encounter this, it is safe to simply rerun that migration withtruffle migrate --network=optimistic_goerli --config=truffle-config.ovm --f 4 --to 4 --skip-dry-run .

Briding Eth Demo¶

The scriptgoerli_bridge_value.mjs demonstrates how to send ETH and DAI across each side of the bridge. To run it, you will also need DAI on the Goerli network. You can go to Uniswap to exchange ETH for DAI on Goerli. Then call:

node ./scripts/goerli_bridge_value.js

Developing for Optimism¶

To learn more about developing for Optimism, see theTruffle Optimism Box

Support¶

Support for this box is available via the Truffle community availablehere .