Build an Aztec Connect-style Uniswap

note Before going through this tutorial, you will need to have completed the Token Bridge tutorial Our goal here is for someone with funds on L2 to be able to swap using L1 Uniswap and then get the swapped assets back to L2. In this tutorial, L1 will refer to Ethereum and L2 will refer to Aztec.

The flow will be:

1. The user withdraws their “input” assets to L1 (i.e. burn them on L2 and create a L2 to L1 message to withdraw)
2. We create an L2 → L1 message to swap on L1
3. On L1, the user gets their input tokens, consumes the swap message, and executes the swap
4. The user deposits the “output” tokens to the output token portal so it can be deposited into L2

We will assume that token portals and token bridges for the input and output tokens must exist. These are what we built in the previous tutorial.

The execution of swap on L1 should be designed such that any 3rd party can execute the swap on behalf of the user.

In this tutorial, we will code both the private and public flow!

We will create:

1. Uniswap Portal - a contract on L1 that talks to the input token portal to withdraw the assets, executes the swap, and deposits the swapped tokens back to L2
2. Uniswap L2 contract - a contract on L2 that creates the needed messages to perform the swap on L1

This diagram describes the private flow.

Let’s get to the setup! Edit this page

Previous Deploy & Call Contracts with Typescript Next Setup & Installation