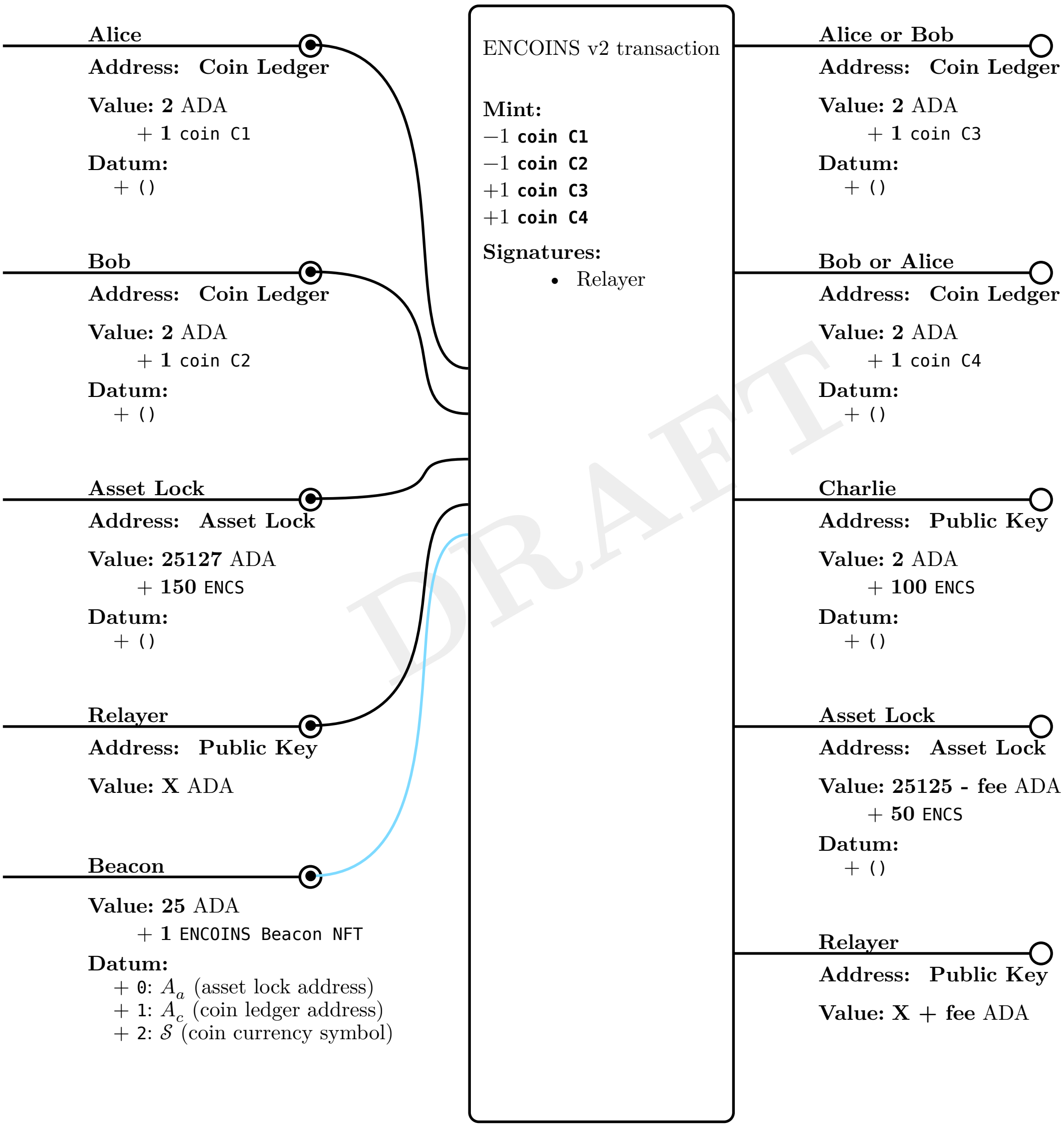
ENCOINS V2 INTEGRATION GUIDE

VLADIMIR SINYAKOV

1. Introduction

In this guide, we will explore how ENCOINS v2 can be integrated with other Cardano protocols. The figure below shows a *typical ENCOINS v2 transaction.*



Note: a typical ENCOINS v2 transaction

The transaction burns two coins that may belong to different users and mints two other coins. If Alice owns the first minted coin, then Bob owns the second one, or vice versa. The values of the minted coins may differ from each other and from the burned coins as long as the balance equation holds. Such a transaction can represent a value transfer from Alice to Bob, from Bob to Alice, or even an atomic swap of different assets between the parties. In this particular instance, a third user, Charlie, receives 50 `ENCS` from either Alice or Bob (or both). For the full details of the ENCOINS v2 protocol, please consult the protocol specification document.

In the following, we will consider two types of integration with the ENCOINS v2 protocol. We call an integration *atomic* if it is done within a single transaction that must satisfy both specifications. In a *sequential* integration, we chain transactions of different protocols that result in a more complex interaction between the two protocols. Sequential integration is generally more costly in terms of transaction fees but a bit easier to implement.

2. Sequential integration

We identify several ways to integrate ENCOINS v2 with your protocol in a sequential way.

- Lock funds in a validator script that requires running the ENCOINS v2 minting script as a spending condition. For example, the validator script may require minting a specific coin and sending it to a specific address as a spending condition.

  NOTE: the address must be different from the *asset lock* and the *coin ledger* addresses of the ENCOINS v2 protocol. This is because sending the funds directly to those addresses will make them unrecoverable.

- Use already minted ENCOINS v2 coins in your protocol. Examples:
  1. the coins could be used for OTC trading of fungible assets on an NFT marketplace. The proof of the value of a coin could be included in the item description (a non-privacy use case) or it could be communicated to a potential buyer in private.
  2. An auction with blind bids. The participants can lock their bids in a validator script that requires running the ENCOINS v2 minting script to unlock. The winning bid gets the auctioned token, and the seller gets the redemption value of the bid.

3. Atomic integration

3.1. **Integration via zkFold Symbolic scripts.** Besides value, every coin contains a script which is a boolean function of transaction data and some user input. These scripts are zkFold Symbolic scripts. In the ENCOINS v2 protocol, for every coin that is being burned in the transaction, the corresponding script must succeed on that transaction and some user input.

This provides the simplest way to integrate with other protocols that are based on zkFold Symbolic. With this approach, your protocol's specification check will be triggered as long as at least one coin with that specification is burned in the transaction.

3.2. **Integration via the Connector script.** If your protocol is not based on zkFold Symbolic, you can still trigger your *spending* or *minting* Plutus scripts using zkFold Symbolic Connector script. The Connector script checks that a particular validator or minting policy must succeed for the transaction to be valid. This way, you may require a specific spending or minting Plutus script to be executed whenever a coin is burned.

4. Acknowledgement

Thanks to Pi Lanningham and the Sundae Swap team for sharing the transaction figure template.