AIP-009: Cross-Shard Token Swap

seo@, lia@ 2020-09-11

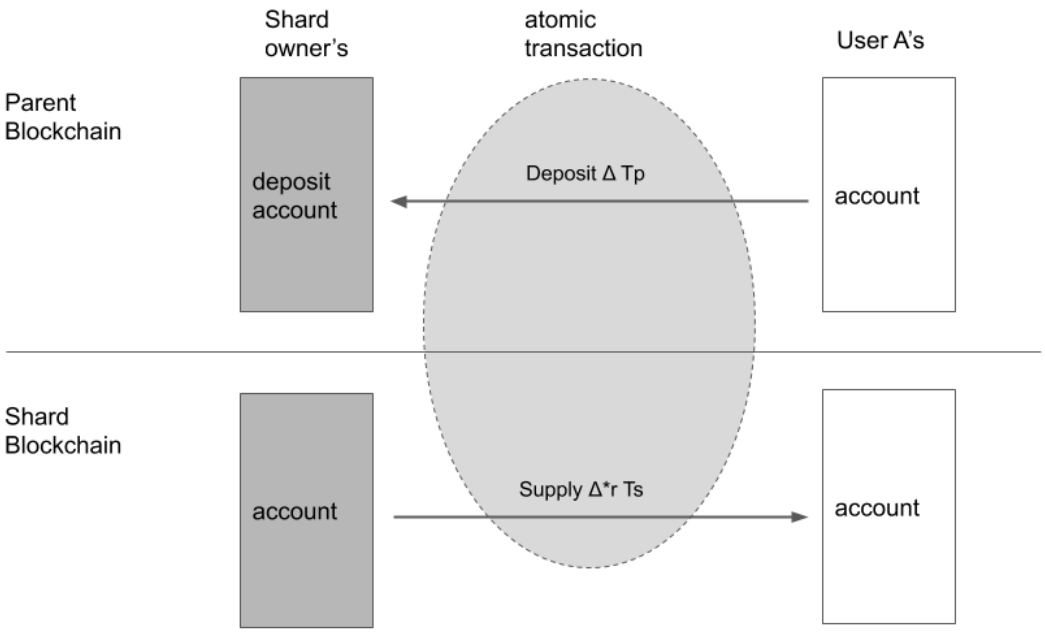
Goals

- Provide a design of cross-shard transactions for a narrow scenario: Cross-shard token swap transactions (check-in and check-out)

Problem Definition

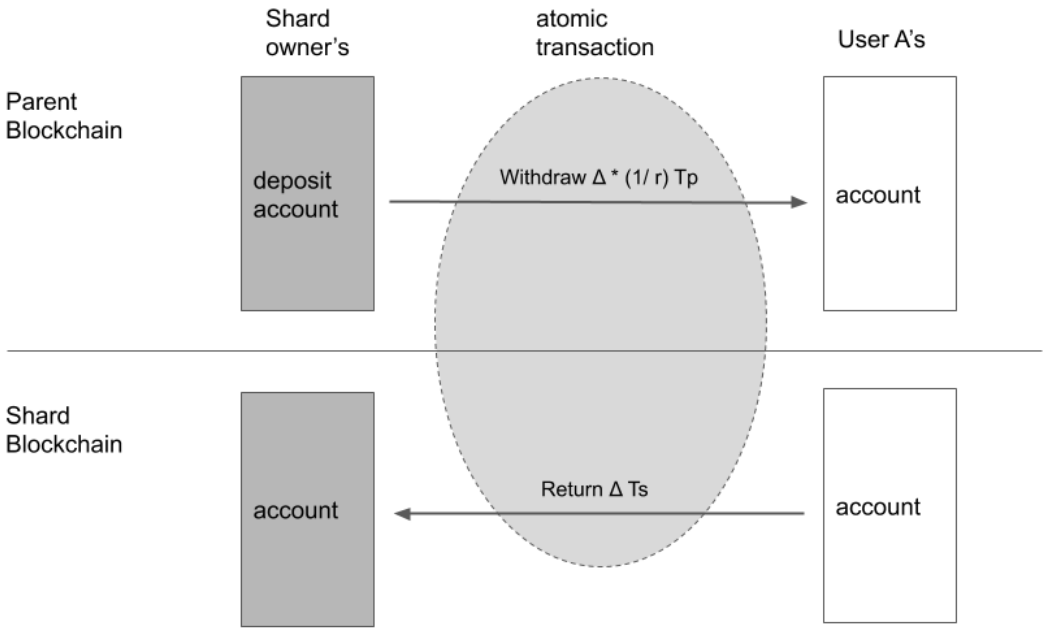
Let the token of the parent blockchain and the shard (or child) blockchain Tp and Ts, respectively.

Check-In



Check-in is a process of swapping a certain amount (say Δ) of Tp with the corresponding amount of Ts. In the process, a user (say A) deposits Δ amount of Tp to the shard owner's deposit account in the parent blockchain, and at the same time the shard owner supplies Δ * r amount of Ts to user A in the shard blockchain where r is the exchange rate.

Check-Out



Check-out is the opposite to the check-in process. By returning Δ amount of Ts to the shard owner in the shard blockchain, user A withdraws Δ * (1/ r) amount of Tp from the shard owner's deposit account in the parent blockchain where r is the exchange rate.

Requirements

- For cross-shard token swap transactions, the token transfer in the parent blockchain and the corresponding token transfer in the shard blockchain should be **atomic**

Proposed Design

Design Principles

- Assume that parent and shard blockchains are operated in trusty ways
- Minimize human intervention as much as possible

Key Ideas

- Use global path transactions to carry multi-blockchain operations in a single transaction
- Introduce token reservation and refund functionality for safety

- Introduce block-triggered native functions to make the check-in / check-out transactions run automatically
- Cancel transactions (TODO(seo))

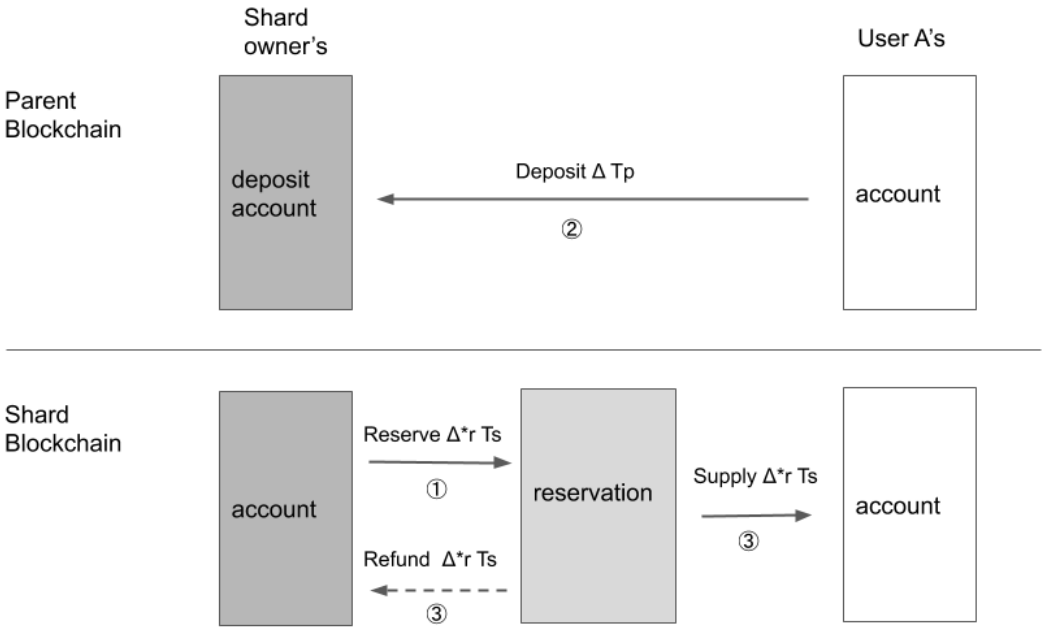
Design Details

Global Path Transactions

To use a signed transaction over blockchains, we need a global namespace of database paths. In shard blockchains, transactions are defined with local paths by default, e.g. in a shard blockchain with sharding path '/apps/afan', a transaction with path '/foo/bar' means '/apps/afan/foo/bar' globally. We can allow transactions with global path with flag is_global = true, e.g.:

```
{
  type: "SET_VALUE",
  ref: "/apps/afan/foo/bar",
  value: 10,
  is_global: true
}
```

Check-In



The check-in transaction is executed in the following steps:

1. User A submits a check-in transaction to the shard blockchain. The transaction contains:

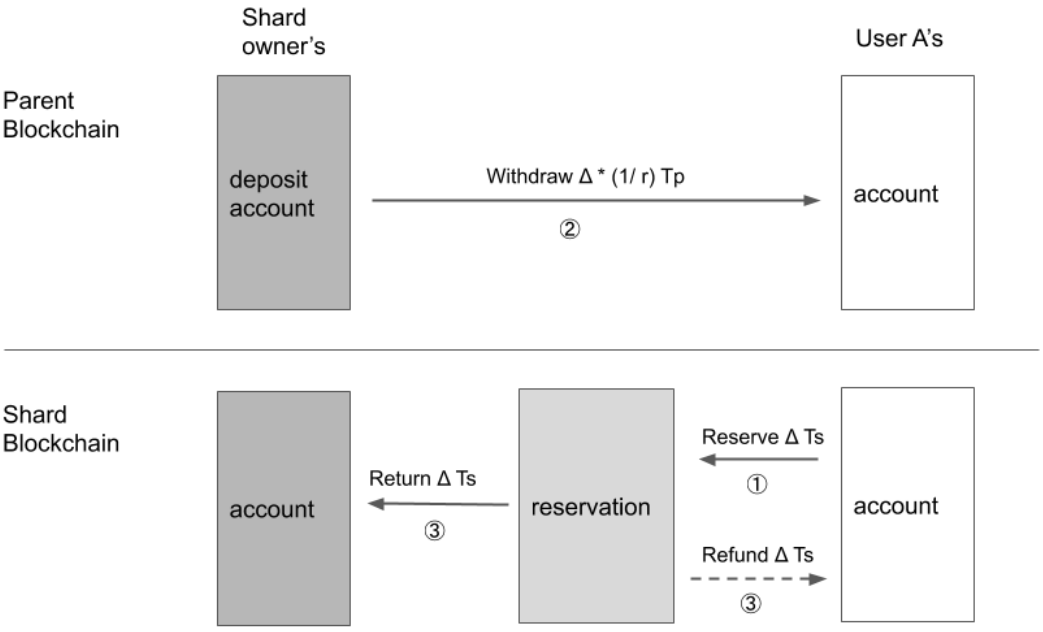
    a. An operation to trigger the check-in process in the shard blockchain
    b. A deposit operation of Δ Tp in the parent blockchain
2. Δ*r amount of Ts is reserved in the shard blockchain
3. The transaction is forwarded to the parent blockchain and wait until the deposit operation is finalized
4. If the transaction is successfully finalized, the reserved token is supplied to user A. Otherwise the reserved token is refunded.

The following is an example of the check-in transactions from users:

```
{
  type: "SET",
  op_list: [
    {
      ref: "/transfer/<User Address>/<Shard Owner Address>/value",
      value: 10,
      type: "SET_VALUE",
      is_global: true
    },
    {
      ref: "/apps/afan/check_in/<User Address>/<Check-in Key>/value",
      value: 10,
      type: "SET_VALUE",
      is_global: true
    },
  ]
}
```

In order to be forwarded, the transaction must have global paths (w/ is_global = true). Note that the sharding path is “/apps/afan” in the example.

Check-Out



The check-out transaction is executed in the following steps:

1. User A submits a check-out transaction to the shard blockchain. The transaction contains:
    a. An operation to trigger the check-out process in the shard blockchain
    b. An operation to reserve Δ Ts in the shard blockchain
2. Δ amount of Ts is reserved in the shard blockchain
3. A transaction containing the following operation is created and submitted to the parent blockchain and wait until the operation is finalized:
    a. An operation to withdraw Δ * (1/ r) Tp to user A in the parent blockchain
4. If the transaction is successfully finalized, the reserved token is returned to the shard owner. Otherwise the reserved token is refunded.

The following is an example of the check-out transactions from users:

```
{
  type: "SET",
  op_list: [
    {
      ref: "/reserve/<User Address>/<Shard Owner Address>/value",
      value: 100,
      type: "SET_VALUE",
    },
    {
```

```
      ref: "/check_out/<User Address>/<Check-out Key>/value",
      value: 100,
      type: "SET_VALUE",
    },
  ]
}
```

This transaction doesn't have to have global paths.

Native Functions

We introduce the following native functions.

Function Name	Process	Triggered By	Does
_open_checkin()	check-in	user	- Reserves Ts
_uplink_checkin()	check-in	new blocks	- Forwards the tx to the parent blockchain
_close_checkin()	check-in	new blocks	- Checks if the tx is finalized in the parent blockchain - Supply or refund Ts
_open_checkout()	check-out	user	- Reserves Ts
_uplink_checkout()	check-out	new blocks	- Creates and sends a tx to the parent blockchain
_close_checkout()	check-out	new blocks	- Checks if the tx is finalized in the parent blockchain - Supply or refund Ts

'Triggered by new blocks' means that the function is called when new blocks are created.

Tx Finalization Checking Native Function

In check-in or check-out processes, we need to check finalization of the uplinked transactions in the parent blockchain. The key idea is to introduce a native function triggered by every new block to perform the finalization check job.

We need the following new features:

- Let last block number is automatically recorded in the database
- An native function that checks finalization of given transactions in the parent blockchain

Shard Configuration

For the cross-shard token swap transactions, shard config includes the following properties:

Property name	Description	Example values
token_exchange_scheme	The token exchange scheme. Default value is “NONE”, which means no scheme is adopted.	“NONE”, “FIXED_RATED”
token_exchange_rate	The exchange rate of the tokens used when the token exchange scheme is “FIXED_RATED”. Value x means x amount of Ts is exchanged with 1 Tp. This value should be an integer number.	10, 100, 1000

Advanced Topics

Token Circulation Policies

As stated in the Design Principles, our cross-shard token swap scheme aims to minimize human intervention. One of the policies we can adopt to achieve the goal is the Single Circulation Source (SCS) Principle, introduced by the authors of Payment Guaranteed Polynomial Exchange Rate Scheme and Its Application to Cryptocurrency Swaps (PG-PERS). By limiting the sources of tokens to one, service providers can reduce the resources required to manage the exchange systems as well as the ecosystem. On top of that, the paper states that SCS guarantees payments in stable exchange systems.

In the cross-shard token swap scheme, either the shard owner’s accounts in parent and child chains or some other reserve pool would serve as the Single Circulation Source, and shard owners or service providers would need to decide on how to supply the tokens through the source. We see two general options here, one is a pre-deposit model and the other is an auto-mint model.

With a pre-deposit model, shard owners would deposit the majority, or all, of the minted Ts tokens into the SCS in the child chain, and deposit the corresponding amount of Tp tokens into the SCS in the parent chain. The initial ratio of the two deposits will determine the initial exchange rate. With an auto-mint model, the system can be configured to calculate the amount of Ts tokens to mint when a user deposits Tp tokens, according to a chosen exchange rate scheme, and transfer the minted Ts to the user. // TODO(lia): initial deposit이 필요한가?

Exchange Rate Schemes

In addition to the Single Circulation Source Principle mentioned in the previous section, more complex exchange rate schemes could be implemented to better reflect the dynamic values of tokens. In the PG-PERS paper, the authors present a consistent, stable, and resilient exchange rate scheme that calculates a given Ts (derived) token's value in terms of the token's circulation. The more Ts token is circulated, the higher its value is, and Ts/Tp rate rises. A function generated by the PG-PERS can be used in place of a fixed exchange rate by adding a few more variables and constants to the initial token swap configuration.

Roadmap

This project will be pushed with the following phases:

- Phase 0: Token reservation
- Phase 1: Block-triggered native function
- Phase 2: Check-in
    - open check-in
    - uplink check-in
    - close check-in
- Phase 3: Check-out
    - open check-out
    - uplink check-out
    - close check-out
- Phase 4: Advanced topics
    - token supply models
    - PG-PERS

Conclusion

- Provided a design for cross-shard token swap processes (check-in and check-out)
- We achieved atomicity of the cross-shard transactions by using global path transactions, token reservation functionality, and transaction finalization checking native function
- We covered advanced topics for token exchange rate schemes and token circulation policies

Links

- Plasma (link)
- Payment Guaranteed Polynomial Exchange Rate Scheme and Its Application to Cryptocurrency Swaps (PG-PERS) (pdf)

Document History

Date	Who	Change	Notes
2020-09-10	seo@	Initial draft	
2020-09-11	seo@, lia@	Internal review	
2020-09-20	lia@	Covered advanced topics	
2020-10-13	seo@	- Removed life-time period - Described global path transactions	
2021-05-07	seo@	Published	