DeBridge Audit

10. 12. 2021

by Ackee Blockchain

Table of Contents

1. Overview 2

2. Scope 4

3. System Overview 6

4. Security Specification 8

5. Findings 10

6. Conclusion 20

Document Revisions

Revision	Date	Description
1.0	1.11.2021	Initial document
2.0	6.12.2021	Re-audit Revision 2
2.1	20.12.2021	Final report of Revision 2

1. Overview

This document presents a security audit of the DeBridge protocol. An initial audit (revision 1) has been performed between 18.10.2021 and 29.10.2021. The time donation was 20 engineering days. Several issues were identified and reported to the client. This audit revision was agreed as an internal audit, therfore we don't comment code maturity or test coverage of this version in the report.

The revision 2 of the audit took place between 22.11.2021 and 3.12.2021 with a time donation of 10 engineering days. The main focus of the second revision was to check whenever reported issues from the first revision were correctly fixed. In the remaining time we focused on two main objectives. First one was a re-audit of code changes caused by refactoring of existing functionality as it could also bring unintended bugs and security issues. The second objective was an audit of newly added functionality.

This document presents our findings of both revisions.

1.1 Ackee Blockchain

Ackee Blockchain is an auditing company based in Prague, Czech Republic, specializing in audits and security assessments. Our mission is to build a stronger blockchain community by sharing knowledge – we run a free certification course Summer School of Solidity and teach at the Czech Technical University in Prague. Ackee Blockchain is backed by the largest VC fund focused on blockchain and DeFi in Europe, Rockaway Blockchain Fund.

1.2 Audit Methodology

1. **Technical specification/documentation** - a brief overview of the system is requested from the client and the scope of the audit is defined.
2. **Manual code review** - the code is checked line by line for vulnerabilities, code duplication, best practices and the code architecture is reviewed.
3. **Local deployment + hacking** - the programs are deployed locally and we try to attack the system and break it.

1.3 Review team

The audit has been performed with a total time donation of 20 + 10 engineering days. The work was divided between the Lead Auditor and second auditor who performed manual code reviews. The whole process was controlled by the Audit Supervisor.

Member's Name	Position
Tibor Tribus	Lead Auditor
Vladimír Marcin	Auditor
Josef Gattermayer, Ph.D.	Audit Supervisor

1.4 Disclaimer

We've put our best effort to find all vulnerabilities in the system, however, our findings shouldn't be considered as a complete list of all existing issues.

2. Scope

This chapter describes the audit scope, contains provided specifications, supporting documentation and sets the main objectives for the audit process.

2.1 Coverage

After an agreement with the client we did not audit any external call logic. If any of this logic was used in the audit scope we have just assumed it's safe.

Files being audited:

- `programs/settings/*`
- `programs/debridge/*`

Code excluded from the scope of the audit:

- `programs/debridge/lib.rs`
  - `pub fn init_external_call_storage(...)`[1]
  - `pub fn update_external_call_storage(...)`[1]
  - `pub fn close_external_call_storage(...)`[1]
  - `pub fn execute_external_call(...)`[1]
  - `mod external_call_storage{...}`

Sources revision used during the re-audit process:

- Repository: https://github.com/debridge-finance/solana-contracts
- Commit: 94b840c5742086de20eaa741eb8958b31574bb1f

2.2 Supporting Documentation

Most of the documentation is located in the Gitbook of the project. It was helpful to understand the overall architecture design.

Another documentation is available in the form of in-code comments (RustDocs) for `/program/settings/lib.rs` and `/program/debridge/lib.rs`. Readme of the project is extensive and covers the basic workflow of the protocol.

[1] We've also excluded the contexts used by the instructions.

2.3 Objectives

We’ve defined the following main objectives of the audit:

- Check the overall code quality and best practices.
- Check functionality of the system.
- Check if nobody unauthorized is able to claim or send assets.
- Check if the discovered issues were correctly fixed and check the correctness of a newly implemented logic for the change balance and keys management.
- Check if the refactored code didn’t bring any new issues.

3. System Overview

This chapter describes the audited system from our understanding.

The basic idea behind DeBridge is to allow a transfer of assets between various blockchains. It is achieved by locking/unlocking an asset on a native chain and minting/burning the wrapped asset on a secondary chain. From a smart contracts perspective, Solana is acting once as a native chain and once as a secondary chain, from this point of view, there are four possible scenarios.

Scenarios:

A. Transfer Solana SPL token from Solana to a secondary chain.
B. Transfer Solana SPL token back to Solana chain from a secondary chain.
C. Release asset on a native chain, and burn deAsset on Solana chain.
D. Transfer asset from a native chain, and mint deAsset on Solana chain.

The verified system behaves differently in each of these scenarios. The behaviour depends on a type of a bridge and a type of an invoked method. We will now discuss these scenarios in more detail.

`SendBridge` ≡ SOLANA is NATIVE:

- A. **locking (staking)**[2]
  - A transferred amount of an asset is locked by sending it to a wallet with which can operate only a smart contract.
  - Mint deAssets on a secondary chain[3].
- B. **releasing**[4]
  - Burn deAsset on a secondary chain[3].
  - Transfer Solana SPL tokens from a wallet owned by a smart contract to a user wallet at the Solana chain.

`MintBridge` ≡ SOLANA is SECONDARY:

- C. **burning**[2]
  - A particular amount of deAsset is burned from a sender wallet on the Solana chain.
  - Release native assets on a native chain[3].
- D. **minting**[4]
  - Lock native assets on a native chain[3].
  - A transferred amount of deAsset is minted and sent to a wallet owned by a user.

[2] Invoking the `send` instruction.
[3] This is happening on other chains (not Solana).
[4] Invoking the `claim` instruction.

3.1 Key components

The project is divided into two programs: `settings`, `debridge`. Settings program is responsible for initialization and managing a state of the protocol. The Debridge program is responsible for asset flow between chains.

We have identified the following key components in the system. And that's what we tried to focus on the most.

Bridge is represented by `Bridge` structure in `settings/src/bridge.rs [loc: 225]`

- Holds information about a type of a bridge (sendBridge, mintBridge), and a state of the bridge.

State is represented by `State` structure in `settings/src/state.rs [loc: 368]`

- Holds all information about the state of the system. Defines who is Authority, StopTap, FeeBeneficiar. Store oracles addresses and information necessary to validate transactions.

Send instruction in `debridge/src/lib.rs [loc: 208]`

- Is invoked by a user when he wants to initialize a transfer of assets from the Solana chain. This instruction can operate in two different modes (locking assets/burning assets) based on a bridge type.

Claim instruction in `debridge/src/lib.rs [loc: 351]`

- Is invoked by a user when he wants to transfer assets to the Solana chain. This instruction can operate in two different modes (releasing assets/minting assets) based on a bridge type.

An overview of all components can be found in diagrams in Appendix D and Appendix E.

4. Security Specification

This section specifies single roles and their relationships in terms of security in our understanding of the audited system. These understandings are later validated.

4.1 Actors

This part describes actors of the system, their roles and permissions.

Protocol Authority

Protocol authority is represented by an account that is associated with a state of the system. It's responsible for initiating and managing the state and also for fees collection. It's the account with the highest permission level in the system.

Protocol

The protocol itself is the owner of a wallet in which the protocol holds a locked asset and is, therefore, the only one who can release the assets. It is also responsible for minting new tokens and only the protocol can give consent for minting.

Stop Tap

The account has the privileges to stop the protocol but does not have the authority to start it. This account can be changed by the Protocol Authority.

User

A user of DeBridge protocol. This account can initialize Bridge, and invoke Send, Claim instructions, however it can manipulate only assets owned by him.

Claimer

This role is used for automatic claims on behalf of the user of DeBridge protocol. But just in case user pays execution fees in advance.

4.2 Trust model

Users have to trust developers that logic and math algorithms are correctly implemented and they receive the correct amount of Assets or deAssets.

The user must also trust the validators not to team up and thus attack the protocol. However, the debridge protocol implements delegated staking and slashing mechanics that act as a backbone for the protocol security and prevent economic incentives for validators to collude.

5. Findings

This chapter shows detailed output of our analysis and testing.

5.1 General Comments

This section presents an overall engineering culture that is a crucial precursor of the right security.

5.1.1 Overall code quality

An overall code quality is solid, the code is well written and understandable. We haven't found any major bad practices.

5.1.2 Commit culture
Developers successfully adopted a standardized commit message format that contains a name of a component and a short description.

Individual commits are small with descriptive commits titles which makes understanding of the repository much easier.

Some examples:

`commit c2777ebef67a43bf66c25e4fe0b1613f286aa9aa`
- 2 changed files, 3 additions, 3 deletions
- Clearly described commit msg

`commit d02d80bb6b7ba091eba00fdffb304adb7b49344b`
- 2 changed files, 1 additions, 4 deletions
- Clearly described commit msg

5.1.3 Comments and documentation

There is sufficient documentation describing how the program is supposed to work in general. Since the initial audit the Gitbook documentation was updated and extended, but we recommend adding more description of different fees that exist in the protocol and from what assets they are deducted. There should be a more detailed description of how validators are selected.

We have found several places in the code where comments didn't match the code as well, but it wasn't crucial for the audit process. We believe those mistakes were introduced during the refactor process.

5.1.4 Release cycle

The project contains a clear roadmap of stable releases. The client has started using the Github project where all the necessary information about the state of the current release cycle can be found. The client has already made the first stable release.

5.1.5 Code Maturity

The code maturity got improved between two audit revisions. The client delivered a commit for the second revision of the audit on time and also added better documentation and extensive test coverage. The code was functional with all tests successfully passing.

5.2 Issues

Using our toolset, manual code review and unit testing we've identified the following issues.

Low

Low severity issues are more comments and recommendations rather than security issues. We provide hints on how to improve code readability and follow best practices. Further actions depend on the development team decision.

ID	Description	Program	Line	Status
L1	Typos in several places in code	settings/src/state.rs, settings/src/lib.rs		Fixed
L2	Bad naming conventions			Fixed
L3	NewTypes or type aliases for primitive types			Fixed
L4	Missing or Unused code	settings/src/event.rs	91	Fixed
L5	Unused accounts	debridge/src/lib.rs settings/src/lib.rs	46, 94, 119, 121, 625	Fixed
L6	Unconstrained `authority`	debridge/src/lib.rs	118	Fixed

L7	Using of `ProgramAccount` struct	debridge/src/lib.rs settings/src/lib.rs		Fixed
L8	Add extra optimizations in `Cargo.toml`			Fixed
L9	Use the latest stable Rust version (1.56)			Fixed
L10	Consider more pedantic `clippy` rules			Fixed
Update rev. 2				
L11	Mistakes in documentation	settings/src/lib.rs debridge/src/lib.rs	227, 285, 388, 575, 1643	New
L12	Ununified approach to system accounts constraint			New
L13	Wrongly used range literal	signature-verifier/src/lib.rs	43	New

L1: Fix typo for all occurrences

- `subsmission → submission`
- `transfered → transferred`
- `old_reserbed_bps → old_min_reserved_bps`

L2: Use better naming

- `is_work_now → is_working`
- `is_exists → exists`

L3: Use NewTypes or type aliases for primitive types to increase readability and reliability.

```
pub old_reserbed_bps: u64,
pub bridge_id: [u8; 32],
pub collected_fee: u64,
```

vs.

```
pub old_reserbed_bps: BasisPoints,
pub bridge_id: BridgeId,
pub collected_fee: Lamports,
```

L4: Missing Instruction should be added or unused code should be removed.

- Unused event `BridgeFeeInfoUpdated` or there's a missing instruction `update_fee_bridge_info`.

L5: Unused accounts unnecessarily increase the size of a transaction and should be removed.

`BridgeCtx::spl_token_program`
- Just set it in the `AmountContextBuilder` structure and then never use it.
- It also lacks a check to see if it is really a token program

`BridgeCtx::system_program`.
- Also missing a check if it is really a system program (but account not used so it is not critical).

L6: The role of this `authority` is not clear, except that it signs the transaction, it is not used anywhere else. Is it necessary? If so, better naming might be appropriate.

L7: Use `Account` instead of `ProgramAccount` as advised by the Anchor framework author:

L8: Add more optimizations to `Cargo.toml`

```
[profile.release]
lto = true
opt-level = 's'   // or 'z' or 3
```

L10: Consider using a command line

```
cargo clippy --workspace -- -W clippy::pedantic -W clippy::nursery -W
clippy::cargo
```

Update rev. 2:

L11 (Revision 2): Fix mistakes in the documentation.

- `excess` → `minimum`
- `fee_beneficiar` → `stop_tap`

- not used `depth` parameter
- `staking_wallet` doesn't belong to `fee_beneficiar`
- not used `submission_bump`

L12 (Revision 2): Unify system program constraints.

```
#[account(address = system_program::ID)]
system_program: AccountInfo<'info>
```

vs.

```
pub system_program: Program<'info, System>
```

L13 (Revision 2): Lower number should be on the left side of the range operator. Alternatively refactor not to use range operator with map if the `depth` param will always be 1.

Medium

Medium severity issues aren't security vulnerabilities, but should be clearly clarified or fixed.

ID	Description	Contract	Line	Status
M1	Use `create_program_address` instead of `find_program_address`			Partially Fixed
M2	Using API calls instead of SysVar			Fixed
M3	Extra SEED during checking, constraint fails even the right account is used	debridge/src/lib.rs	108	Fixed
M4	Comparing bad PubKeys	debridge/src/lib.rs	113	Fixed
M5	Badly calculated rent exempt for one day	settings/src/lib.rs	353 - 355	Fixed
M6	`BridgeCtx::staking_wallet` bad constraint	debridge/src/lib.rs	74	Fixed

M1: Call `create_program_address` with `BUMP` instead of `find_program_address` on-chain. There's a risk of exceeding a compute budget. We recommend computing the `BUMP` off-chain.

```
      #[account(
          constraint = Pubkey::find_program_address(
              &[State::SEED],
              &settings::ID
          ).eq(&(state.key(), state.state_bump)),
          has_one = fee_beneficiar,
      )]
      pub state: Account<'info, State>,
...
```

vs.

```
        #[account(
             constraint = Pubkey::create_program_address(
                 &[State::SEED, &[state.state_bump]],
                 &settings::ID
             ).eq(&Ok(state.key())),
             has_one = fee_beneficiar,
        )]
        pub state: Account<'info, State>,
...
```

M2: As of `solana-program` version 1.6.5, sysvars can also be accessed without being passed into an entrypoint as an account. It improves security (no need for additional checks) and performance (smaller transactions).

```
self.state.state.check_confirmation_adequacy(
            &Clock::get()?, // instead of `&self.clock`
            self.signature_storage.verify_and_iter(
                signature_storage_key,
                &settings::ID,
                &submission_id.to_bytes(),
            )?,
        )?;
```

M3: `BridgeFeeInfo::SEED` missing during a creation of `bridge_fee` account at `settings/src/lib.rs [loc: 696]`. Use `BridgeFeeInfo::SEED` in the initialization of the `bridge_fee` account.

M4: Bad PubKeys used for a constraint check. Use `bridge_fee.key()` instead of `chain_support_info.key()`.

M5: The comment in the code says that signatures should be stored for one day, but the code calculates an hour_rent. In addition, the calculation uses subtraction instead of division. Use a simple division for calculating `day_rent` instead of `checked_sub()`. Instead of calling two times `checked_sub()`, you should calculate rent for one day as `year_rent / 365`.

```
/// Temporary storage for signatures
/// Stores signatures for one day. Access is obtained by a signed message bytes
/// [`OraclesKeys`] inside account data
```

M6: For Mint Bridge it belongs to a `fee_beneficiar`. For Send Bridge it is owned by `` `Self::mint_authority` ``. But in `BridgeCtx`, it is checked whether an owner is a `bridge`. In our opinion, the validation of the owner should depend on the type of `bridge`.

Update rev. 2:

M1 (Revision 2): There are still several places where `create_program_address` with `BUMP` can be used instead of `find_program_address` on-chain:

- `setting/src/lib.rs [loc: 1039]`
- `setting/src/bridge.rs [loc: 459]`

High

High severity issues are security vulnerabilities, which require specific steps and conditions to be exploited, or bugs that make a system unusable or unreliable. These issues have to be fixed.

ID	Description	Program	Line	Status
H1	`InitSendBridge` computational Budget	settings/src/lib.rs	1304	Fixed
H2	Custom program errors in `settings` program.	settings/src/lib.rs	701, 702, 775, 899,	Fixed
H3	The program violates the stack size at runtime, an `AccessViolation error`	debridge/src/lib.rs	145	Fixed
Update rev. 2				
H4	Protocol doesn't collected native fix fee	debridge/src/state.rs	177	New

H1: The `initialize_send_bridge` instruction consumed all computational units. The problem occurred during CPI from the `init_token_staking_wallet()` function. Can be fixed by creating a `staking_wallet` off-chain. You can find a test instruction in Appendix A.

```
"Program BkE5dgkHHhQZZCEjqDsRSLrkfZGYbeSWAjBzRSd2vyzh consumed 200000
of 200000 compute units"
```

H2: These errors were caused by allocating insufficient space for accounts being created. Fixed by increasing the allocated space (`mem::size_of`). Also, one of these errors was caused by a use of a bad constraint. You can find test instruction in Appendix B.

```
constraint = bridge_fee.to_account_info().is_not_exists()
```

H3: This error was caused by calling `send()` instruction. You can find test instruction in Appendix C.

Update Rev. 2:

H4 (Revision 2): The fix fee is not transferred to the `fee_beneficiar` wallet if fix fees are charged in a native asset. We believe it should be accumulated to `execution_fee` or it should happen in the `take_native_fix_fee` function .

Critical

Direct critical security threats, which could be instantly misused to attack the system. These issues have to be fixed.

✓ We haven't found any critical severity issues.

5.3 Testing & Verification

The project implements two types of tests: Unit Tests, Integration Tests.

- Unit Tests
    - The project improved unit tests coverage.
- Integration Tests
    - The project improved integration test coverage.
    - However they mainly cover happy path scenarios.

General code coverage got improved. Both Unit tests and Integration tests got extended. But before a production release we recommend adding tests for non-happy path scenarios.

6. Conclusion

The code is mature in quality, with sufficient test coverage and supporting documentation.

The client correctly fixed all issues discovered in Revision 1 except issue M1, which was addressed correctly but not all of its occurrences.
Revision 2 of the audit discovered three low, one medium and one high issues.

Appendix A - InitSentBridge instruction

```
Instruction {
      program_id: BkE5dgkHHhQZZCEjqDsRSLrkfZGYbeSWAjBzRSd2vyzh,
      accounts: [
        { pubkey: DwGXPVMDyqKn8mrdwZmRPExHn7nfMUk1N9vKx99UQBaW, is_signer: false, is_writable: true },
        { pubkey: 7YxsoxwJN4fWVKLXGCPtKcbZEvYxhnCusV1tMVPnrSsk, is_signer: false, is_writable: true },
        { pubkey: 3j3LuxfD7Mcfb3EvS3YZxBataL4kXFBxwdKrWtj3T8K8, is_signer: false, is_writable: true },
        { pubkey: 87so1rooD2gzzyivC824wvJmhDtyk6bkaDeuU4Uq8Qra, is_signer: false, is_writable: false },
        { pubkey: GRCHaiCmMxS7eYj2E7Qbit5pfUv3rAnZirvYp26QvdbN, is_signer: false, is_writable: false },
        { pubkey: 11111111111111111111111111111111, is_signer: false, is_writable: false },
        { pubkey: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA, is_signer: false, is_writable: false },
        { pubkey: 3z2acQDyVn5pUSdmVR4VAq9ZTejdxVwXCmX6fKcNX539, is_signer: true, is_writable: false },
        { pubkey: SysvarRent111111111111111111111111111111111, is_signer: false, is_writable: false },
        { pubkey: ATokenGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL, is_signer: false, is_writable: false }],
      data: [185, 145, 74, 79, 45, 146, 231, 31, 254, 252]
}
```

Appendix B1 - InitFeeBridgeInfo instruction

```
Instruction {
      program_id: BkE5dgkHHhQZZCEjqDsRSLrkfZGYbeSWAjBzRSd2vyzh,
      accounts: [
        { pubkey: AFm6TsQRcDisR5hyFnb6yRZ1B9gzqhahu5HibmG1RAGj, is_signer: false, is_writable: false },
        { pubkey: GxXs76JYYjdby499AU3Gdy9ftqcyGEPGVJz1tQsppJYJ, is_signer: false, is_writable: false },
        { pubkey: 7YxsoxwJN4fWVKLXGCPtKcbZEvYxhnCusV1tMVPnrSsk, is_signer: false, is_writable: true },
        { pubkey: 2Rp8GkdM7sgzMAK47SNpVtTRrzKMEiBrFh8vZEHYK9XA, is_signer: false, is_writable: true },
        { pubkey: DuFfamGLQEvPeToog1aFUfBohiRVTrGysaPiEkmEVE5s, is_signer: true, is_writable: false },
        { pubkey: 11111111111111111111111111111111, is_signer: false, is_writable: false },
        { pubkey: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA, is_signer: false, is_writable: false }],
      data: [4, 238, 238, 167, 87, 40, 237, 132, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 101, 255, 10, 0, 0, 0, 0, 0, 0, 0] }
```

Appendix B2 - InitChainSupportInfo instruction

```
Instruction {
      program_id: BkE5dgkHHhQZZCEjqDsRSLrkfZGYbeSWAjBzRSd2vyzh,
      accounts: [
        { pubkey: 7YxsoxwJN4fWVKLXGCPtKcbZEvYxhnCusV1tMVPnrSsk, is_signer: false, is_writable: false },
        { pubkey: DuFfamGLQEvPeToog1aFUfBohiRVTrGysaPiEkmEVE5s, is_signer: true, is_writable: false },
        { pubkey: DuFfamGLQEvPeToog1aFUfBohiRVTrGysaPiEkmEVE5s, is_signer: true, is_writable: false },
        { pubkey: J96yUe2LFmw4xYmTBeud8zazxWLYCznCdgayvqzs13Ey, is_signer: false, is_writable: true },
        { pubkey: 11111111111111111111111111111111, is_signer: false, is_writable: false }],
      data: [146, 14, 216, 153, 206, 170, 83, 177, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 101, 255, 1, 100, 0, 0, 0, 0, 0, 0, 0, 232, 3, 0, 0, 0, 0, 0, 0]
}
```

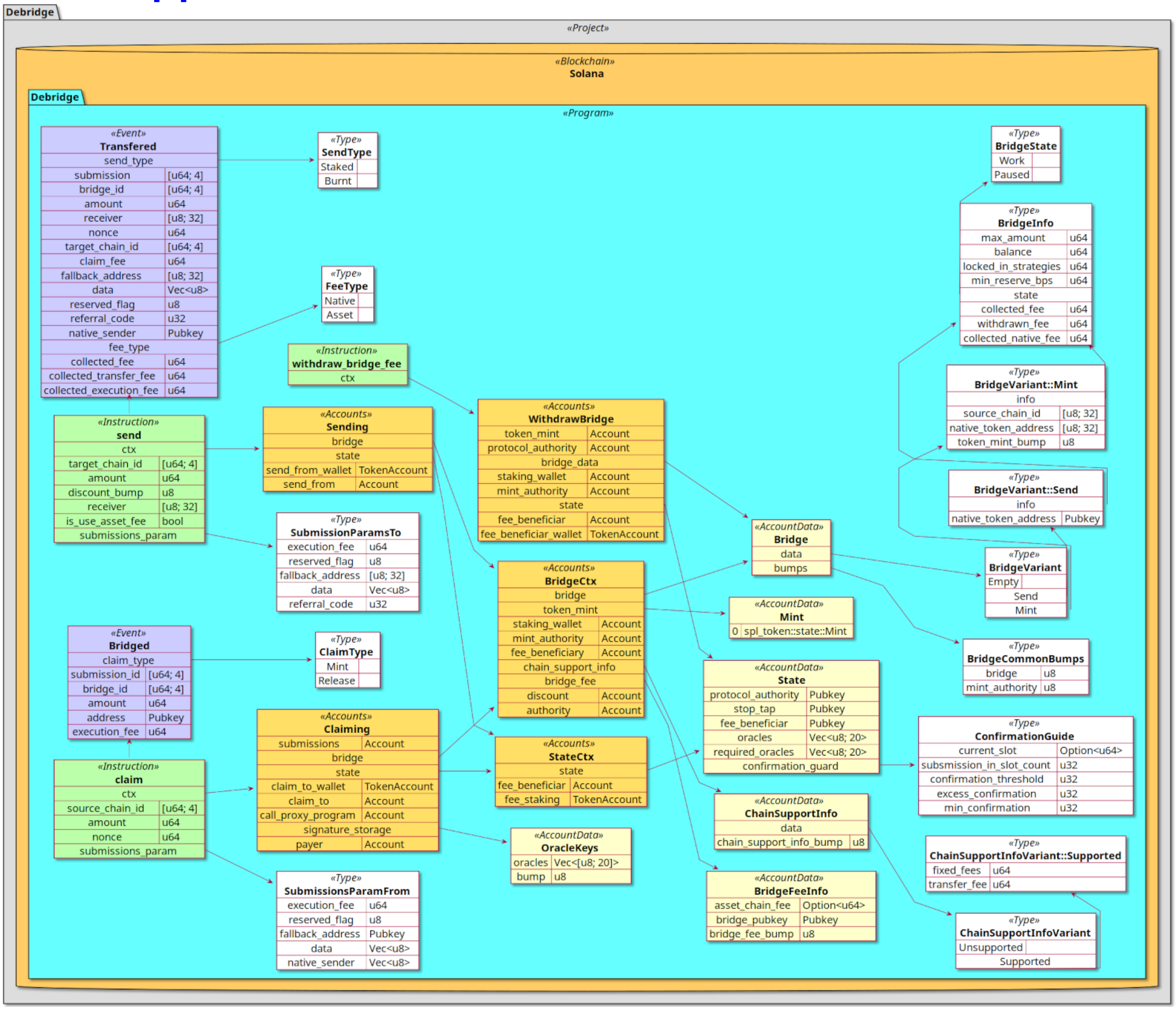
Appendix B3 - InitDiscountInfo instruction

```
Instruction {
      program_id: BkE5dgkHHhQZZCEjqDsRSLrkfZGYbeSWAjBzRSd2vyzh,
      accounts: [
        { pubkey: 7YxsoxwJN4fWVKLXGCPtKcbZEvYxhnCusV1tMVPnrSsk, is_signer: false, is_writable: false },
        { pubkey: 5gL3EH72siRwFjpRbwBZbgcAh9mYWajRdnfViTCTgugv, is_signer: false, is_writable: true },
        { pubkey: 3z2acQDyVn5pUSdmVR4VAq9ZTejdxVwXCmX6fKcNX539, is_signer: false, is_writable: false },
        { pubkey: 11111111111111111111111111111111, is_signer: false, is_writable: false },
        { pubkey: DuFfamGLQEvPeToog1aFUfBohiRVTrGysaPiEkmEVE5s, is_signer: true, is_writable: false }],
      data: [133, 68, 55, 173, 101, 223, 74, 14, 255, 2, 0, 2, 0]
}
```

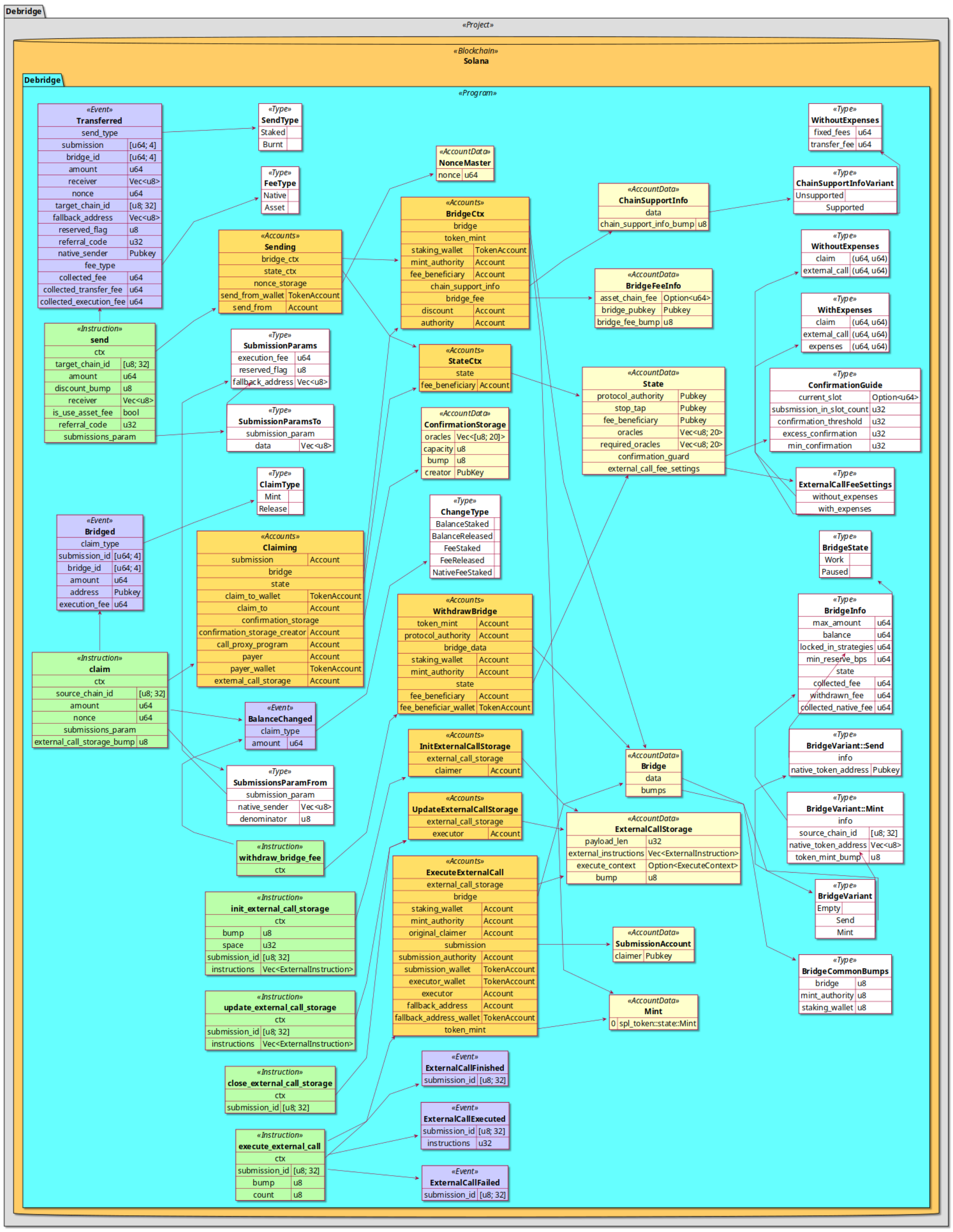
Appendix C - Send instruction

```
Instruction {
      program_id: 2dhGsWUzy5YKUsjZdLHLmkNpUDAXkNa9MYWsPc4Ziqzy,
      accounts: [
        { pubkey: D3yHckgBcZC65k8sSPc4TnfrwxByn3NizGavFkXaTmwE, is_signer: false, is_writable: true },
        { pubkey: CUq3ddZMRtzULcBKFELADWDxWBkvM3UfTTE5BaCWvzqE, is_signer: false, is_writable: false },
        { pubkey: FaKTCzVYj61A37NbX59omCZqFjr3HSUzvJop8TR1bLk9, is_signer: false, is_writable: false },
        { pubkey: FN87rkMYTnDTxUnAZMu7fQEvePbnL12ieo4FNPStS2kV, is_signer: false, is_writable: false },
        { pubkey: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA, is_signer: false, is_writable: false },
        { pubkey: J96yUe2LFmw4xYmTBeud8zazxWLYCznCdgayvqzs13Ey, is_signer: false, is_writable: false },
        { pubkey: 2wpXj1q5ELVcF9ufLCAorfvL3jAtk5AkGTWCJFWv1LvZ, is_signer: false, is_writable: false },
        { pubkey: 5gL3EH72siRwFjpRbwBZbgcAh9mYWajRdnfViTCTgugv, is_signer: false, is_writable: false },
        { pubkey: 3z2acQDyVn5pUSdmVR4VAq9ZTejdxVwXCmX6fKcNX539, is_signer: true, is_writable: false },
        { pubkey: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA, is_signer: false, is_writable: false },
        { pubkey: SysvarRent111111111111111111111111111111111, is_signer: false, is_writable: false },
        { pubkey: 11111111111111111111111111111111, is_signer: false, is_writable: false },
        { pubkey: 7YxsoxwJN4fWVKLXGCPtKcbZEvYxhnCusV1tMVPnrSsk, is_signer: false, is_writable: false },
        { pubkey: EdKBnC98VgW2EjVqeg9SYKPvJmQn2LRhzLuA4qEXCbSG, is_signer: false, is_writable: false },
        { pubkey: 4F4sVd5z4cmJqtmaVjumHJ8MtoReCQegtw2xcUZU5nFe, is_signer: false, is_writable: true },
        { pubkey: 3z2acQDyVn5pUSdmVR4VAq9ZTejdxVwXCmX6fKcNX539, is_signer: true, is_writable: false },
        { pubkey: 11111111111111111111111111111111, is_signer: false, is_writable: false }
      ],
      data: [102, 251, 20, 187, 65, 75, 12, 69, 101, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0]
}
```

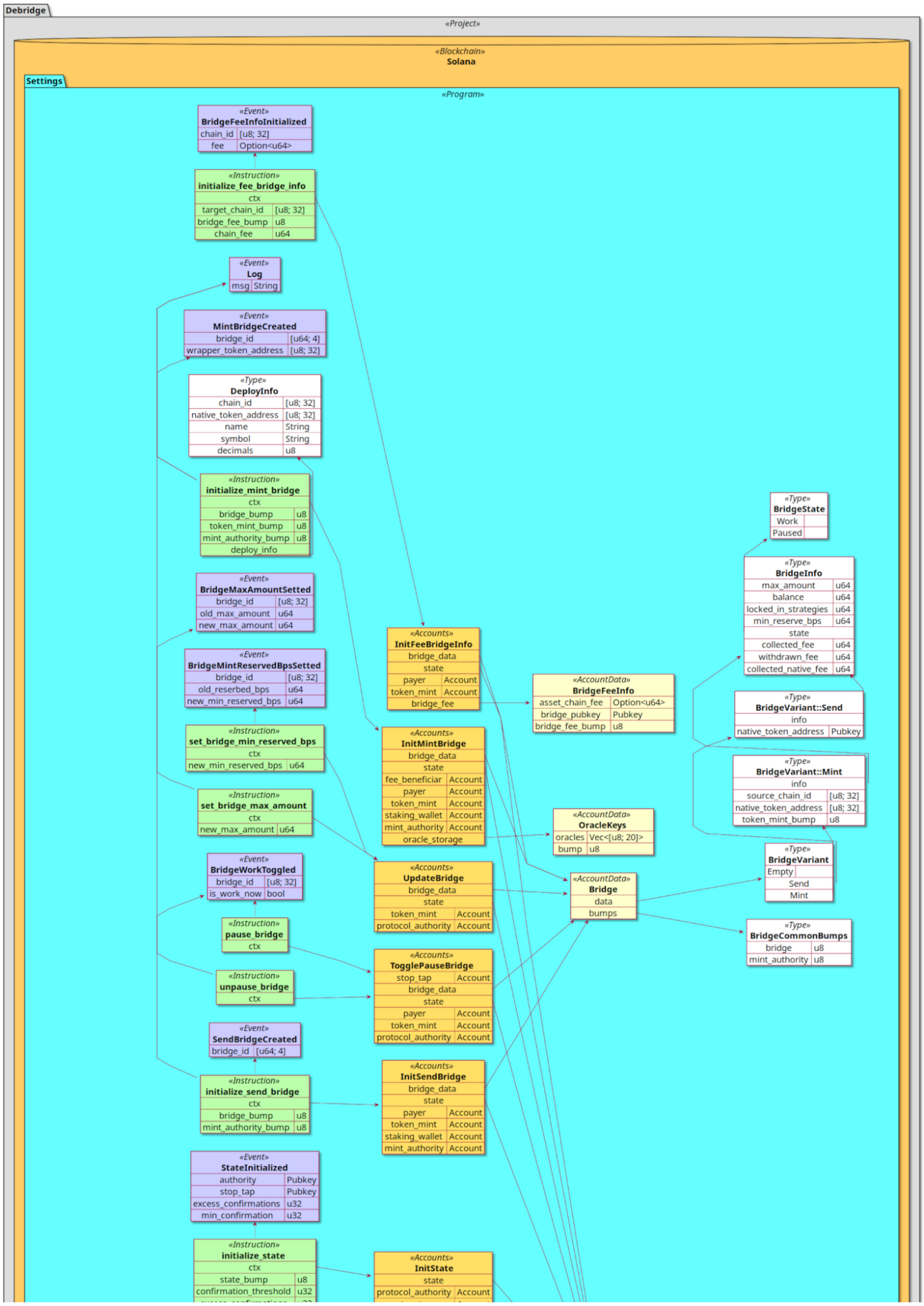
Appendix D


Debridge
«Project»
«Blockchain»
Solana
Debridge
«Program»
«Event» Transfered
send_type
submission [u64; 4]
bridge_id [u64; 4]
amount u64
receiver [u8; 32]
nonce u64
target_chain_id [u64; 4]
claim_fee u64
fallback_address [u8; 32]
data Vec<u8>
reserved_flag u8
referral_code u32
native_sender Pubkey
fee_type
collected_fee u64
collected_transfer_fee u64
collected_execution_fee u64
«Type» SendType
Staked
Burnt
«Type» FeeType
Native
Asset
«Instruction» withdraw_bridge_fee
ctx
«Instruction» send
ctx
target_chain_id [u64; 4]
amount u64
discount_bump u8
receiver [u8; 32]
is_use_asset_fee bool
submissions_param
«Accounts» Sending
bridge
state
send_from_wallet TokenAccount
send_from Account
«Type» SubmissionParamsTo
execution_fee u64
reserved_flag u8
fallback_address [u8; 32]
data Vec<u8>
referral_code u32
«Event» Bridged
claim_type
submission_id [u64; 4]
bridge_id [u64; 4]
amount u64
address Pubkey
execution_fee u64
«Type» ClaimType
Mint
Release
«Instruction» claim
ctx
source_chain_id [u64; 4]
amount u64
nonce u64
submissions_param
«Accounts» Claiming
submissions Account
bridge
state
claim_to_wallet TokenAccount
claim_to Account
call_proxy_program Account
signature_storage
payer Account
«Type» SubmissionParamFrom
execution_fee u64
reserved_flag u8
fallback_address Pubkey
data Vec<u8>
native_sender Vec<u8>
«Accounts» WithdrawBridge
token_mint Account
protocol_authority Account
bridge_data
staking_wallet Account
mint_authority Account
state
fee_beneficiar Account
fee_beneficiar_wallet TokenAccount
«Accounts» BridgeCtx
bridge
token_mint
staking_wallet Account
mint_authority Account
fee_beneficiary Account
chain_support_info
bridge_fee
discount Account
authority Account
«Accounts» StateCtx
state
fee_beneficiar Account
fee_staking TokenAccount
«AccountData» OracleKeys
oracles Vec<[u8; 20]>
bump u8
«AccountData» Bridge
data
bumps
«AccountData» Mint
0 spl_token::state::Mint
«AccountData» State
protocol_authority Pubkey
stop_tap Pubkey
fee_beneficiar Pubkey
oracles Vec<u8; 20>
required_oracles Vec<u8; 20>
confirmation_guard
«AccountData» ChainSupportInfo
data
chain_support_info_bump u8
«AccountData» BridgeFeeInfo
asset_chain_fee Option<u64>
bridge_pubkey Pubkey
bridge_fee_bump u8
«Type» BridgeState
Work
Paused
«Type» BridgeInfo
max_amount u64
balance u64
locked_in_strategies u64
min_reserve_bps u64
state
collected_fee u64
withdrawn_fee u64
collected_native_fee u64
«Type» BridgeVariant::Mint
info
source_chain_id [u8; 32]
native_token_address [u8; 32]
token_mint_bump u8
«Type» BridgeVariant::Send
info
native_token_address Pubkey
«Type» BridgeVariant
Empty
Send
Mint
«Type» BridgeCommonBumps
bridge u8
mint_authority u8
«Type» ConfirmationGuide
current_slot Option<u64>
submission_in_slot_count u32
confirmation_threshold u32
excess_confirmation u32
min_confirmation u32
«Type» ChainSupportInfoVariant::Supported
fixed_fees u64
transfer_fee u64
«Type» ChainSupportInfoVariant
Unsupported
Supported

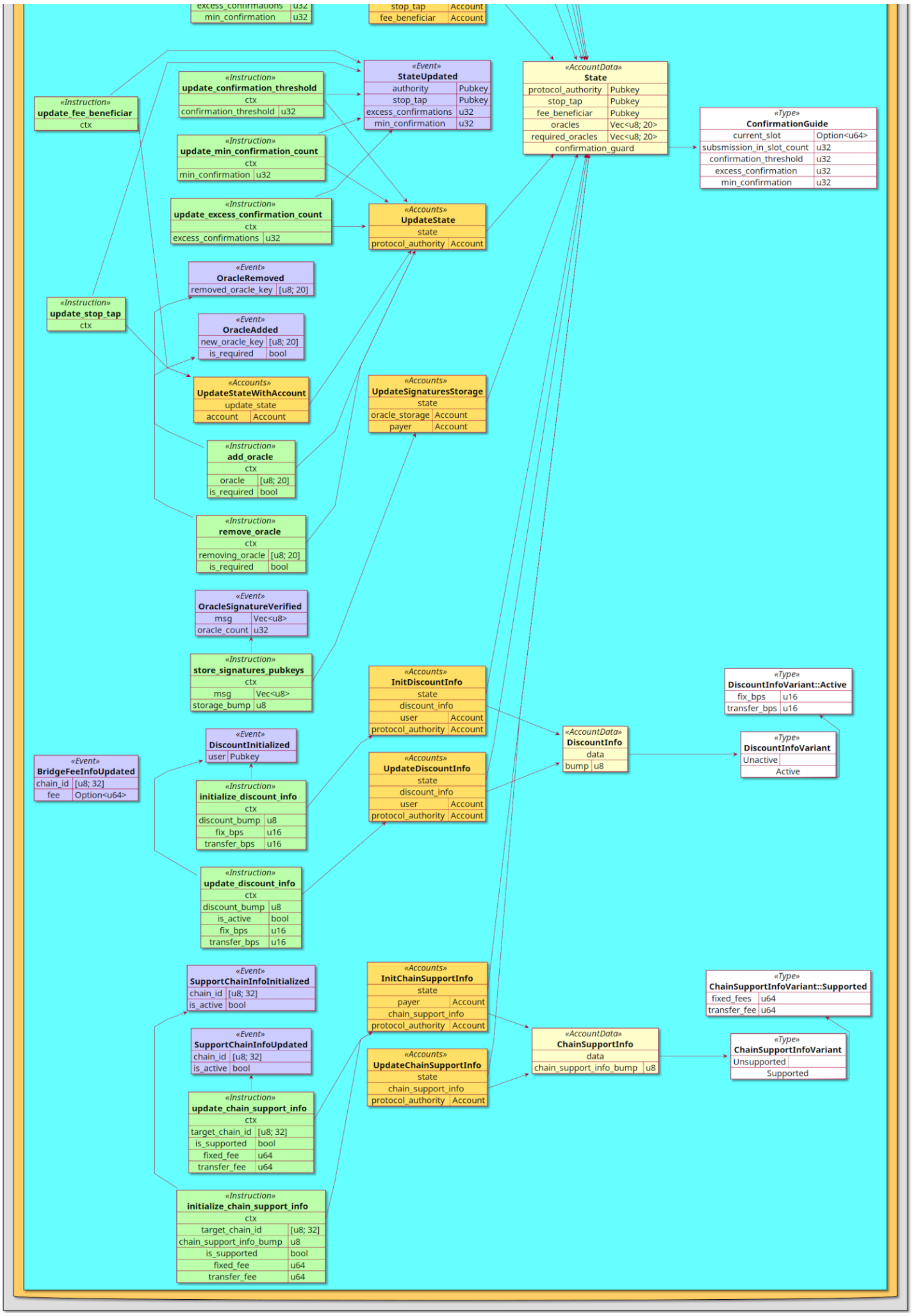
Update rev. 2


Debridge
«Project»
«Blockchain» Solana
Debridge
«Program»
«Event» Transferred
send_type
submission [u64; 4]
bridge_id [u64; 4]
amount u64
receiver Vec<u8>
nonce u64
target_chain_id [u8; 32]
fallback_address Vec<u8>
reserved_flag u8
referral_code u32
native_sender Pubkey
fee_type
collected_fee u64
collected_transfer_fee u64
collected_execution_fee u64
«Type» SendType
Staked
Burnt
«Type» FeeType
Native
Asset
«AccountData» NonceMaster
nonce u64
«Accounts» BridgeCtx
bridge
token_mint
staking_wallet TokenAccount
mint_authority Account
fee_beneficiary Account
chain_support_info
bridge_fee
discount Account
authority Account
«AccountData» ChainSupportInfo
data
chain_support_info_bump u8
«Type» WithoutExpenses
fixed_fees u64
transfer_fee u64
«Type» ChainSupportInfoVariant
Unsupported
Supported
«Type» WithoutExpenses
claim (u64, u64)
external_call (u64, u64)
«Type» WithExpenses
claim (u64, u64)
external_call (u64, u64)
expenses (u64, u64)
«Accounts» Sending
bridge_ctx
state_ctx
nonce_storage
send_from_wallet TokenAccount
send_from Account
«AccountData» BridgeFeeInfo
asset_chain_fee Option<u64>
bridge_pubkey Pubkey
bridge_fee_bump u8
«Instruction» send
ctx
target_chain_id [u8; 32]
amount u64
discount_bump u8
receiver Vec<u8>
is_use_asset_fee bool
referral_code u32
submissions_param
«Type» SubmissionParams
execution_fee u64
reserved_flag u8
fallback_address Vec<u8>
«Accounts» StateCtx
state
fee_beneficiary Account
«AccountData» State
protocol_authority Pubkey
stop_tap Pubkey
fee_beneficiary Pubkey
oracles Vec<u8; 20>
required_oracles Vec<u8; 20>
confirmation_guard
external_call_fee_settings
«Type» ConfirmationGuide
current_slot Option<u64>
submission_in_slot_count u32
confirmation_threshold u32
excess_confirmation u32
min_confirmation u32
«Type» SubmissionParamsTo
submission_param
data Vec<u8>
«AccountData» ConfirmationStorage
oracles Vec<[u8; 20]>
capacity u8
bump u8
creator PubKey
«Type» ExternalCallFeeSettings
without_expenses
with_expenses
«Type» ClaimType
Mint
Release
«Type» ChangeType
BalanceStaked
BalanceReleased
FeeStaked
FeeReleased
NativeFeeStaked
«Event» Bridged
claim_type
submission_id [u64; 4]
bridge_id [u64; 4]
amount u64
address Pubkey
execution_fee u64
«Accounts» Claiming
submission Account
bridge
state
claim_to_wallet TokenAccount
claim_to Account
confirmation_storage
confirmation_storage_creator Account
call_proxy_program Account
payer Account
payer_wallet TokenAccount
external_call_storage Account
«Type» BridgeState
Work
Paused
«Accounts» WithdrawBridge
token_mint Account
protocol_authority Account
bridge_data
staking_wallet Account
mint_authority Account
state
fee_beneficiary Account
fee_beneficiar_wallet TokenAccount
«Type» BridgeInfo
max_amount u64
balance u64
locked_in_strategies u64
min_reserve_bps u64
state
collected_fee u64
withdrawn_fee u64
collected_native_fee u64
«Instruction» claim
ctx
source_chain_id [u8; 32]
amount u64
nonce u64
submissions_param
external_call_storage_bump u8
«Event» BalanceChanged
claim_type
amount u64
«Accounts» InitExternalCallStorage
external_call_storage
claimer Account
«AccountData» Bridge
data
bumps
«Type» BridgeVariant::Send
info
native_token_address Pubkey
«Type» SubmissionsParamFrom
submission_param
native_sender Vec<u8>
denominator u8
«Accounts» UpdateExternalCallStorage
external_call_storage
executor Account
«AccountData» ExternalCallStorage
payload_len u32
external_instructions Vec<ExternalInstruction>
execute_context Option<ExecuteContext>
bump u8
«Type» BridgeVariant::Mint
info
source_chain_id [u8; 32]
native_token_address Vec<u8>
token_mint_bump u8
«Instruction» withdraw_bridge_fee
ctx
«Accounts» ExecuteExternalCall
external_call_storage
bridge
staking_wallet Account
mint_authority Account
original_claimer Account
submission
submission_authority Account
submission_wallet TokenAccount
executor_wallet TokenAccount
executor Account
fallback_address Account
fallback_address_wallet TokenAccount
token_mint
«Type» BridgeVariant
Empty
Send
Mint
«Instruction» init_external_call_storage
ctx
bump u8
space u32
submission_id [u8; 32]
instructions Vec<ExternalInstruction>
«AccountData» SubmissionAccount
claimer Pubkey
«Type» BridgeCommonBumps
bridge u8
mint_authority u8
staking_wallet u8
«Instruction» update_external_call_storage
ctx
submission_id [u8; 32]
instructions Vec<ExternalInstruction>
«AccountData» Mint
0 spl_token::state::Mint
«Instruction» close_external_call_storage
ctx
submission_id [u8; 32]
«Event» ExternalCallFinished
submission_id [u8; 32]
«Event» ExternalCallExecuted
submission_id [u8; 32]
instructions u32
«Instruction» execute_external_call
ctx
submission_id [u8; 32]
bump u8
count u8
«Event» ExternalCallFailed
submission_id [u8; 32]

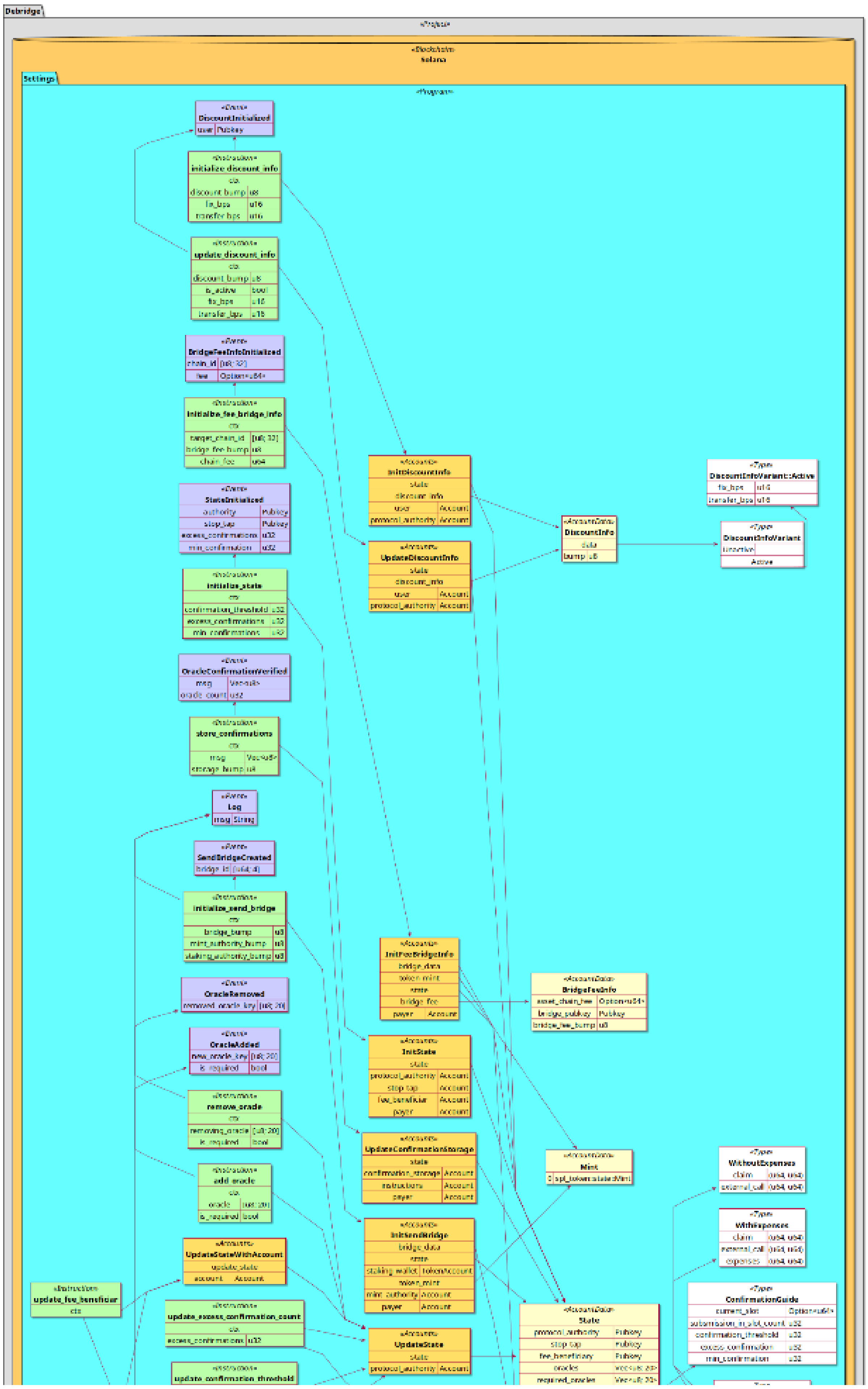
Appendix E1


Debridge
«Project»
«Blockchain»
Solana
Settings
«Program»
«Event»
BridgeFeeInfoInitialized
chain_id [u8; 32]
fee Option<u64>
«Instruction»
initialize_fee_bridge_info
ctx
target_chain_id [u8; 32]
bridge_fee_bump u8
chain_fee u64
«Event»
Log
msg String
«Event»
MintBridgeCreated
bridge_id [u64; 4]
wrapper_token_address [u8; 32]
«Type»
DeployInfo
chain_id [u8; 32]
native_token_address [u8; 32]
name String
symbol String
decimals u8
«Instruction»
initialize_mint_bridge
ctx
bridge_bump u8
token_mint_bump u8
mint_authority_bump u8
deploy_info
«Event»
BridgeMaxAmountSetted
bridge_id [u8; 32]
old_max_amount u64
new_max_amount u64
«Event»
BridgeMintReservedBpsSetted
bridge_id [u8; 32]
old_reserbed_bps u64
new_min_reserved_bps u64
«Instruction»
set_bridge_min_reserved_bps
ctx
new_min_reserved_bps u64
«Instruction»
set_bridge_max_amount
ctx
new_max_amount u64
«Event»
BridgeWorkToggled
bridge_id [u8; 32]
is_work_now bool
«Instruction»
pause_bridge
ctx
«Instruction»
unpause_bridge
ctx
«Event»
SendBridgeCreated
bridge_id [u64; 4]
«Instruction»
initialize_send_bridge
ctx
bridge_bump u8
mint_authority_bump u8
«Event»
StateInitialized
authority Pubkey
stop_tap Pubkey
excess_confirmations u32
min_confirmation u32
«Instruction»
initialize_state
ctx
state_bump u8
confirmation_threshold u32
«Accounts»
InitFeeBridgeInfo
bridge_data
state
payer Account
token_mint Account
bridge_fee
«Accounts»
InitMintBridge
bridge_data
state
fee_beneficiar Account
payer Account
token_mint Account
staking_wallet Account
mint_authority Account
oracle_storage
«Accounts»
UpdateBridge
bridge_data
state
token_mint Account
protocol_authority Account
«Accounts»
TogglePauseBridge
stop_tap Account
bridge_data
state
payer Account
token_mint Account
protocol_authority Account
«Accounts»
InitSendBridge
bridge_data
state
payer Account
token_mint Account
staking_wallet Account
mint_authority Account
«Accounts»
InitState
state
protocol_authority Account
«AccountData»
BridgeFeeInfo
asset_chain_fee Option<u64>
bridge_pubkey Pubkey
bridge_fee_bump u8
«AccountData»
OracleKeys
oracles Vec<[u8; 20]>
bump u8
«AccountData»
Bridge
data
bumps
«Type»
BridgeState
Work
Paused
«Type»
BridgeInfo
max_amount u64
balance u64
locked_in_strategies u64
min_reserve_bps u64
state
collected_fee u64
withdrawn_fee u64
collected_native_fee u64
«Type»
BridgeVariant::Send
info
native_token_address Pubkey
«Type»
BridgeVariant::Mint
info
source_chain_id [u8; 32]
native_token_address [u8; 32]
token_mint_bump u8
«Type»
BridgeVariant
Empty
Send
Mint
«Type»
BridgeCommonBumps
bridge u8
mint_authority u8

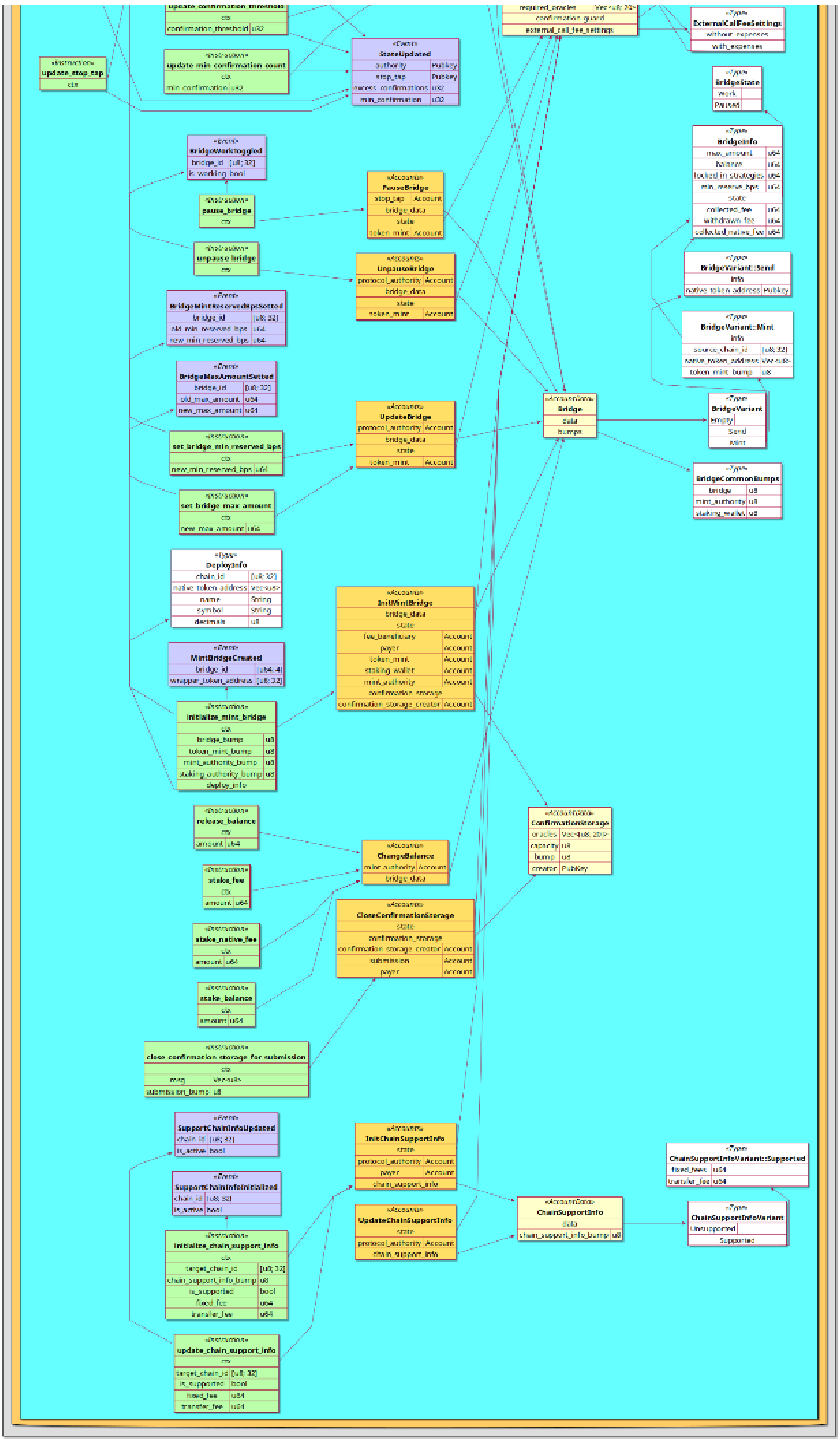
Appendix E2


excess_confirmations u32
min_confirmation u32
stop_tap Account
fee_beneficiar Account
«Instruction» update_fee_beneficiar ctx
«Instruction» update_confirmation_threshold ctx confirmation_threshold u32
«Event» StateUpdated authority Pubkey stop_tap Pubkey excess_confirmations u32 min_confirmation u32
«AccountData» State protocol_authority Pubkey stop_tap Pubkey fee_beneficiar Pubkey oracles Vec<u8; 20> required_oracles Vec<u8; 20> confirmation_guard
«Type» ConfirmationGuide current_slot Option<u64> subsmission_in_slot_count u32 confirmation_threshold u32 excess_confirmation u32 min_confirmation u32
«Instruction» update_min_confirmation_count ctx min_confirmation u32
«Instruction» update_excess_confirmation_count ctx excess_confirmations u32
«Accounts» UpdateState state protocol_authority Account
«Event» OracleRemoved removed_oracle_key [u8; 20]
«Instruction» update_stop_tap ctx
«Event» OracleAdded new_oracle_key [u8; 20] is_required bool
«Accounts» UpdateStateWithAccount update_state account Account
«Accounts» UpdateSignaturesStorage state oracle_storage Account payer Account
«Instruction» add_oracle ctx oracle [u8; 20] is_required bool
«Instruction» remove_oracle ctx removing_oracle [u8; 20] is_required bool
«Event» OracleSignatureVerified msg Vec<u8> oracle_count u32
«Instruction» store_signatures_pubkeys ctx msg Vec<u8> storage_bump u8
«Accounts» InitDiscountInfo state discount_info user Account protocol_authority Account
«Type» DiscountInfoVariant::Active fix_bps u16 transfer_bps u16
«AccountData» DiscountInfo data bump u8
«Type» DiscountInfoVariant Unactive Active
«Event» DiscountInitialized user Pubkey
«Event» BridgeFeeInfoUpdated chain_id [u8; 32] fee Option<u64>
«Instruction» initialize_discount_info ctx discount_bump u8 fix_bps u16 transfer_bps u16
«Accounts» UpdateDiscountInfo state discount_info user Account protocol_authority Account
«Instruction» update_discount_info ctx discount_bump u8 is_active bool fix_bps u16 transfer_bps u16
«Event» SupportChainInfoInitialized chain_id [u8; 32] is_active bool
«Accounts» InitChainSupportInfo state payer Account chain_support_info protocol_authority Account
«Type» ChainSupportInfoVariant::Supported fixed_fees u64 transfer_fee u64
«Event» SupportChainInfoUpdated chain_id [u8; 32] is_active bool
«AccountData» ChainSupportInfo data chain_support_info_bump u8
«Type» ChainSupportInfoVariant Unsupported Supported
«Accounts» UpdateChainSupportInfo state chain_support_info protocol_authority Account
«Instruction» update_chain_support_info ctx target_chain_id [u8; 32] is_supported bool fixed_fee u64 transfer_fee u64
«Instruction» initialize_chain_support_info ctx target_chain_id [u8; 32] chain_support_info_bump u8 is_supported bool fixed_fee u64 transfer_fee u64

Update rev. 2



Debridge
«Project»
«Blockchain»
solana
Settings
«Program»
«Event» DiscountInitialized
user | Pubkey
«Instruction» initialize_discount_info
ctx
discount_bump | u8
fix_bps | u16
transfer_bps | u16
«Instruction» update_discount_info
ctx
discount_bump | u8
is_active | bool
fix_bps | u16
transfer_bps | u16
«Event» BridgeFeeInfoInitialized
chain_id | [u8; 32]
fee | Option<u64>
«Instruction» initialize_fee_bridge_info
ctx
target_chain_id | [u8; 32]
bridge_fee_bump | u8
chain_fee | u64
«Event» StateInitialized
authority | Pubkey
stop_tap | Pubkey
excess_confirmations | u32
min_confirmation | u32
«Instruction» initialize_state
ctx
confirmation_threshold | u32
excess_confirmations | u32
min_confirmations | u32
«Event» OracleConfirmationVerified
msg | Vec<u8>
oracle_count | u32
«Instruction» store_confirmations
ctx
msg | Vec<u8>
storage_bump | u8
«Event» Log
msg | String
«Event» SendBridgeCreated
bridge_id | [u64; 4]
«Instruction» initialize_send_bridge
ctx
bridge_bump | u8
mint_authority_bump | u8
staking_authority_bump | u8
«Event» OracleRemoved
removed_oracle_key | [u8; 20]
«Event» OracleAdded
new_oracle_key | [u8; 20]
is_required | bool
«Instruction» remove_oracle
ctx
removing_oracle | [u8; 20]
is_required | bool
«Instruction» add_oracle
ctx
oracle | [u8; 20]
is_required | bool
«Accounts» UpdateStateWithAccount
update_state
account | Account
«Instruction» update_fee_beneficiar
ctx
«Instruction» update_excess_confirmation_count
ctx
excess_confirmations | u32
«Instruction» update_confirmation_threshold
«Accounts» InitDiscountInfo
state
discount_info
user | Account
protocol_authority | Account
«Accounts» UpdateDiscountInfo
state
discount_info
user | Account
protocol_authority | Account
«AccountData» DiscountInfo
data
bump | u8
«Type» DiscountInfoVariant::Active
fix_bps | u16
transfer_bps | u16
«Type» DiscountInfoVariant
Inactive
Active
«Accounts» InitFeeBridgeInfo
bridge_data
token_mint
state
bridge_fee
payer | Account
«AccountData» BridgeFeeInfo
asset_chain_fee | Option<u64>
bridge_pubkey | Pubkey
bridge_fee_bump | u8
«Accounts» InitState
state
protocol_authority | Account
stop_tap | Account
fee_beneficiar | Account
payer | Account
«Accounts» UpdateConfirmationStorage
state
confirmation_storage | Account
instructions | Account
payer | Account
«AccountData» Mint
0 | spl_token::state::Mint
«Accounts» InitSendBridge
bridge_data
state
staking_wallet | TokenAccount
token_mint
mint_authority | Account
payer | Account
«Accounts» UpdateState
state
protocol_authority | Account
«AccountData» State
protocol_authority | Pubkey
stop_tap | Pubkey
fee_beneficiary | Pubkey
oracles | Vec<[u8; 20]>
required_oracles | Vec<[u8; 20]>
«Type» WithoutExpenses
claim | (u64, u64)
external_call | (u64, u64)
«Type» WithExpenses
claim | (u64, u64)
external_call | (u64, u64)
expenses | (u64, u64)
«Type» ConfirmationGuide
current_slot | Option<u64>
submission_in_slot_count | u32
confirmation_threshold | u32
excess_confirmation | u32
min_confirmation | u32


«Instruction» update_confirmation_threshold
confirmation_threshold u32
«Instruction» update_stop_tap
«Instruction» update_min_confirmation_count
min_confirmation u32
«Event» StateUpdated
authority Pubkey
stop_tap Pubkey
excess_confirmations u32
min_confirmation u32
required_oracles Vec<[u8; 20]>
confirmation_guard
external_call_fee_settings
«Type» ExternalCallFeeSettings
without_expenses
with_expenses
«Type» BridgeState
Work
Paused
«Type» BridgeInfo
max_amount u64
balance u64
locked_in_strategies u64
min_reserve_bps u64
state
collected_fee u64
withdrawn_fee u64
collected_native_fee u64
«Event» BridgeWorkToggled
bridge_id [u8; 32]
is_working bool
«Instruction» pause_bridge
«Accounts» PauseBridge
stop_tap Account
bridge_data
state
token_mint Account
«Instruction» unpause_bridge
«Accounts» UnpauseBridge
protocol_authority Account
bridge_data
state
token_mint Account
«Type» BridgeVariant::Send
info
native_token_address Pubkey
«Type» BridgeVariant::Mint
info
source_chain_id [u8; 32]
native_token_address Vec<u8>
token_mint_bump u8
«Event» BridgeMinReserveBpsUpdated
bridge_id [u8; 32]
old_min_reserved_bps u64
new_min_reserved_bps u64
«Event» BridgeMaxAmountSetted
bridge_id [u8; 32]
old_max_amount u64
new_max_amount u64
«AccountData» Bridge
data
bumps
«Type» BridgeVariant
empty
Send
Mint
«Accounts» UpdateBridge
protocol_authority Account
bridge_data
state
token_mint Account
«Instruction» set_bridge_min_reserved_bps
new_min_reserved_bps u64
«Instruction» set_bridge_max_amount
new_max_amount u64
«Type» BridgeCommonBumps
bridge u8
mint_authority u8
staking_wallet u8
«Type» DeployInfo
chain_id [u8; 32]
native_token_address Vec<u8>
name String
symbol String
decimals u8
«Accounts» InitMintBridge
bridge_data
state
fee_beneficiary Account
payer Account
token_mint Account
staking_wallet Account
mint_authority Account
confirmation_storage
confirmation_storage_creator Account
«Event» MintBridgeCreated
bridge_id [u64; 4]
wrapper_token_address [u8; 32]
«Instruction» initialize_mint_bridge
bridge_bump u8
token_mint_bump u8
mint_authority_bump u8
staking_authority_bump u8
deploy_info
«AccountData» ConfirmationStorage
oracles Vec<[u8; 20]>
capacity u8
bump u8
creator Pubkey
«Instruction» release_balance
amount u64
«Accounts» ChangeBalance
mint_authority Account
bridge_data
«Instruction» stake_fee
amount u64
«Accounts» CloseConfirmationStorage
state
confirmation_storage
confirmation_storage_creator Account
submission Account
payer Account
«Instruction» stake_native_fee
amount u64
«Instruction» stake_balance
amount u64
«Instruction» close_confirmation_storage_for_submission
msg Vec<u8>
submission_bump u8
«Event» SupportChainInfoUpdated
chain_id [u8; 32]
is_active bool
«Accounts» InitChainSupportInfo
state
protocol_authority Account
payer Account
chain_support_info
«Type» ChainSupportInfoVariant::Supported
fixed_fees u64
transfer_fee u64
«Event» SupportChainInfoInitialized
chain_id [u8; 32]
is_active bool
«Accounts» UpdateChainSupportInfo
state
protocol_authority Account
chain_support_info
«AccountData» ChainSupportInfo
data
chain_support_info_bump u8
«Type» ChainSupportInfoVariant
Unsupported
Supported
«Instruction» initialize_chain_support_info
target_chain_id [u8; 32]
chain_support_info_bump u8
is_supported bool
fixed_fee u64
transfer_fee u64
«Instruction» update_chain_support_info
target_chain_id [u8; 32]
is_supported bool
fixed_fee u64
transfer_fee u64

Thank You

Ackee Blockchain a.s.

Prague, Czech Republic

hello@ackeeblockchain.com

https://discord.gg/ZZNyQ23R