MVP: Main case flow conceptualization

**Summary**:

|  |
| --- |
| The user approves pending Deriveum smart contracts and initiates a new one |

**Involved parties**:

1. User: An insurer or an insured institution
2. Counterparty: the institution on the other side of the deal
3. Verifier: person/company/algorithm that will trigger the contract in case of credit event materializing
4. Resolver: person/company/algorithm that will terminate the contract in case of egregious mistakes, insufficient or incomplete payment, instructed to do so by court or competent authority

**Preconditions:**

1. All involved parties have already reviewed and agreed on the parameters of the deal
2. All documents for the CDS deal (credit default swap agreement, triggering conditions, termination conditions, etc.) are already agreed upon and duly signed by the parties

An additional pre-condition in a real-life use case is …

1. All parties have been duly identified and have the legal capacity to operate with Deriveum (are financial institutions)

**Set-up/ context for LBChain implementation:**

The context helps to clarify the roles of participants that need to be filled during for this proof of concept use case.

1. The context is a Minimum Viable Product that will lock a number of tokens for a period of time, will be triggered by a pre-determined verifier and the contract will be destroyed by a resolver;
2. In the context of the MVP all users are representatives of financial institutions or notaries that agree to back government debt in a cross-border complex insurance contract

**Basic flow/ use case to be implemented (contract runs its course):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **User** | **Counterparty** | **Verifier** | **Resolver** | **Corda network** |
| 1 | Agrees terms of the CDS (off-chain contract) | Agrees terms of the CDS (off-chain contract) | Agrees to his/her role and off-chain payment conditions | Agrees to his/her role and off-chain payment conditions | Off-chain step, no involvement |
| 2 | Initiates creation of a new smart contract[[1]](#footnote-1) , where (s)he fills in the amount and duration and chooses the other parties from his/her contacts list. Also attaches the agreed documents in step 1 for reference by the other parties. |  |  |  | Creates space or route for the named parties to reach the uploaded by the initiator documents; once uploaded a document cannot be modified, but parties may add additional documents. Creates “Pending contracts” entry on the management console of the parties. |
| 3 | Signs the smart contract | Signs the smart contract agreeing with its terms | Signs the smart contract | Signs the smart contract | Certifies that the initiator has sufficient tokens in its account. Accepts and records the agreement by all parties. |
| 5 |  |  |  |  | Creates temporary account that has four exits[[2]](#footnote-2) and transfers the amount of tokens under the deal in that account |
| 6 |  |  |  |  | Certifies that the duration of the smart contract as agreed by the parties has expired |
| 7 |  |  |  |  | Returns the tokens back to the account of the initiator |

*Explanation: We anticipate that around 80-85% of the deals will run its course uneventfully and that the financial institution will get back its tokens.*

**Alternative flow (credit event):**

In the alternative flow the Verifier is notified by the counterparty that a credit event has taken place in accordance with the underlining off-chain agreement.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **User** | **Counterparty** | **Verifier** | **Resolver** | **Corda network** |
| 6 |  | By using the management console the counterparty requests triggering of the smart contract |  |  | Corda records the triggering request and an e-mail is send to the verifier |
| 7 |  |  | The verifier makes a judgment off line, based on the contracting terms and the fact on the ground if the credit even has actually taken place |  |  |
| 8 |  |  | Following the judgment that there is a need to trigger the contract the verifier signs the triggering |  | Records the verifier’s triggering of the contract |
| 9 |  |  |  |  | Moves the tokens from the temporary account to the account of the counterparty |

*Explanation: We expect that in around 5% of the cases a need for payment would arise under the CDS agreement.*

**Alternative flow (lack of payment):**

In the alternative flow the Resolver is notified by the initiator that a payment hasn’t been made in accordance with the underlining off-chain agreement.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **User** | **Counterparty** | **Verifier** | **Resolver** | **Corda network** |
| 6 | By using the management console the initiator requests termination of the smart contract |  |  |  | Corda records the termination request and an e-mail is send to the resolver |
| 7 |  |  |  | The resolver makes a judgment off line, based on the contracting terms and the fact on the ground if the conditions for termination has taken place |  |
| 8 |  |  |  | Following the judgment that there is a need to terminate the contract the resolver signs the termination | Records the resolver’s termination of the contract |
| 9 |  |  |  |  | Moves the tokens from the temporary account to the account of the initiator (user) |

*Explanation: We expect that in around 5% of the cases conditions necessitating termination – such as lack of payment, instruction by competent authority or mistake in the agreed terms - may take place.*

**Alternative flow (mutual agreement):**

In the alternative flow the Initiators wants to terminate the contract in order to be able to enter new agreement with the counterparty or a third person, where the initiator and the counterparty settle offline for the damages that the initiator would pay for terminating the contract.

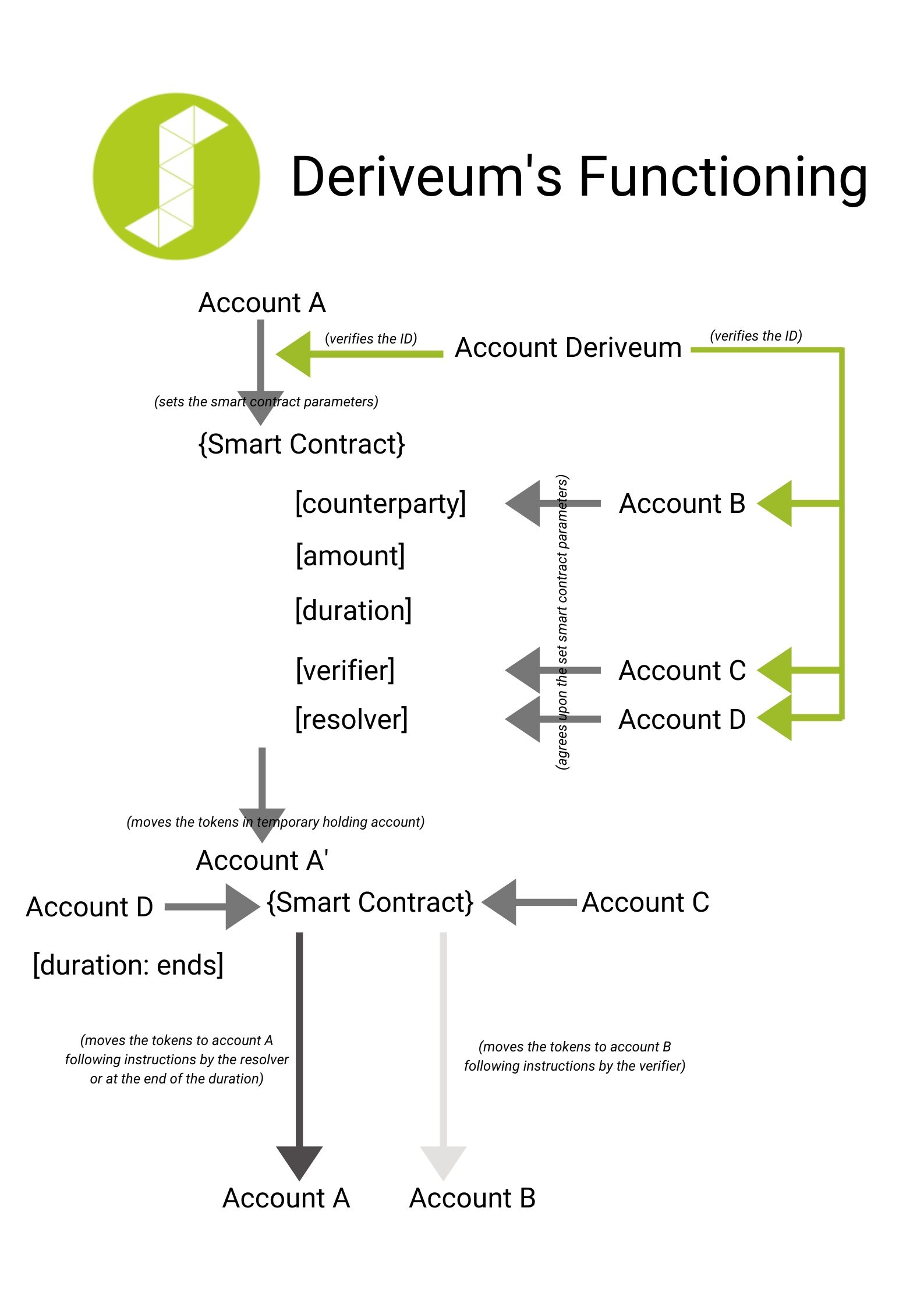
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **User** | **Counterparty** | **Verifier** | **Resolver** | **Corda network** |
| 6 | By using the management console the user requests premature termination of the smart contract |  |  |  | Corda records the termination request and an e-mail is send to the counterparty |
| 7 |  | Following off-line/off-chain settlement the counterparty uses the management console to agree on the premature termination |  |  | Records the counterparty’s triggering of the contract |
| 8 |  |  |  |  | Moves the tokens from the temporary account to the account of the initiator |

*Explanation: We expect that in around 5% of the cases will be terminated by mutual agreement by the parties.*

**Postconditions:**

*Special cases: not envisioned in the MVP*

**Graphic description of flow:**



The Initiator

The Verifier

The Resolver

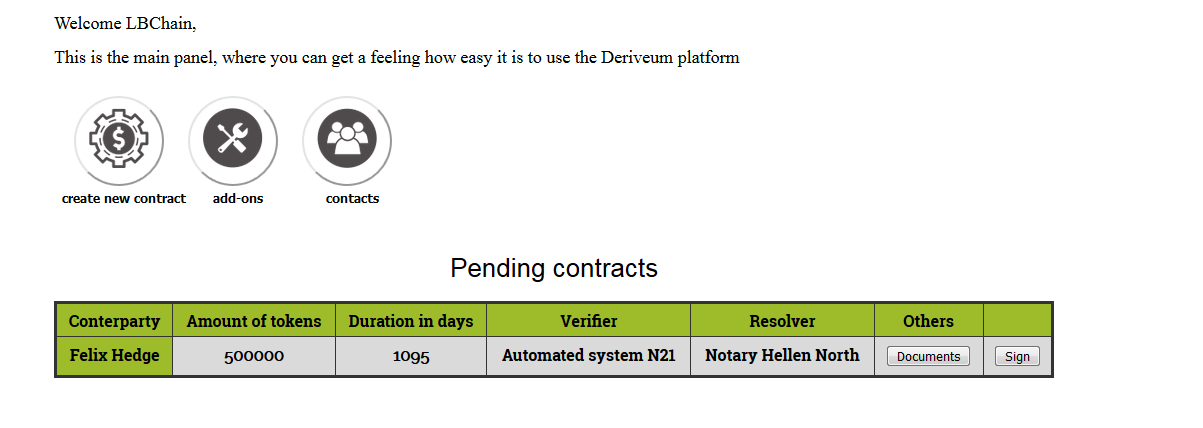
The Verifier

The Counterparty

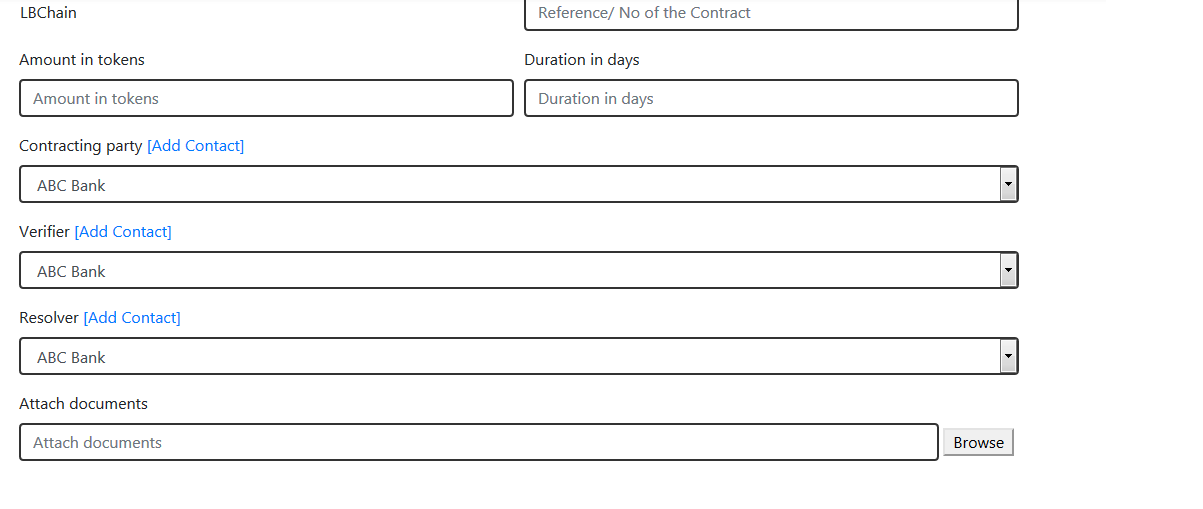
The Initiator

**Graphic description of the interface:**

Main panel with pending contracts



Create new contract



## **Thank you for Your Interest in our Project.**

1. The smart contract in the MVP has the following parameters: initiator, counterparty, amount (number of tokens), duration (number of days), verifier and a resolver. [↑](#footnote-ref-1)
2. Either the duration ends, the verifier or resolver triggers the contract or the parties both agree to terminate the contract [↑](#footnote-ref-2)