# Token Factory Application

Business Process

Token Factory

The **macro process** shows how the create token application works. There are two steps:

- the **open area**, where the user defines the token and makes its registration;
- the **logged area**, where the user defines the licenses, the payment method and manages the token.

The application also needs to manage the status of these licenses
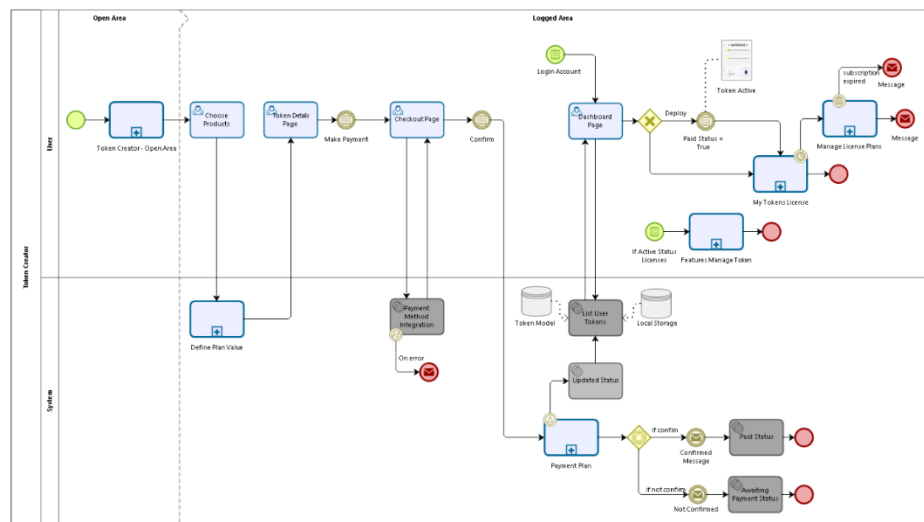
In the **open area** the user will define his token, a summary of the token to be created is displayed. After filling in the account registration information, a wallet is created and associated with that user registration.
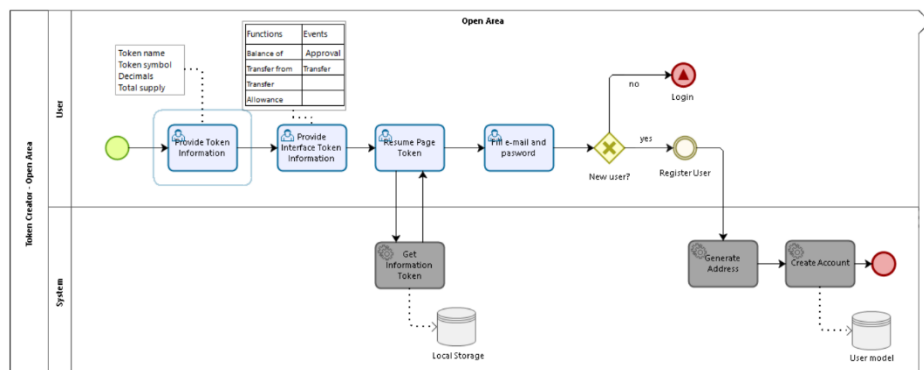
The setup fee is paid only when creating the token. The purchase of other products after the creation of the token does not include the value of the setup.
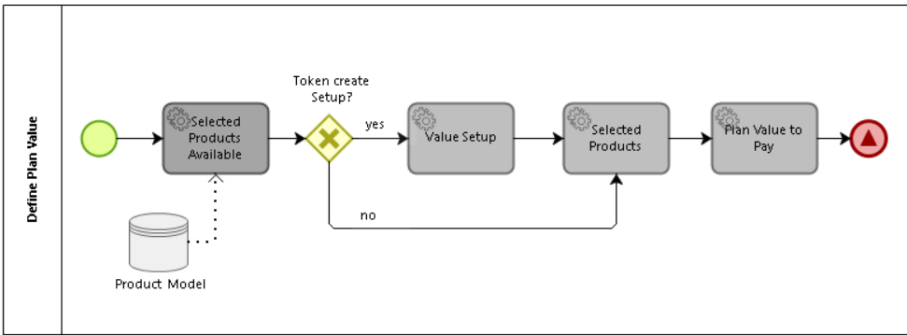
After confirmation of payment, the products selected for your plan are available for use.
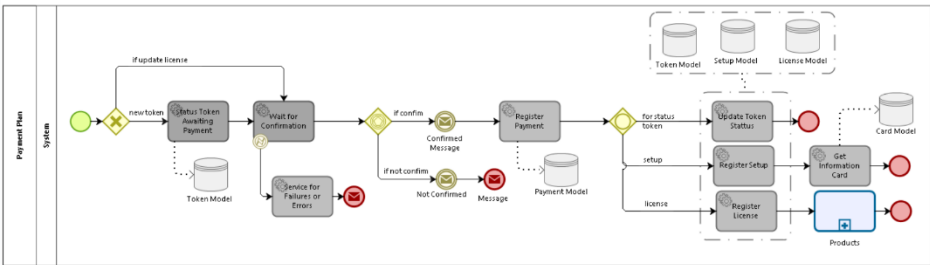
Features to Manage Token

To manage the token, an application with the main features is available to the user. Currently it is possible to mint and burn the created token.
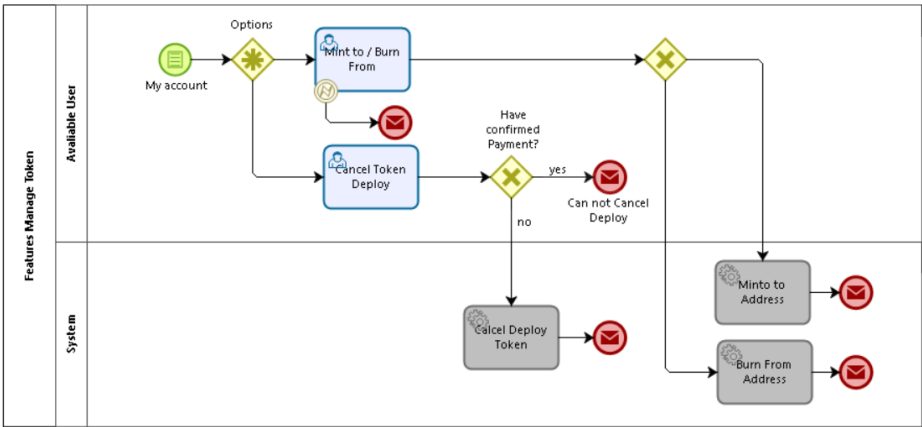
Manage Licenses by User

The user can manage his plan by canceling products or adding others at any time.
The cancellation checks the remaining usage time.
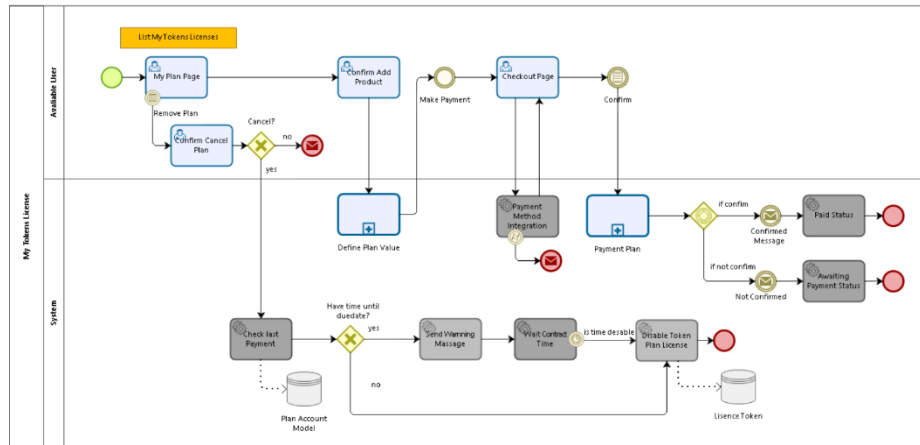
Plan Accounts Manager

The license management generates the recurring payment and checks whether the license is close
to the expiration date to notify the token owner. If the license expires and there is no payment, it
is canceled.

Acceptance criteria

- **An account** can create **multiples tokens** contracts
- Each new account need generate an address to be a owner of the Tokens
- User need fill with e-mail and password to create an account
- Encrypted storage of generated wallet information
- User does not receive private key and mneumonic
- The token symbol need to be unique in the environment
- Create Token smart contracts **ERC-20** standard Ethereum Token, mintable and burnable, Ownable and Pausable
- User buy licences to use the avaliable products
- User need to pay a setup value once
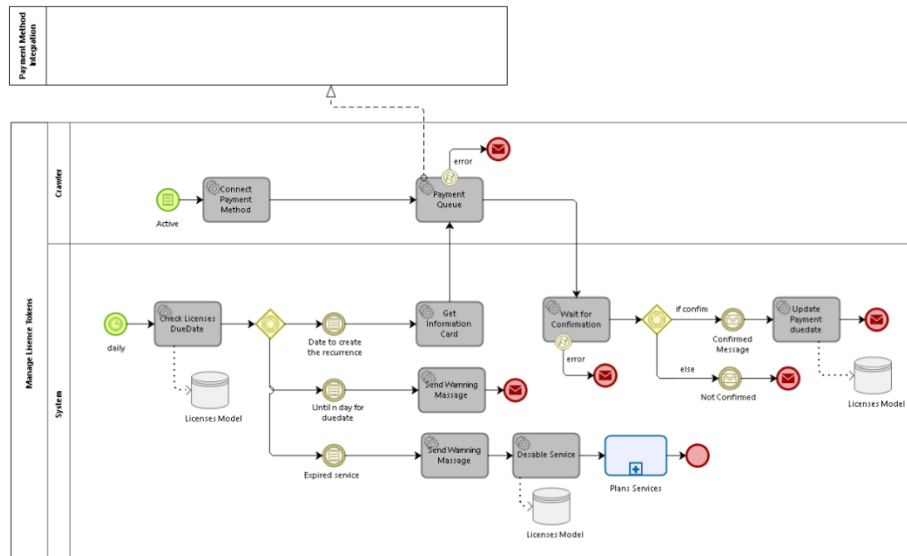- Each new token need to be a subscribe to a service. Is for token and not for account.
- Deploy only to the mainnet
- Internationalization (PT-BR and English)
- Status to create a Token:

Status	Description
Wainting for Payment	Pending confirmation of license payment.
Paid	After confirmation of license payment
Deployed	After user makes the token deploy.

Interface to create and manage Token

- Open area:
  - Form with infos about token creation

Field	Descryption
Token Name	String (3 - 25 symbols. Alphanumerical characters, space, and hyphen are accepted.)
Token Symbol	String (3-4 letters and need to be unique)
Decimals	Integer (0 - 18 integer)
Type	Value ERC-20
Total Suply	Integer

- Logged area
  - Manage token application

Menu	Descryption
Menu Dashboard	Card with Information about created token (with details created token)
Menu My Plan	Show information about the contracted license
Payment Details	List the payment details for each token.
My Tokens	List all my created tokens
Menu Mint / Burn	Enabled after the token is in active status, user can mint/burn to token created

How it works

Create your Token without code and in a few steps.
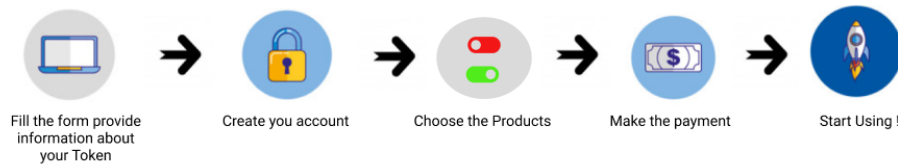
1. With an intuitive form you provide information about the token you want to create: name, symbol, number of decimal places and the total suply. All information has a clear explanation.
2. You create your account. Whith this account you can manage your Token easily with a intuitive interface.
3. Make the payment. Afther payment confirmation we made deploy
4. Then start to use

Proccess Steps

Start Create Token

If you want to generate your own token, we have created a tool that will enable you to deploy your own token in a few minutes, without coding skills. You only needs define the name and symbol of the token, as well as initial supply.

Step1 - Provide Token Information

- Fill the form with information about your Token:
  - o **Name** -> the name of your token, it is important to give an identity. 3-25 symbols. Alphanumerical characters, space, and hyphen are accepted.
  - o **Symbol** -> the symbol represents your brand, it is usually 3-4 letters and need to be unique
  - o **Type** -> ERC-20 is a proposed and widely adopted standard for creating tokens. It's a set of rules implemented in a smart contract that is deployed on the Must network
  - o **Decimal places** -> The divisibility will help us determine the lowest possible value of the token.
    A divisibility of 0 will mean that the lowest value of the token is 1.
    A divisibility of 2, on the other hand, means its lowest value will be 0.01.
    The maximum number of decimal places allowed is 18.

o **Total supply** -> Number of tokens that will exist in the created ecosystem. This amount can be changed according to market needs, and can be increased or decreased (from the Mint and Burn functions respectively)

Step2 - Provide Interface Information

- The choice of functions defines how your token will work.
  All created smart contracts are **ERC-20** standard Token, mintable and burnable, with owner access permissions and module pausable.

With this Token you can generate more tokens to increse the Total Supply of your token.

Function	Description	Default ERC-20
Transfer	Transfers amount of tokens from the caller's account to recipient.	yes
Allowance (approve, increase and decrease)	Checks the limit that an address_spender (approved by the token owner) can spend from _owner.	yes
Transfer from	Moves amount tokens from sender to recipient.	yes
Balance of	Show the amount of tokens owned by your account	yes

- **Events** are important because they facilitate communication between smart contracts and their user interfaces.

Event	Description	Default ERC-20
Transfer	Emitted when value tokens are moved from one account (from) to another (to).	yes
Approval	Emitted when the allowance of a spender for an owner is set by a call to approve.	yes

Step3 - Resume Token

- A summary is displayed with the selected information of the token that will be created. It is possible to return and change if necessary.

Step4 - Register new account

- To access the token management functionality and the other benefits offered, it is necessary to create an account.

Step5 - Choose the plan

- We provide the best management and control tools for your token in each available plan.

Step6 - Make the payment

- Choose your payment method

Features

```
Feature: Create Token
    As a user I want to fill a form and then use my new token.

Scenario: Verify if description form is empty
    Given I am at create-token-description page
    When I click <next> button
    And the field Name filled with <Name>
    And the field Symbol filled with <Symbol>
    And the field Decimal filled with <Decimal>
    And the field Total Supply filled with <totalSupply>
    Then I should see a <message>
```

Name	Symbol	Decimal	type	totalSupply	message
empty	"TKN"	18	ERC-20	0	Field needs to be filled
"Token Name"	empty	18	ERC-20	0	Field needs to be filled
"Token Name"	"TKN"	empty	ERC-20	0	Field needs to be filled
"Token Name"	"TKN"	18	ERC-20	empty	Field needs to be filled

```
Scenario: verify if all itens are selectable
    Given I am at create-token-interface page
    And all itens <itens> enabled
    When I check an enabled item <itens>
    Then this must be desable

Scenario: verify return data token
    Given I filled the form to create token
    When I access create-token-resume page
    Then I see all the information about the token

@register_User
Background: I am at create-token-register page

Scenario: Register user with invalid e-mail
    Given I fill the field e-mail with <email>
    And I fill the field password with <password>
    And I fill the field Confirm password with <confirmPassword>
    When I click Register button
    Then I can't proceed
```

```
Scenario: Register user with invalid password
    Given I fill the field e-mail with <email>
    And I fill the field password with <password>
    And I fill the field Confirm password with <confirmPassword>
    When I click Register button
    Then I can't proceed
|email           | password  |confirmPassword|
|xpto.gmail.     | 123456789 |123456789      |
|xpto@gmail.com  | 123456789 |123456         |


@payment_method
    Given context
    When event
    Then outcome
Feature: Managment Token

    As I am a token owner I want to a easy experience to manage my token

@token_features

Background: logged in Token Creator application
    Given I am login page Token Creator application
    When I fill my <username>
    And my <password>
    Then I am at dashboard page

Scenario: Mint token
    Given I click mint button
    And I fill field "Address to" with <addressTo>
    And the field "Amount" with <amount>
    When I click mint button
    Then Then I see a message <message>

Scenario: Burn token
    Given I click burn button
    And I fill field "Address from" with <addressFrom>
    And the field "Amount" with <amount>
    When I click burn button
    Then I see a message <message>

Scenario: Cancel Deploy
    Given my created token has "Wainting for Deploy" status
    When I click cancel deploy button
    Then I see a message <message>

Scenario: Change the token plan
    Given I click change plan button
    When I am at my plan page
    And change my current plan
    Then I click make the payment button
    And I see payment page

Scenario: Access payment Details
    Given I need to know about my token plan
    When I select payment details at menu itens
    Then I see a list of my tokens and details plan
```