

Everscale Self-Sovereign Identity Framework (Stage 4). Identix.Pass (Public)

Everscale SSI Framework Contest. Motivation

Deliverables for Stage 4

What had been changed for Stage 4 of the SSI Framework contest:

Further directions:

What is Identix.Pass?

Screencast of Identix.Pass

Identix.Pass for end-users

Identix.Pass for issuers

Identix.Pass for verifiers

Identix.Pass as identity provider (Identix.SSO)

Identix.Pass walkthrough

External Links

License (MIT)

Everscale SSI Framework Contest. Motivation

The self-sovereign identity (SSI) concept appears to be one of the most prominent identity technologies of this decade. The main goal of SSI is to shift the locus of control over users' data and authorization functions. It gives back the power of profile control to end-users while preserving a prominent level of privacy and security. Various consortiums and state authorities have paid attention to the SSI development and have already undertaken certain actions to set up proper agenda and to hold the best positions at the frontier of the technology.

Being technology-agnostic the SSI, however, perfectly fits any blockchain ecosystem due to its decentralized and trustless character. Blockchain networks can perform the role of decentralized virtual machines for SSI-based smart contracts or even store the data of certain SSI components.

After the previous stages devoted to the theoretical justification and the DID method implementation, we continued the development of the SSI framework with a Verifiable Credentials (VC) infrastructure. The staged implementation of the contest led to the creation of the Identix.PASS product, which we intend to develop further.

Deliverables for Stage 4



Technical documentation for Stage 4 may be found in this repo: <https://github.com/identix-space/everscale-ssi-contest-stage4>



Product information is contained at the Notion public page <https://jealous-shroud-6ff.notion.site/Everscale-Self-Sovereign-Identity-Framework-Stage-4-Identix-Pass-Public-f0c199cf8a8c4b8e860dca5dc94f130e>

What had been changed for Stage 4 of the SSI Framework contest:

- Deployed Identix SSO service: <https://sso.identix.space/>
- Deployed Identix DID Registry
`0:fc8ca01354b618a10d91a2d6dce88596eadb06a9143e8696b63fa5fb582d0eb1`
- Deployed Identix.Pass service: <https://pass.identix.space/>
- Deployed Identix VC fabric smart contract at Everscale:
`0:6c33236486c501865c88a67c59f50a74568ef7492c22db400bccf179483c0880`
- Code repositories:
[everscale-ssi-contest-stage4](#) - the contest documentation
[identix-pass-service](#), [identix-pass-service-ui](#) - Identix.Pass web app
[identix-wallets](#) - custodial SSI wallets to control DIDs/VCs
[identix-sso-service](#), [identix-sso-ui](#) - Single Sign-On solution
[identix-ssi-contracts](#) - Everscale smart contracts and VC schemas

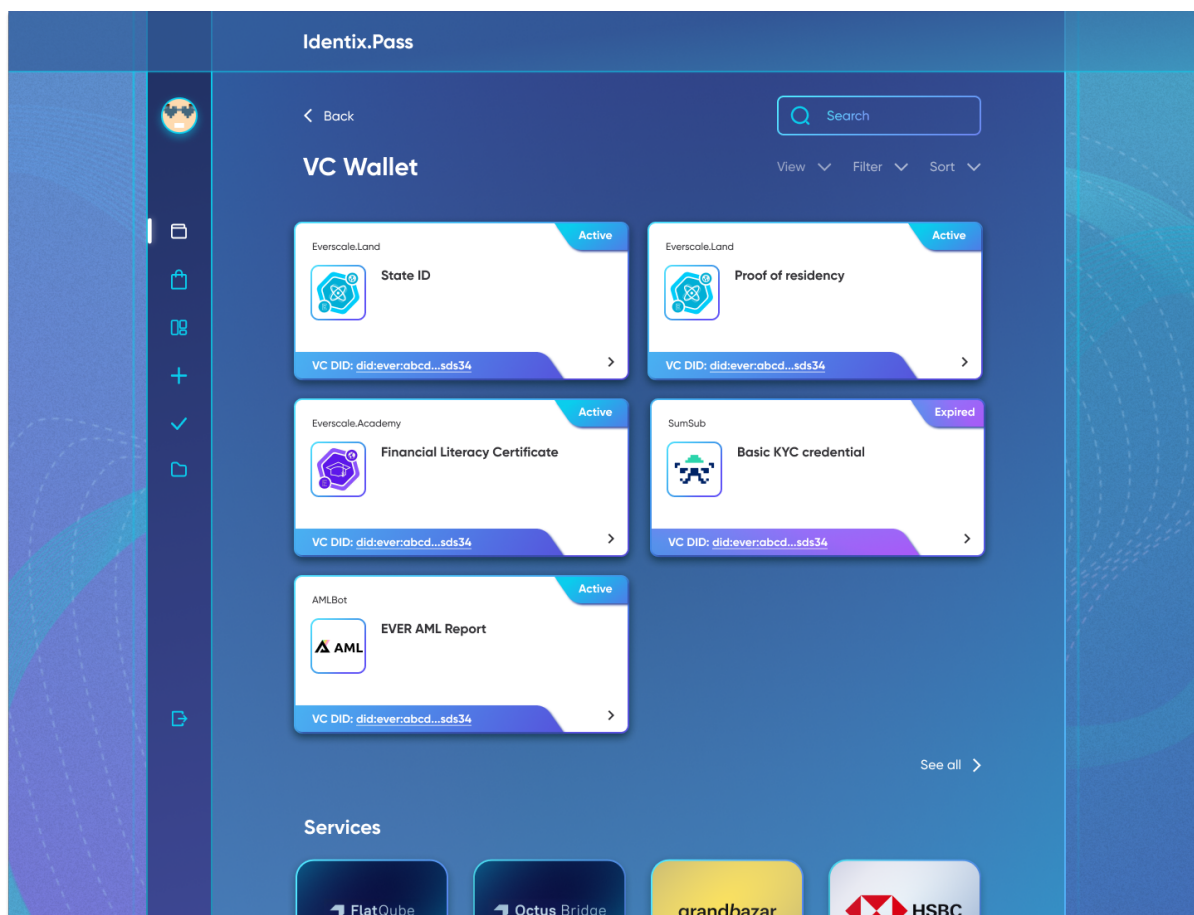
Further directions:

- VC Brokerage with multiple issuers and trust chains
- VC Registry: types, schemas, issuers, verifiers
- VC Schema constructor
- Advanced brokerage: issuance and verifications requests, credential demands
- Advanced roles: Proxy, Delegate, Arbiter

What is Identix.Pass?



The prototype is located at this address: <https://pass.identix.space/>



Identix.Pass is a solution that serves as an entry point for users to the Everscale SSI stack. It is a platform where SSI agents reach each other and exchange VCs.

This prototype continues the development of [Everscale.id](#), which we prepared for the [previous stage](#) of the Everscale SSI Framework (DID method implementation). We complemented it with VC issuance, storage, and verification tools, as well as a custodial wallet solution for creating an SSI profile in a few clicks.

In comparison with the previous prototype, enhanced user authentication options appeared. Identix.Pass allows users to create their SSI Profile in several clicks as it supports custodial authentication methods (Google, Facebook) to create Identix.Pass profile. In such a case the user's DID is created and managed by Identix.Pass, which has control over the user's key pair. At the same time, full-fledged web3 authentication remains available through EVER Wallet. This authentication solution (as **Identix.SSO** component) can be used by third-party services, providing them with DID authentication and VC capabilities.

Stage 4 of the contest implies the basic implementation of SSI Trust Triangle relations with issuance and verification procedures possible. In this regard, we propose the concept of **Verifiable Credential Brokerage Protocol** which is the prototype of a public standard for digital trust relations. Identix.Pass has basic interfaces to let users go through the full VC lifecycle:

1. Issue a credential (performed by Issuer).
2. Hold it in the wallet (performed by User).
3. Provide it to the Verifier (performed by User).

4. Verify the VC (performed by Verifier).

We suppose that Identix.Pass will continue to evolve in parallel with Everscale VC infrastructure. Most likely, it will take the role of VC Marketplace on the market - the place where VC mappings between providers of verifiable credentials and services that accept them will be established. This requires additional development of several tools and protocols ((in particular, in the field of VC schema management).

At this stage we decided not to deliver the VC marketplace function because it would require additional architectural components that will be delivered in later stages of the Everscale SSI Framework.

Screencast of Identix.Pass

<https://youtu.be/TUqbsdfnWJU>

Identix.Pass for end-users

For end-users, Identix.Pass provides the following scenarios:

1. **Creating SSI Profile.** Identix.SSO allows users to create their DID both through the EVER wallet and using web2 Identity providers. It is assumed that the user in any case will be able to change the DID controller to himself.
2. **Store verifiable credentials in VC Wallet.** Currently, this is a custodial wallet on the Identix.Pass side. Allows to observe the credentials, and view their claims and statuses.
3. **Sharing credentials to Services (Verifiers).** As part of the implementation for Stage 4, we have mocked one Verifier (FlatQube DEX platform) and the list of VCs it may accept in the Services section. Here, the user sends the VC for verification and observes the request processing result.

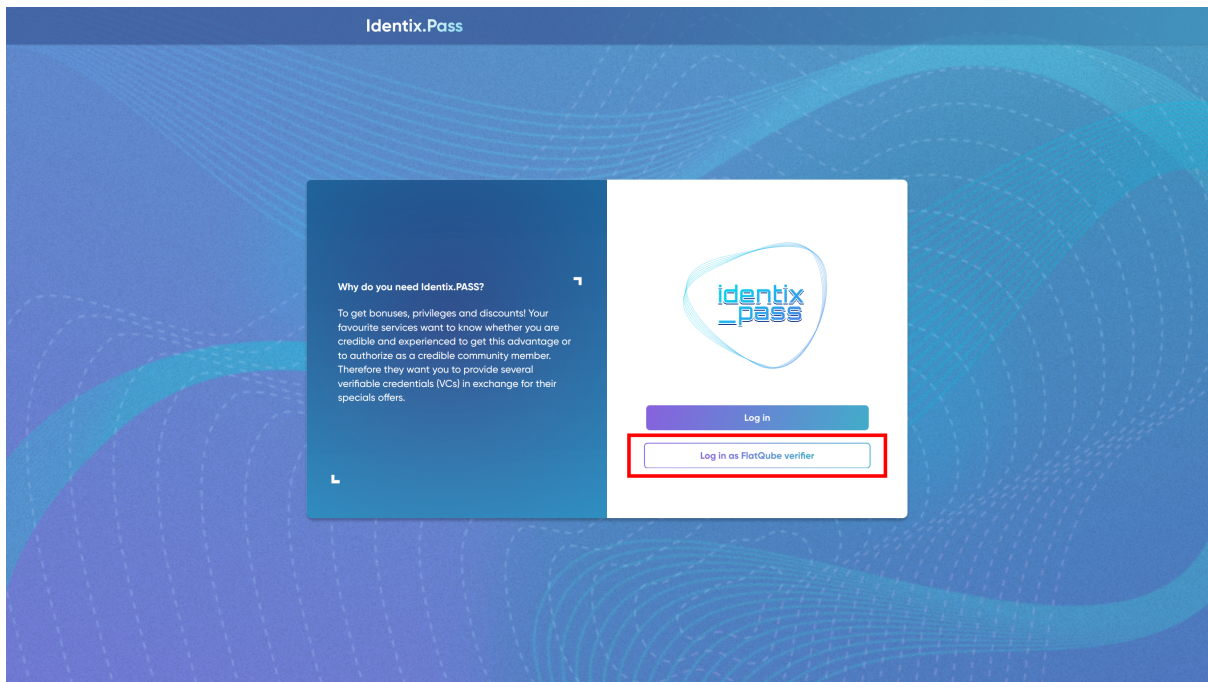
Identix.Pass for issuers

Currently, an issuer can select a VC schema from a predefined list and issue a VC on the specified DID. More advanced scenarios (requests processing, VC schema creation and deployment) are planned for future iterations.

Identix.Pass for verifiers

For Verifiers Identix.Pass provides interfaces for processing incoming verification requests and reviewing VCs for the purpose of verification and making an appropriate decision.

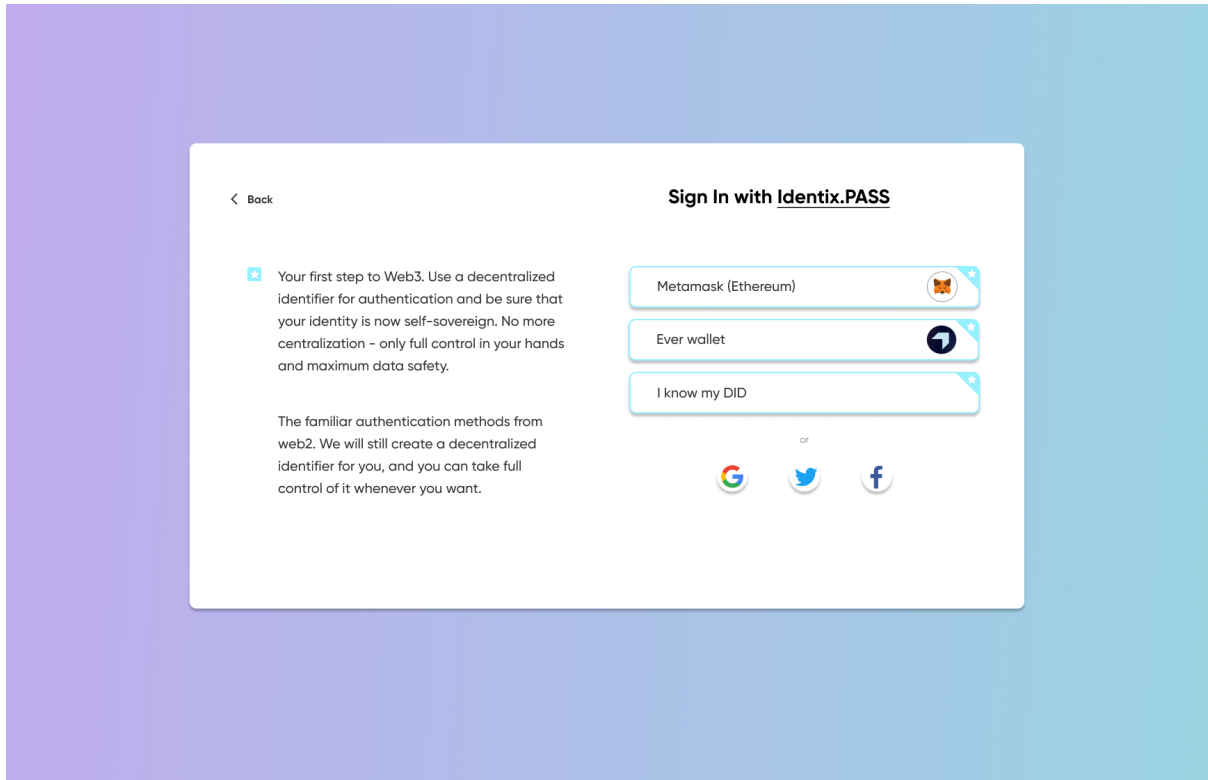
NOTE: since we have not launched VC Marketplace, the Verifiers listed in Services are hardcoded. To test VC verification you should log in as a FlatQube verifier (which is the account with a specific hardcoded DID). Then, you will see verification requests sent by other users and accept or decline them.



Identix.Pass as identity provider (Identix.SSO)

We provide Identix.SSO authentication solution for any site that wishes to implement decentralized authentication (and optionally an SSI profile) in their service. For users, this looks like a typical SSO button. The user who clicks on the button has to complete 2-3 simple steps to create a DID and to be authenticated as if he were doing it through traditional identity providers like Google, Twitter, or Facebook.

Connected services obtain a custodial wallet that allows to onboard users in several clicks and enable them to interact with blockchain features (sign messages, send transactions, manage DID and VCs).



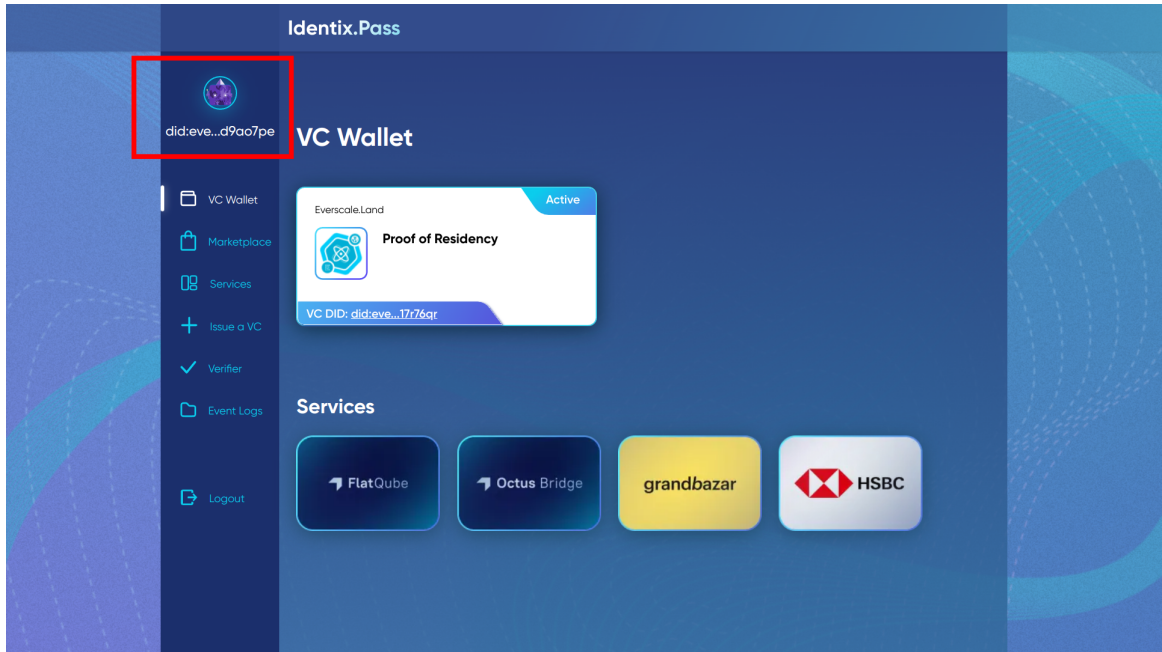
Identix.Pass walkthrough

1. **First, you need to create your Identix.Pass profile.** This implies the creation of your decentralized identifier (DID). This DID is the core of your SSI profile: this is the identifier that will be used while anchoring your VCs to the Everscale network.

You have 2 options here:

- a. Use an Identity provider (Google or Facebook).
- b. Use EVER Wallet for web3 authentication.

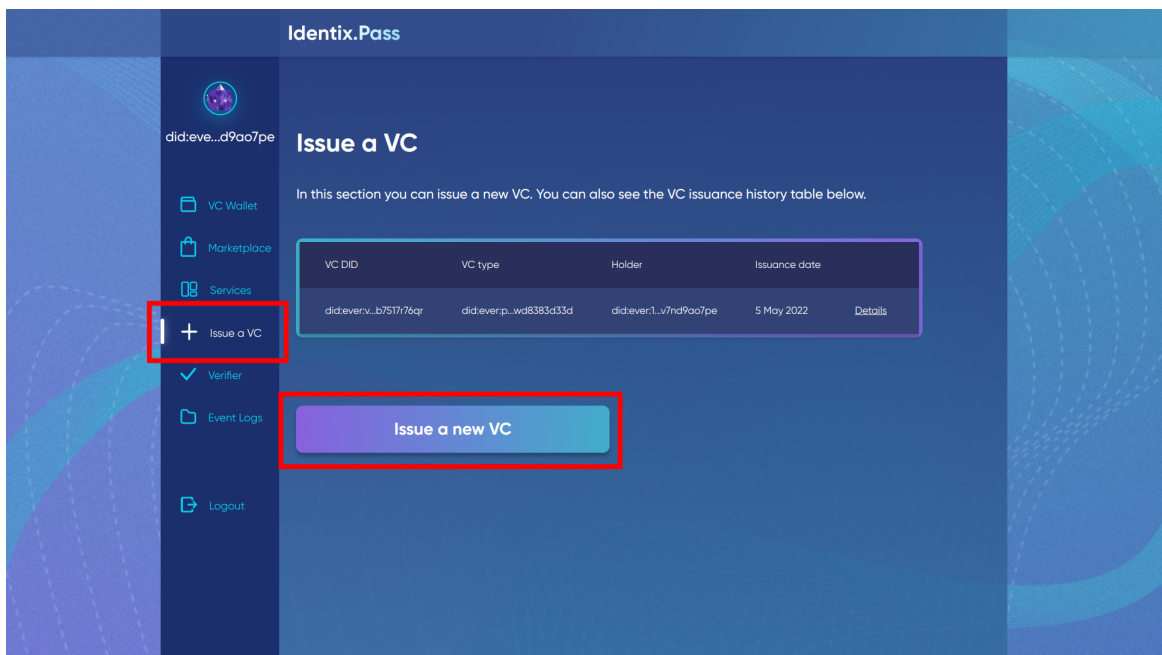
Go to <https://pass.identix.space/> to start. Once you have logged in, in the upper left corner of the main menu **you can see your DID**. You can click on it to copy.



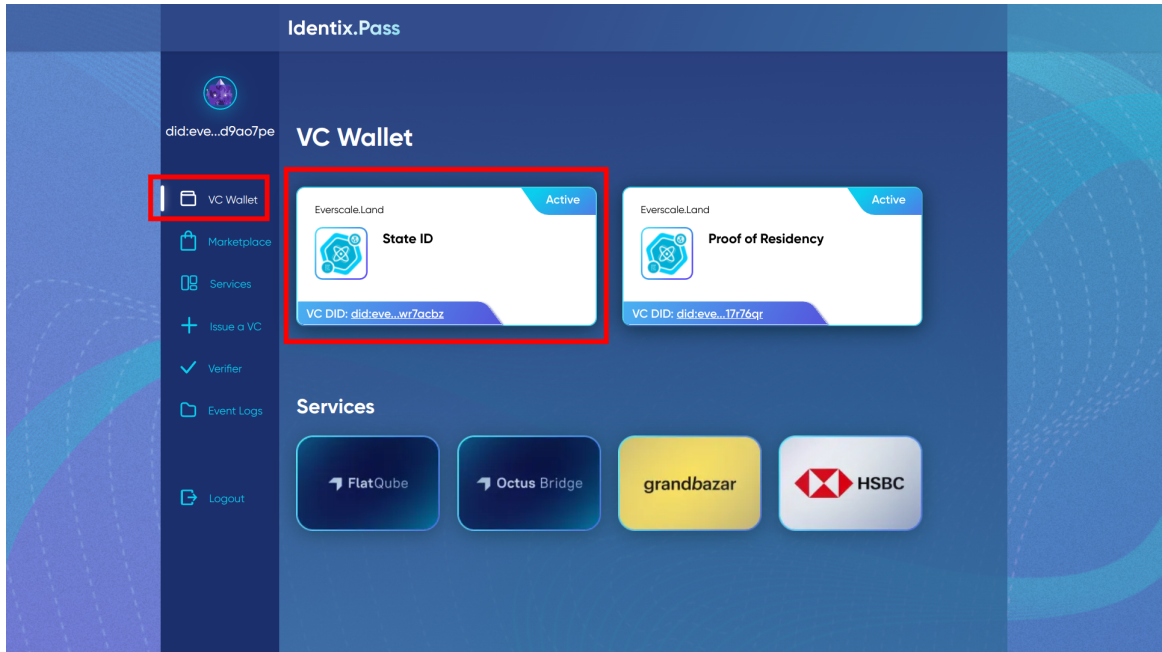
2. **Next, you need to receive your first verifiable credential.** There are currently no restrictions on VC issuance, so it doesn't matter if you issue a VC on your DID yourself or ask someone else. For simplicity, we will show the first option. But usually you, will request them from issuers.

To obtain a VC, follow a few simple steps:

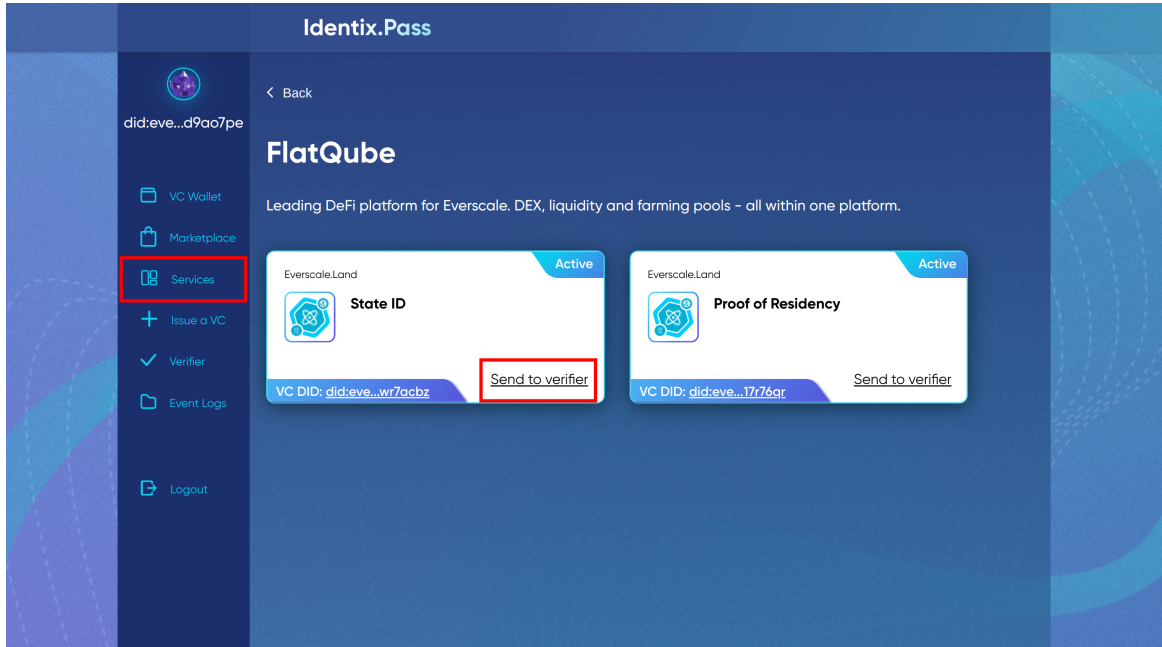
- Go to the **“Issue a VC”** section from the main menu.
- Specify your DID when choosing the VC holder.
- Choose the VC type and fill in all the fields.
- Review your VC data. Sign the document if everything is ok.



The received credential will appear in the wallet in a few seconds (**VC Wallet** section).

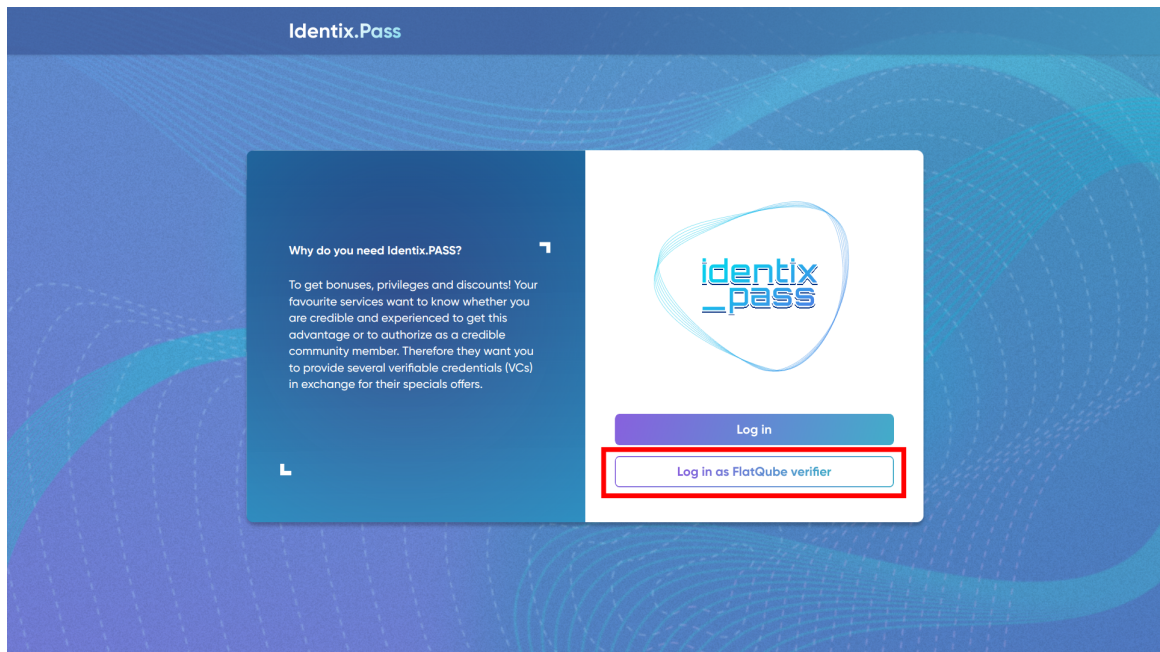


- Now you should provide the VC to the verifier.** Go to the Services section from the main menu. Here, for demonstration purposes, we have mocked one DeFI service - FlatQube - which supposedly accepts your VC for KYC purposes. Click on the “Send to verifier” button to provide the VC for verification.

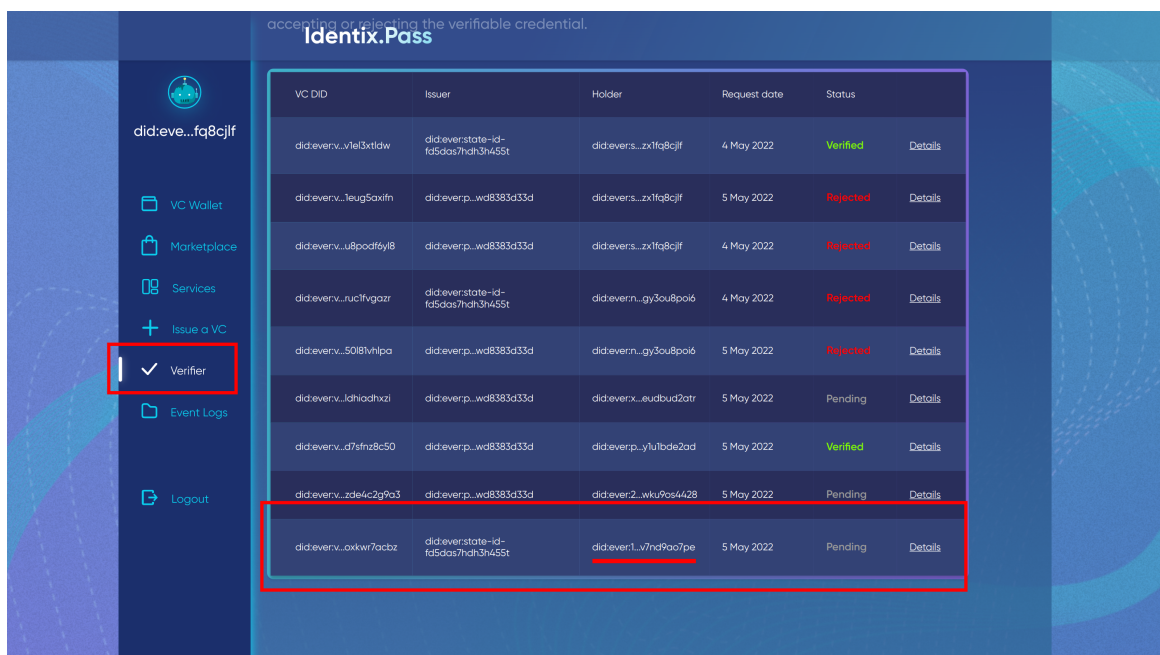


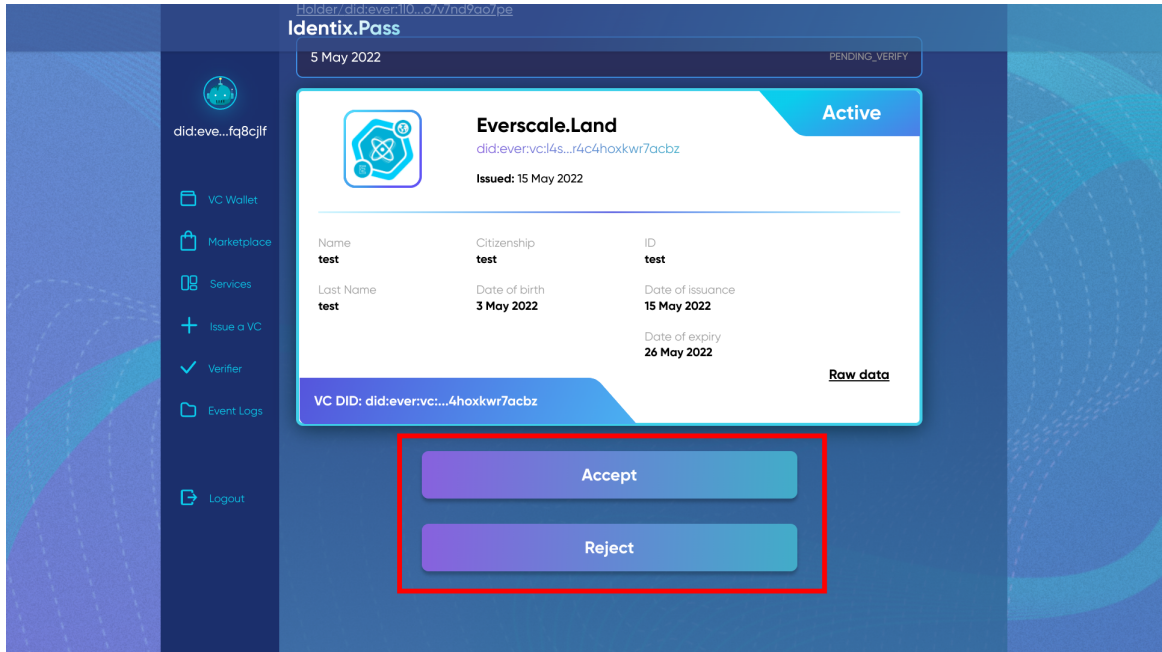
- The next step is to log in as a verifier.** In general, each SSI agent can verify the VC: for this you have the Verifier section in the main menu. However, since you sent VCs to the fake FlatQube agent for demonstration purposes, now you must log in on his side and process the verification request.

Logout from current profile. You will see the “Log in as FlatQube verifier” button, which will log you in with the right DID:

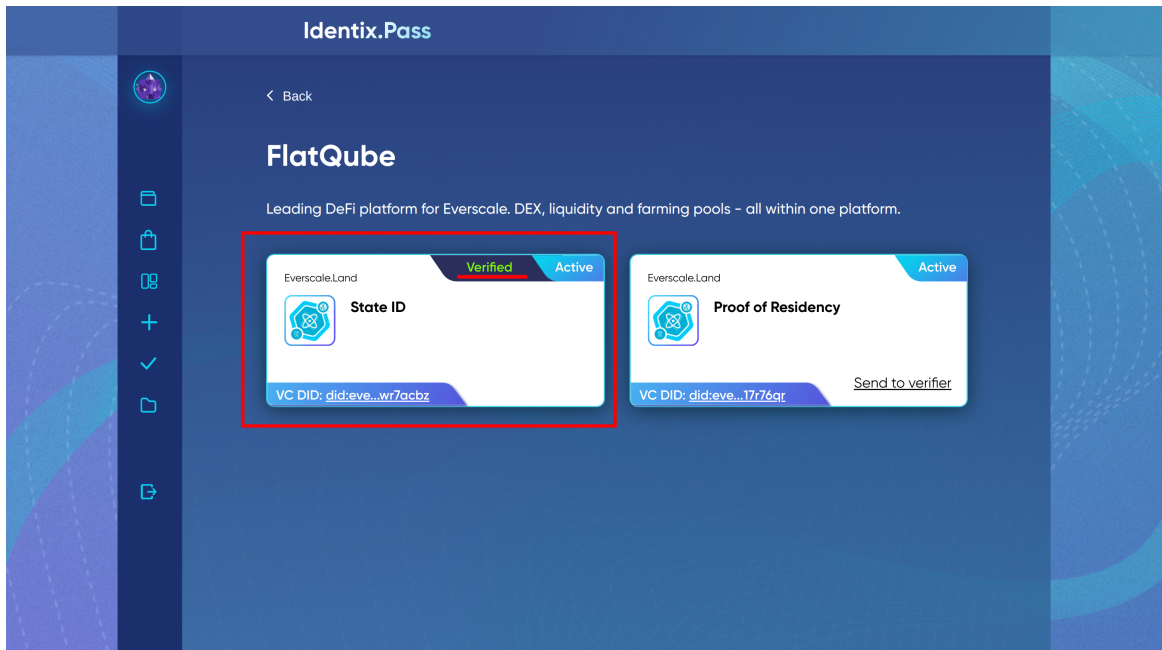


Once you are logged in, go to the Verifier section from the main panel. Select the required request in the table and process it.





5. **Leave the verifier profile.** Now when you log in to your profile again, in the Services section you will see that your verification request has been processed.



External Links

Notion public page: <https://jealous-shroud-6ff.notion.site/Everscale-Self-Sovereign-Identity-Framework-Stage-4-Identix-Pass-Public-f0c199cf8a8c4b8e860dca5dc94f130e>

Web app - <https://pass.identix.space/>

Github repos - <https://github.com/identix-space>

Technical documentation for Stage 4 - <https://github.com/identix-space/Everscale-SSI-Contest-Stage4>

Showcase - <https://youtu.be/TUqbsdfnWJU>

Contacts (Telegram): @avarab (Alex Varabei)

License (MIT)

Copyright (c) [Identix.Space](#) team

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.