We are a team of Veterans, Creators, and Innovators dedicated to ensuring Veterans and their families get access to the benefits they've earned and deserve.

VA.gov Identity Team Release Notes

February 16th - 28th, 2023

Overview:

During this release we focused on enhancing the Engineer experience within VA.gov by starting down the path of creating a Mocked Authentication feature for localhost and dev. We've also released several enhancements to our SiS, again mostly related to the Engineer experience that we know will set up the next phase of adoption.

For more information on previous release reports please visit our Github Page.

New Features and Improvements:

- Initial version of Mocked Authentication
- Sign in Service Microservice Infrastructure

Bug Fixes:

- Flipper Refresh for SiS

Table of Contents

Overview: 3

New Features and Improvements: 3

Bug Fixes: 3

Featured Releases and Updates 4
Mocked Authentication: 4
Sign in Service Sample Application: 6
Sign in Service Infrastructure Preparation: 7
Datadog Resource Import and Custom Pipelines: 7
Login.gov Adoption Web Comment Analysis: 8
Sign in Service (SiS) Enhancements: 8
Frontend Configs for Mocked Authentication: 9
Notable Documentation Updates: 9

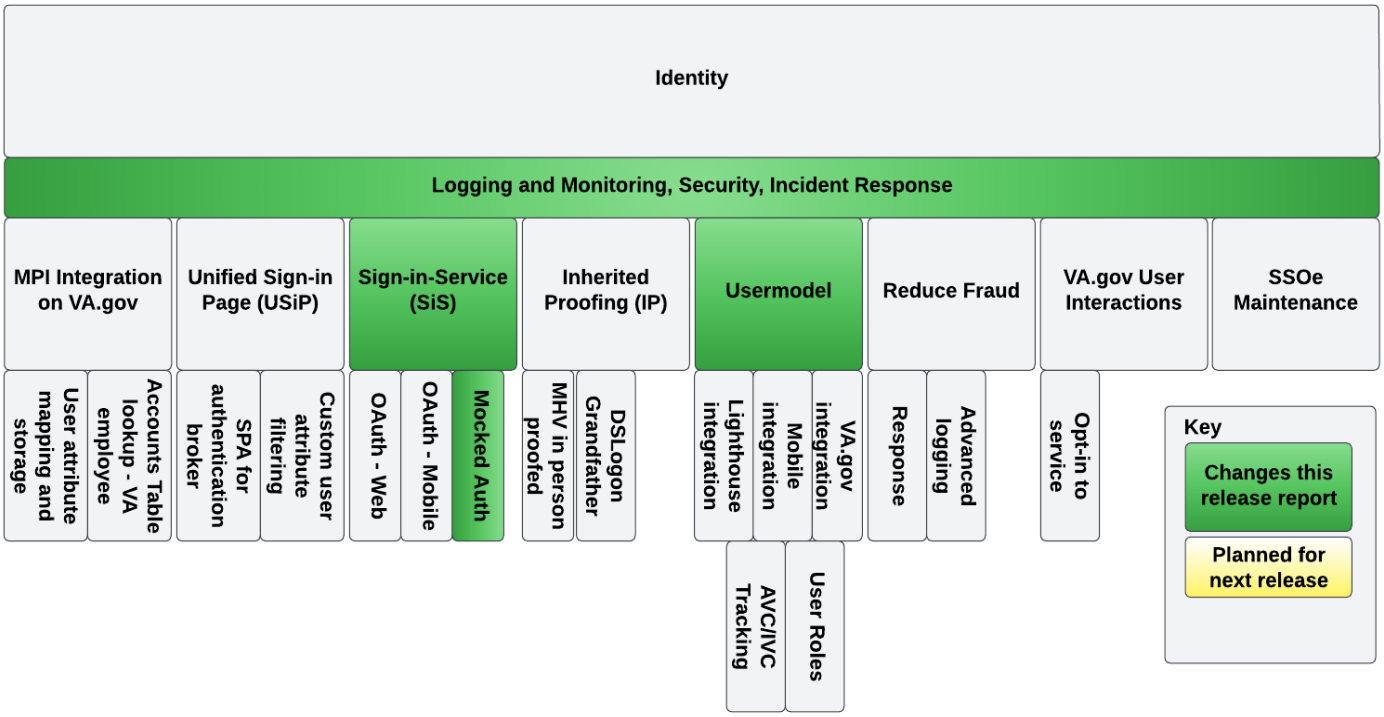
PM Update: 10

Issues/Limitations/Challenges: 10
How to provide feedback: 10

APPENDIX A- Ticket Issues Completed 11

Refer to the Key below to see the changes in this Release Report highlighted in green.

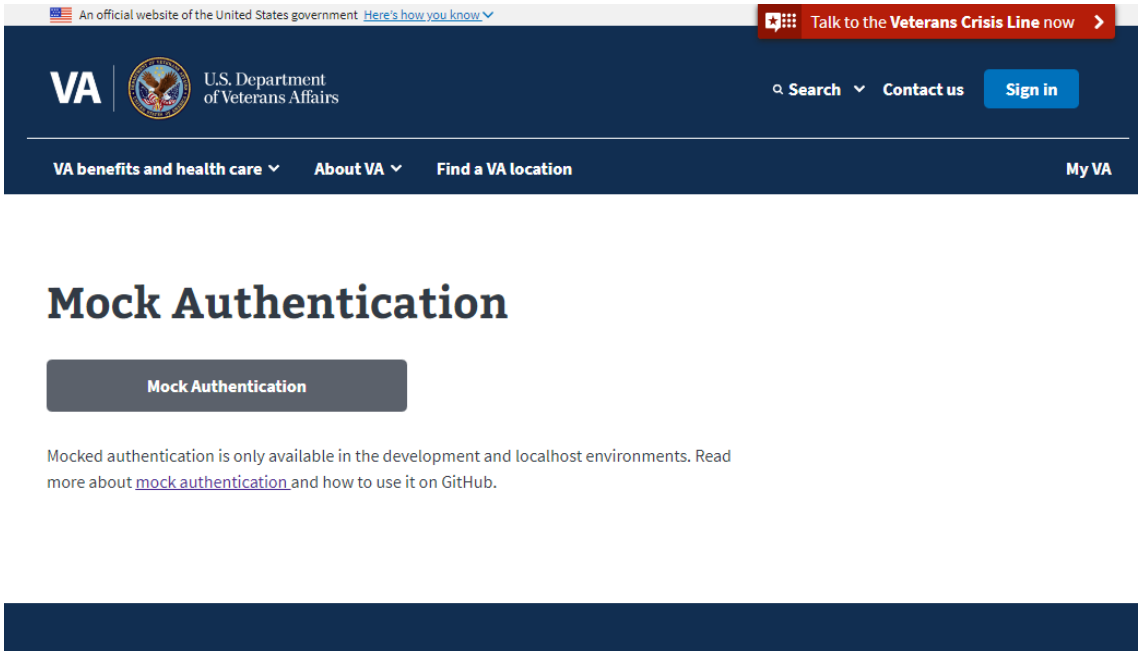
Note: See **link** to Identity Capabilities definitions.



Featured Releases and Updates

Mocked Authentication:

Summary: Mocked Auth (Mocked Authentication) provides the ability for Engineers to authenticate without explicitly entering credentials on external credential provider sites. It is available only for localhost and dev.va.gov stacks. This new feature will provide Engineers with an easier way to test authenticated endpoints.
How to use it: When an Engineer selects Mocked Auth, they will then be able to choose a credential provider and define a payload representing the mocked credential user information. This gives the vets-api backend the ability to communicate with this mocked auth entity, as if it were a 'real credential provider'. See below Image 3: Proposed Mocked Auth.



Mocked Auth will work for the "Sign in Service" method of authentication, on the localhost and development stacks only. It will be accessible through the unified sign in page, as an extra button that can be selected. There will also be a way to utilize the Mocked Auth feature using only an API based flow. This enables automated testing on the dev.va.gov stack to utilize Mocked Auth to test arbitrary authenticated routes.

Who is impacted: Engineers and Product Testers

What are the benefits: The solution will make it easier to test and develop around authenticated services.

Why did we make the change: Engineer feedback requested a feature that allows for a simple and secure method to bypass and mock authentication on localhost and dev.

Version, and planned release dates: Initial version, first release with selectable users planned for end of March 2023.

What's next: Create a frontend page and integrate it with backend API.

Sign in Service Sample Application:

Summary: The Sign in Service (SiS) Sample Application is a Ruby Sinatra application that serves as an example of how a client might interact with the SiS. This example application will allow a Engineer to investigate how the SiS API works from a client perspective, and give them an easily built application with the following features:

List	Features
I.	Ability to navigate to a root page and see a Log in/Sign in form
II.	Ability to navigate to a root page and see a Log in/Sign in form
III.	Ability to authenticate with Sign in Service
IV.	Ability to inspect relevant attributes once authentication has occurred
V.	Ability to inspect session behavior once authentication has occurred
VI.	Automatic session refresh capability
VII.	Logout functionality

Note: This example application will be able to integrate with Sign in Service localhost (where both example app and Sign in Service are built on the same Engineer machine).

How to use it: An Engineer will be able to download the sample application and build/run it, then use it to see a sample integration with Sign in Service. More integration/how to documentation to follow as the feature is completed.

Note: The public repository is located here: Sign in Service Sinatra Client

Who is impacted: Engineers and future Sign in Service Clients.

What are the benefits: Entities interested in Sign in Service will have an example of how to easily integrate with the service.

Why did we make the change: An example application will allow for teams that plan to integrate with Sign in Service to more effectively discover and implement the required changes to integrate.

Version, and planned release dates: Initial version, first release planned by end of March 2023.

What's next (most immediate next sprint): Complete backend API integrations and frontend authentication page.

Sign in Service Infrastructure Preparation:

Summary: Before moving Sign in Service out of vets-api, work was done to test some of the necessary infrastructure elements such as Elasticache Redis and RDS Postgres. This implementation is important as it will help reveal details we may have missed during initial research.

Note: More information can be found in the repository linked here: Create Sign in Service Infra Repository / Investigate Infra Dependencies for Ruby App

Who is impacted: Engineers (backend, front-end, and devops Engineers) and future Sign in Service Clients.

What are the benefits: Allows for separation of concerns between SiS and vets-api so that we can own everything related to SiS and just SiS as well as separate everything into their own repositories making things such as getting approvals easier.

Version, and planned release dates: Too early to have a delivery date, stay tuned!

What's next (most immediate next sprint): Working on creating a Dockerfile using what's in https://github.com/department-of-veterans-affairs/sign-in-service.

Datadog Resource Import and Custom Pipelines:

Summary: Datadog is used for accessing metrics, logs, and application performance monitoring. Below are some of the changes made to make things such as querying logs as well as importing changes made in the GUI easier. More information can be found in the PRs linked below:

- Script to Import Datadog Resources to Maintain Changes via GUI
- Adds Custom Pipeline(s) Making Attribute Querying Easier

Note: Documentation can be found using the link Datadog Logging Documentation.

Who is impacted: Engineers (backend, front-end, devops, security Engineers and/or anyone who needs to see under the hood of SiS).

What are the benefits: Allows for easier querying of attributes in Datadog as well as streamlines the process for importing Datadog changes that are done via the GUI maintaining their history via Terraform.

Login.gov Adoption Web Comment Analysis:

Summary: There was feedback provided in the comment section of the VA news/blog post regarding the release of Login.gov to express their feelings on Login.gov, the act of logging in, and credential providers. An analysis of the comments on the Login.gov Provides Access to VA Digital Products, Services article was completed. The feedback could be valuable, so long as the limitations of this data are taken into consideration.

Who is impacted: UX, PMs, Partner credential providers.

What are the benefits:

- UX could use this as an opportunity to conduct further research in order to gather more information about these pain points: better understand negative impressions of new CSP, to gather data about which CSPs Veterans are experiencing issues with the most, and how to define login "ease of use" for the Veteran population.
- UX and Product could also use this as an opportunity to collaborate with content writers and improve FAQs and create self-service documentation.
- There is also an opportunity to work with support teams to gather additional feedback on the issues summarized in the analysis. We want to know if we are receiving similar comments.

What's next (most immediate next sprint):

- DS Logon Migration Research
- In Person Proofing Research

Sign in Service (SiS) Enhancements:

Summary: Major changes include automatic refresh for Flipper UI and an overhaul in authenticating with the Sign in Service via Call to Actions, Sign in buttons, and next redirecting.

Who is impacted: Engineers and Product Testers.

What are the benefits:

Engineers and Product Testers will no longer be signed out after 5 minutes when trying to enable/disable Flipper toggles.

Buttons and links used to authenticate on VA.gov will have the same functionality across the board when Sign in Service is rolled out.

Why did we make the change: To ensure continuity and modernization of VA.gov functionality.

Version, and planned release dates: First full week of March 2023.

What's next (most immediate next sprint):

Mocked Authentication integration.

Logout of all devices (requires UX research & Collab Cycle before integration).

Frontend Configs for Mocked Authentication:

Summary: Adding configurations for Mocked Authentication and creation of Mocked Authentication child route.

Who is impacted: Engineers and Product Testers

What are the benefits: Engineers and Product Owners will be able to navigate to a Mocked Authentication route on the Unified Sign in Page (USiP) to test the new functionality.

Why did we make the change: To allow Engineers and Product Owners to focus more on their applications rather than the implementation of sessions.

Version, and planned release dates: Next sprint (Early March 2023)

What's next (most immediate next sprint): Integrate Mock Authentication with backend.

Notable Documentation Updates:

Documentation describing persistent data related to the authenticated user and the credential they authenticate with:

(User Model) Specific attributes stored on the persistent User Account and Credential data: user attributes.

(User Model) Creation flow for persistent User Account and Credential data: creation flow.

Product Managers Update:

Onboarded five (5) new employees to the Identity Core team, one Backend Engineer remaining to onboard.

Met with stakeholders to identify top priorities for Identity team tasking.

Began creating tickets and epics to align work with top priorities.

Constructed year out product roadmap for Identity team.

Issues/Limitations/Challenges:

Everything is running smoothly this sprint as planned.

How to provide feedback:

Open a feature request ticket

Basic feedback can be discussed within the Slack

APPENDIX A- Ticket Issues Completed

- ***Removed virtual_agent_user_access_records (#11890)*** - This table and model were not used and were identified for removal. It also would have attempted to store SSN unencrypted, so it was essential to remove.
- ***Created configuration table for sign in service client configuration (#11861)*** - Sign in Service was using a set of hard coded hashes that describe what we should do with different 'client' configurations. We were rapidly hitting the limit of this implementation, as we will soon add two more client_ids, and we only want them on certain stacks, so set up a database-driven approach.
- ***Created Gem modules directory for Mocked Authentication (#53905)*** - in order to properly have a frontend component for Mocked Authentication, we needed to mount the Mocked Authentication functionality as a Gem or Engine. We created a module in vets-api for mocked-authentication that the backend and frontend components can go into.
- ***Completed write up for SiS Adoption (#53870)***
- ***Create SiS Infrastructure Repository (#54122)*** - In order to prepare for the eventual deployment of SiS to EKS, we created a repo that will house the respective infrastructure resources.
- ***Created plan for Sign in Service sample app (#52961)***
- ***Created new Sign in Service client_id for Mocked Authentication (#53052)***
- ***Created docker localhost setup and health check for example app (#53993)*** - created a Dockerfile that an Engineer can run and use to easily set up the example app. Additionally, a health check route should be set up for compatibility.
- ***Create Continuous Integration settings that must pass before PRs can be merged (#53989)*** - Created a setting in which PRs in the Sign in Service Example Application repo should not be allowed to merge to master unless:
    - An appropriate review is made (this should already be implemented in another ticket).
    - Continuous Integration tests pass. This should include an Rspec run at minimum. We may need to identify if other checks need to be added.
- ***Use StatsD in UserAcceptableVerifiedCredentialCountsJob instead of Rails.Logger (#53525)*** - Logs can be parsed and added to facets and measures in Datadog but it is not in the way we desire. Using StatsD instead of Rails.Loggger. We updated UserAcceptableVerifiedCredentialCountsJob to use StatsD for each query and added a count of removed/decreased to no_avc_and_no_ivc.
- ***Create public repository for Sign in Service Example App (#53987)*** - Created a functionality where an arbitrary Github user should be able to navigate to a public repository describing a Sign in Service Client Application.
- ***Update UserAcceptableVerifiedCredentialCountsJob to use StatsD (#11780)*** - Created a job in which we can run Sidekiq cron job daily to set StatsD gauges with user_avc totals. User_avc_totals.{provider}.{scope}.total and use StatsD gauge for user_avc Metrics
- ***Create Sinatra framework for Sign in Service Client app (#53988)*** - Created a Sinatra framework implementation that we can leverage the rest of the Sign in Service Client app functionality onto. This framework should allow for API calls to be made and Simple views accessible at the application root.