Project Plan - aries-vcx based message mediator service

Abstract

Mediator service is like a drop box / post box that Aries agents can use to receive and store encrypted messages in their stead.

Basic flow:
- people can send you messages at the mediator address if you are not always available to receive messages
- you can pick them up later from mediator service using pick up protocol

Mediator/ Agency are useful for scenarios like mobile phone agents which can't always receive messages. Due to reasons including:
  - network interruption
  - sleep /power off
  - power conservation optimizations

There exist a agency/mediator service for Aries, but it has limitations:
  - Written in NodeJS. A Rust based mediator service would be desirable for better integration with vcx
  - Does not support pick up protocol (uses custom data exchange)

A new Rust based Aries mediator service would offer.
  - High performance
  - Horizontal scalability
  - Simpler and more robust integration with Aries-vcx
  - Support for newer standard protocols like Pick Up protocol

Deliverables

Main Deliverables:
- Specialized high performance Aries agent (mediator to-be) that can store “forward messages” received through connection protocol.
- Pickup protocol data structs and impl inside aries-vcx
- Mediator service that can answer pick up protocol requests to authenticated peers
- CI to build and test on new commits.

Extra Deliverables:
- CI to make releases based on tags and provide binaries or docker image in Github's release section.
- (stretch goal): design/document mediator service component that can notify registered peers on inbox message available.

Project Plan

Stage 1: HTTP Server
- Research and choose rust based web framework, suitable for project goals
- Implement simple http service with some endpoints
- Include logging library (env_logger is popular choice)
- Include E2E test - make request against the endpoints, expect appropriate response
- CI: Lint, build, run & test each commit

Stage 2: Storage
- Understand pick-up protocol specification
- Based on the specification, design trait for retrieving messages and storing messages
- Provide mysql implementation
- Include integration tests
Stage 3: Integrate Aries-vcx
- Include aries-vcx as project dependency
- Initialize aries-vcx library (create wallet)
- Add endpoint for clients to fetch connection invitation (built using aries-vcx)

Stage 4: Support establishing didcomm connections
- Decide if there is need to alter storage traits, or to just create new implementations, and act accordingly.
- Implement endpoint to fetch mediator's connection invitation
- Support establishing didcomm connection between client and mediator using aries-vcx Connection state machine
- Create e2e test for establishing didcomm connection between test client and the mediator

Stage 5: Receiving and storing didcomm messages
- Enable receiving messages (mediation) on behalf of a connected client (which established connection and requested mediation - stage 4)
- Add mediator e2e test where mediator receives and stores didcomm message

Stage 6: Add pickup protocol support
- Add pick-up protocol data model to aries-vcx messages crate
- Add pick-up protocol messages parsing to mediator
- Add mediator e2e test performing message pick-up

Stage 7: Bonus - websocket
- Implement websocket interface for pickup protocol (think can be broken down further if we start working on this)

Stage 8: Bonus - notifications
- Implement pub/sub mechanism in distributed environment to assure instant notification of connected websocket clients about a newly received message (think can be broken down further if we start working on this)

Milestones

Evaluation 1 (July 11):
- Development environment setup
- Rust learning, http framework research
- Evaluation of Stage 1

Evaluation 2 (August 22):
- Evaluation of Stage 2, 3, 4

Evaluation 3 (October 6):
- Evaluation of Stage 5, 6

Evaluation 4 (November 14):
- Evaluation of Stage 7 / 8 or overall solution polishing & refactoring
Timeline

<table>
<thead>
<tr>
<th>Week</th>
<th>Activity</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 1 - June 30</td>
<td>Onboarding, research, rust learning</td>
<td></td>
</tr>
<tr>
<td>July 1 - July 11</td>
<td>Stage 1 (server)</td>
<td></td>
</tr>
<tr>
<td>Evaluation 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>July 11 - July 25</td>
<td>Stage 2 (storage)</td>
<td></td>
</tr>
<tr>
<td>July 26 - August 2</td>
<td>Stage 3 (aries-vcx as dependency)</td>
<td></td>
</tr>
<tr>
<td>August 3 - August 22</td>
<td>Stage 4 (didcomm connections)</td>
<td></td>
</tr>
<tr>
<td>Evaluation 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>August 23 - September 5</td>
<td>Stage 5 (Receiving&amp;storing didcomm messages)</td>
<td></td>
</tr>
<tr>
<td>September 6 - October 3</td>
<td>Stage 6 (Add pickup protocol)</td>
<td></td>
</tr>
<tr>
<td>Evaluation 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>October 4 - November 14</td>
<td>Refactoring / Stage 7, 8 / Other improvements / catch-ups</td>
<td></td>
</tr>
<tr>
<td>Evaluation 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Future plans

Design and document ways in which persistent connection based notification service could be integrated into mediator service project.

Benchmark the service.

Add support for did:peer:4 based did exchange.

Participants (Mentors/Mentees)

Mentors

<table>
<thead>
<tr>
<th>Name</th>
<th>Hyperledger Discord</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patrik Stas</td>
<td>Patrik Stas#7722</td>
<td>Absa Group</td>
</tr>
<tr>
<td>Bogdan Mircea</td>
<td>bobozaur#5997</td>
<td>Absa Group</td>
</tr>
<tr>
<td>Miroslav Kovar</td>
<td>mirgee#3763</td>
<td>Absa Group</td>
</tr>
<tr>
<td>George Mulhearn</td>
<td>gmulhearn#0356</td>
<td></td>
</tr>
</tbody>
</table>

Mentee

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naian G</td>
<td><a href="mailto:ng.p.ref+hlfwiki@mailbox.org">ng.p.ref+hlfwiki@mailbox.org</a></td>
</tr>
</tbody>
</table>

Processes

Internal communication and updates:

- Slack is used for regular communications and updates
- Weekly Zoom meetings are scheduled to discuss the progress, blocks, questions, advice
Biweekly mentee-mentee meetings for knowledge sharing, active collaboration.

Mentee maintains a blog with frequent updates:

- Aries VCX Diaries

Other useful information can be found on this page

- Learnings and Deliverables - aries-vcx based Aries Mediator Project
- Links: Useful research, analysis and general guide.