PI Planning

PI Objectives

Continuous Improvements:

Trainings, Standups, One-to-one meetings, Performance reviews and so

Features	Enhancements	Continuous Improvements
• Visualization for the current policies,roles,permissions • Real Time changes for the policies,roles,permissions • Roles and their privileges • Resources & Permissions mapping	• Deployment of the service on already running services • Envoy exploration • Rollback to old versions • Policies Code generation to rego ◦ RBAC ◦ ABAC ◦ PBAC	• Stand-up ◦ 12:30 pm → 1:30 pm • Mentors availability ◦ 4 → 5 pm

Breakout

We have decided to use the beta three point system to estimate the time of tasks in hours. Please find the individual expected hours in the Github Projects page

Sprint 1(Discovery) (8 Aug -> 14 Aug)				
Objective	Objective in Detail	Capacity Percentage	Risks	Dependencies
Team Capacity: 5 days=> man-days = 2*5 = 10 days				
Mini Task	you have 3 roles (from your choice) and you have a list of permissions each role has access to some of these permissions your opa server accepts a role and a permission as input your opa server			Research about OPA & Rego

	has an allow rule that determines whether the user is allowed or not to this permission Bonus: what if the opa server receives an http request instead of {role,permission}			
Brainstorm in achieving the dynamic access management in real time • how to solve the issue of reflecting changes in realtime server in the code • how to make automated tests for the new/updated policies & permissions • auto-upload the predefined/saved policies to the server • how to secure the opa service			• Low experience in OPA • May need more time than the discovery sprint	
Research of market in using OPA for dynamic access management				
Features & Stories definition with their timeline				Brainstorming tasks

Sprint 2 (15 Aug -> 27 Aug)				
Objective	Objective in Detail	Capacity Percentage	Risks	Dependencies
Team Capacity: 2		87/84 = 1.03		

individuals * 6 hours * 7 days = 84 hours				
UI design for the frontend	Before working on the frontend, we need to have a design ready with the shared components defined.		Getting familiar with Figma may take longer than expected and hinder the progress of actual development	Knowledge of Figma and UI/UX
Visualization for the current policies,roles,permissions	This objective requires us to visualize the roles, the scopes, and the permissions available in the system. This requires us to settle on a design to show them to the user		• Data Representation: Designing an effective and understandable way to represent complex policy, role, and permission relationships visually	UI design
Real Time changes for the policies,roles,permissions	This objective requires a frontend to allow editing for select types of users. The change these users want should be reflected in realtime to the policy engine. This also requires running automated tests whenever policy is changed		• Third-Party Dependencies • Rollback Strategy • Automated Test Failures • Data Conflicts	Visualization of roles,policies, and permissions

Sprint 3 (28 Aug -> 10 Sep)				
Objective	Objective in Detail	Capacity Percentage	Risks	Dependencies
Team Capacity: 2*10				
Configuring ABAC policies through the GUI	This objective requires a user-friendly interface that allows a user to define his ABAC policies through the GUI. This will further require handling how to define the attributes through the GUI			Complete implementation of RBAC policy system
Provide an IDE for code editing in the dashboard	This objective requires adding an IDE to the dashboard that will allow an experienced user to write policies as code		• High risk of being postponed till sprint 4	Realtime changes working correctly

Sprint 4 (11 Sep -> 24 Sep)				
Objective	Objective in Detail	Capacity Percentage	Risks	Dependencies
Team Capacity: 2*10				
Login and SSO	After coordination with the Infrastructure team, we need to integrate SSO into our project		Retrieving roles from the integration may not be as smooth as expected	Proper mocking of role retrieval on our side
Bug Solving	Go over the project modules that each module is working as expected as part of the entire product			Each individual module working, tested, and documented

Risk Roaming

1. Resolved

The determined risk isn’t a threat at this time. No action is required.

2. Owned

Because the risk can’t be resolved during the meeting, a team member is chosen to “own” the resolving of the risk. This team member is responsible for the management of the risk.

3. Accepted

The risk in this category can’t be resolved, and the only action to take is to accept it as it is and to deal with the risk as necessary.

4. Mitigated

A plan has to be formulated in order to eradicate the threat of the risk

Resolved • Data representation: Drawing inspiration from the market survey should resolve this risk	Owned • UI design taking longer than expected: After discussing the open source solutions first, Gamal will handle the UI design as agreed upon
Mitigated	Accepted • Rollback strategy and automated test failures: Will need research as part of the task

PI Objectives and Confidence Levels

Confidence Vote:

Is an activity done in the Program Increment Planning session after the risk assessment has been done.

How is it done:

Team members raise their hands and with their fingers show their vote of confidence. There are five levels of confidence, each of which can be represented by the fingers where 1 finger being the lowest and 5 fingers showing the highest level of confidence. If the average number of fingers is 3, then the PI Objectives are accepted and go onwards with the plan

PI Objective	Avg, Confidence Level	Total