VOTING Stimmunterlagen Online

Authentication Model

Author	Abraxas Informatik AG
Classification	public
Version	1.1
Date	09.10.2024

Für die digitale Schweiz. Mit Sicherheit.

abraxas

Contents

1. VOTING IAM

The VOTING IAM software developed by Abraxas is used as the IAM solution. This includes user management, role management and the use of VOTING IAM as an identity provider.
The setup of applications, roles and clients is done by VOTING IAM administrators (employees of Abraxas Informatik AG). However, user and rights management can be delegated to responsible persons per tenant. They can, for example, create new users or change the permissions of a user within their tenant. It is not possible for such tenant administrators to authorize users for other tenants (or applications or roles of other tenants). Only VOTING IAM administrators can perform this kind of advanced privileged actions.
VOTING IAM does fully support the OpenID Connect (OIDC) standard.

2. Procedure

2.1 Login Flow

The user login flow in the frontend uses the authorization code flow with PKCE (Proof Key for Code Exchange - https://tools.ietf.org/html/rfc7636). After successful authentication, the frontend application (user agent) is provided with an access token - JSON Web Token (JWT) - that can be used to authenticate against the backend services. Among with the access token the user receives further identity or authorization information from the identity and permission services used by the frontend for authorization and visual representation purposes.

2.2 Logout Flow

After a successful logout, all tokens and other user specific information is deleted from the client's user agent (frontend) and revoked on the identity provider. When the user logs out, their active session cookies, access tokens, and refresh tokens are automatically invalidated, enhancing security by preventing unauthorized access to their account.

2.3 Refresh Flow

The frontend uses the token refresh flow to prevent long-lived access tokens. Refresh tokens are valid for 30 days and revoked at any time the user applies security specific operations such as logout, password change and other operations.

2.4 Access Token Validation

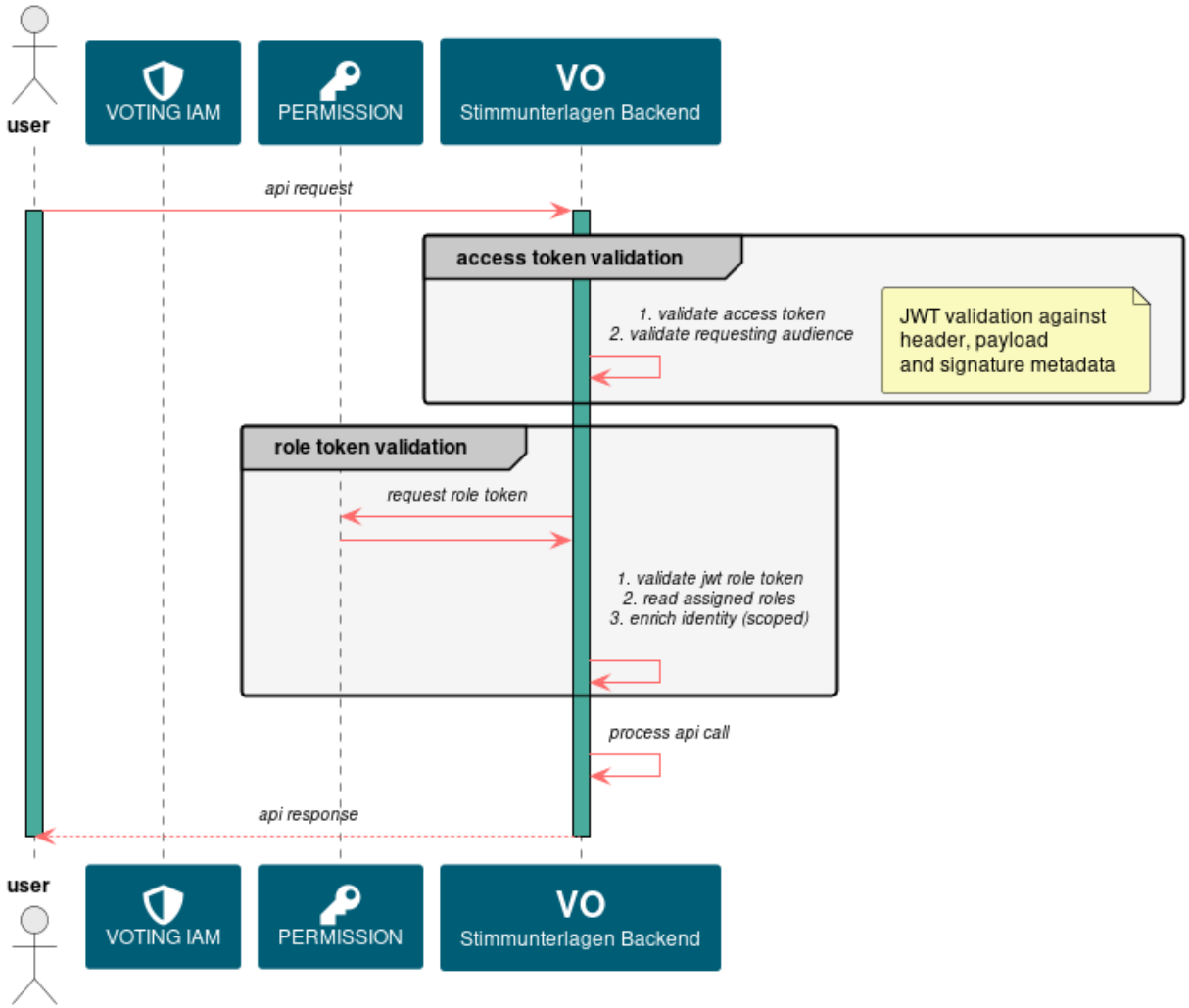
The VOTING Stimmunterlagen backend requires the following information as an HTTP header to be able to fully authenticate a user:

HTTP Header	Description	Example value
Authorization	Access token of the user	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc...
x-app	Application(s) to be accessed	VOTING-STIMMUNTERLAGEN
x-tenant	Tenant, in which context the API call should be executed. The x-tenant header corresponds to the selected tenant within the web application.	132543547687919

The steps to a successful authentication are as follows:

1. The user's access token is checked for validity (not expired, valid audience, etc.).
2. The requested application(s) scopes are verified to see if they are included in the configured whitelist of audiences.
3. The access token and the application(s) are sent to the token endpoint.
4. In response, a role token is returned containing the user's roles for the listed applications.
5. The role token is checked for validity.
6. The roles are extracted from the token and matched with the corresponding application as defined by x-app (roles in the role token are only filtered by application, not by tenant).
7. If all validation succeeded, the user is considered authenticated.
8. Users, tenants and roles are saved for the current request to prove valid authentication against further evaluation logic.

2.4.1 Validation Flow



2.5 Authentication

IAM Instance	Application Scope	Authentication Flow	Client App	Requesting Application Audience(s)	Use Case(s)
SHARED	VO Stimmunterlagen Service	Client Credential	VOTING-STIMMUNTERLAGEN	n/a	• Authenticate for CONNECT uploader service endpoint.
VOTING	VO Stimmunterlagen WebApp	Code with PKCE	VOTING-STIMMUNTERLAGEN	VOTING-STIMMREGISTER	• Query permission and identity information for visualizing user context information • Authorize for exporting VOTING Stimmregister versions through the backend API. • Call protected backend services through gRPC and/or REST endpoints.
VOTING	VO Stimmunterlagen Service	Client Credential	VOTING-STIMMUNTERLAGEN	n/a	• Query permission and identity information to authenticate and authorize incoming requests.