

VOTING Wahlvorschlag

Authorization Model

Author	Abraxas Informatik AG
Classification	public
Version	1.1
Date	September 22, 2025

Contents

1.	IAM System	3
1.1	Applications	3
1.2	Roles	3
1.2.1	Permission Inheritance	3
1.3	Tenants	4
1.4	Users	4
2.	Authorization Table	5
2.1	API Specification V1	5
2.1.1	Conditions	5
2.1.2	Authorization.....	6

1. IAM System

1.1 Applications

The following applications are available for VOTING Wahlvorschlag:

Application ID	Description
EAWV	Application for managing master data for voting eligibility to be able to obtain information on voting rights daily.

1.2 Roles



A role is always linked to a tenant context and therefore requires explicit authorization to access tenant data. Exceptions are explicitly mentioned in the description of the role. The following roles are available for the application VOTING Wahlvorschlag:

Application ID	Role	Description
EAWV	Wahlverwalter	<p>A user with the role <i>Wahlverwalter</i> is responsible for preparing and validating elections and list proposals, as well as managing master data through the in-app user interface. This includes authorizing political parties and users, requesting new political tenants, creating or deleting users and modifying user authorization scopes within specific political parties.</p> <p>Assignment of the <i>Wahlverwalter</i> role is restricted exclusively to users of the superior authority.</p>
	Benutzer	<p>A user with the role <i>Benutzer</i> is responsible for creating new election lists and capturing candidates as part of an election proposal list. In coordination with the superior authority, the lists are released for review as part of a lifecycle process. After approval, the lists are exported and used either for signing or for importing into VOTING Basis based (subsystem) on the eCH-0157 standard.</p> <p>Assignment of the <i>Benutzer</i> role is restricted exclusively to users of the political party.</p>

1.2.1 Permission Inheritance

The roles do not inherit from each other. Every role is explicitly authorized for the desired actions.

✔ Indicated that this role applies permissions directly.

	Wahlverwalter	Benutzer
Wahlverwalter		-
Benutzer	-	

1.3 Tenants

In the context of VOTING, the tenant is reflected by a political unit such as a superior authority (canton) or party (Volkspartei, Einzelperson). An example configuration is provided below:

Tenant	Application ID	Tenant Id	Available Roles	Unit	Responsible Authority (Parent Tenant Id)
Canton A <i>Kanton A</i>	EA WV	549462173064135111	Wahlverwalter	Superior authority	-
Party A1 <i>Partei A1</i>	EA WV	549462118219416450	Benutzer	Party	549462173064135111
Party A2 <i>Partei A2</i>	EA WV	549462172057502594	Benutzer	Party	549462173064135111

1.4 Users

User accounts for the superior authority are managed directly through SECURE Connect, while accounts for parties are handled via the in-app user interface by the superior authority's designated responsibilities. An example configuration is provided below:

User	Tenant	Application ID	Role ID
Barbara Burger	State Chancellery of the Canton A	EA WV	Wahlverwalter
Franz Fürer	Party A1	EA WV	Benutzer
Laurenz Marty	Party A1	EA WV	Benutzer
Laurenz Marty	Party A2	EA WV	Benutzer

2. Authorization Table

2.1 API Specification (Version 1)

This chapter provides an overview of all actions that can be executed on the frontend and API level. The listed actions can only be executed when all conditions are fulfilled. If a condition is empty, it is considered fulfilled.

2.1.1 Conditions

The following conditions are re-used for several API calls. They are described once and referenced by corresponding keys in the chapter *Authorization*.

Reference Key	ACL Expression (pseudo code)	Comments
HasReadAccessToList	(IsWahlverwalter && IsOwnerOfReferencedElection) (IsResponsiblePartyTenantForList && (IsAssignedToList IsCreator))	A user has access to a list if the referencing election is either owned by the calling tenant with role <i>Wahlverwalter</i> or the calling party with role <i>Benutzer</i> is assigned as a responsible user or is the creator of the list himself. Connection between an election and list: An election is always managed by the superior authority and the lists are managed by the parties. Several lists can be assigned to an election.
HasWriteAccessToList	HasReadAccessToList && (IsWahlverwalter ListIsNotLocked)	As a user with role <i>Wahlverwalter</i> , the list can be modified at any time, while users with role <i>Benutzer</i> may only modify a list if the list status is not locked.
HasReadAccessToElection	IsWahlverwalter && IsOwnerOfElection	A user has read access to a list if the election is owned by the calling tenant with role <i>Wahlverwalter</i> .
HasWriteAccessToElection	(HasReadAccessToElection) (IsPartyChildOfOwningElectionTenant)	A user has write access to a list if the election is owned by the calling tenant with role <i>Wahlverwalter</i> or the calling party is a child of the tenant owning the election.

2.1.2 Authorization

Action	API	Conditions		Comments
{Name[de]} {service}	{method[en]}	User role	Additional Conditions (ACL)	
Wahlen ElectionController				
Wahl lesen	GetElections	Wahlverwalter Benutzer	HasReadAccessToElection	
Wahlen lesen	GetElection	Wahlverwalter Benutzer	HasReadAccessToElection	
Wahl erstellen	CreateElection	Wahlverwalter	-	
Wahl aktualisieren	UpdateElection	Wahlverwalter	HasReadAccessToElection	
Wahl löschen	DeleteElection	Wahlverwalter	HasReadAccessToElection	
Wahlkreis (Stammdaten) DomainOfInfluenceController				
Wahlkreise lesen	GetDomainOfInfluences	Wahlverwalter Benutzer	TenantId == ParentTenantId TenantId == CurrentTenantId	
Wahlkreis lesen	GetDomainOfInfluence	Wahlverwalter Benutzer	TenantId == ParentTenantId TenantId == CurrentTenantId	
Wahlkreis erstellen	CreateDomainOfInfluence	Wahlverwalter	-	
Wahlkreis aktualisieren	UpdateDomainOfInfluence	Wahlverwalter	TenantId == CurrentTenantId	
Wahlkreis löschen	DeleteDomainOfInfluence	Wahlverwalter	TenantId == CurrentTenantId	
Infotexte (Stammdaten) InfoTextController				
Infotexte lesen	GetInfoTexts	Wahlverwalter Benutzer	TenantId == ParentTenantId TenantId == CurrentTenantId	
Infotext lesen	GetInfoText	Wahlverwalter Benutzer	TenantId == ParentTenantId TenantId == CurrentTenantId	
Infotexte nach Schlüssel lesen	GetInfoTextByKey	Wahlverwalter Benutzer	TenantId == ParentTenantId TenantId == CurrentTenantId	

Action {Name[de]} {service}	API {method[en]}	Conditions		Comments
		User role	Additional Conditions (ACL)	
Infotext erstellen oder aktualisieren	CreateOrUpdateInfoText	Wahlverwalter	HasReadAccessToElection	
Infotexte erstellen oder aktualisieren	CreateOrUpdateInfoTextBatch	Wahlverwalter	HasReadAccessToElection	
Einstellungen (Stammdaten) SettingController				
Einstellungen lesen	GetSetting	Wahlverwalter Benutzer	TenantId == ParentTenantId TenantId == CurrentTenantId	Information for ballots, additional information and client settings
Einstellungen aktualisieren	UpdateSetting	Wahlverwalter	TenantId == ParentTenantId TenantId == CurrentTenantId	
Parteien (Stammdaten) PartiesController				
Parteien lesen	GetParties	Wahlverwalter	EAWV.Parent = CurrentTenantId	
Partei erstellen	CreateParty	Wahlverwalter	-	
Partei löschen	DeleteParty	Wahlverwalter	PartyToDelete[EAWV.Parent] = CurrentTenantId	
Benutzer (Stammdaten) UsersController				
Alle Benutzer lesen	GetUsers	Wahlverwalter	EAWV.Parent = CurrentTenantId	
Benutzer für Mandant lesen	GetUsersForTenant	Wahlverwalter Benutzer	tenantId = CurrentTenantId (IsWahlverwalter && EAWV.Parent = CurrentTenantId)	
Benutzer lesen	GetUser	Wahlverwalter Benutzer	tenantId = CurrentTenantId (IsWahlverwalter && EAWV.Parent = CurrentTenantId)	
Benutzer erstellen	CreateUser	Wahlverwalter	AssertChildParent	
Benutzer aktualisieren	UpdateUser	Wahlverwalter	AssertChildParent	
Benutzerrechte löschen	RemoveUserAccess	Wahlverwalter	AssertChildParent	Only the rights in the VOTING Wahlvorschlag context are deleted for the user. The user

Action {Name[de]} {service}	API {method[en]}	Conditions		Comments
		User role	Additional Conditions (ACL)	
				itself remains in the IAM system.
Wahldokumente BallotDocumentController				
Dokument lesen	GetDocument	Wahlverwalter Benutzer	Document.Election.TenantId == ParentTenantId Document.Election.TenantId == CurrentTenantId	
Dokument erstellen	CreateDocument	Wahlverwalter	HasReadAccessToElection	
Dokument löschen	DeleteDocument	Wahlverwalter	HasReadAccessToElection	
Liste ListController				
Listen lesen	GetLists	Wahlverwalter Benutzer	HasReadAccessToList	
Liste lesen	GetList	Wahlverwalter Benutzer	HasReadAccessToList	
Liste erstellen	CreateList	Wahlverwalter Benutzer	HasWriteAccessToElection	
Liste aktualisieren	UpdateList	Wahlverwalter Benutzer	HasWriteAccessToList	
Listen-Lifecycle aktualisieren	UpdatePartialList	Wahlverwalter Benutzer	HasWriteAccessToList	
Liste löschen	DeleteList	Wahlverwalter Benutzer	HasWriteAccessToList	
Listen-Kommentare ListCommentController				
Kommentar lesen	GetComments	Wahlverwalter Benutzer	HasReadAccessToList	
Kommentar erstellen	CreateComment	Wahlverwalter Benutzer	HasReadAccessToList	

Action {Name[de]} {service}	API {method[en]}	Conditions		Comments
		User role	Additional Conditions (ACL)	
Kommentar aktualisieren	UpdateComment	Wahlverwalter Benutzer	HasReadAccessToList	
Kommentar löschen	DeleteComment	Wahlverwalter Benutzer	HasReadAccessToList	
Listen-Verbindung ListUnionController				
Unterlistenverbindung erstellen oder aktualisieren	AddOrUpdateListSubUnion	Wahlverwalter	HasReadAccessToElection	
Listenverbindung erstellen oder aktualisieren	AddOrUpdateListUnion	Wahlverwalter	HasReadAccessToElection	
Unterlistenverbindung löschen	DeleteFromSubUnion	Wahlverwalter	HasReadAccessToList	
Listenverbindung löschen	DeleteFromUnion	Wahlverwalter	HasReadAccessToList	
Listen-Exports ListExportController				
Export ausführen	Export	Benutzer	HasReadAccessToList && ExportType != "WabstiCandidates"	Available exports: Formular Kandidaturen (PDF)
		Wahlverwalter	HasReadAccessToList	Available exports: Formular Kandidaturen (PDF), Export Wabsti (XLSX)
Wahlen-Exports ElectionExportController				
Export ausführen	Export	Benutzer	HasReadAccessToElection && ExportType NOT IN ["FederalChancellery", "EmptyCandidates", "EmptySignatories"]	
		Wahlverwalter	HasReadAccessToElection	
Kandidat CandidateController				
Kandidaten lesen	GetCandidates	Wahlverwalter Benutzer	HasReadAccessToList	
Kandidaten erstellen oder aktualisieren	UpdateAllCandidates	Wahlverwalter Benutzer	HasWriteAccessToList	

Action <code>{Name[de]} {service}</code>	API <code>{method[en]}</code>	Conditions		Comments
		User role	Additional Conditions (ACL)	
Wahlkreis (Wahlen) DomainOfInfluenceElectionController				
Wahlkreis erstellen	CreateDomainOfInfluenceElection	Wahlverwalter	HasReadAccessToElection	
Wahlkreis aktualisieren	UpdateDomainOfInfluenceElection	Wahlverwalter	HasReadAccessToElection	
Wahlkreis löschen	DeleteDomainOfInfluenceElection	Wahlverwalter	HasReadAccessToElection	