Broadcast

Requirements, User Experience, and Architecture Specification

Dr. Ralf S. Engelschall

Version: 0.9.0
Created: 2026-06-18
Modified: 2026-10-04

Broadcast is a Web application which complements a broadcasted video stream with several features to achieve an enterprise-grade "townhall" event experience. It embeds the video stream with an external video player and surrounds it with additional user interface controls for chat and stream control. This allows the attendees to select between several stream variants, mainly meant to carry different language translations of the audio content.

Broadcast offers a chat as a backchannel from the audience to the presenters. It allows an attendee to send messages and see messages sent by other attendees or a moderator. Sent messages may be reviewed by an artificial or human operator before they are allowed to spread to the audience. Attendees may like specific messages to signal their support. Additionally, a moderator can select questions coming from the audience which are then forwarded to the video production studio to be answered by the presenters with a small time delay. **Broadcast** can also optionally embed an external application, for instance to provide live voting. Here the attendee's choice is transferred back to the video production studio, aggregated there and presented to moderators and presenters as the audience's opinion.

Broadcast optionally enforces a closed group policy by only allowing a certain list of people (identified by their e-mail addresses) to join. It seamlessly switches to an alternative video stream provider if requested by the people overseeing the event production. This helps to overcome video quality or scaling issues without interrupting the broadcasting of the program. It supports an audience of up to 10,000 attendees, specifically employing a fast join process and swift delivery of chat messages.

Broadcast provides several statistics to assess the technical reception of the event program by the audience (number of viewers, used client setup, event-wide counts). None of this data can be linked to specific persons. **Broadcast** complies with data protection rules by reliably deleting personal data after the event. Chat content including likes may be saved for later review (e.g. for assessing information needs, detecting unanswered questions), but here, too, no link to specific persons is ever provided.

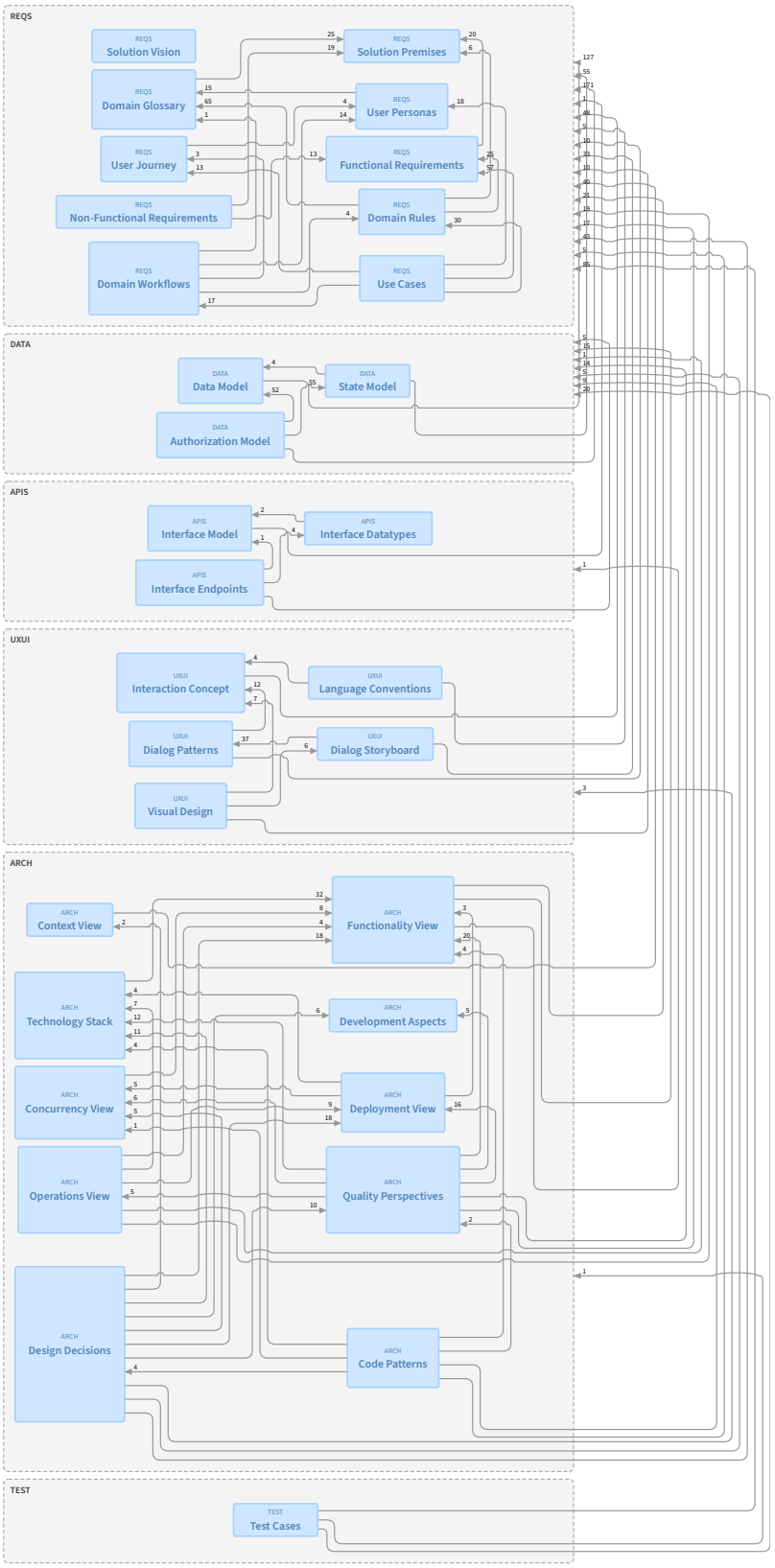
Table of Contents

01 ● **REQS:** Solution Vision 6
02 ● **REQS:** Solution Premises 7
03 ● **REQS:** Domain Glossary 11
04 ● **REQS:** User Personas 20
05 ● **REQS:** User Journey 24
05.01 ● **JOURNEY:** Attend a Broadcast event 24
05.02 ● **JOURNEY:** Operate a Broadcast Event 25
05.03 ● **JOURNEY:** Moderate a Broadcast Event 25
06 ● **REQS:** Functional Requirements 26
07 ● **REQS:** Non-Functional Requirements 39
08 ● **REQS:** Domain Rules 47
09 ● **REQS:** Domain Workflows 56
09.01 ● **WORKFLOW:** Run Broadcast Event 56
10 ● **REQS:** Use Cases 59
10.01 ● **USE-CASE:** Join Event and Watch Stream 60
10.02 ● **USE-CASE:** Authenticate via Email Token 61
10.03 ● **USE-CASE:** Ask a Question 63
10.04 ● **USE-CASE:** Chat During the Event 65
10.05 ● **USE-CASE:** Moderate and Forward Messages 67
10.06 ● **USE-CASE:** Moderate the Chat Conversation 69
10.07 ● **USE-CASE:** Steer the Presenter 70
10.08 ● **USE-CASE:** Configure the Event 71
10.09 ● **USE-CASE:** Switch Streaming Provider 73
10.10 ● **USE-CASE:** Create Event from Registration Import 73
10.11 ● **USE-CASE:** Publish, Start, and Finish Event 74
10.12 ● **USE-CASE:** Export Anonymized Event Data 75
10.13 ● **USE-CASE:** Present Forwarded Questions 76
10.14 ● **USE-CASE:** Administer Any Event 78
11 ● **DATA:** Data Model 79
11.01 ● **ENTITY:** Event 80
11.02 ● **ENTITY:** AgendaPoint 85
11.03 ● **ENTITY:** Channel 85
11.04 ● **ENTITY:** Resource 86
11.05 ● **ENTITY:** ResourceProviderParam 87
11.06 ● **ENTITY:** Role 88
11.07 ● **ENTITY:** User 89
11.08 ● **ENTITY:** Message 91

11.09 ENTITY: MessageText 93
11.10 ENTITY: QuestionTag 94
11.11 ENTITY: AuthorizationToken 95
11.12 ENTITY: SessionToken 96
11.13 ENTITY: EventStatistic 97
11.14 ENTITY: ChannelStatistic 98
11.15 ENTITY: UserStatistic 98
12 DATA: State Model 99
12.01 LIFECYCLE: Event 100
12.02 LIFECYCLE: Message 101
12.03 LIFECYCLE: AuthorizationToken 103
12.04 LIFECYCLE: UserStatistic 104
13 DATA: Authorization Model 105
14 APIS: Interface Model 127
15 APIS: Interface Datatypes 127
15.01 TYPE: LogLevel 127
15.02 TYPE: Configuration 128
16 APIS: Interface Endpoints 129
16.01 ENDPOINT: broadcast-server 129
17 UXUI: Interaction Concept 131
18 UXUI: Language Conventions 136
19 UXUI: Dialog Patterns 137
20 UXUI: Dialog Storyboard 142
20.01 STORYBOARD: Attendee Joins and Watches 142
20.02 STORYBOARD: Attendee Asks a Tagged Question 143
20.03 STORYBOARD: Attendee Reads and Posts Chat 143
20.04 STORYBOARD: Attendee Gets Support from a Moderator 144
20.05 STORYBOARD: Chat Moderator Triages Messages 145
20.06 STORYBOARD: Moderator Forwards to Presenter 146
20.07 STORYBOARD: Manager Switches Provider Live 147
20.08 STORYBOARD: Manager Imports Registration Sheet 148
21 UXUI: Visual Design 148
21.01 MOCKUP: Attendee Chat View 152
21.02 MOCKUP: Attendee Q&A View 153
21.03 MOCKUP: Attendee Support View 153
21.04 MOCKUP: Administrator Event Creation 154
21.05 MOCKUP: Chat Moderation Board 155
21.06 MOCKUP: Q&A Moderation Board 156
21.07 MOCKUP: Presenter View 157
22 ARCH: Context View 158

23 ARCH: Functionality View 161
24 ARCH: Technology Stack 168
24.01 TIER: Common 168
24.02 TIER: Client 172
24.03 TIER: Middleware 175
24.04 TIER: Server 176
24.05 TIER: Database 179
25 ARCH: Development Aspects 180
26 ARCH: Concurrency View 182
27 ARCH: Deployment View 185
28 ARCH: Operations View 191
29 ARCH: Quality Perspectives 194
30 ARCH: Design Decisions 201
30.01 DECISION: Use MQTT over WebSockets as the live transport 201
30.02 DECISION: Self-host on Hetzner instead of public cloud 202
30.03 DECISION: Privacy by design with no permanent user accounts 203
30.04 DECISION: Email one-time token as the authentication factor 203
30.05 DECISION: Decouple logical Channels from physical provider Resources 204
30.06 DECISION: Embed provider-delivered video instead of self-hosted streaming 204
30.07 DECISION: One modular service process instead of microservices 205
30.08 DECISION: TypeScript with a shared common module across client and server 206
30.09 DECISION: Vue.js single-page client with headless widgets 207
30.10 DECISION: HAProxy as a two-stage router and proxy edge 207
30.11 DECISION: Three network segments with an isolated database subnet 208
30.12 DECISION: Single-host Docker Compose instead of a cluster orchestrator 209
30.13 DECISION: PostgreSQL as the single self-hosted relational store 209
30.14 DECISION: Typed SQL builder instead of a full ORM 210
30.15 DECISION: Logical dumps instead of continuous archiving for backup 211
30.16 DECISION: Serve the static client from a CDN edge outside the data center 211
31 ARCH: Code Patterns 212
31.01 CODE-PATTERN: Declarative Permission Check 212
31.02 CODE-PATTERN: DAO per Entity 213
31.03 CODE-PATTERN: Version-Checked Write 214
32 TEST: Test Cases 215

Diagram of Contents



01 ● REQS: Solution Vision (SV) ⚓

○ PURPOSE	**Broadcast** exists to let organizations stream live video events to thousands of attendees while keeping the audience engaged through moderated chat, questions, and likes, all under full control of who may watch and interact.
○ TARGET-AUDIENCE	The solution serves event organizers (managers), live-event moderators and presenters, and large invited audiences of attendees, ranging from small internal lectures of 75 people up to digital town halls of 10,000 simultaneous attendees.
○ SOLUTION	**Broadcast** is a live-event platform that joins an embedded video player of an external streaming provider with an email-verified access control and a moderated interaction channel in a single web application.
○ VALUE-PROPOSITION	Organizers keep full control over who watches and who may interact, and receive all attendee input in a form they can moderate, forward to the presenter, and export afterwards. Attendees reach the event through a single personal link in any recent browser, without installation or accounts, and see their questions reach the presenters within seconds.
○ DIFFERENTIATORS	Compared to public streaming portals, **Broadcast** admits only invited (and optionally authenticated) attendees and enforces per-attendee access limits. It stays independent of any single streaming provider and switches providers live during a running event without user interaction. Compared to the status quo of renting a third-party event platform per event, it runs on self-operated infrastructure, keeps all personal data under the organization's own GDPR responsibility, and anonymizes it on event finish.
○ SCOPE	**Broadcast** covers anonymous and authenticated event access, video playback, moderated interaction, and event statistics. It deliberately neither produces nor encodes nor hosts the video streams itself — ingest and delivery stay with the external streaming providers — and it is neither a conferencing tool with attendee audio and video, nor a video-on-demand archive.
○ SUCCESS-CRITERIA	**Broadcast** is achieved once an organization can run its large-scale events on it end-to-end: events of up to 10,000 simultaneous attendees are carried without a stream interruption visible to the audience.
○ FUTURE-STATE	Once adopted, live events are no longer bought per event from external platforms but can be held on an organization's own channel: the attendee data never leaves its control, the streaming provider becomes an exchangeable supplier instead of the platform, and organizers reuse one familiar event setup instead of a different tool for every occasion.

02 ● REQS: Solution Premises (SP)

Registered Audience Known in Advance
TYPE: Assumption
FAILURE-LIKELIHOOD: Low
FAILURE-IMPACT: High

Attendees Read Their Email While Logging In
TYPE: Assumption
FAILURE-LIKELIHOOD: Medium
FAILURE-IMPACT: Medium

Ten Thousand Attendees Is the Upper Bound
TYPE: Assumption
FAILURE-LIKELIHOOD: Low
FAILURE-IMPACT: High

Attendee Networks Pass WebSocket Traffic
TYPE: Assumption
FAILURE-LIKELIHOOD: Medium
FAILURE-IMPACT: High

German and English Cover All Events
TYPE: Assumption
FAILURE-LIKELIHOOD: Low
FAILURE-IMPACT: Low

Streaming Providers Deliver the Video
TYPE: Dependency
FAILURE-LIKELIHOOD: Medium
FAILURE-IMPACT: High

Registration System Exchanges Attendee Data
TYPE: Dependency
FAILURE-LIKELIHOOD: Low
FAILURE-IMPACT: Medium

Email Gateway Delivers Tokens Within Minutes
TYPE: Dependency
FAILURE-LIKELIHOOD: Medium
FAILURE-IMPACT: High

Translation Service Available During Events
TYPE: Dependency
FAILURE-LIKELIHOOD: Medium
FAILURE-IMPACT: Low

EU Hosting on Self-Operated Infrastructure
TYPE: Dependency
FAILURE-LIKELIHOOD: Low
FAILURE-IMPACT: High

Connection Surge at Event Start
TYPE: Risk
FAILURE-LIKELIHOOD: High
FAILURE-IMPACT: High

Access URLs Forwarded Beyond the Audience
TYPE: Risk
FAILURE-LIKELIHOOD: Medium
FAILURE-IMPACT: Medium

Message Texts Contain Personal Data
TYPE: Risk
FAILURE-LIKELIHOOD: High
FAILURE-IMPACT: Medium

Abusive Interaction from the Audience
TYPE: Risk
FAILURE-LIKELIHOOD: Medium
FAILURE-IMPACT: Medium

● PREMISE	Properties & Description	
Registered Audience Known in Advance	Properties	
	○ TYPE	Assumption
	○ FAILURE-LIKELIHOOD	Low
	○ FAILURE-IMPACT	High
	○ CONSEQUENCE	Without an attendee list ahead of the event, no individual access URLs can be generated and returned, so the event has to fall back onto an access email pattern or onto anonymous access.
	○ MITIGATION	Access email patterns and anonymous access remain available as fallback access modes of an event.
	Description	
	The audience of an event is registered in the Event Registration System before the event takes place, so that its attendee list is available for import while the event is planned — **BECAUSE** the organization invites its events through the registration system and never on the day of the event itself.	
Attendees Read Their Email While Logging In	Properties	
	○ TYPE	Assumption
	○ FAILURE-LIKELIHOOD	Medium
	○ FAILURE-IMPACT	Medium
	○ CONSEQUENCE	An attendee without access to their mailbox at login time cannot consume the authorization token within its validity of 5 minutes and is locked out of the event.
	○ MITIGATION	Personalized and automatic access URLs let a pre-verified attendee skip the token step, and the token validity is configurable per event.

● PREMISE	Properties & Description	
	Description	
	An attendee has their mailbox at hand on the device (or next to it) while logging in, so that the one-time authorization token reaches them within its validity — **BECAUSE** the invitation itself arrives by email and attendees join from the same workplace or home context.	
Ten Thousand Attendees Is the Upper Bound ⚓	Properties	
	○ **TYPE**	Assumption
	○ **FAILURE-LIKELIHOOD**	Low
	○ **FAILURE-IMPACT**	High
	○ **CONSEQUENCE**	An event exceeding 10,000 simultaneous attendees exhausts the dimensioned relay and proxy capacity, degrading or refusing the connections of the excess audience.
	○ **MITIGATION**	The horizontal scaling of proxy, relay, and server instances leaves headroom, which is confirmed by a load test before each larger event.
	Description	
	No single event gathers more than 10,000 simultaneous attendees, the size of the largest digital town hall of the organization — **BECAUSE** the invited audience is bounded by the workforce of the organization and its registration figures.	
Attendee Networks Pass WebSocket Traffic ⚓	Properties	
	○ **TYPE**	Assumption
	○ **FAILURE-LIKELIHOOD**	Medium
	○ **FAILURE-IMPACT**	High
	○ **CONSEQUENCE**	An attendee behind a proxy or firewall which blocks long-lived WebSocket connections sees the video but receives neither the live interaction nor the configuration updates.
	○ **MITIGATION**	All connections use the standard HTTPS port with a WebSocket upgrade, and the client reports a failed real-time connection to the attendee.
	Description	
	The corporate and home networks of the attendees permit long-lived WebSocket connections over HTTPS — **BECAUSE** the live interaction and the configuration propagation are carried over MQTT-over-WebSocket rather than over plain HTTP requests.	
German and English Cover All Events ⚓	Properties	
	○ **TYPE**	Assumption
	○ **FAILURE-LIKELIHOOD**	Low
	○ **FAILURE-IMPACT**	Low
	○ **CONSEQUENCE**	An event held in a third language needs a further translation target and user interface locale, which the solution does not offer.
	Description	
	Every event is held in German or English and its audience reads at least one of the two — **BECAUSE** these are the working languages of the organization.	
Streaming Providers Deliver the Video ⚓	Properties	
	○ **TYPE**	Dependency
	○ **FAILURE-LIKELIHOOD**	Medium
	○ **FAILURE-IMPACT**	High
	○ **CONSEQUENCE**	Without a working provider resource there is no video, and the event degrades to an interaction channel without a stream.

● PREMISE	Properties & Description	
	○ MITIGATION	Every event is configured with at least one fallback provider resource, and the operators switch providers live without user interaction.
	Description	
	The external streaming providers (such as msg Filmstudio, YouTube, Cloudflare, or 3Q) ingest, encode, and deliver the video and expose an embeddable playback URL in the required 1080p30 quality — **BECAUSE** the solution deliberately neither produces nor hosts the video streams itself.	
Registration System Exchanges Attendee Data ⚓	Properties	
	○ TYPE	Dependency
	○ FAILURE-LIKELIHOOD	Low
	○ FAILURE-IMPACT	Medium
	○ CONSEQUENCE	A changed export format of the registration system breaks the attendee import, and the invitation emails cannot carry the individual access URLs.
	○ MITIGATION	The exchange uses plain Excel sheets in an agreed column layout, which the manager can adjust by hand in the worst case.
	Description	
	The Event Registration System exports the attendee registrations of an event as an Excel sheet, accepts the generated access URLs in return, and sends the invitation emails itself — **BECAUSE** the organization provisions its events from its existing registration system and the solution does not send invitations.	
Email Gateway Delivers Tokens Within Minutes ⚓	Properties	
	○ TYPE	Dependency
	○ FAILURE-LIKELIHOOD	Medium
	○ FAILURE-IMPACT	High
	○ CONSEQUENCE	A delayed or undelivered token email locks the attendee out, as the token expires after 5 minutes, and at the event start thousands of attendees are affected at once.
	○ MITIGATION	The token validity is configurable per event, and automatic access URLs let pre-verified attendees enter without a token email.
	Description	
	The external email gateway accepts the one-time authorization token emails at the burst rate of an event start and delivers them to the attendees within a minute — **BECAUSE** the email token is the first-factor authentication of the solution.	
Translation Service Available During Events ⚓	Properties	
	○ TYPE	Dependency
	○ FAILURE-LIKELIHOOD	Medium
	○ FAILURE-IMPACT	Low
	○ CONSEQUENCE	Without the service, messages remain in their source language only, so the attendees of the other language read untranslated chat and questions.
	○ MITIGATION	Messages are stored and shown in their source language regardless, and the translation is added asynchronously once the service responds.
	Description	
	The external AI/LLM service translates message texts between German and English on the fly within a few seconds — **BECAUSE** the two-language audience expects to read chat and questions in their own language.	

● PREMISE	Properties & Description	
EU Hosting on Self-Operated Infrastructure ⚓	Properties	
	○ **TYPE**	Dependency
	○ **FAILURE-LIKELIHOOD**	Low
	○ **FAILURE-IMPACT**	High
	○ **CONSEQUENCE**	Hosting outside the EU or on a public cloud without an EU data-processing agreement violates the GDPR obligations of the operator and voids the cost advantage of the solution.
	○ **MITIGATION**	The hosting contract fixes the data center location, and the containerized deployment is portable to any other EU hoster.
	Description	
	The Hetzner data center in Nürnberg, Germany, provides the virtual machines the solution runs on, within the EU and at a price below the equivalent public cloud offerings — **BECAUSE** self-hosting within the EU is both the legal and the economic foundation of the solution.	
Connection Surge at Event Start ⚓	Properties	
	○ **TYPE**	Risk
	○ **FAILURE-LIKELIHOOD**	High
	○ **FAILURE-IMPACT**	High
	○ **CONSEQUENCE**	Thousands of attendees logging in within the same minute overload the token issuing, the email gateway, and the relays, so that the first minutes of the event are lost for a part of the audience.
	○ **MITIGATION**	Proxy, relay, and server instances are scaled out before the event, and personalized and automatic access URLs let most attendees bypass the token step.
	Description	
	The audience of a large event connects within a very short window around the announced start rather than spread over time — **BECAUSE** attendees open their invitation link only when the event begins.	
Access URLs Forwarded Beyond the Audience ⚓	Properties	
	○ **TYPE**	Risk
	○ **FAILURE-LIKELIHOOD**	Medium
	○ **FAILURE-IMPACT**	Medium
	○ **CONSEQUENCE**	Uninvited persons watch a confidential event and take part in its interaction under the identity of an invited attendee.
	○ **MITIGATION**	Access is granted to listed or pattern-matching email addresses only, each attendee holds a single active session, and unguessable URLs with time-limited tokens shorten the window of a leaked link.
	Description	
	An invited attendee forwards their individual access URL to colleagues or posts it publicly — **BECAUSE** personal links are easily shared and attendees do not perceive them as secrets.	
Message Texts Contain Personal Data ⚓	Properties	
	○ **TYPE**	Risk
	○ **FAILURE-LIKELIHOOD**	High
	○ **FAILURE-IMPACT**	Medium

● PREMISE	Properties & Description	
	○ CONSEQUENCE	Personal data written into chat and questions is exported to the organizers and sent to the external translation service, so that a retention or transfer beyond the given consent violates the GDPR.
	○ MITIGATION	Attendees give explicit consent before interacting, the finish procedure anonymizes all attendee data, and the translation service is contracted under an EU data-processing agreement.
	Description	
	Attendees write personal data of themselves or of others into chat messages and questions — **BECAUSE** free-text interaction cannot be constrained to non-personal content.	
Abusive Interaction from the Audience ⚓	Properties	
	○ TYPE	Risk
	○ FAILURE-LIKELIHOOD	Medium
	○ FAILURE-IMPACT	Medium
	○ CONSEQUENCE	Spam, floods, or offensive messages drown the legitimate questions and put the moderators under pressure during the live event.
	○ MITIGATION	Messages are moderated before becoming visible, submissions are throttled per attendee, and the sentiment analysis flags offensive content automatically.
	Description	
	A part of the audience misuses the chat and question channels with spam, floods, or offensive content — **BECAUSE** large invited audiences are never free of disruptive participants.	

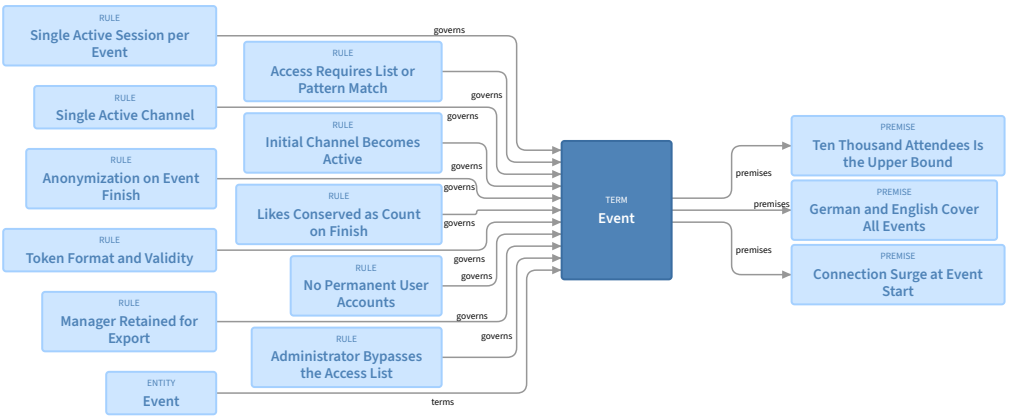
03 ● REQS: Domain Glossary (DG) ⚓

Event TYPE: Entity	Attendee TYPE: Actor	User TYPE: Entity	Manager TYPE: Actor
Moderator TYPE: Actor	Presenter TYPE: Actor	Administrator TYPE: Actor	Role TYPE: Entity
Channel TYPE: Entity	Resource TYPE: Entity	Streaming Provider TYPE: Actor	Message TYPE: Entity
Question TYPE: Entity	Chat TYPE: Entity	Support TYPE: Entity	Like TYPE: Entity
Authorization Token TYPE: Entity	Session Token TYPE: Entity	Access List TYPE: Entity	Event Registration System TYPE: Actor
Anonymization TYPE: Activity	Sentiment Analysis TYPE: Activity		

● TERM	Properties & Description

Event

Diagram



Properties

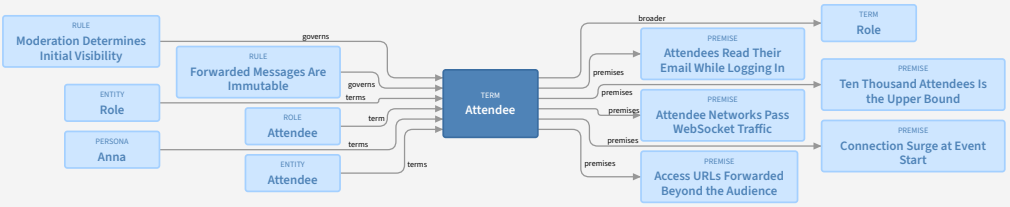
○ **TYPE**	Entity
○ **PREMISES**	● **PREMISE:** Ten Thousand Attendees Is the Upper Bound, ● **PREMISE:** German and English Cover All Events, ● **PREMISE:** Connection Surge at Event Start

Description

The central, organizer-defined live broadcast occasion to which an audience is invited; it carries all configuration, ● **Channel**s, the ● **Access List**, and ● **Message**s, and progresses from planning through running to finished

Attendee

Diagram



Properties

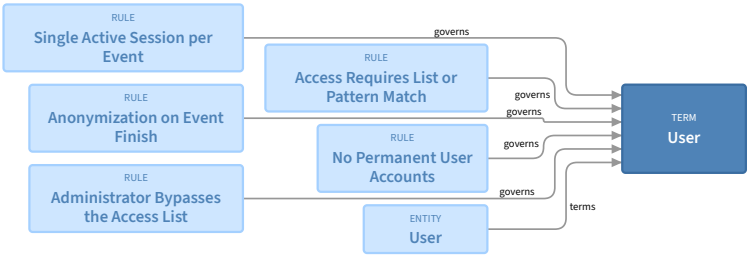
○ **TYPE**	Actor
○ **SYNONYMS**	Participant, Viewer
○ **BROADER**	● **TERM:** Role
○ **PREMISES**	● **PREMISE:** Attendees Read Their Email While Logging In, ● **PREMISE:** Ten Thousand Attendees Is the Upper Bound, ● **PREMISE:** Attendee Networks Pass WebSocket Traffic, ● **PREMISE:** Connection Surge at Event Start, ● **PREMISE:** Access URLs Forwarded Beyond the Audience

Description

A ● **User** invited to and logged into a specific ● **Event** who watches the stream and may use the interaction channels; an Attendee exists only for the duration of the ● **Event**

User

Diagram



● TERM	Properties & Description

Properties

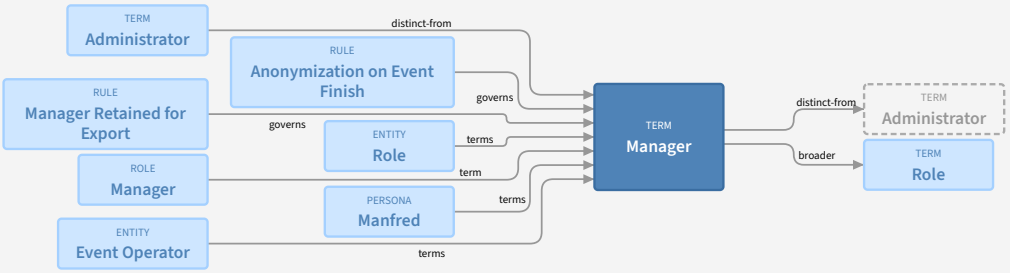
○ TYPE	Entity

Description

A person identified by their email address within a specific ● **Event**, holding the ● **Role**s granted to them and existing only while granted a ● **Role**, present on the ● **Access List**, or joining via the access email pattern of the ● **Event**, except the holder of the ● **Administrator** ● **Role**, who exists permanently and outside any ● **Event**; any other User is never a permanent account, and an ● **Attendee** is a User logged into the ● **Event**

Manager

Diagram



Properties

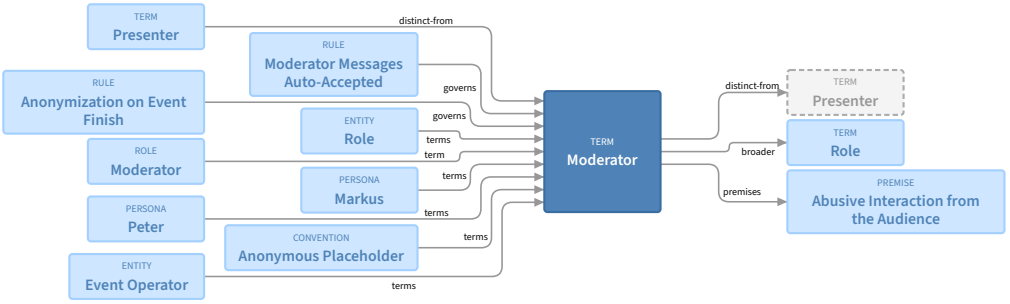
○ TYPE	Actor
○ BROADER	● TERM: Role
○ DISTINCT-FROM	● TERM: Administrator

Description

An event-level ● **Role** that can edit, start, stop, and delete an ● **Event** and export its anonymized data; the Manager role is retained after the ● **Event** finishes until the ● **Event** is deleted

Moderator

Diagram



Properties

○ TYPE	Actor
○ BROADER	● TERM: Role
○ DISTINCT-FROM	● TERM: Presenter
○ PREMISES	● PREMISE: Abusive Interaction from the Audience

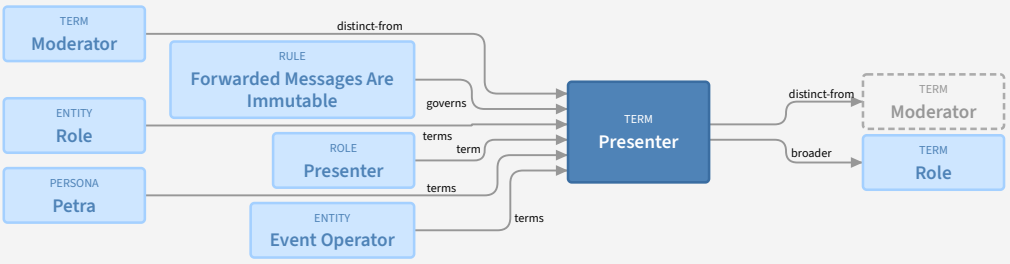
Description

An event-specific ● **Role** that moderates chat and question ● **Message**s by rejecting, approving, and forwarding them, and that supports the ● **Presenter** with hints and curated input

● TERM	Properties & Description

Presenter

Diagram



Properties

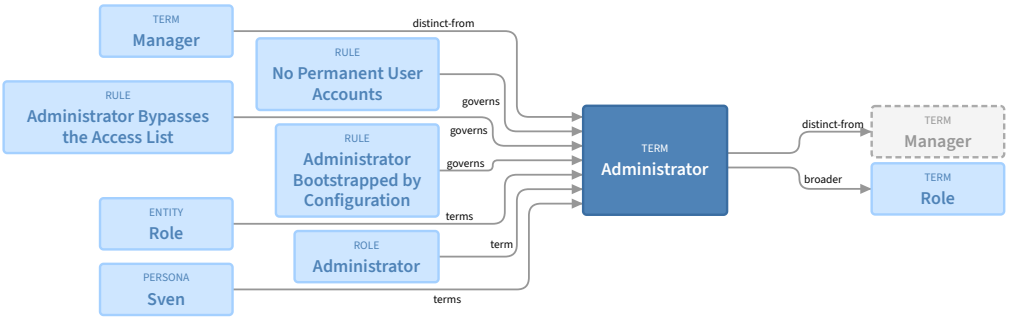
○ **TYPE**	Actor
○ **BROADER**	● **TERM: Role**
○ **DISTINCT-FROM**	● **TERM: Moderator**

Description

An event-specific ● **Role** held by the person on stage in a recorded video ● **Event**, receiving the ● **Messages** forwarded by the ● **Moderator** and marking them answered or suspended after processing them live; unlike the ● **Moderator**, the Presenter curates nothing and sees only the forwarded subset

Administrator

Diagram



Properties

○ **TYPE**	Actor
○ **SYNONYMS**	Software Administrator, Hardware Administrator
○ **BROADER**	● **TERM: Role**
○ **DISTINCT-FROM**	● **TERM: Manager**

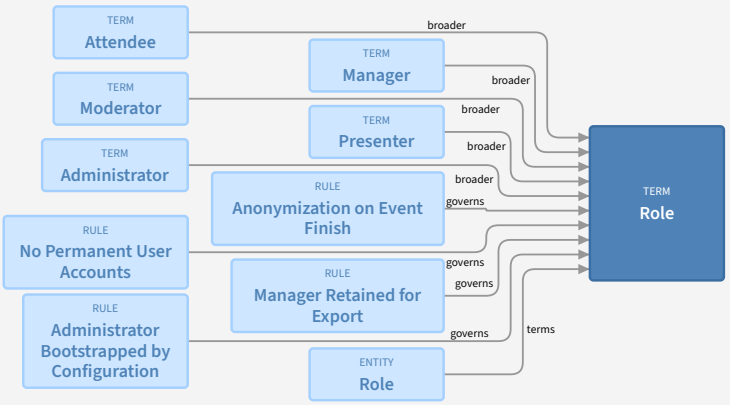
Description

A permanent ● **Role** independent of any ● **Event** and bootstrapped from the software configuration, whose holder needs no ● **Access List** entry and may perform every operation on every entity of every ● **Event**; the Hardware Administrator manages the physical server while the Software Administrator manages the software configuration and creates ● **Events**

● TERM	Properties & Description

Role

Diagram



Properties

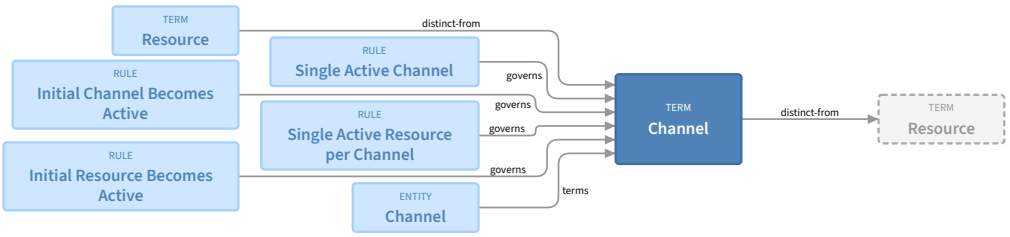
○ **TYPE**	Entity
○ **SYNONYMS**	Event Role

Description

A grant of rights held by a ● **User** (usually within a specific ● **Event**), of type ● **Attendee**, ● **Manager**, ● **Moderator**, ● **Presenter** or ● **Administrator**. A role is a permanent entity

Channel

Diagram



Properties

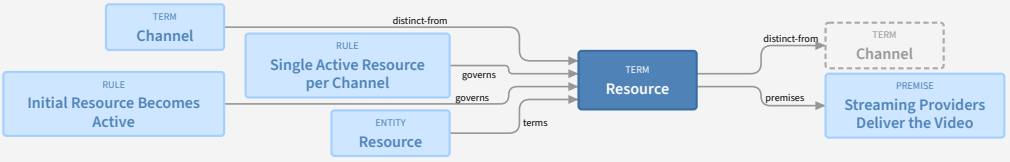
○ **TYPE**	Entity
○ **DISTINCT-FROM**	● **TERM: Resource**

Description

A logical content stream of an ● **Event** grouping language- and resolution-specific ● **Resources**; exactly one Channel of an ● **Event** is active at a time

Resource

Diagram



Properties

○ **TYPE**	Entity
○ **DISTINCT-FROM**	● **TERM: Channel**
○ **PREMISES**	● **PREMISE: Streaming Providers Deliver the Video**

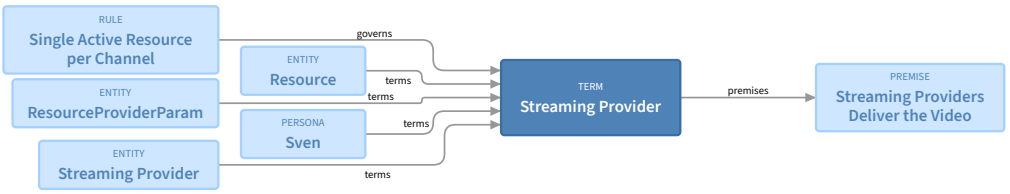
Description

A physical content delivery endpoint backing a ● **Channel**, such as a provider video stream or a static website; exactly one Resource of a ● **Channel** is active at a time

● TERM	Properties & Description

Streaming Provider

Diagram



Properties

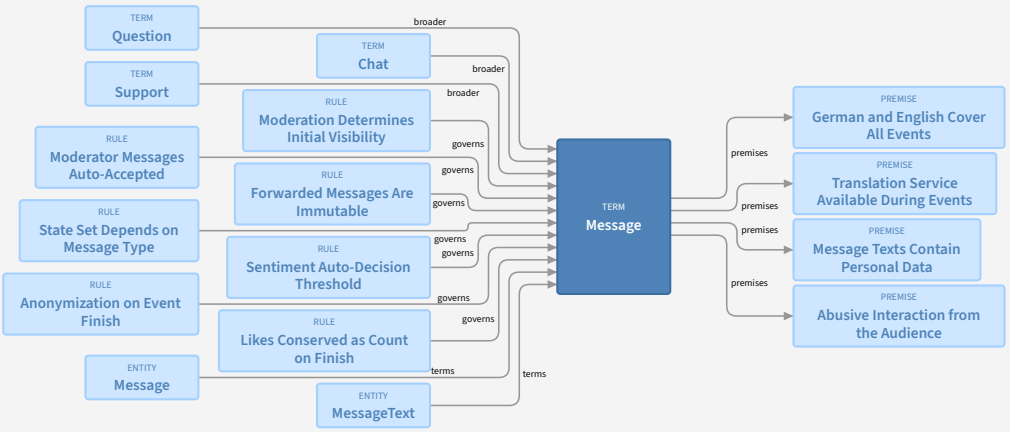
○ TYPE	Actor
○ SYNONYMS	Provider, CDN Provider
○ PREMISES	● PREMISE: Streaming Providers Deliver the Video

Description

An external service such as msg Filmstudio, YouTube, Cloudflare, Twitch, or 3Q that ingests and delivers the video stream addressed by a ● **Resource** through configured provider parameters

Message

Diagram



Properties

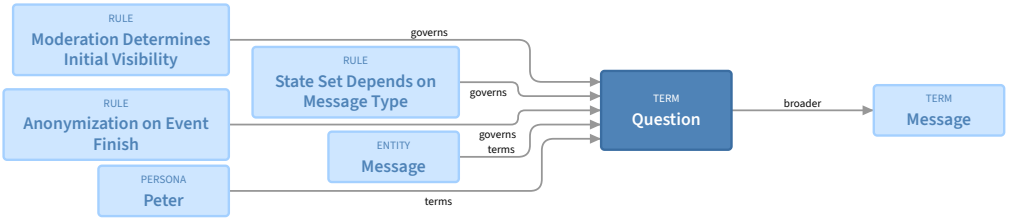
○ TYPE	Entity
○ PREMISES	● PREMISE: German and English Cover All Events, ● PREMISE: Translation Service Available During Events, ● PREMISE: Message Texts Contain Personal Data, ● PREMISE: Abusive Interaction from the Audience

Description

A single unit of ● **Event** interaction of type ● **Chat**, ● **Question**, or ● **Support**, carrying language-specific texts and moving through a moderation and presentation lifecycle

Question

Diagram



Properties

○ TYPE	Entity
○ SYNONYMS	Question Message
○ BROADER	● TERM: Message

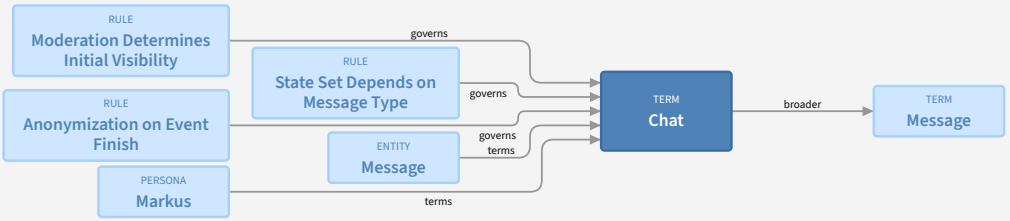
● TERM	Properties & Description

Description

A ● **Message** submitted by an ● **Attendee** as structured audience input for the Q&A rounds; it passes through the full moderation and presentation lifecycle, may carry tags, and may receive ● **Like**s

Chat

Diagram



Properties

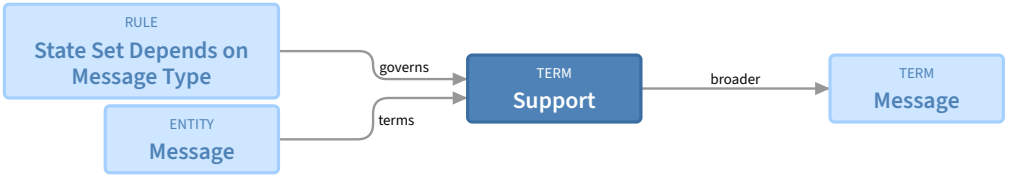
○ **TYPE**	Entity
○ **SYNONYMS**	Chat Message
○ **BROADER**	● **TERM: Message**

Description

A ● **Message** sent by an ● **Attendee** as direct live commentary on the ● **Event**, visible to the audience once accepted; it may receive ● **Like**s and, where enabled, replies

Support

Diagram



Properties

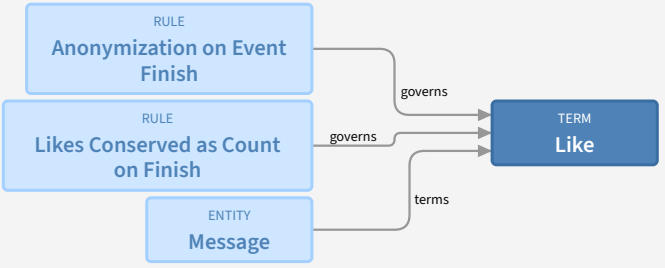
○ **TYPE**	Entity
○ **SYNONYMS**	Support Message
○ **BROADER**	● **TERM: Message**

Description

A ● **Message** exchanged privately between an ● **Attendee** and the ● **Moderator**s for clarification and help; it is never visible to the audience and needs no moderation

Like

Diagram



Properties

○ **TYPE**	Entity

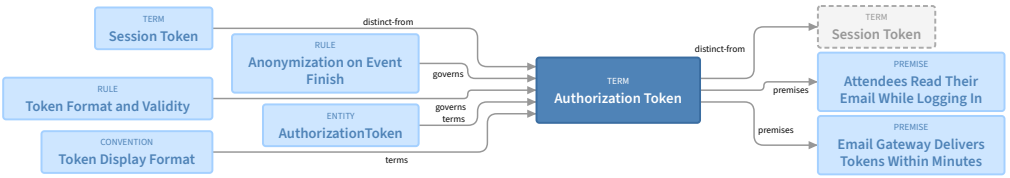
Description

The marking of a ● **Question** or ● **Chat** as relevant by an ● **Attendee**, tracked per ● **User** and revocable until the ● **Event** finishes, thereafter conserved as a bare count by the ● **Anonymization**

● TERM	Properties & Description

Authorization Token

Diagram



Properties

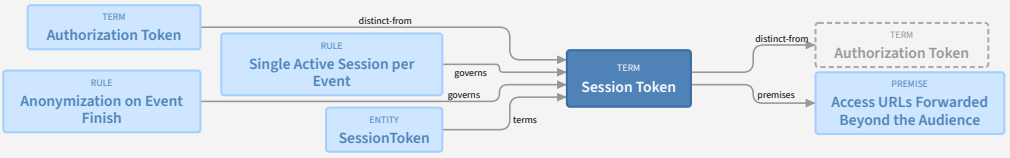
○ **TYPE**	Entity
○ **SYNONYMS**	Access Token
○ **DISTINCT-FROM**	● **TERM: Session Token**
○ **PREMISES**	● **PREMISE: Attendees Read Their Email While Logging In**, ● **PREMISE: Email Gateway Delivers Tokens Within Minutes**

Description

A one-time, time-limited “NNN-NNN” six-digit second factor proving an ● **Attendee** controls their email address; it may be pre-generated for automatic access

Session Token

Diagram



Properties

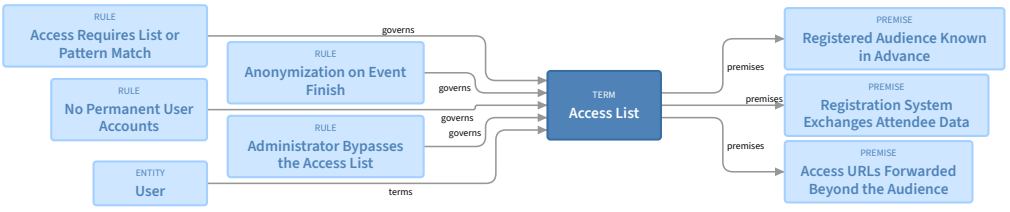
○ **TYPE**	Entity
○ **DISTINCT-FROM**	● **TERM: Authorization Token**
○ **PREMISES**	● **PREMISE: Access URLs Forwarded Beyond the Audience**

Description

The result of a successful login granting an ● **Attendee** an active connection to an ● **Event**; only one Session Token per User per ● **Event** may be active at a time

Access List

Diagram



Properties

○ **TYPE**	Entity
○ **PREMISES**	● **PREMISE: Registered Audience Known in Advance**, ● **PREMISE: Registration System Exchanges Attendee Data**, ● **PREMISE: Access URLs Forwarded Beyond the Audience**

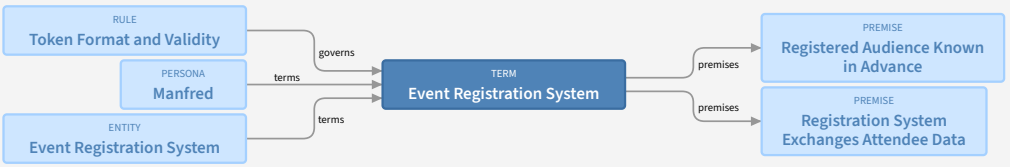
Description

The set of Users, identified by email, invited to a specific ● **Event**; together with the access email pattern it determines who may be granted ● **Event** access

● TERM	Properties & Description

Event Registration System

Diagram



Properties

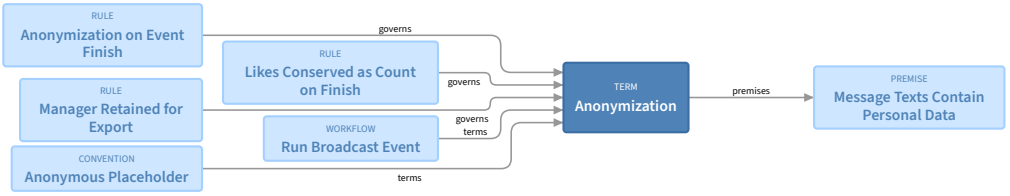
○ TYPE	Actor
○ PREMISES	● PREMISE: Registered Audience Known in Advance, ● PREMISE: Registration System Exchanges Attendee Data

Description

The external registration platform from which ● **Attendee** data is imported via Excel and to which generated access URLs are returned

Anonymization

Diagram



Properties

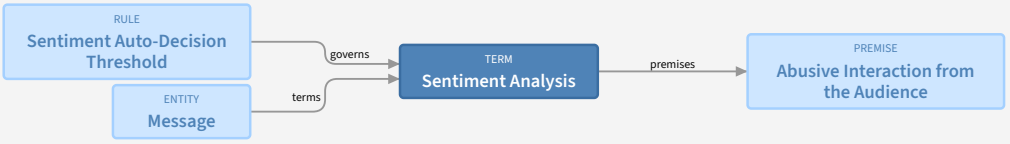
○ TYPE	Activity
○ SYNONYMS	Privacy by Design Finish
○ PREMISES	● PREMISE: Message Texts Contain Personal Data

Description

The automated procedure on ● **Event** finish that reduces ● **Messages** to like counts, anonymizes sender names, drops personal relations, deletes tokens and Users, unlinks the viewer statistics from their Users, and removes ● **Moderator** roles

Sentiment Analysis

Diagram



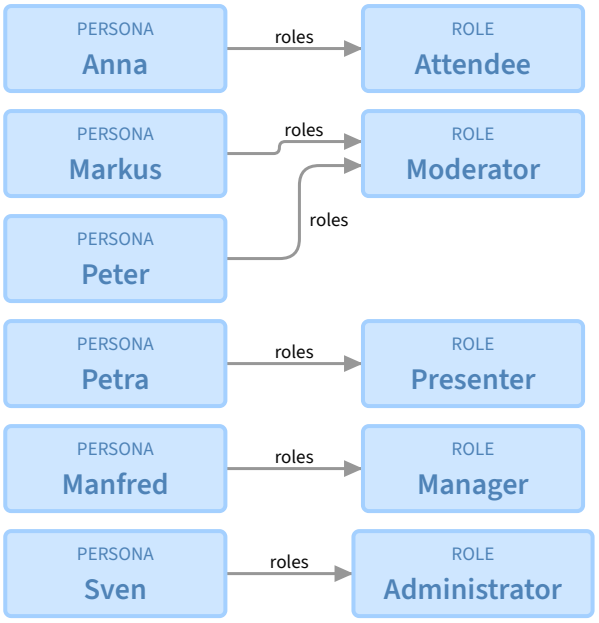
Properties

○ TYPE	Activity
○ SYNONYMS	Profanity Check
○ PREMISES	● PREMISE: Abusive Interaction from the Audience

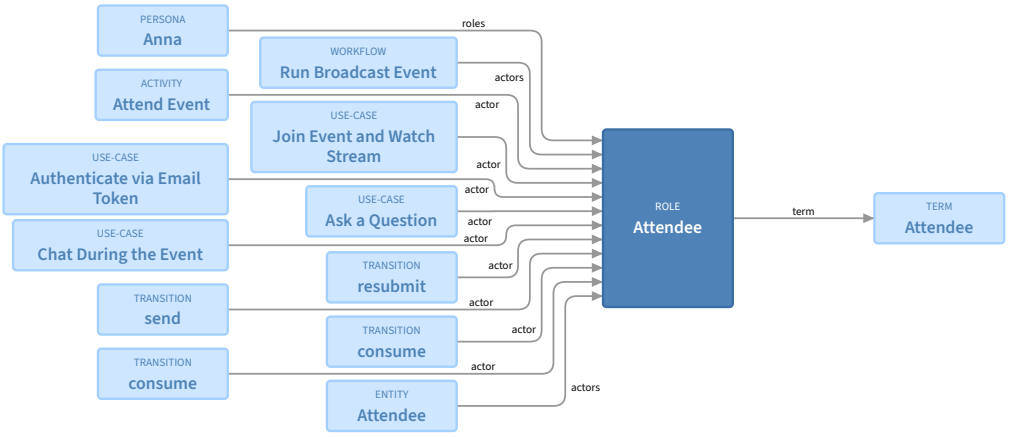
Description

The optional client-side or server-side evaluation of ● **Message** text yielding a sentiment score used to prevent, auto-accept, or auto-reject improper input

04 ● REQS: User Personas (UP)



● ROLE	Properties & Description
Attendee	Diagram

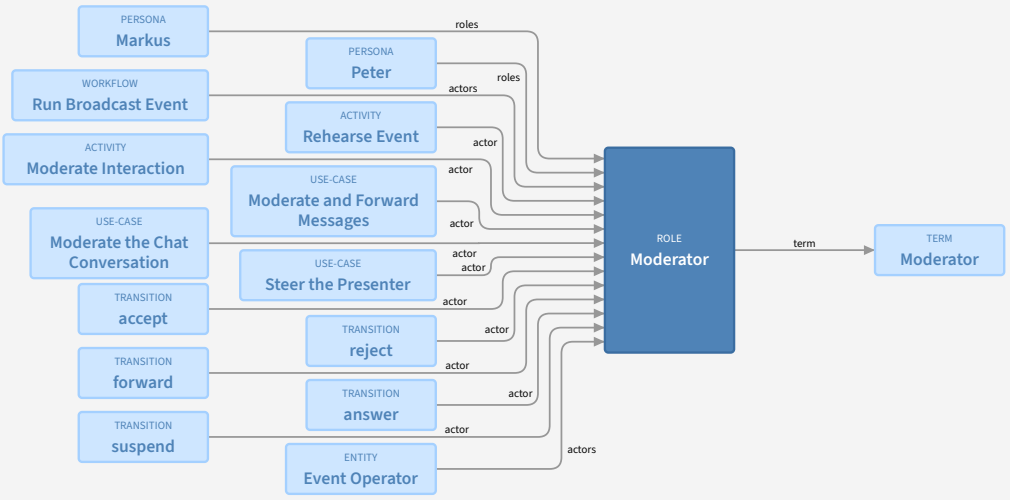


○ TYPE	○ TERM	Description
Person	● **TERM: Attendee**	The attendee watches the stream of the event they were invited to and takes part in its interaction with messages and likes of their own — **BECAUSE** the audience has to participate without ever gaining a view on the moderation or on the personal data of other attendees.

● ROLE	Properties & Description

Moderator

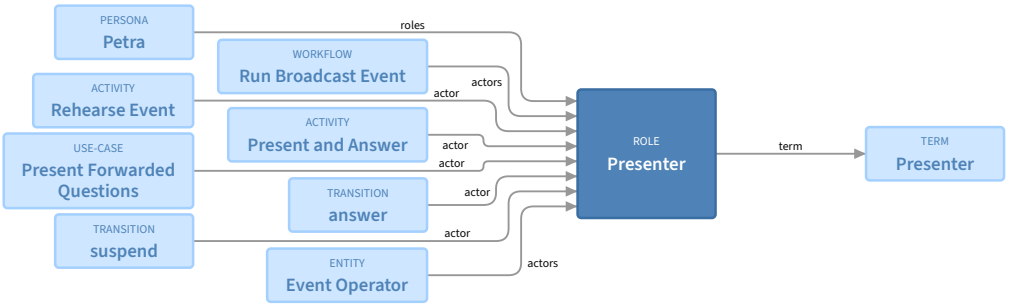
Diagram



○ TYPE	○ TERM	Description
Person	● **TERM: Moderator**	The moderator gatekeeps the interaction of a running event by deciding, answering, forwarding, and seeding messages and by steering the presenter, where the Q&A moderator and the chat moderator hold the very same role and split the work by message type by convention only — **BECAUSE** the audience has to see only what the organizers approve while the presenter receives a curated selection.

Presenter

Diagram



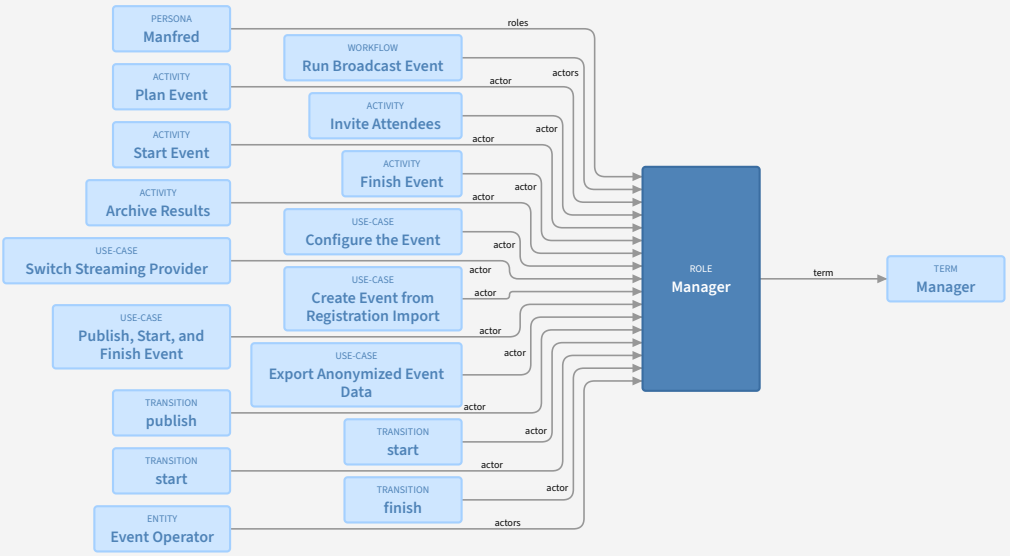
○ TYPE	○ TERM	Description
Person	● **TERM: Presenter**	The presenter receives the curated questions on stage and records their processing, seeing nothing else of the interaction — **BECAUSE** the person on air has to be shielded from the raw audience input.

● ROLE

Manager

Properties & Description

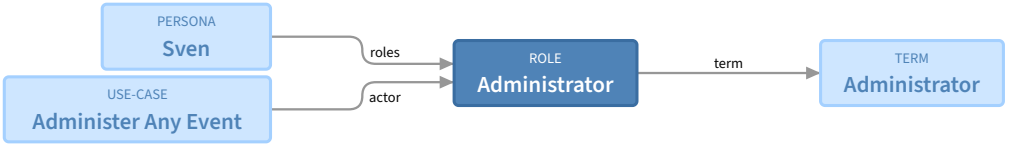
Diagram



○ TYPE	○ TERM	Description
Person	● **TERM: Manager**	The manager owns an event from its configuration through its live run to its archived export, without taking part in the moderation — **BECAUSE** one accountable role has to control the visibility, the audience, and the lifecycle of the event.

Administrator

Diagram



○ TYPE	○ TERM	Description
Person	● **TERM: Administrator**	The administrator provisions the events and the streaming providers through the configuration of the software and, as the permanent bootstrap identity, enters every event without an access list entry and edits everything in it — **BECAUSE** the operation of the platform has to be reproducible from configuration and recoverable by one role that depends on no event-specific grant.

● PERSONA

Anna

Properties & Description

○ ROLES	○ TERMS	○ GOAL	○ FRUSTRATION
● **ROLE: Attendee**	● **TERM: Attendee**	Watch the live event effortlessly and get her questions answered.	Streaming tools that demand accounts, apps, or plugins before she can even watch.

Description

"I just want to watch the town hall and ask my question without jumping through hoops."

● PERSONA	Properties & Description			
Markus	○ ROLES	○ TERMS	○ GOAL	○ FRUSTRATION
	● **ROLE:** Moderator	● **TERM:** Moderator, ● **TERM:** Chat	Keep the chat conversation civil and on-topic with minimal effort.	Chat tools without moderation support, letting spam and off-topic noise drown the conversation.
	Description			
	"I keep the Chat conversation clean."			
Peter	○ ROLES	○ TERMS	○ GOAL	○ FRUSTRATION
	● **ROLE:** Moderator	● **TERM:** Moderator, ● **TERM:** Question	Filter the incoming questions and forward only the relevant ones to the presenter.	Unfiltered question streams where duplicates and noise bury the questions that truly matter.
	Description			
	"I keep the Q&A funnel clean and feed the presenter only questions which truly matter."			
Petra	○ ROLES	○ TERMS	○ GOAL	○ FRUSTRATION
	● **ROLE:** Presenter	● **TERM:** Presenter	Deliver her talk while receiving the curated questions at the right moment.	Being distracted on stage by raw chat noise instead of a few curated questions.
	Description			
	"On stage I need the right questions in front of me at the right moment, and nothing else."			
Manfred	○ ROLES	○ TERMS	○ GOAL	○ FRUSTRATION
	● **ROLE:** Manager	● **TERM:** Manager, ● **TERM:** Event Registration System	Set up events quickly and hand the access URLs back to the Event Registration System.	Manual, error-prone event setup with access URLs that have to be copied around by hand.
	Description			
	"Setting up an event and getting the access URLs back to the Event Registration System has to be quick and repeatable."			
Sven	○ ROLES	○ TERMS	○ GOAL	○ FRUSTRATION
	● **ROLE:** Administrator	● **TERM:** Administrator, ● **TERM:** Streaming Provider	Provision events and streaming providers reproducibly via configuration, and fix anything in any event when the organizers are stuck.	Platforms that require per-event manual clicks and lock him out of the events he provisioned.
	Description			
	"I provision events and streaming providers via config, and when something is stuck I step into any event and fix it."			

05 ● REQS: User Journey (UJ) ⚓

Attend a Broadcast event	Operate a Broadcast Event	Moderate a Broadcast Event
ACTORS: Anna	ACTORS: Manfred	ACTORS: Markus, Peter

05.01 ● JOURNEY: Attend a Broadcast event ⚓

○ ACTORS	● PERSONA: Anna ⚭

● STEP	Properties		
Receive personal event invitation URL ⚓	○ STAGE	○ GOAL	○ TOUCHPOINT
	Awareness	Learn that an event is happening and that access is granted	Invitation email containing the individual event URL
	○ ACTION	○ EMOTION	○ PAIN-POINT
	The attendee reads the invitation and clicks the personalized event link.	Curious (+1)	∅
	○ OPPORTUNITY		
	∅		
Open event page and review login info ⚓	○ STAGE	○ GOAL	○ TOUCHPOINT
	Consideration	Understand what the event is and how to join	Event landing screen in the web browser
	○ ACTION	○ EMOTION	○ PAIN-POINT
	The attendee reads the optional login message and any links before deciding to log in.	Interested (+1)	Uncertainty whether the link is genuine and what data will be collected.
	○ OPPORTUNITY		
	The event landing screen states organizer, purpose, and data handling upfront, so the attendee can trust the link.		
Authenticate via email token ⚓	○ STAGE	○ GOAL	○ TOUCHPOINT
	Decision	Gain access to the live video stream	Login dialog with email and token challenge
	○ ACTION	○ EMOTION	○ PAIN-POINT
	The attendee enters the email, receives a 6-digit token by mail, and submits it.	Slightly impatient (-1)	Waiting for the token email and re-checking the inbox adds friction before the event.
	○ OPPORTUNITY		
	The token email arrives within seconds and the token field accepts a pasted value, so the wait before the event stays minimal.		
Watch stream and interact ⚓	○ STAGE	○ GOAL	○ TOUCHPOINT
	Usage	Follow the event and contribute questions, chat, and likes	Main attendee screen with video and interaction sidebar

● STEP	Properties		
	○ ACTION	○ EMOTION	○ PAIN-POINT
	The attendee watches the stream, posts questions or chat messages, and likes others' messages.	Engaged (+2)	Messages may be delayed by moderation, so the attendee is unsure whether input arrived.
	○ OPPORTUNITY		
	Submitted messages appear immediately in a pending state, so the attendee knows the input arrived.		

05.02 ● JOURNEY: Operate a Broadcast Event

○ ACTORS	● PERSONA: Manfred

● STEP	Properties		
Configure and run the event	○ STAGE	○ GOAL	○ TOUCHPOINT
	Usage	Set up, start, and control an event end-to-end	Manager configuration screen and Event Registration System Excel import/export
	○ ACTION	○ EMOTION	○ PAIN-POINT
	The manager imports attendees, configures channels and options, and starts the event.	Focused (0)	Coordinating streaming providers, access lists, and tokens before going live is error-prone.
	○ OPPORTUNITY		
	The configuration screen validates the imported attendee list and the streaming settings before the event goes live.		
Export anonymized results after event	○ STAGE	○ GOAL	○ TOUCHPOINT
	Usage	Capture and share the event's interaction outcomes	Manager export function producing a downloadable file
	○ ACTION	○ EMOTION	○ PAIN-POINT
	The manager exports anonymized messages with timestamps, states, and like counts.	Satisfied (+2)	∅
	○ OPPORTUNITY		
	∅		

05.03 ● JOURNEY: Moderate a Broadcast Event

○ ACTORS	● PERSONA: Markus, ● PERSONA: Peter

● STEP	Properties		
Moderate and support the presenter	○ STAGE	○ GOAL	○ TOUCHPOINT
	Usage	Keep interaction clean and forward the best questions to the presenter	Moderator Kanban board of chat and question messages

● STEP	Properties		
	○ ACTION	○ EMOTION	○ PAIN-POINT
	The moderator approves, rejects, and forwards messages and sends hints to the presenter.	Pressured (-2)	High message volume under time pressure makes it hard to keep an overview.
	○ OPPORTUNITY		
	The Kanban board groups messages by state and highlights new arrivals, so the moderator keeps an overview under load.		

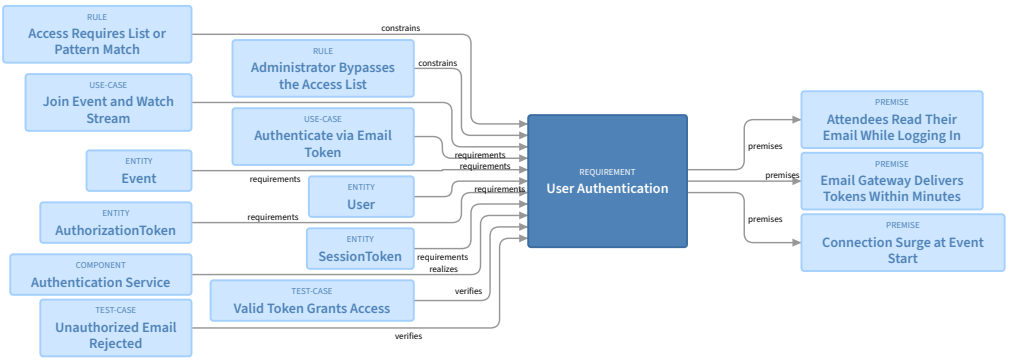
06 ● REQS: Functional Requirements (FR) ⚓

User Authentication PRIORITY: MUST	Limit Parallel Access PRIORITY: MUST	Ask Questions PRIORITY: MUST	Chat Messages PRIORITY: MUST
Like Messages PRIORITY: MUST	Reply to Chat Messages PRIORITY: SHOULD	Configurable Name Appearance PRIORITY: MUST	Personal Display Filtering PRIORITY: SHOULD
Client-Side Sentiment Analysis PRIORITY: COULD	Server-Side Sentiment Analysis PRIORITY: COULD	Edit and Delete Messages PRIORITY: SHOULD	Tag Question Messages PRIORITY: SHOULD
Moderate Chat and Questions PRIORITY: MUST	Forward Inputs to Presenter PRIORITY: MUST	Answer Attendee Inputs PRIORITY: SHOULD	Moderator-Authored Messages PRIORITY: SHOULD
Export Attendee Inputs PRIORITY: MUST	Presenter Hints and Alerts PRIORITY: SHOULD	Sort and Filter Messages PRIORITY: SHOULD	Manage Embedded Application PRIORITY: COULD
Browser Access PRIORITY: MUST	Individual Event Access URL PRIORITY: MUST	Personalized Event Access URL PRIORITY: SHOULD	Automatic Event Access URL PRIORITY: SHOULD
Direct Resource Access URL PRIORITY: SHOULD	GDPR-Compliant EU Processing PRIORITY: MUST	Explicit User Consent PRIORITY: MUST	Login and Interaction Messages PRIORITY: SHOULD
Multiple Streaming Providers PRIORITY: MUST	Live Provider Switching PRIORITY: MUST	Import Registered Attendees PRIORITY: MUST	Return Access URLs to Registration PRIORITY: MUST
Import and Export Event PRIORITY: SHOULD	General Event Statistics PRIORITY: MUST	Debugging Statistics PRIORITY: SHOULD	Channel Statistics PRIORITY: SHOULD
User Statistics PRIORITY: SHOULD	Periodic Statistics Snapshots PRIORITY: MUST	Live Configuration Propagation PRIORITY: MUST	Presenter Dashboard PRIORITY: COULD
Multiple-Choice Feedback PRIORITY: COULD	Mobile Device Support PRIORITY: COULD	In-Session Language Switching PRIORITY: SHOULD	Light and Dark Theme PRIORITY: COULD
Deleted Message Placeholder PRIORITY: SHOULD			

● REQUIREMENT	Properties & Description

User Authentication

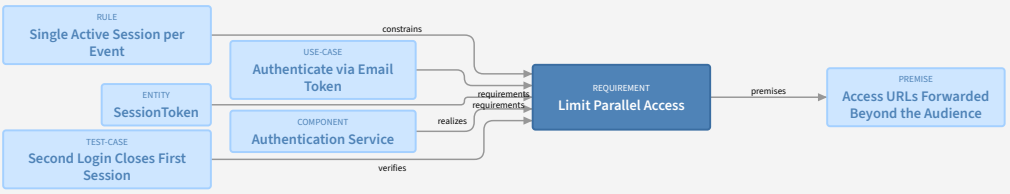
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE:** Attendees Read Their Email While Logging In, ● **PREMISE:** Email Gateway Delivers Tokens Within Minutes, ● **PREMISE:** Connection Surge at Event Start	The system MUST authenticate participants via their email address before showing the video stream or, at minimum, before allowing participation in questions and chat, re-checking on every access that the event still lists the user even when the user is already authenticated — **BECAUSE** access must be restricted to invited participants to minimize foreign viewers

Limit Parallel Access

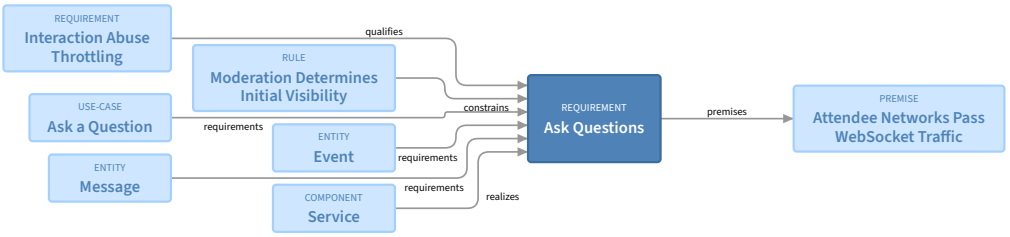
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE:** Access URLs Forwarded Beyond the Audience	When participants must be authenticated to watch, the system MUST allow an event to be viewed only once per participant at a time, closing the prior connection on a new login — **BECAUSE** single concurrent sessions minimize unauthorized outside viewing

Ask Questions

Diagram

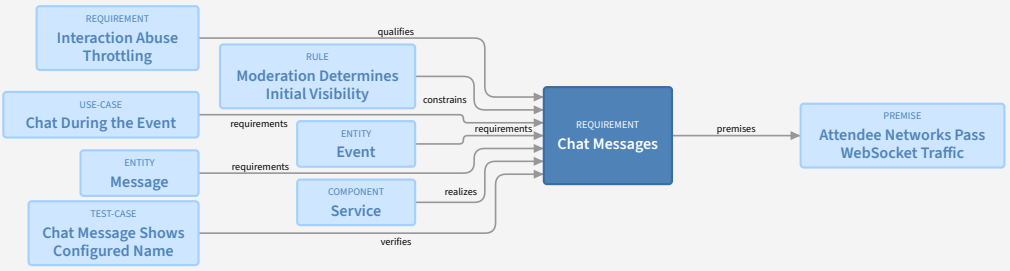


○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE:** Attendee Networks Pass WebSocket Traffic	The system MUST let attendees submit questions as a dedicated feedback channel feeding the Q&A rounds — **BECAUSE** structured audience questions are a core interaction the event format depends on

● REQUIREMENT	Properties & Description

Chat Messages

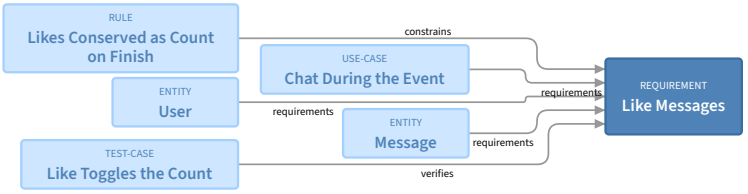
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE: Attendee Networks Pass WebSocket Traffic**	The system MUST let attendees send chat messages for direct comment on the event — **BECAUSE** live commentary keeps the audience engaged during the broadcast

Like Messages

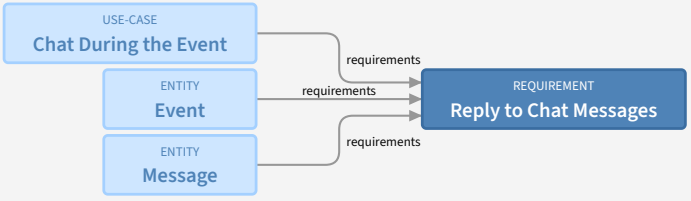
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	∅	The system MUST let attendees like questions and chat messages and undo the like — **BECAUSE** liking surfaces the most relevant audience input for moderators and presenters

Reply to Chat Messages

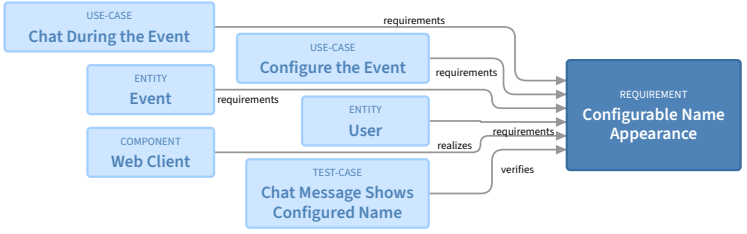
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	Where replies are enabled for an event, the system SHOULD let attendees reply to chat messages — **BECAUSE** threaded replies aid discussion but are not appropriate for every event format

Configurable Name Appearance

Diagram



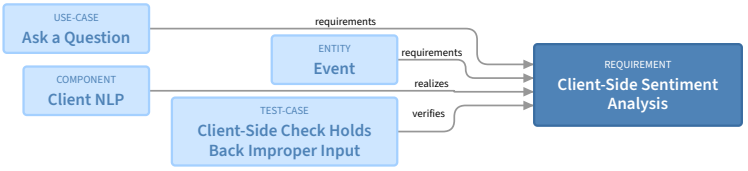
○ PRIORITY	○ PREMISES	Description
MUST	∅	The system MUST display an attendee's name on messages as full "Firstname Lastname", first name only, or

● REQUIREMENT	Properties & Description		
			anonymous per event configuration, always exposing the email address (at least on hover) and optionally deriving the name from the email if none is given — **BECAUSE** events differ in how much attendee identity should be revealed
Personal Display Filtering ⚓	Diagram		

○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD let an attendee choose to see only their own messages and questions or also those of others — **BECAUSE** attendees want to track their own contributions without distraction

Client-Side Sentiment Analysis ⚓

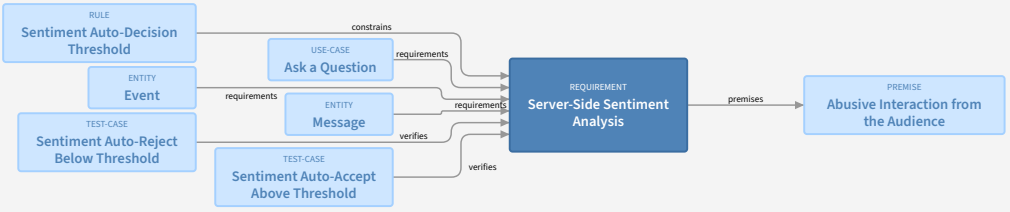
Diagram



○ PRIORITY	○ PREMISES	Description
COULD	∅	The system COULD perform configurable client-side sentiment analysis that checks attendee input before it is sent and may prevent improper submissions — **BECAUSE** filtering at the source reduces server moderation load and discourages misconduct

Server-Side Sentiment Analysis ⚓

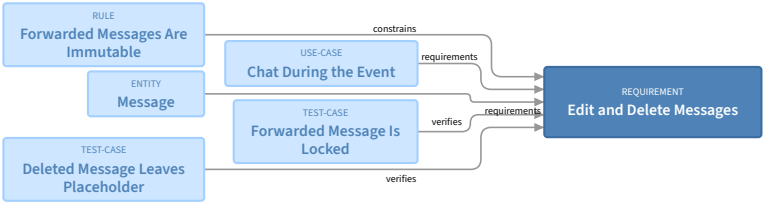
Diagram



○ PRIORITY	○ PREMISES	Description
COULD	● **PREMISE: Abusive Interaction from the Audience** ⚭	The system COULD perform configurable server-side sentiment analysis that checks attendee input before it is stored, auto-accepting positive and auto-rejecting negative input — **BECAUSE** automated triage assists moderators with high message volume

Edit and Delete Messages ⚓

Diagram

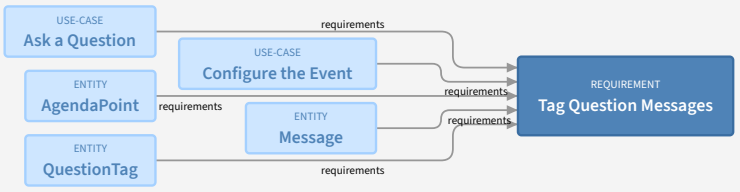


○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD let attendees edit and delete their own chat and question messages, mark edited messages for others, and re-run moderation on edits where required

● REQUIREMENT	Properties & Description		
			— **BECAUSE** attendees must correct mistakes without bypassing moderation

Tag Question Messages ⚓

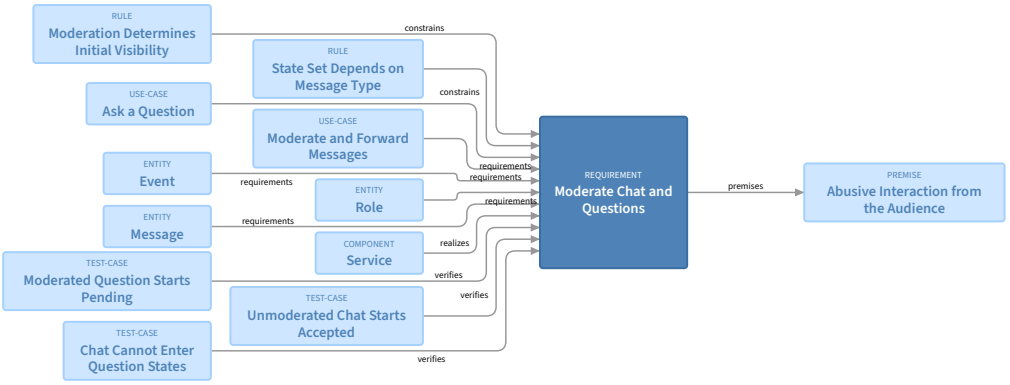
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD let attendees tag question messages with predefined tags to give context such as the addressed person or agenda point — **BECAUSE** tags help moderators and presenters route and group questions

Moderate Chat and Questions ⚓

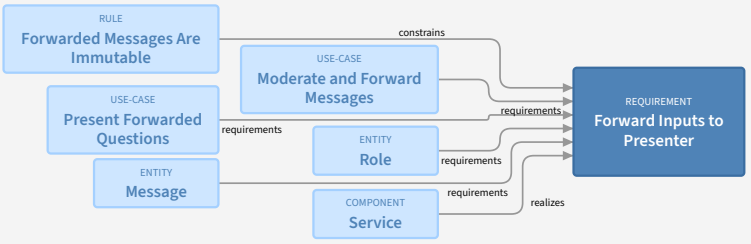
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE: Abusive Interaction from the Audience** ⚭	The system MUST support optional moderation in which moderators reject, approve, or forward attendee input, with messages held pending until approved when moderation is enabled — **BECAUSE** organizers need control over what becomes publicly visible

Forward Inputs to Presenter ⚓

Diagram

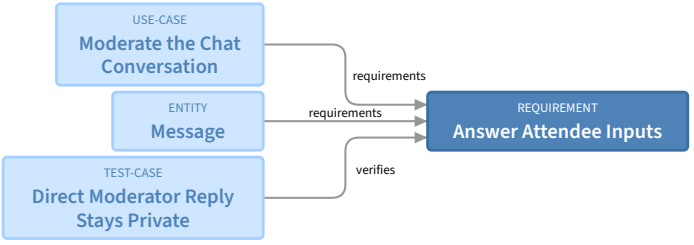


○ PRIORITY	○ PREMISES	Description
MUST	∅	The system MUST let moderators forward approved inputs to a presenter so the presenter sees only a curated subset rather than all approved messages — **BECAUSE** the presenter on stage can only process a focused set of items

● REQUIREMENT	Properties & Description

Answer Attendee Inputs ⚓

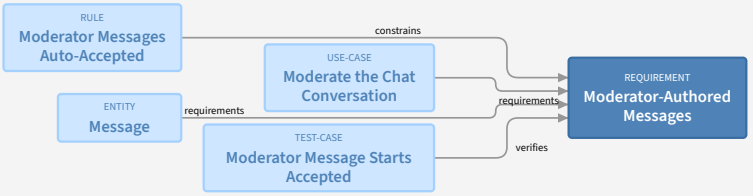
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD let moderators reply to an attendee's input, including non-visible direct messages — **BECAUSE** moderators need to clarify details or provide support privately

Moderator-Authored Messages ⚓

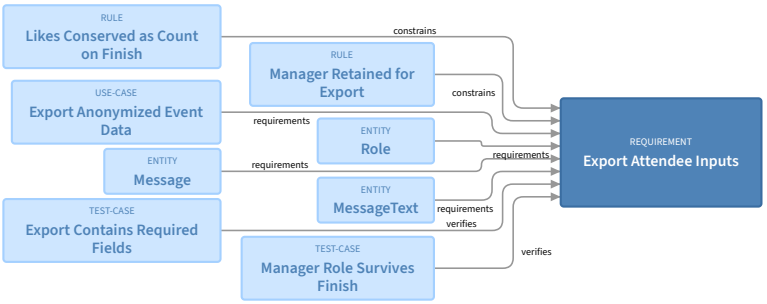
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD let moderators author chats and questions visible to others under a sender name defaulting to "Moderator" but overridable on send — **BECAUSE** prepared questions are needed to seed Q&A sessions

Export Attendee Inputs ⚓

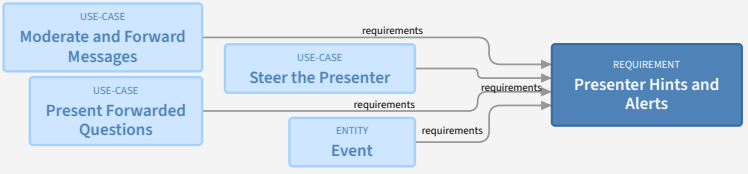
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	∅	After an event finishes, the system MUST let managers export all attendee inputs including at least timestamp, state, number of likes, and message text — **BECAUSE** organizers need a durable record of audience interaction

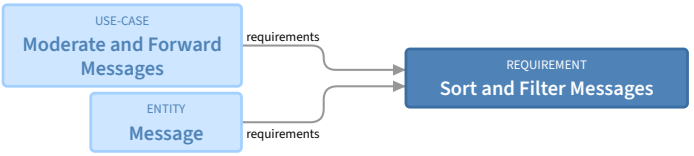
Presenter Hints and Alerts ⚓

Diagram



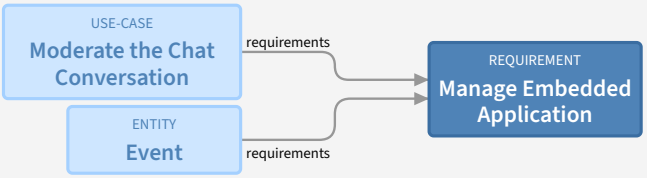
○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD let moderators send textual hints to the presenter and raise an alert the presenter can confirm

● REQUIREMENT	Properties & Description		
			— **BECAUSE** the moderator must steer the live presenter with timing and routing guidance
Sort and Filter Messages ⚓	Diagram		



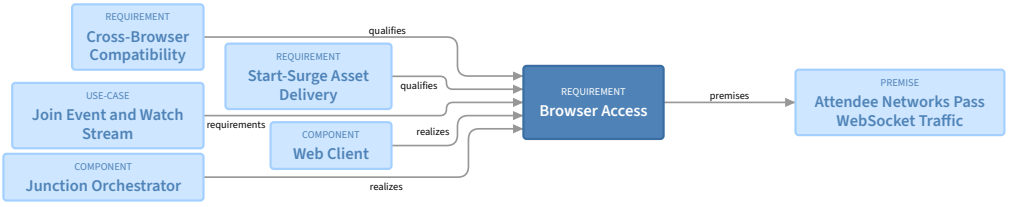
○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD let moderators filter messages and manually order forwarded questions — **BECAUSE** moderators need a clear overview and a defined order for the presenter to ask questions

Manage Embedded Application ⚓	Diagram		



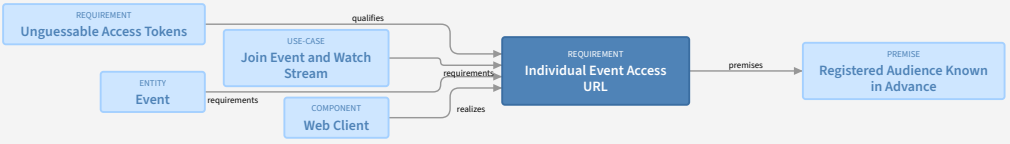
○ PRIORITY	○ PREMISES	Description
COULD	∅	The system COULD let moderators administer an embedded third-party application when it exposes an administration view — **BECAUSE** some events integrate interactive apps that the moderator must control

Browser Access ⚓	Diagram		



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE: Attendee Networks Pass WebSocket Traffic** ⚭	The system MUST be accessible from any reasonably recent web browser without installation — **BECAUSE** attendees use both managed and unmanaged devices with no common client software

Individual Event Access URL ⚓	Diagram		

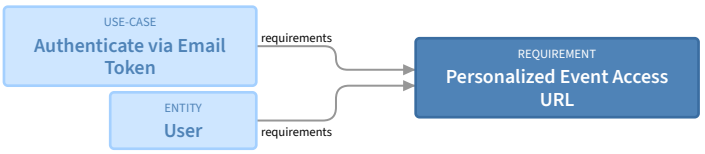


○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE: Registered Audience Known in Advance** ⚭	The system MUST allow accessing an event via an individual, unguessable URL of the form #/event=<event> — **BECAUSE** an obscure entry point minimizes access by external viewers

● REQUIREMENT	Properties & Description

Personalized Event Access URL ⚓

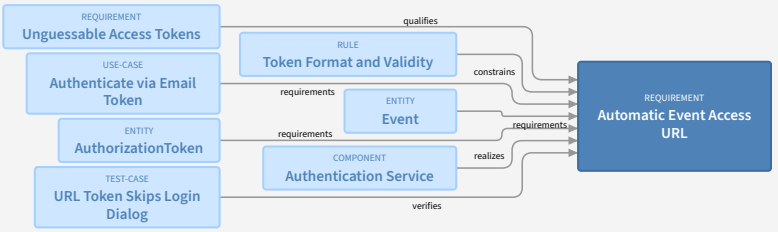
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD support an event URL that also carries the user email (`&user=<email>`), making the email read-only in the login dialog — **BECAUSE** pre-filling the identity streamlines the attendee login

Automatic Event Access URL ⚓

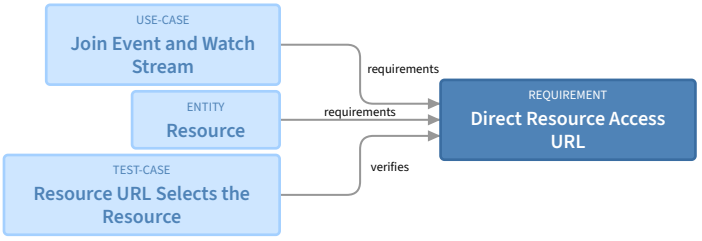
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD support an event URL carrying a pre-generated access token (`&token=<token>`) that auto-authenticates the user without a login dialog, with single or multiple use governed by event settings — **BECAUSE** automated access enables frictionless joining for trusted distributions

Direct Resource Access URL ⚓

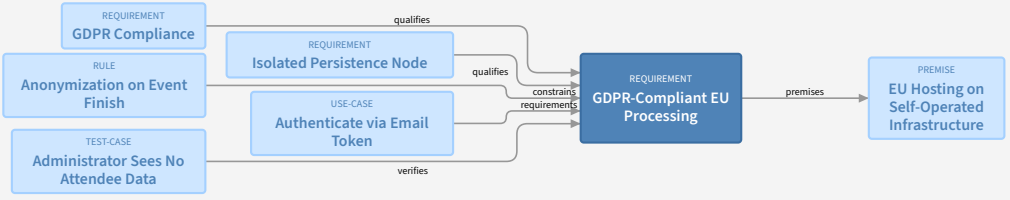
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD support a `&resource=<resource>` URL parameter that selects a specific event resource on access — **BECAUSE** attendees reloading the page must return to the same stream or static resource

GDPR-Compliant EU Processing ⚓

Diagram

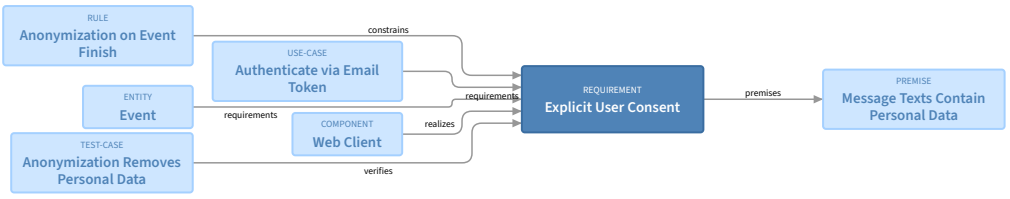


○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE: EU Hosting on Self-Operated**	The system MUST provide all services in compliance with GDPR within the EU — **BECAUSE** the audience and operator are subject to European data-protection law

● REQUIREMENT	Properties & Description		
		Infrastructure	

Explicit User Consent

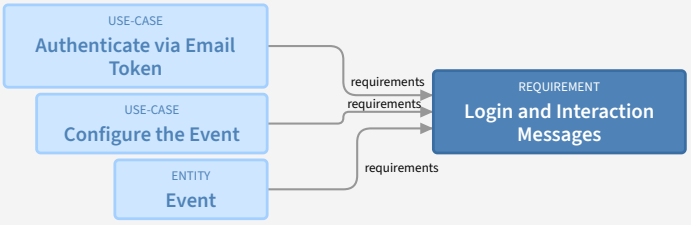
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE: Message Texts Contain Personal Data**	The system MUST require event attendees to give explicit consent before participating under defined conditions, in addition to prior platform consent — **BECAUSE** consent is both a legal safeguard and a psychological hurdle against prohibited stream sharing

Login and Interaction Messages

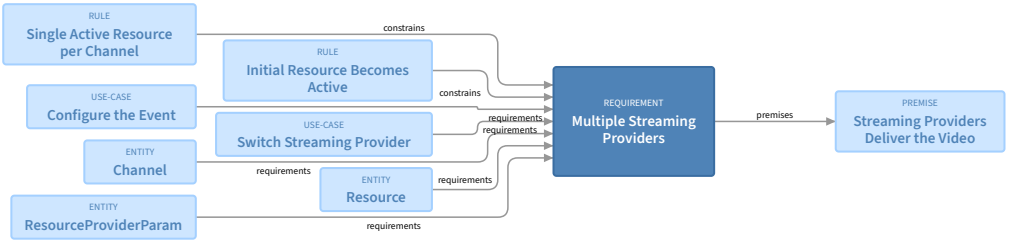
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD optionally display configurable text with one or two links at login and at interaction time — **BECAUSE** organizers communicate event information and conduct rules at these moments

Multiple Streaming Providers

Diagram

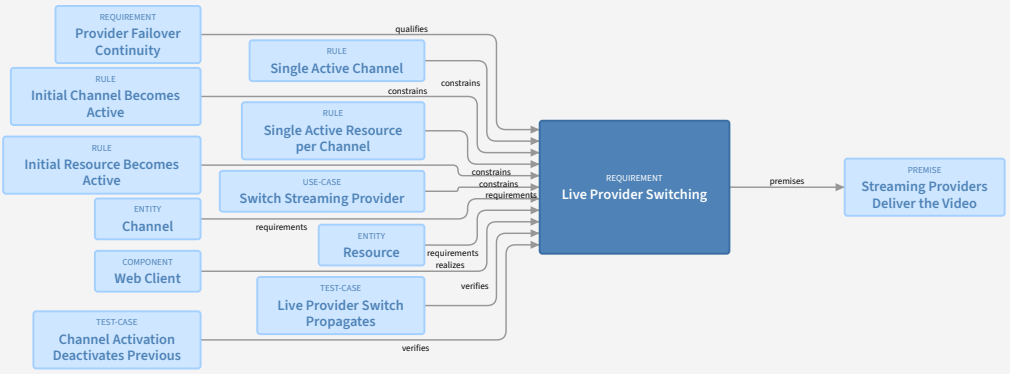


○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE: Streaming Providers Deliver the Video**	The system MUST allow multiple streaming providers to be configured per event — **BECAUSE** a configured fallback is required to recover from provider problems even during a running event

● REQUIREMENT	Properties & Description

Live Provider Switching

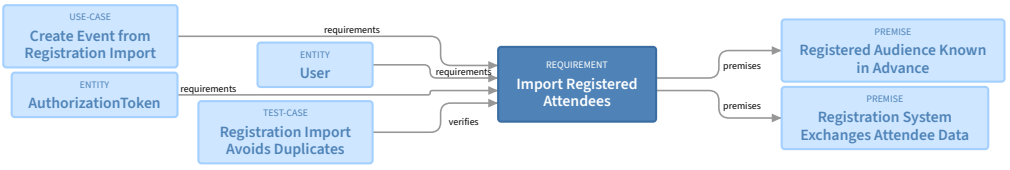
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE:** **Streaming Providers Deliver the Video**	The system MUST allow switching the active streaming provider during an event, with attendee clients following the switch without user interaction — **BECAUSE** provider outages must be mitigated without disrupting the audience

Import Registered Attendees

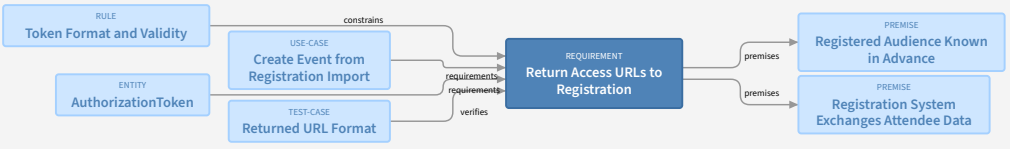
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE:** **Registered Audience Known in Advance**, ● **PREMISE:** **Registration System Exchanges Attendee Data**	The system MUST import an Excel sheet of the Event Registration System to populate an event's access list and generate authorization tokens, avoiding duplicate invitations on repeated imports — **BECAUSE** events are provisioned from the Event Registration System

Return Access URLs to Registration

Diagram

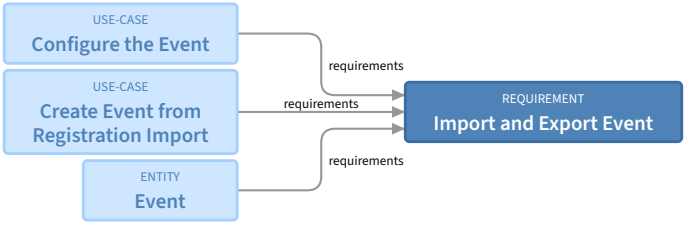


○ PRIORITY	○ PREMISES	Description
MUST	● **PREMISE:** **Registered Audience Known in Advance**, ● **PREMISE:** **Registration System Exchanges Attendee Data**	The system MUST generate each attendee's personal access URL containing event, user, and a "NNN-NNN" six-digit token and return it in an Excel sheet to the Event Registration System — **BECAUSE** the Event Registration System distributes the join URLs to attendees

● REQUIREMENT	Properties & Description

Import and Export Event ⚓

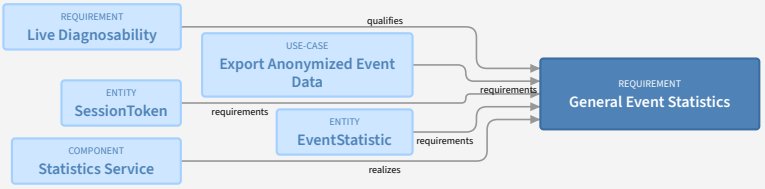
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD export an event with its related entities to a serialized format such as YAML and re-import it — **BECAUSE** this eases development, demonstration, and creation of recurring production events

General Event Statistics ⚓

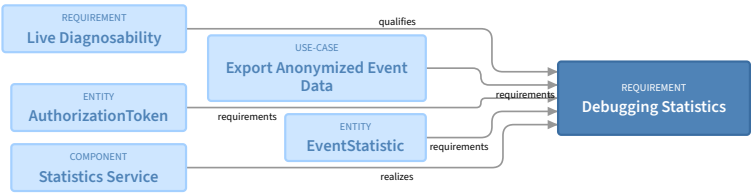
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	∅	The system MUST provide an attendee statistics curve of logged-in users over time — **BECAUSE** organizers need to observe audience size and trend during the event

Debugging Statistics ⚓

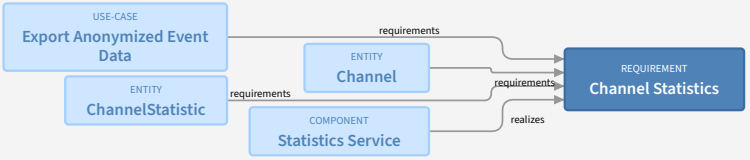
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD record login-challenge statistics including issued, sent, and used authorization tokens — **BECAUSE** operators need visibility into the authentication flow for diagnosis

Channel Statistics ⚓

Diagram

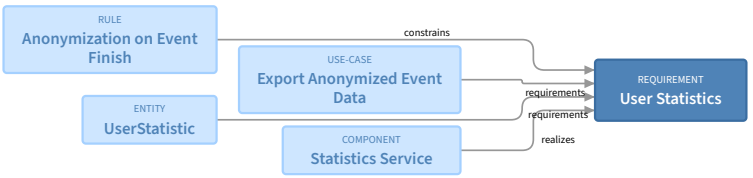


○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD show the number of viewers per channel of an event — **BECAUSE** organizers want to know the relative popularity of language and resolution variants

● REQUIREMENT	Properties & Description

User Statistics ⚓

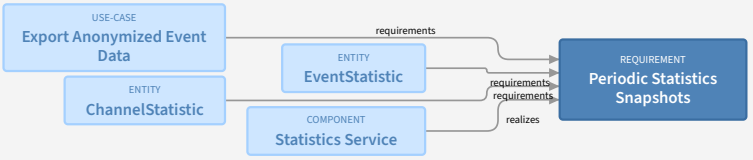
Diagram



○ PRIORITY	○ PREMISES	Description
SHOULD	∅	The system SHOULD record viewer statistics including country, browser, device, and viewport, using country to select the default application language on first use — **BECAUSE** audience composition informs both reporting and localization

Periodic Statistics Snapshots ⚓

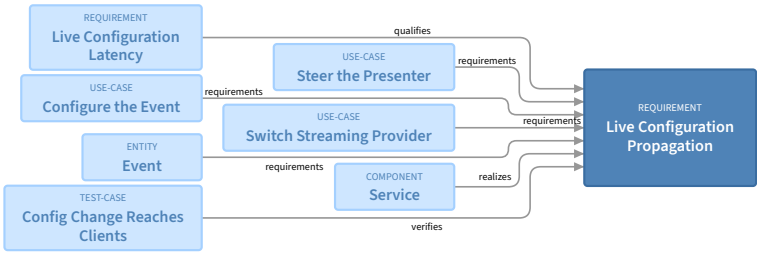
Diagram



○ PRIORITY	○ PREMISES	Description
MUST	∅	While an event is running, the system MUST capture a cumulative snapshot of its event and channel statistics every five minutes, independent of any user activity — **BECAUSE** a trend curve needs regularly sampled counts and not just the counts at the moments someone looks

Live Configuration Propagation ⚓

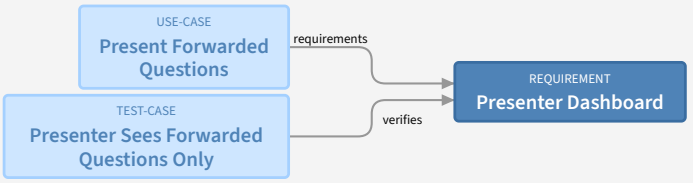
Diagram



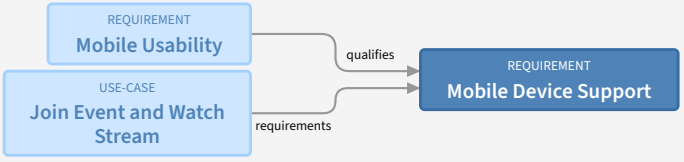
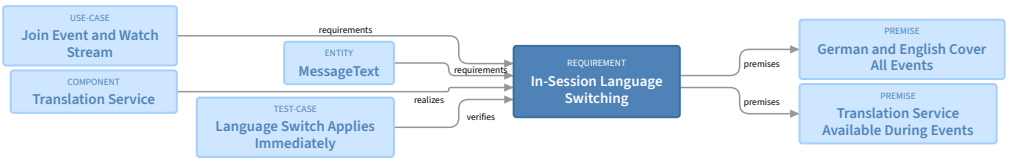
○ PRIORITY	○ PREMISES	Description
MUST	∅	The system MUST propagate event configuration changes directly to all connected participants — **BECAUSE** toggles such as enabling chat must take effect for the live audience without a reload

Presenter Dashboard ⚓

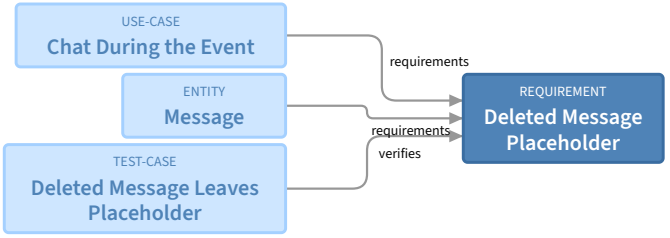
Diagram



○ PRIORITY	○ PREMISES	Description
COULD	∅	The system COULD provide a dedicated presenter dashboard showing forwarded inputs for processing on stage — **BECAUSE** the presenter needs a focused working view, though its design is not yet finalized

● REQUIREMENT	Properties & Description		
Multiple-Choice Feedback ⚓	Diagram		
	○ PRIORITY	○ PREMISES	Description
	COULD	∅	The system COULD provide real-time multiple-choice questions answered by attendees — **BECAUSE** polls add interaction during the event
Mobile Device Support ⚓	Diagram		
			
	○ PRIORITY	○ PREMISES	Description
	COULD	∅	The system COULD let participants take part on mobile phones, at least for video with possible interaction restrictions — **BECAUSE** many attendees join from mobile devices
In-Session Language Switching ⚓	Diagram		
			
	○ PRIORITY	○ PREMISES	Description
	SHOULD	● **PREMISE: German and English Cover All Events** ⚭, ● **PREMISE: Translation Service Available During Events** ⚭	The system SHOULD let an attendee switch the display and content language between German and English at any time from the header, applying the change immediately — **BECAUSE** attendees expect to read the interface and translated messages in their preferred one of the two supported languages
Light and Dark Theme ⚓	Diagram		
	○ PRIORITY	○ PREMISES	Description
	COULD	∅	The system COULD let an attendee toggle a light or dark theme from the header, applying it immediately across the interface — **BECAUSE** attendees view events under varied lighting and prefer a comfortable appearance

● REQUIREMENT	Properties & Description
Deleted Message Placeholder ⚓	Diagram

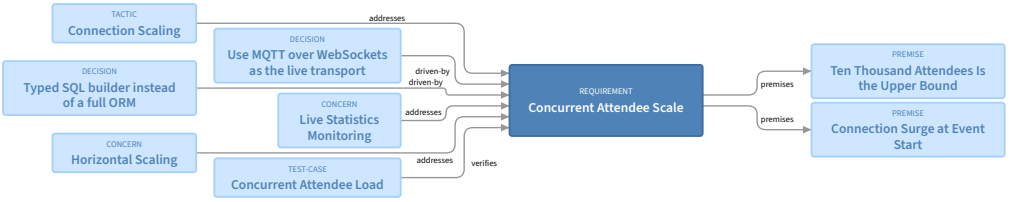


○ PRIORITY	○ PREMISES	Description
SHOULD	∅	When an attendee deletes a message, the system SHOULD retain its position in the stream and show a “This message was deleted” placeholder instead of removing it — **BECAUSE** preserving the gap keeps the surrounding conversation coherent for readers

07 ● REQS: Non-Functional Requirements (NR) ⚓

Concurrent Attendee Scale PRIORITY: MUST CATEGORY: Performance	**Horizontal Scalability** PRIORITY: MUST CATEGORY: Performance	**Streaming Quality** PRIORITY: MUST CATEGORY: Performance	**Provider Failover Continuity** PRIORITY: MUST CATEGORY: Reliability
Cross-Browser Compatibility PRIORITY: MUST CATEGORY: Compatibility	**GDPR Compliance** PRIORITY: MUST CATEGORY: Compliance	**Privacy by Design** PRIORITY: MUST CATEGORY: Security	**Unguessable Access Tokens** PRIORITY: MUST CATEGORY: Security
Isolated Persistence Node PRIORITY: MUST CATEGORY: Security	**Event Setup Recovery** PRIORITY: MUST CATEGORY: Reliability	**Event-Safe Maintenance** PRIORITY: MUST CATEGORY: Reliability	**Live Configuration Latency** PRIORITY: SHOULD CATEGORY: Performance
Per-Event Cost Efficiency PRIORITY: SHOULD CATEGORY: Constraint	**Interaction Abuse Throttling** PRIORITY: SHOULD CATEGORY: Security	**Start-Surge Asset Delivery** PRIORITY: SHOULD CATEGORY: Performance	**Compile-Time Contract Safety** PRIORITY: SHOULD CATEGORY: Maintainability
Live Diagnosability PRIORITY: SHOULD CATEGORY: Maintainability	**Mobile Usability** PRIORITY: COULD CATEGORY: Usability		

● REQUIREMENT	Properties & Description
Concurrent Attendee Scale ⚓	Diagram

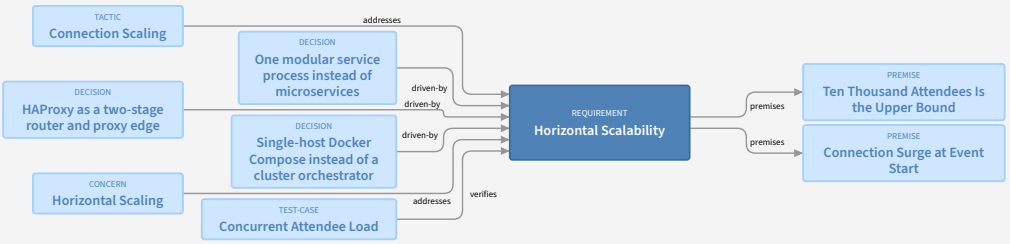


○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Performance	∅
○ PREMISES	**○ METRIC**	**Description**
● PREMISE: Ten Thousand Attendees Is the Upper Bound	2500 to 10000 concurrently connected attendees per event	The system MUST support between 2500 and 10000

● REQUIREMENT	Properties & Description		
	⚭, ● **PREMISE: Connection Surge at Event Start** ⚭		attendees connected to a single event at the same time — **BECAUSE** msg Filmstudio video productions must reach an audience of this size simultaneously

Horizontal Scalability ⚓

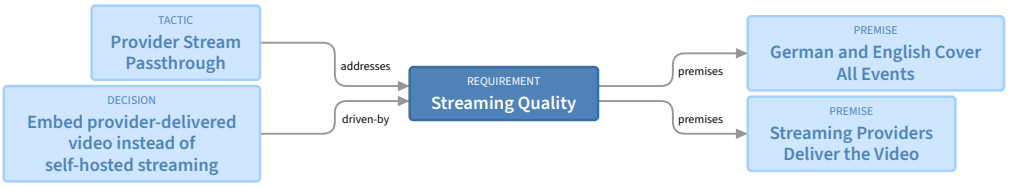
Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Performance	∅
○ PREMISES	**○ METRIC**	Description
● **PREMISE: Ten Thousand Attendees Is the Upper Bound** ⚭, ● **PREMISE: Connection Surge at Event Start** ⚭	10000 concurrent WebSocket connections per event	The system MUST scale horizontally by running multiple proxy, relay, and server instances per runtime environment so that 10000 concurrent WebSocket connections are served per event — **BECAUSE** a single instance cannot handle the required connection volume and response times

Streaming Quality ⚓

Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Performance	∅
○ PREMISES	**○ METRIC**	Description
● **PREMISE: German and English Cover All Events** ⚭, ● **PREMISE: Streaming Providers Deliver the Video** ⚭	1920x1080 pixels at 30 fps in 2 languages	The system MUST process and distribute video at 1080p30 (1920x1080 at 30 fps) in both German and English — **BECAUSE** productions are delivered in this defined quality and the two required languages

● REQUIREMENT	Properties & Description

Provider Failover Continuity

Diagram

○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Reliability	● **REQUIREMENT: Live Provider Switching**
○ PREMISES	**○ METRIC**	**Description**
● **PREMISE: Streaming Providers Deliver the Video**	all attendee clients on the fallback provider within 10 seconds	The system MUST switch to a fallback streaming provider during a running event with attendee clients following the switch automatically within seconds and without user interaction — **BECAUSE** provider outages must not interrupt the live audience

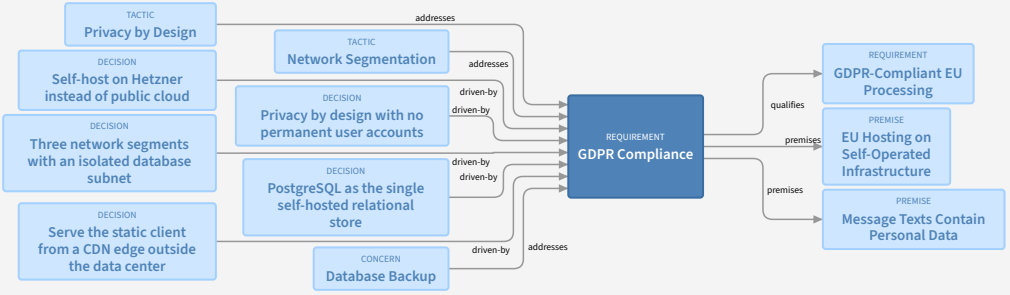
Cross-Browser Compatibility

Diagram

○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Compatibility	● **REQUIREMENT: Browser Access**
○ PREMISES	**○ METRIC**	**Description**
● **PREMISE: Attendee Networks Pass WebSocket Traffic**	the last 2 major versions of Chrome, Edge, Firefox, and Safari	The system MUST run in any reasonably recent version of the major web browsers (Chrome, Edge, Firefox, Safari) without plugins — **BECAUSE** attendees use heterogeneous managed and unmanaged devices

GDPR Compliance

Diagram

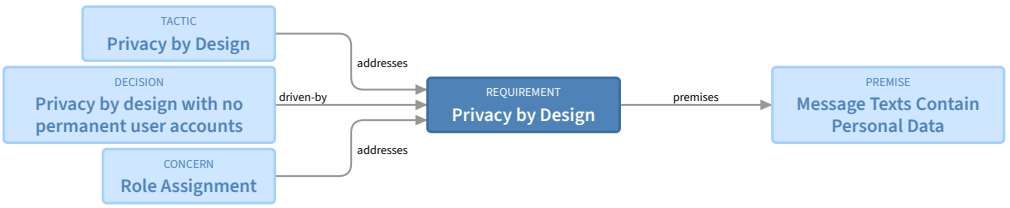


○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Compliance	● **REQUIREMENT: GDPR-Compliant EU Processing**

● REQUIREMENT	Properties & Description		
	○ PREMISES	○ METRIC	Description
	● **PREMISE: EU Hosting on Self-Operated Infrastructure**, ● **PREMISE: Message Texts Contain Personal Data**	100 % of personal data processed and hosted within the EU	The system MUST process all personal data in compliance with GDPR and host it exclusively within the EU (data center in Nürnberg, Germany) — **BECAUSE** the operator is legally bound to European data-protection requirements

Privacy by Design

Diagram

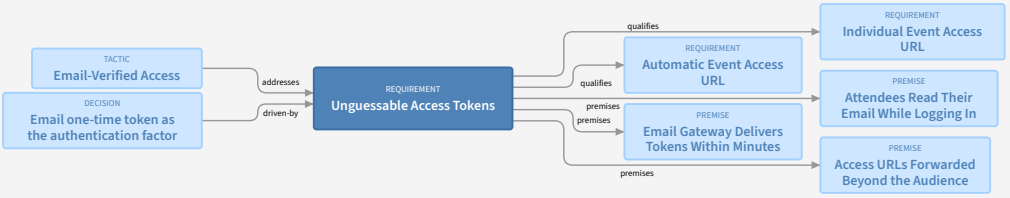


○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Security	∅

○ PREMISES	○ METRIC	Description
● **PREMISE: Message Texts Contain Personal Data**	0 attendee personal-data records retained after the event finish procedure	The system MUST retain attendee personal data only while an event runs, anonymizing or deleting all of it within the automated finish procedure when the event finishes — **BECAUSE** minimizing personal-data retention is the core privacy guarantee of the product

Unguessable Access Tokens

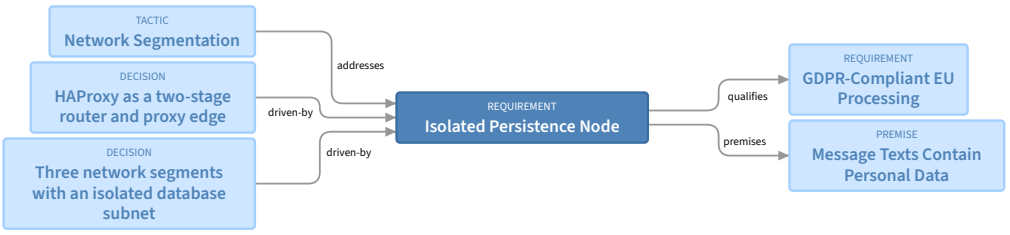
Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Security	● **REQUIREMENT: Individual Event Access URL**, ● **REQUIREMENT: Automatic Event Access URL**

○ PREMISES	○ METRIC	Description
● **PREMISE: Attendees Read Their Email While Logging In**, ● **PREMISE: Email Gateway Delivers Tokens Within Minutes**, ● **PREMISE: Access URLs Forwarded Beyond the Audience**	token expiry <= 5 minutes by default	The system MUST use unguessable event URLs and time-limited authorization tokens that by default expire within 5 minutes — **BECAUSE** weak access secrets would let unauthorized viewers join the event

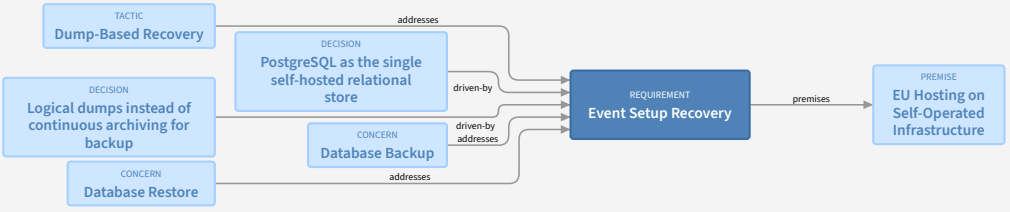
● REQUIREMENT	Properties & Description
Isolated Persistence Node ⚓	Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Security	● **REQUIREMENT: GDPR-Compliant EU Processing** ⚭
○ PREMISES	**○ METRIC**	Description
● **PREMISE: Message Texts Contain Personal Data** ⚭	0 routes to the database port from outside the business services	The system MUST keep the persistence node reachable from the business services alone, with no route from the Internet or from the connection-handling tiers — **BECAUSE** the persisted personal data is protected best by a node an attacker of the public entry point cannot even address

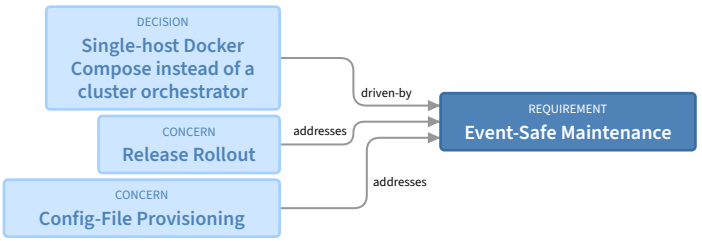
Event Setup Recovery ⚓

Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Reliability	∅
○ PREMISES	**○ METRIC**	Description
● **PREMISE: EU Hosting on Self-Operated Infrastructure** ⚭	RPO <= 24 hours (<= 1 hour right before an event) and RTO <= 2 hours	The system MUST allow the event setups and their messages to be restored after a database loss within 2 hours from a backup at most 24 hours old, and at most 1 hour old right before an event — **BECAUSE** a prepared event must not have to be re-entered under time pressure on the day of the broadcast

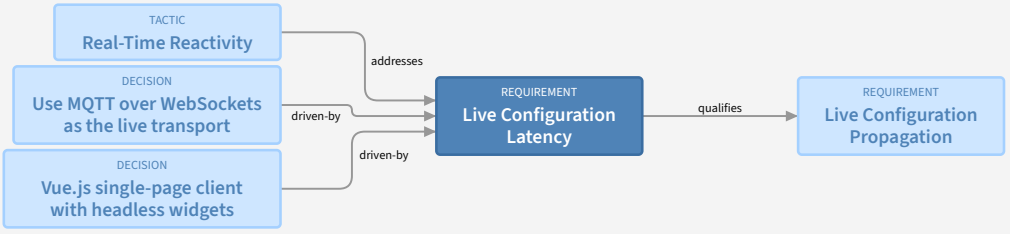
● REQUIREMENT	Properties & Description		
Event-Safe Maintenance ⚓	Diagram		



○ PRIORITY	○ CATEGORY	○ QUALIFIES
MUST	Reliability	∅
○ PREMISES	**○ METRIC**	Description
∅	0 rollouts or reconfigurations while an event runs	The system MUST be upgraded and reconfigured only while no event runs — **BECAUSE** a live audience cannot be asked to wait for a restart

Live Configuration Latency ⚓

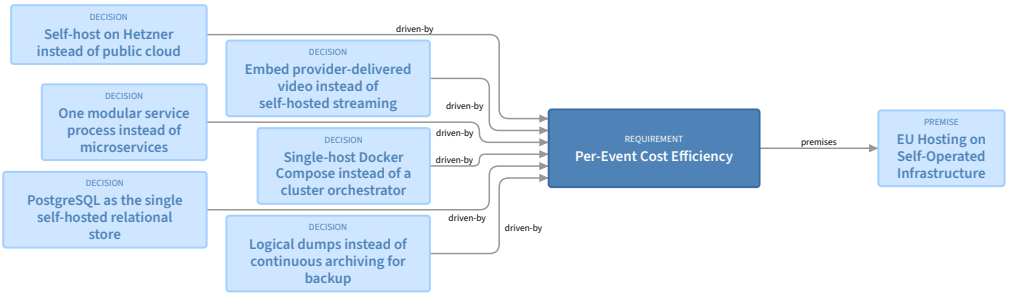
Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
SHOULD	Performance	● **REQUIREMENT: Live Configuration Propagation** ⚭
○ PREMISES	**○ METRIC**	Description
∅	propagation to all connected clients <= 2 seconds	The system SHOULD propagate an event configuration change to all connected clients within 2 seconds — **BECAUSE** operators expect toggles such as enabling chat to take effect near-instantly for the live audience

Per-Event Cost Efficiency ⚓

Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
SHOULD	Constraint	∅
○ PREMISES	**○ METRIC**	Description
● **PREMISE: EU Hosting on Self-Operated Infrastructure**	recurring cost per event below the equivalent Azure or AWS	The system SHOULD minimize the recurring cost per event by

● REQUIREMENT	Properties & Description		
		hosting cost	self-hosting on Hetzner infrastructure rather than Azure or AWS, excluding one-off development cost — **BECAUSE** low operating cost per event is a primary economic justification for building the solution
Interaction Abuse Throttling	Diagram		

○ PRIORITY	○ CATEGORY	○ QUALIFIES
SHOULD	Security	● **REQUIREMENT: Chat Messages**, ● **REQUIREMENT: Ask Questions**
○ PREMISES	**○ METRIC**	Description
● **PREMISE: Abusive Interaction from the Audience**	<= 10 submissions per user per minute by default, configurable per event	The system SHOULD throttle chat and question submissions to a configurable maximum per user per minute — **BECAUSE** rate limiting prevents denial-of-service abuse of the interaction channels

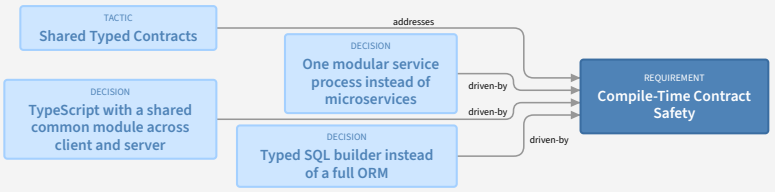
Start-Surge Asset Delivery	Diagram

○ PRIORITY	○ CATEGORY	○ QUALIFIES
SHOULD	Performance	● **REQUIREMENT: Browser Access**
○ PREMISES	**○ METRIC**	Description
● **PREMISE: Connection Surge at Event Start**	client bundle delivered within 3 seconds at 10000 simultaneous joins	The system SHOULD deliver the static client bundle to each of 10000 attendees joining within the same minute in at most 3 seconds — **BECAUSE** the audience arrives in a surge right at the start of the event and must not be held back by the download

● REQUIREMENT	Properties & Description

Compile-Time Contract Safety

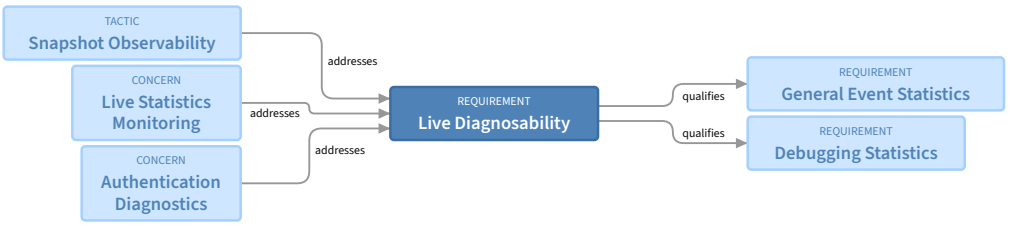
Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
SHOULD	Maintainability	∅
○ PREMISES	**○ METRIC**	Description
∅	100 % of the client/server message contracts type-checked at build time	The system SHOULD catch every mismatch between the message contracts of client and server at build time instead of at runtime — **BECAUSE** a contract slip surfacing only during a live event is the costliest defect the solution can have

Live Diagnosability

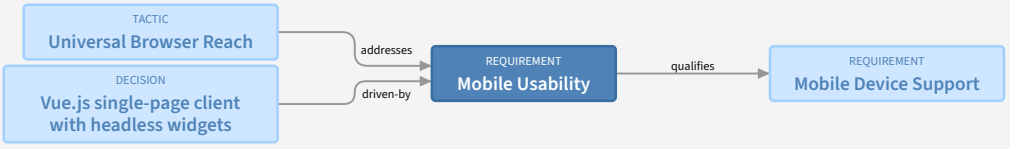
Diagram



○ PRIORITY	○ CATEGORY	○ QUALIFIES
SHOULD	Maintainability	● **REQUIREMENT:** General Event Statistics ⚭, ● **REQUIREMENT:** Debugging Statistics ⚭
○ PREMISES	**○ METRIC**	Description
∅	event counters at most 5 minutes old, every login failure traceable from token states and log	The system SHOULD let the administrator see the attendee, channel, and authentication-flow counters of a running event with at most 5 minutes delay and trace every login failure from the recorded token states and the log — **BECAUSE** capacity and access problems must be diagnosed while the audience is still online

Mobile Usability

Diagram



● REQUIREMENT	Properties & Description		
	○ PRIORITY	○ CATEGORY	○ QUALIFIES
	COULD	Usability	● REQUIREMENT: Mobile Device Support
	○ PREMISES	○ METRIC	Description
	∅	attendee screens fully usable at 360 px viewport width in portrait and landscape	The system COULD render the attendee experience responsively for mobile phones in landscape and portrait orientation — **BECAUSE** a substantial share of attendees join from mobile devices

08 ● REQS: Domain Rules (DR)

Single Active Session per Event CATEGORY: Constraint SOURCE: Business	**Access Requires List or Pattern Match** CATEGORY: Constraint SOURCE: Business	**Single Active Channel** CATEGORY: Constraint SOURCE: Domain	**Initial Channel Becomes Active** CATEGORY: Derivation SOURCE: Domain
Single Active Resource per Channel CATEGORY: Constraint SOURCE: Domain	**Initial Resource Becomes Active** CATEGORY: Derivation SOURCE: Domain	**Moderation Determines Initial Visibility** CATEGORY: Action SOURCE: Business	**Moderator Messages Auto-Accepted** CATEGORY: Action SOURCE: Business
Forwarded Messages Are Immutable CATEGORY: Constraint SOURCE: Business	**State Set Depends on Message Type** CATEGORY: Constraint SOURCE: Domain	**Sentiment Auto-Decision Threshold** CATEGORY: Derivation SOURCE: Business	**Anonymization on Event Finish** CATEGORY: Action SOURCE: Law
Likes Conserved as Count on Finish CATEGORY: Action SOURCE: Business	**Token Format and Validity** CATEGORY: Constraint SOURCE: Business	**No Permanent User Accounts** CATEGORY: Constraint SOURCE: Law	**Manager Retained for Export** CATEGORY: Constraint SOURCE: Business
Administrator Bypasses the Access List CATEGORY: Constraint SOURCE: Business	**Administrator Bootstrapped by Configuration** CATEGORY: Constraint SOURCE: Business		

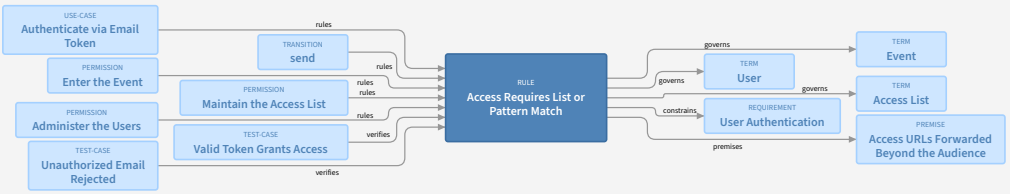
● RULE	Properties & Description			
Single Active Session per Event	Diagram			

○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Business	● TERM: User, ● TERM: Session Token, ● TERM: Event	● REQUIREMENT: Limit Parallel Access

● RULE	Properties & Description	
	○ PREMISES	Description
	● PREMISE: Access URLs Forwarded Beyond the Audience	A ● **User** MUST have at most one active ● **Session Token** per ● **Event** at any time; a new login MUST delete any existing ● **Session Token** and close its connection — **BECAUSE** concurrent sessions would allow access sharing beyond the invited ● **Attendee**.

Access Requires List or Pattern Match

Diagram

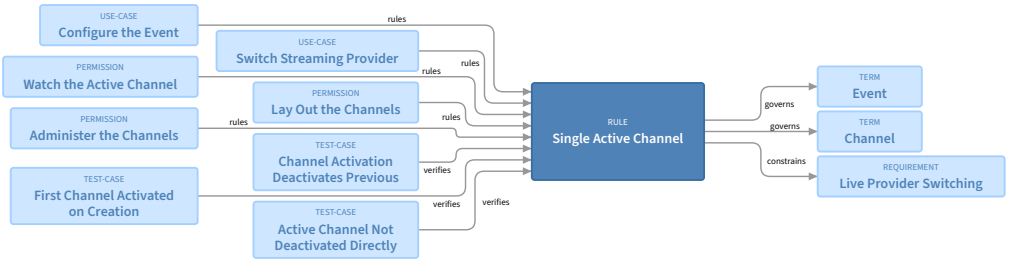


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Business	● TERM: User, ● TERM: Event, ● TERM: Access List	● REQUIREMENT: User Authentication

○ PREMISES	Description
● PREMISE: Access URLs Forwarded Beyond the Audience	A ● **User** not holding the ● **Administrator** ● **Role** (see ● **Administrator Bypasses the Access List**) MUST be granted access to an ● **Event** only according to the first matching case: 1. The email of the ● **User** is on the ● **Access List** of the ● **Event**: access is granted. 2. The email of the ● **User** matches the access email pattern of the ● **Event**: access is granted. 3. The ● **Event** allows anonymous access: access is granted. Otherwise the ● **User** MUST NOT be granted access — **BECAUSE** access must be limited to the audience the organizer intends.

Single Active Channel

Diagram



○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Domain	● TERM: Channel, ● TERM: Event	● REQUIREMENT: Live Provider Switching

○ PREMISES	Description
∅	Exactly one ● **Channel** of an ● **Event** MUST be active at any time once the ● **Event** has at least one ● **Channel**, so the active ● **Channel** MUST NOT be deactivated directly and MUST NOT be deleted while another ● **Channel** of the ● **Event** exists — **BECAUSE** ● **Attendee**s follow one logical content stream that defines the current ● **Event** feed.

● RULE

Initial Channel Becomes Active ⚓

Properties & Description

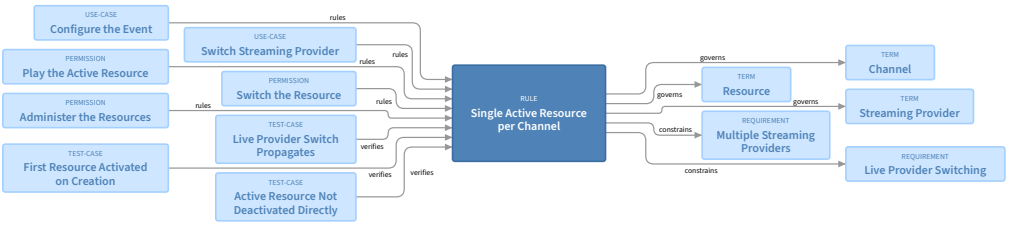
Diagram

○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Derivation	Domain	● **TERM:** Channel ⚭, ● **TERM:** Event ⚭	● **REQUIREMENT:** Live Provider Switching ⚭

○ PREMISES	Description
∅	A ● **Channel** created for an ● **Event** without an active ● **Channel** MUST become the active ● **Channel** of the ● **Event** — **BECAUSE** ● **Single Active Channel** has to hold from the first ● **Channel** on, without a ● **Manager** having to activate it.

Single Active Resource per Channel ⚓

Diagram

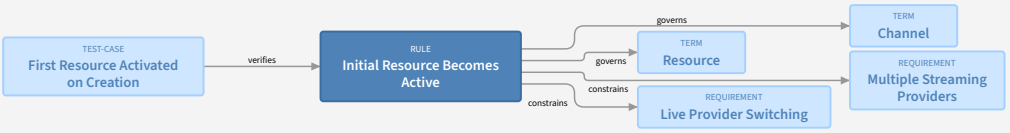


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Domain	● **TERM:** Resource ⚭, ● **TERM:** Channel ⚭, ● **TERM:** Streaming Provider ⚭	● **REQUIREMENT:** Live Provider Switching ⚭, ● **REQUIREMENT:** Multiple Streaming Providers ⚭

○ PREMISES	Description
∅	Exactly one ● **Resource** of a ● **Channel** MUST be active at any time once the ● **Channel** has at least one ● **Resource**, so the active ● **Resource** MUST NOT be deactivated directly and MUST NOT be deleted while another ● **Resource** of the ● **Channel** exists — **BECAUSE** switching the active ● **Resource** is the mechanism by which the live ● **Streaming Provider** is changed.

Initial Resource Becomes Active ⚓

Diagram



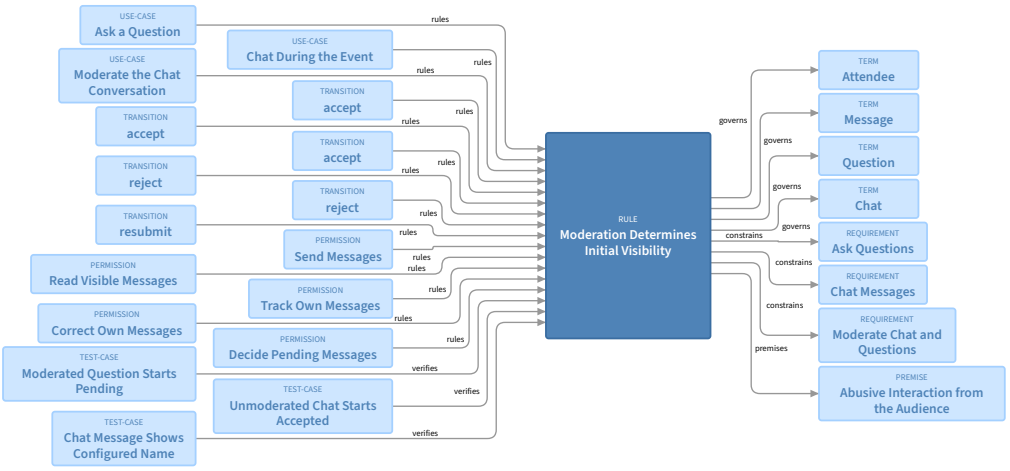
○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Derivation	Domain	● **TERM:** Resource ⚭, ● **TERM:** Channel ⚭	● **REQUIREMENT:** Live Provider Switching ⚭, ● **REQUIREMENT:** Multiple Streaming Providers ⚭

● RULE	Properties & Description

○ PREMISES	Description
∅	A ● **Resource** created for a ● **Channel** without an active ● **Resource** MUST become the active ● **Resource** of the ● **Channel** — **BECAUSE** ● **Single Active Resource per Channel** has to hold from the first ● **Resource** on, without a ● **Manager** having to activate it.

Moderation Determines Initial Visibility ⚓

Diagram

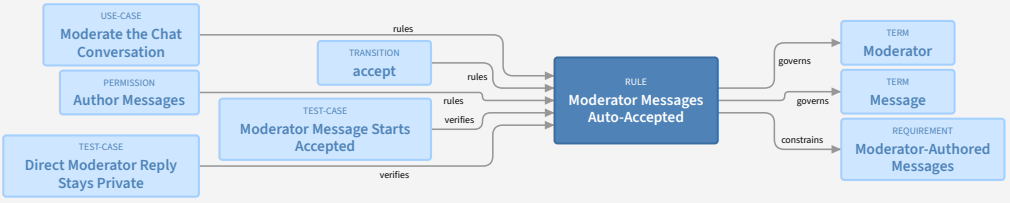


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Action	Business	● **TERM:** Message ⚭, ● **TERM:** Chat ⚭, ● **TERM:** Question ⚭, ● **TERM:** Attendee ⚭	● **REQUIREMENT:** Moderate Chat and Questions ⚭, ● **REQUIREMENT:** Chat Messages ⚭, ● **REQUIREMENT:** Ask Questions ⚭

○ PREMISES	Description
● **PREMISE:** Abusive Interaction from the Audience ⚭	When moderation is enabled for ● **Chat**s or ● **Question**s, a ● **Message** of that type sent by an ● **Attendee** MUST stay in state pending and become visible only once accepted; when disabled, it MUST be accepted at once upon creation — **BECAUSE** organizers must control what the audience sees.

Moderator Messages Auto-Accepted ⚓

Diagram



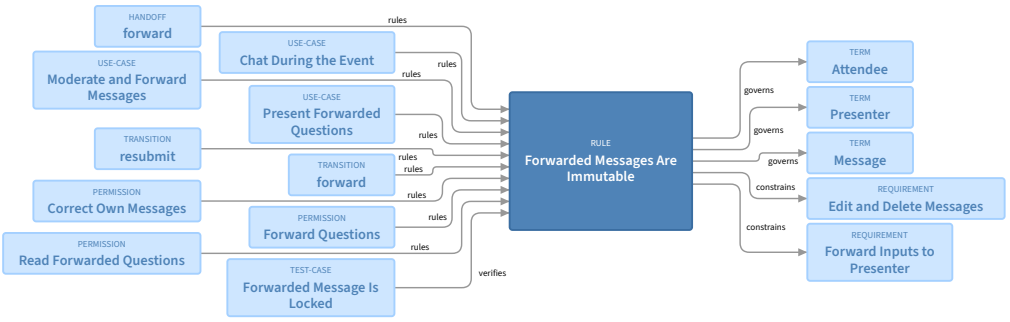
○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Action	Business	● **TERM:** Message ⚭, ● **TERM:** Moderator ⚭	● **REQUIREMENT:** Moderator-Authored Messages ⚭

○ PREMISES	Description
∅	A ● **Message** authored by a ● **Moderator** MUST be accepted at once upon creation and carry the sender name “Moderator” unless overridden — **BECAUSE** ● **Moderator**-authored content is trusted and needs no further approval.

● RULE	Properties & Description

Forwarded Messages Are Immutable ⚓

Diagram

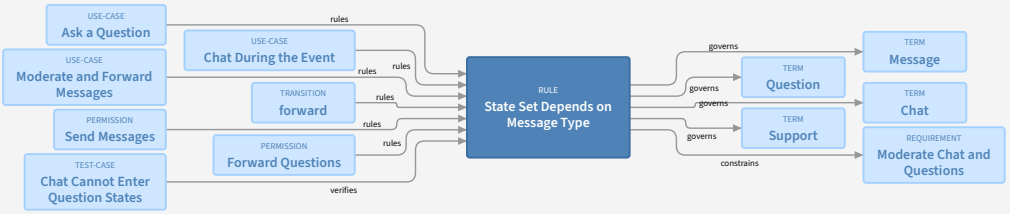


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Business	● **TERM:** Message ⚭, ● **TERM:** Attendee ⚭, ● **TERM:** Presenter ⚭	● **REQUIREMENT:** Edit and Delete Messages ⚭, ● **REQUIREMENT:** Forward Inputs to Presenter ⚭

○ PREMISES	Description
∅	Once a ● **Message** reaches state forwarded, the ● **Attendee** MUST NOT edit or delete it — **BECAUSE** the ● **Presenter** relies on the forwarded content remaining stable for live processing.

State Set Depends on Message Type ⚓

Diagram

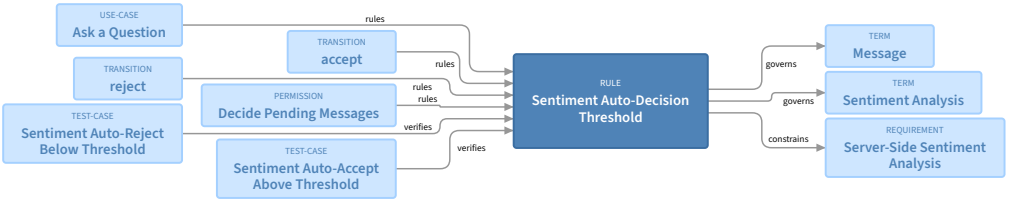


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Domain	● **TERM:** Message ⚭, ● **TERM:** Question ⚭, ● **TERM:** Chat ⚭, ● **TERM:** Support ⚭	● **REQUIREMENT:** Moderate Chat and Questions ⚭

○ PREMISES	Description
∅	A ● **Message** MUST use only the states its type permits: 1. The ● **Message** is a ● **Question**: pending, accepted, rejected, forwarded, answered, and suspended. 2. The ● **Message** is a ● **Chat**: pending, accepted, and rejected. 3. The ● **Message** is a ● **Support** message: accepted. No other combination of ● **Message** type and state is permitted — **BECAUSE** each ● **Message** type has a different interaction lifecycle.

Sentiment Auto-Decision Threshold ⚓

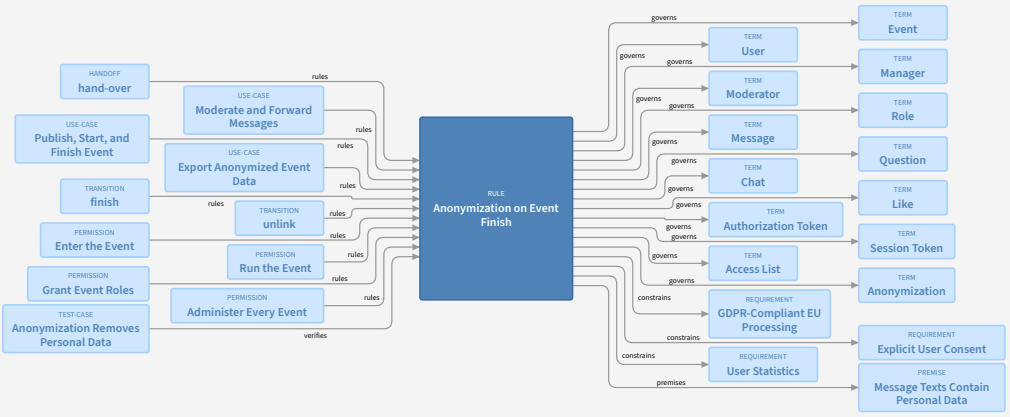
Diagram



● RULE	Properties & Description			
	○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
	Derivation	Business	● TERM: Sentiment Analysis ⚭, ● TERM: Message ⚭	● REQUIREMENT: Server-Side Sentiment Analysis ⚭
	○ PREMISES	Description		
	∅	When server-side ● **Sentiment Analysis** is enabled, a sentiment score of a ● **Message** below -0.1 MUST be treated as improper and a score at or above it as proper for auto-accept and auto-reject decisions — **BECAUSE** a defined threshold makes automated moderation predictable.		

Anonymization on Event Finish ⚓

Diagram



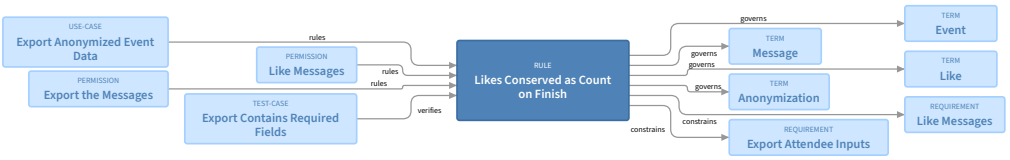
○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Action	Law	● TERM: Anonymization ⚭, ● TERM: Event ⚭, ● TERM: Message ⚭, ● TERM: Chat ⚭, ● TERM: Question ⚭, ● TERM: Like ⚭, ● TERM: Authorization Token ⚭, ● TERM: Session Token ⚭, ● TERM: Access List ⚭, ● TERM: User ⚭, ● TERM: Role ⚭, ● TERM: Moderator ⚭, ● TERM: Manager ⚭	● REQUIREMENT: Explicit User Consent ⚭, ● REQUIREMENT: GDPR-Compliant EU Processing ⚭, ● REQUIREMENT: User Statistics ⚭
○ PREMISES	Description		
● PREMISE: Message Texts Contain Personal Data ⚭	When an ● **Event** finishes, its ● **Anonymization** MUST reduce its ● **Messages** to a bare ● **Like** count with ● **Chat** and ● **Question** sender names set to “Anonymous” and liker and sender relations dropped, delete its ● **Authorization Tokens** and ● **Session Tokens**, clear its ● **Access List** and delete the referenced ● **Users** while retaining their viewer statistics unlinked from any ● **User**, and delete its ● **Moderator** ● **Roles** while retaining its ● **Manager** ● **Roles** — **BECAUSE** personal data must not be retained beyond the ● **Event** (GDPR Art. 5(1)(e), storage limitation).		

● RULE

Likes Conserved as Count on Finish ⚓

Properties & Description

Diagram

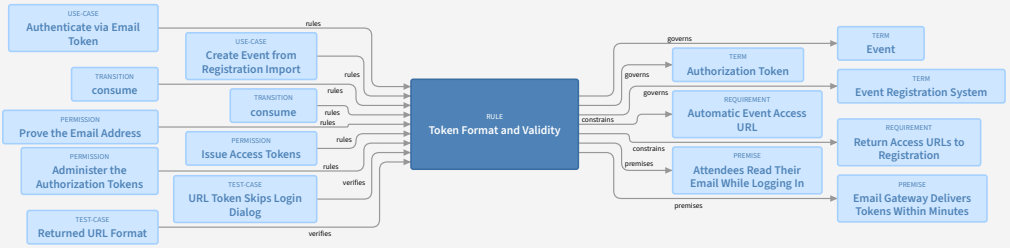


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Action	Business	● **TERM:** Like ⚭, ● **TERM:** Message ⚭, ● **TERM:** Event ⚭, ● **TERM:** Anonymization ⚭	● **REQUIREMENT:** Like Messages ⚭, ● **REQUIREMENT:** Export Attendee Inputs ⚭

○ PREMISES	Description
∅	When an ● **Event** finishes, the number of ● **Like**s of each ● **Message** MUST be computed and stored as its bare ● **Like** count before the liker relations are removed — **BECAUSE** ● **Like** totals must survive the ● **Anonymization** for the exported record.

Token Format and Validity ⚓

Diagram



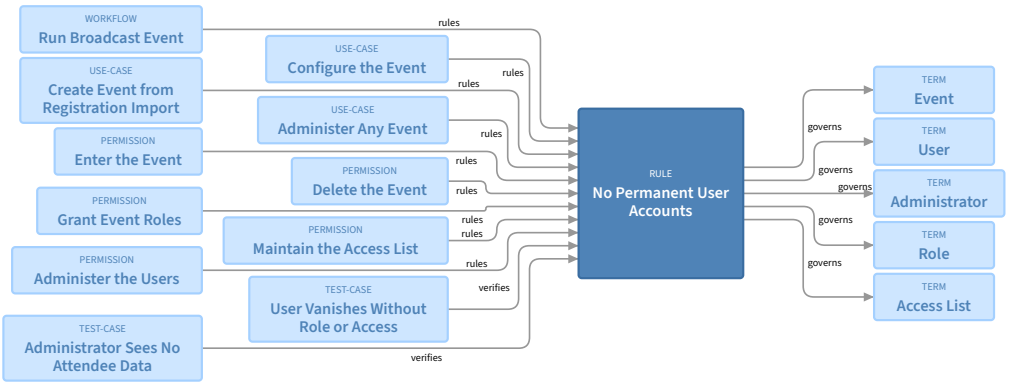
○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Business	● **TERM:** Authorization Token ⚭, ● **TERM:** Event ⚭, ● **TERM:** Event Registration System ⚭	● **REQUIREMENT:** Return Access URLs to Registration ⚭, ● **REQUIREMENT:** Automatic Event Access URL ⚭

○ PREMISES	Description
● **PREMISE:** Attendees Read Their Email While Logging In ⚭, ● **PREMISE:** Email Gateway Delivers Tokens Within Minutes ⚭	An ● **Authorization Token** MUST be formatted as six digits “NNN-NNN”; a normal login token MUST expire within 5 minutes while a token pre-generated for the ● **Event Registration System** lasts until the ● **Event** ends, expiring on first use only if the ● **Event** so configures — **BECAUSE** token strength and lifetime balance security against convenient automated access.

● RULE	Properties & Description

No Permanent User Accounts ⚓

Diagram

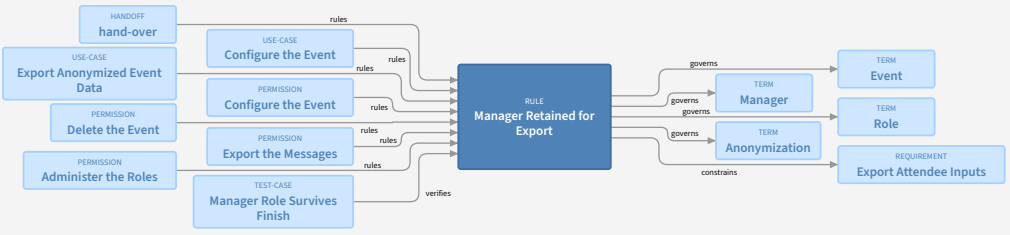


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Law	● **TERM: User** ⚭, ● **TERM: Role** ⚭, ● **TERM: Administrator** ⚭, ● **TERM: Event** ⚭, ● **TERM: Access List** ⚭	∅

○ PREMISES	Description
∅	A ● **User** MUST NOT exist as a permanent account, except the holder of the ● **Administrator** ● **Role** bootstrapped from the configuration: any other ● **User** exists only while granted a ● **Role**, present on the ● **Access List** of an ● **Event**, or joining via a matching access email pattern — **BECAUSE** privacy by design (GDPR Art. 25) forbids persistent personal data beyond operational need.

Manager Retained for Export ⚓

Diagram

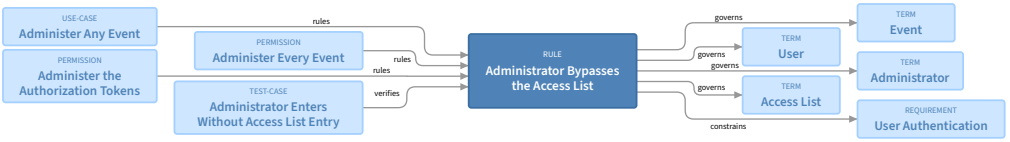


○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Business	● **TERM: Role** ⚭, ● **TERM: Manager** ⚭, ● **TERM: Event** ⚭, ● **TERM: Anonymization** ⚭	● **REQUIREMENT: Export Attendee Inputs** ⚭

○ PREMISES	Description
∅	A ● **Manager** ● **Role** MUST be retained after an ● **Event** finishes until the ● **Event** is deleted entirely — **BECAUSE** ● **Manager**s must still export the ● **Event** data left by the ● **Anonymization**.

Administrator Bypasses the Access List ⚓

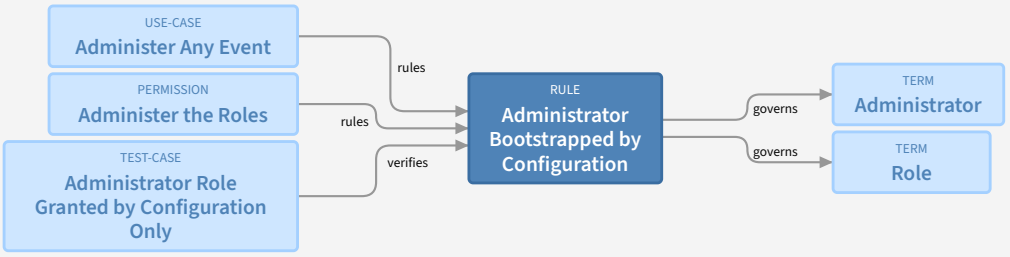
Diagram



● RULE	Properties & Description			
	○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
	Constraint	Business	● TERM: Administrator ⚭, ● TERM: User ⚭, ● TERM: Event ⚭, ● TERM: Access List ⚭	● REQUIREMENT: User Authentication ⚭
	○ PREMISES	Description		
	∅	A ● **User** holding the ● **Administrator** ● **Role** MUST be granted access to every ● **Event** through the ordinary login challenge of the configured email address, without an ● **Access List** entry, an access email pattern match, or anonymous access — **BECAUSE** the ● **Administrator** exists before any ● **Event** and hence cannot be enrolled on its ● **Access List**.		

Administrator Bootstrapped by Configuration ⚓

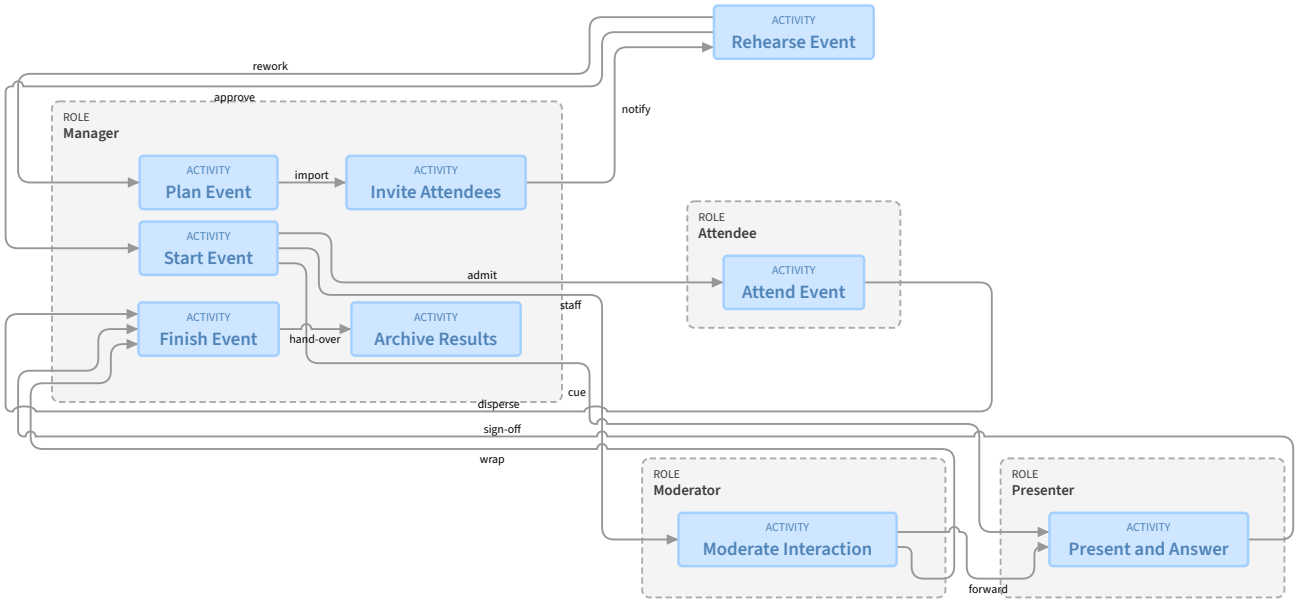
Diagram



○ CATEGORY	○ SOURCE	○ GOVERNS	○ CONSTRAINS
Constraint	Business	● TERM: Administrator ⚭, ● TERM: Role ⚭	∅
○ PREMISES	Description		
∅	An ● **Administrator** ● **Role** MUST be granted and revoked through the software configuration only, never through the solution itself — **BECAUSE** the ● **Administrator** is the bootstrap identity from which every ● **Event** and ● **Manager** originates and hence cannot depend on a grant inside the solution.		

09 ● REQS: Domain Workflows (DW)

09.01 ● WORKFLOW: Run Broadcast Event



○ ACTORS	● ROLE: Manager, ● ROLE: Presenter, ● ROLE: Moderator, ● ROLE: Attendee
○ TRIGGER	The manager receives the attendee list of an upcoming event, exported by the Event Registration System.
○ OUTCOME	The event took place, its personal data is anonymized, and its interaction results are exported.
○ JOURNEYS	● JOURNEY: Attend a Broadcast event, ● JOURNEY: Operate a Broadcast Event, ● JOURNEY: Moderate a Broadcast Event
○ TERMS	● TERM: Anonymization
○ RULES	● RULE: No Permanent User Accounts

The manager prepares and populates the event, the operators rehearse it, the audience attends it live while the moderators and the presenter handle the interaction, and the manager finishes and archives it afterwards — **BECAUSE** a broadcast event is a collaboration of several roles across days, of which each sitting with the solution covers only a part.

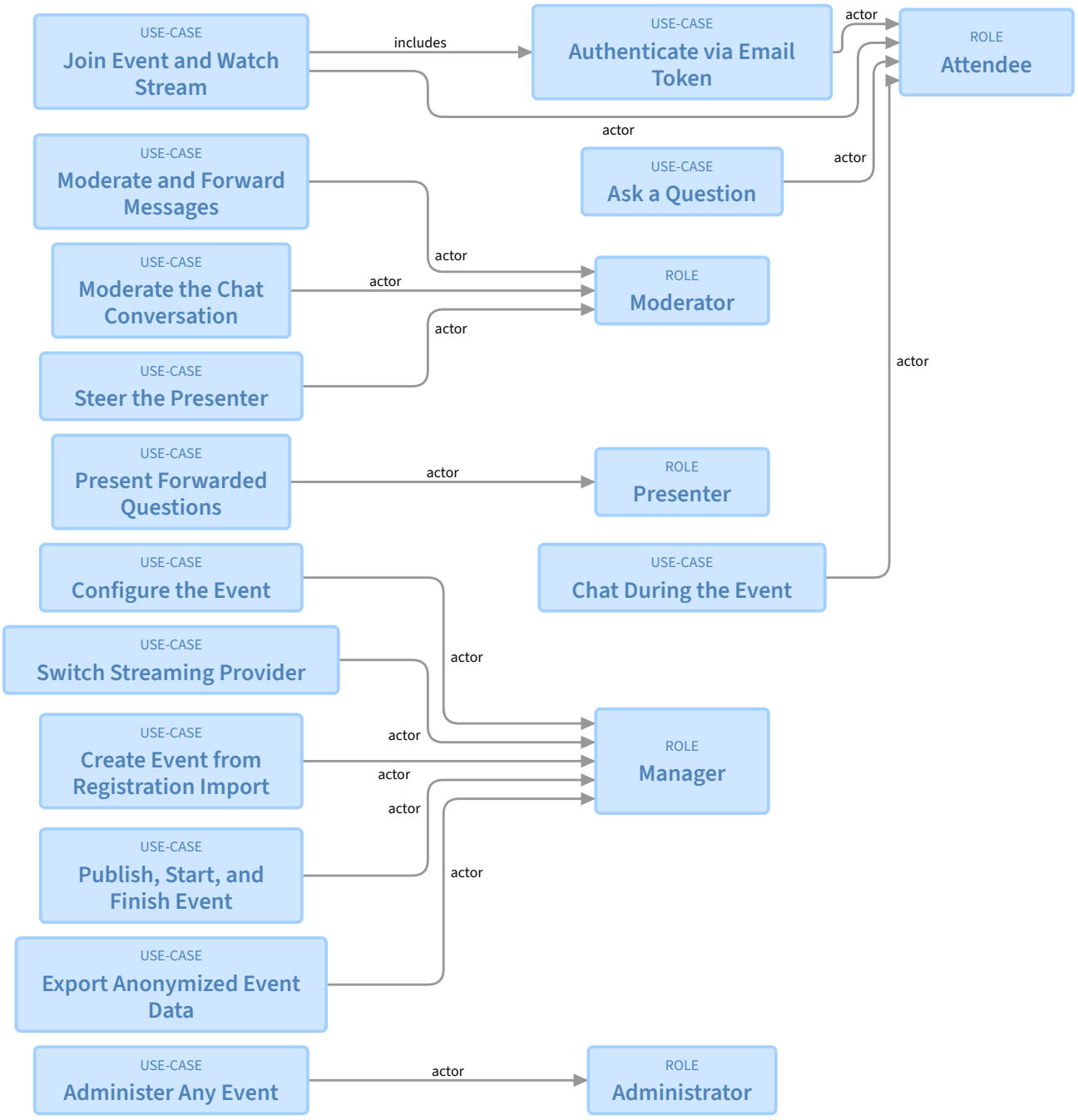
● ACTIVITY	○ ACTOR	○ INITIAL	○ FINAL	Description
Plan Event	● ROLE: Manager	true	∅	The manager configures the event and its channel and imports the attendee list of the Event Registration System, which yields the individual access URLs
Invite Attendees	● ROLE: Manager	∅	∅	The manager hands the individual access URLs back to the Event Registration System, which sends the invitation emails (outside the solution)
Rehearse Event	● ROLE: Presenter, ● ROLE: Moderator	∅	∅	The presenter and the moderators walk through the unpublished event together, checking the stream, the question flow, and the chat

● ACTIVITY	○ ACTOR	○ INITIAL	○ FINAL	Description
Start Event	● ROLE: Manager	∅	∅	The manager publishes and starts the event when the stream goes on air and keeps the streaming resource healthy
Attend Event	● ROLE: Attendee	∅	∅	The attendees join via their individual URLs, watch the stream, and raise questions and chat messages
Moderate Interaction	● ROLE: Moderator	∅	∅	The moderators accept, reject, answer, and forward the incoming messages throughout the live event
Present and Answer	● ROLE: Presenter	∅	∅	The presenter delivers the talk and answers the forwarded questions on air
Finish Event	● ROLE: Manager	∅	∅	The manager finishes the event once the stream is off air, which closes the access and anonymizes the personal data
Archive Results	● ROLE: Manager	∅	true	The manager exports the anonymized messages and statistics and hands them over to the organizer

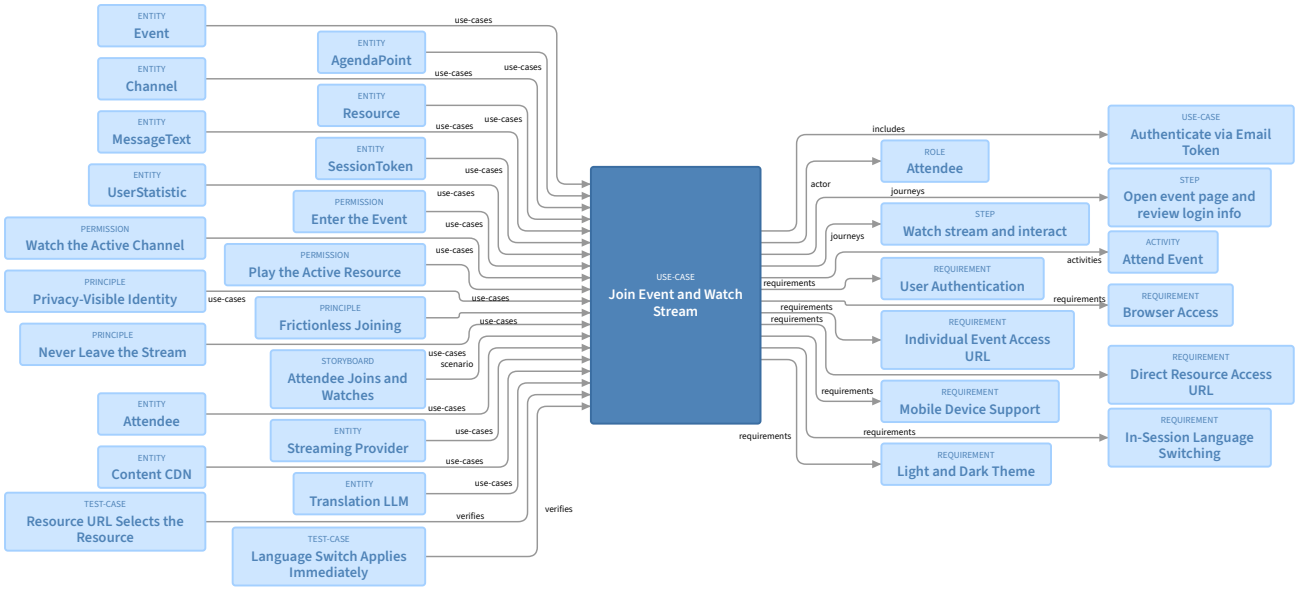
● HANDOFF	○ FROM	○ TO	○ CONDITION	○ RULES	Description
import	● ACTIVITY: Plan Event	● ACTIVITY: Invite Attendees	∅	∅	The generated individual access URLs are returned as a sheet the manager passes on to the Event Registration System
notify	● ACTIVITY: Invite Attendees	● ACTIVITY: Rehearse Event	∅	∅	The presenter and the moderators receive their access URLs and the rehearsal date by email
rework	● ACTIVITY: Rehearse Event	● ACTIVITY: Plan Event	The rehearsal revealed a defect in the event configuration.	∅	The presenter reports the defect to the manager, who corrects the configuration before a further rehearsal
approve	● ACTIVITY: Rehearse Event	● ACTIVITY: Start Event	The rehearsal passed.	∅	The presenter confirms the readiness of the event to the manager
admit	● ACTIVITY: Start Event	● ACTIVITY: Attend Event	∅	∅	The started event admits the invited attendees through their individual URLs
staff	● ACTIVITY: Start Event	● ACTIVITY: Moderate Interaction	∅	∅	The started event opens the moderation views to the moderators
cue	● ACTIVITY: Start Event	● ACTIVITY: Present and Answer	∅	∅	The manager cues the presenter once the stream is on air

● HANDOFF	○ FROM	○ TO	○ CONDITION	○ RULES	Description
forward	● **ACTIVITY: Moderate Interaction**	● **ACTIVITY: Present and Answer**	∅	● **RULE: Forwarded Messages Are Immutable**	The moderator forwards an accepted question, which appears in the presenter's view
disperse	● **ACTIVITY: Attend Event**	● **ACTIVITY: Finish Event**	∅	∅	The audience leaves when the stream ends, which the manager sees in the attendance statistics
wrap	● **ACTIVITY: Moderate Interaction**	● **ACTIVITY: Finish Event**	∅	∅	The moderators report the interaction as settled to the manager
sign-off	● **ACTIVITY: Present and Answer**	● **ACTIVITY: Finish Event**	∅	∅	The presenter signs off and the stream goes off air
hand-over	● **ACTIVITY: Finish Event**	● **ACTIVITY: Archive Results**	∅	● **RULE: Anonymization on Event Finish**, ● **RULE: Manager Retained for Export**	The finished event keeps the anonymized data and the manager role for the export

10 ● REQS: Use Cases (UC) ⚓



10.01 ● USE-CASE: Join Event and Watch Stream ⚓



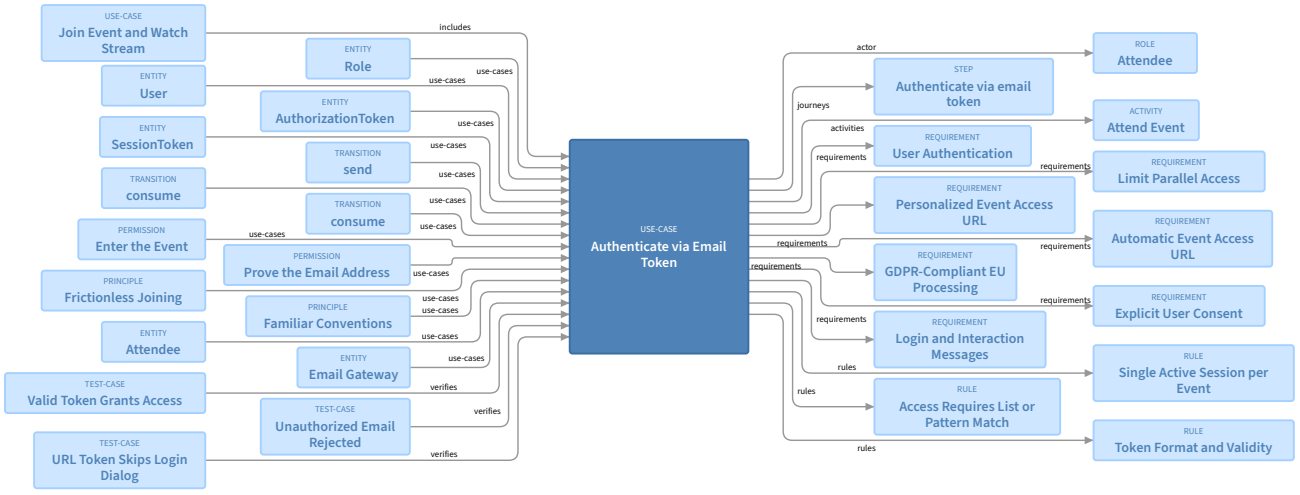
○ ACTOR	● ROLE: Attendee ⚭
○ JOURNEYS	● STEP: Open event page and review login info ⚭, ● STEP: Watch stream and interact ⚭
○ ACTIVITIES	● ACTIVITY: Attend Event ⚭
○ REQUIREMENTS	● REQUIREMENT: Individual Event Access URL ⚭, ● REQUIREMENT: User Authentication ⚭, ● REQUIREMENT: Browser Access ⚭, ● REQUIREMENT: Direct Resource Access URL ⚭, ● REQUIREMENT: Mobile Device Support ⚭, ● REQUIREMENT: In-Session Language Switching ⚭, ● REQUIREMENT: Light and Dark Theme ⚭
○ INCLUDES	● USE-CASE: Authenticate via Email Token ⚭
○ PRE-CONDITION	The attendee holds an event URL and their email is granted access.
○ TRIGGER	The attendee opens the individual event URL received by invitation.
○ POST-CONDITION	The attendee holds an active session token and sees the stream.

The attendee opens the individual event URL in a web browser, on a desktop or a mobile phone, authenticates via their email, and is admitted to the live stream of the event, adjusting language and theme to their liking — **BECAUSE** watching the live stream is the very reason the attendee was invited.

● SCENARIO	Properties & Description		
Join via Individual URL ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The attendee opens the individual event URL in any recent web browser, without installing software. 2. The attendee authenticates (● **Authenticate via Email Token**). 3. The system grants the attendee access to the live video stream.	
Return via Direct Resource URL ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The attendee sees the resource the URL names instead of the	1. The attendee opens an event URL carrying a resource parameter, typically by reloading the page.	

● SCENARIO	Properties & Description		
	default stream of the event.	2. The attendee authenticates (● **Authenticate via Email Token**), unless a session is still active. 3. The system grants access and selects the named stream or static resource of the event.	
Join on a Mobile Phone ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The attendee watches the stream on the phone, with the interaction possibly restricted.	1. The attendee opens the individual event URL in the browser of a mobile phone. 2. The attendee authenticates (● **Authenticate via Email Token**). 3. The system grants access to the live video stream in a layout fitting the phone screen in portrait or landscape orientation.	
Switch the Language ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	3
	○ OUTCOME	Description	
	The attendee sees the interface and the translated messages in the chosen language.	3. The system grants the attendee access to the live video stream in the default language. 4. The attendee switches the language between German and English in the header. 5. The system applies the language immediately to the interface and the translated messages.	
Toggle the Theme ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	3
	○ OUTCOME	Description	
	The attendee sees the interface in the chosen light or dark theme.	3. The system grants the attendee access to the live video stream in the default theme. 4. The attendee toggles between the light and the dark theme in the header. 5. The system applies the theme immediately across the whole interface.	

10.02 ● USE-CASE: Authenticate via Email Token ⚓



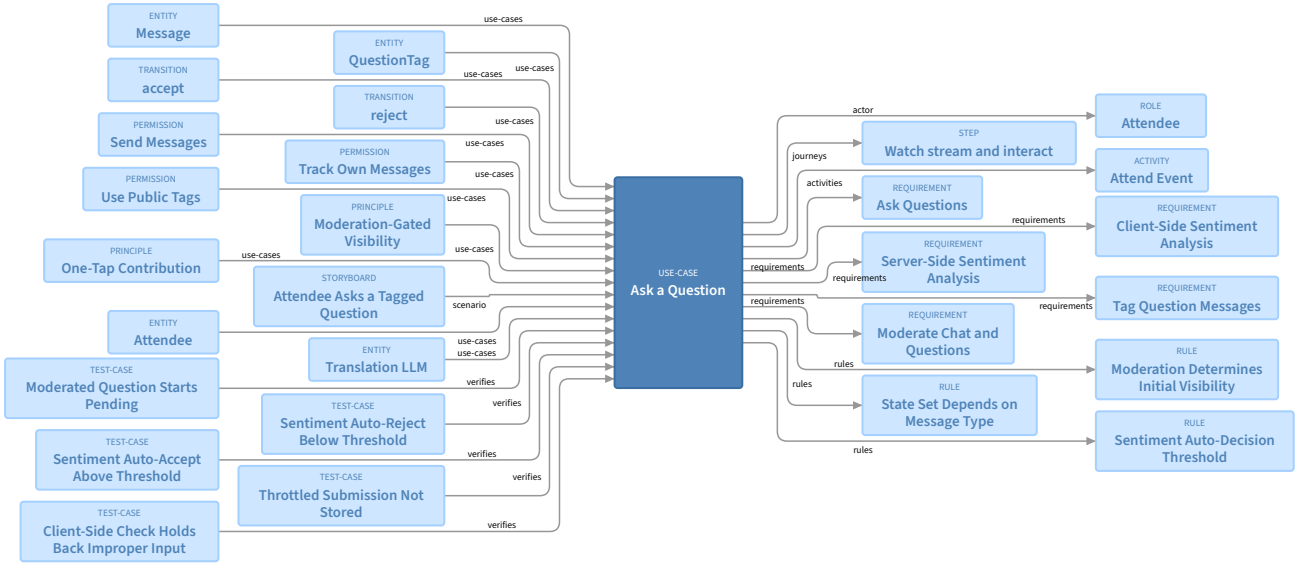
○ ACTOR	● ROLE: Attendee
○ JOURNEYS	● STEP: Authenticate via email token
○ ACTIVITIES	● ACTIVITY: Attend Event
○ REQUIREMENTS	● REQUIREMENT: User Authentication, ● REQUIREMENT: Explicit User Consent, ● REQUIREMENT: Limit Parallel Access, ● REQUIREMENT: Personalized Event Access URL, ● REQUIREMENT: Automatic Event Access URL, ● REQUIREMENT: Login and Interaction Messages, ● REQUIREMENT: GDPR-Compliant EU Processing
○ RULES	● RULE: Access Requires List or Pattern Match, ● RULE: Single Active Session per Event, ● RULE: Token Format and Validity
○ PRE-CONDITION	The attendee's email is granted access to the event.
○ TRIGGER	The attendee requests access to an event without holding an active session.
○ POST-CONDITION	The attendee holds an active session token and any prior session of the same user is closed.

The attendee proves control of their email via a one-time token, accepts any required consent, and receives a session while any prior session for the same user is closed; this flow is included by every use case requiring an authenticated attendee — **BECAUSE** only a proven identity entitles the attendee to the event they were invited to.

● SCENARIO	Properties & Description		
Login with Emailed Token	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The system asks for the email address and presents the optional login message with its links. 2. The attendee enters their email address. 3. The system sends a six-digit authorization token to that email. 4. The attendee enters the received token. 5. The system validates the token and any consent the event requires for GDPR-compliant participation. 6. The system issues a session token and closes any prior session of the user.	
Email Pre-Filled from URL	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Resume	2
	○ OUTCOME	Description	
	The email address is fixed by the personalized event URL, and the flow resumes at step 3.	2. The system finds the email address embedded in the event URL and presents it read-only instead of asking for it.	
Automatic Token in URL	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The attendee holds an active session token, obtained without entering email or token.	1. The system finds the email and a pre-generated token embedded in the event URL. 2. The system validates the embedded token against the event settings. 3. The system issues a session token without asking for the email address.	
Email Not Authorized	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Exceptional	Resume	3

● SCENARIO	Properties & Description	
	○ OUTCOME	Description
	No session exists and no token was sent, and the attendee may retry with another email address (resumes at step 1).	3. The system finds the email neither on the access list nor matching the access pattern. 4. The system denies access and informs the attendee that they are not authorized.

10.03 ● USE-CASE: Ask a Question ⚓



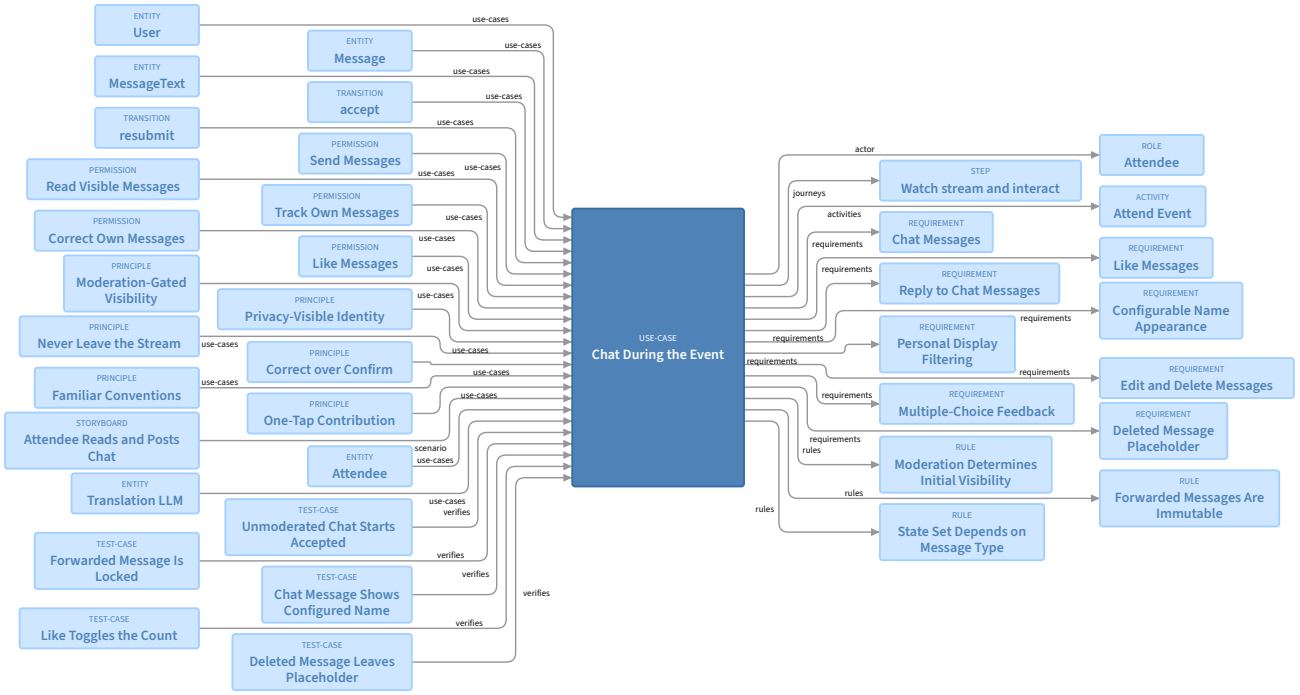
○ ACTOR	● ROLE: Attendee ⚭
○ JOURNEYS	● STEP: Watch stream and interact ⚭
○ ACTIVITIES	● ACTIVITY: Attend Event ⚭
○ REQUIREMENTS	● REQUIREMENT: Ask Questions ⚭, ● REQUIREMENT: Tag Question Messages ⚭, ● REQUIREMENT: Moderate Chat and Questions ⚭, ● REQUIREMENT: Server-Side Sentiment Analysis ⚭, ● REQUIREMENT: Client-Side Sentiment Analysis ⚭
○ RULES	● RULE: Moderation Determines Initial Visibility ⚭, ● RULE: State Set Depends on Message Type ⚭, ● RULE: Sentiment Auto-Decision Threshold ⚭
○ PRE-CONDITION	The attendee has an active session and questions are enabled.
○ TRIGGER	The attendee decides to raise a question during the running event.
○ POST-CONDITION	The question is stored in state pending and awaits moderation.

The attendee writes a question, optionally tags it, and submits it; the system stores it and routes it through moderation if the event requires approval before it becomes visible — **BECAUSE** getting their own question answered turns the attendee from a passive viewer into a participant.

● SCENARIO	Properties & Description		
	○ TYPE	○ RESULT	○ AT-MAIN-STEP
Submit Moderated Question ⚓	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The attendee starts a new question. 2. The attendee writes the question and optionally assigns predefined tags.	

● SCENARIO	Properties & Description		
		3. The attendee submits the question. 4. The system stores the question in state pending. 5. The system shows the question to moderators for approval.	
Auto-Accepted by Sentiment ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	4
	○ OUTCOME	Description	
	The question is stored in state accepted and visible to the audience without moderation.	4. The system runs server-side sentiment analysis on the text. 5. The system finds the sentiment proper and auto-accept is enabled. 6. The system stores the question in state pending and accepts it at once, without a moderator decision.	
Auto-Rejected by Sentiment ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Exceptional	Failure	4
	○ OUTCOME	Description	
	The question is stored in state rejected and hidden from the audience, without a moderator decision.	4. The system runs server-side sentiment analysis on the text. 5. The system finds the sentiment improper and auto-reject is enabled. 6. The system stores the question in state pending and rejects it at once, informing the attendee.	
Throttled Submission ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Exceptional	Failure	4
	○ OUTCOME	Description	
	The question is not stored, and the attendee may submit it again once the limit window has passed.	4. The system detects that the attendee exceeded the per-minute submission limit. 5. The system rejects the new submission and informs the attendee to wait.	
Held Back by Client-Side Check ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Exceptional	Resume	3
	○ OUTCOME	Description	
	The question was not sent, and the attendee may revise its wording (resumes at step 2).	3. The system runs the client-side sentiment analysis on the text before sending it. 4. The system finds the sentiment improper and prevents the submission, informing the attendee.	

10.04 ● USE-CASE: Chat During the Event ⚓



○ ACTOR	● ROLE: Attendee ⚭
○ JOURNEYS	● STEP: Watch stream and interact ⚭
○ ACTIVITIES	● ACTIVITY: Attend Event ⚭
○ REQUIREMENTS	● REQUIREMENT: Chat Messages ⚭, ● REQUIREMENT: Like Messages ⚭, ● REQUIREMENT: Reply to Chat Messages ⚭, ● REQUIREMENT: Edit and Delete Messages ⚭, ● REQUIREMENT: Deleted Message Placeholder ⚭, ● REQUIREMENT: Personal Display Filtering ⚭, ● REQUIREMENT: Configurable Name Appearance ⚭, ● REQUIREMENT: Multiple-Choice Feedback ⚭
○ RULES	● RULE: Moderation Determines Initial Visibility ⚭, ● RULE: State Set Depends on Message Type ⚭, ● RULE: Forwarded Messages Are Immutable ⚭
○ PRE-CONDITION	The attendee has an active session and chat is enabled for the event.
○ TRIGGER	The attendee wants to comment on the running event.
○ POST-CONDITION	The chat message is stored in the state the moderation setting determines and shown under the attendee's configured name appearance.

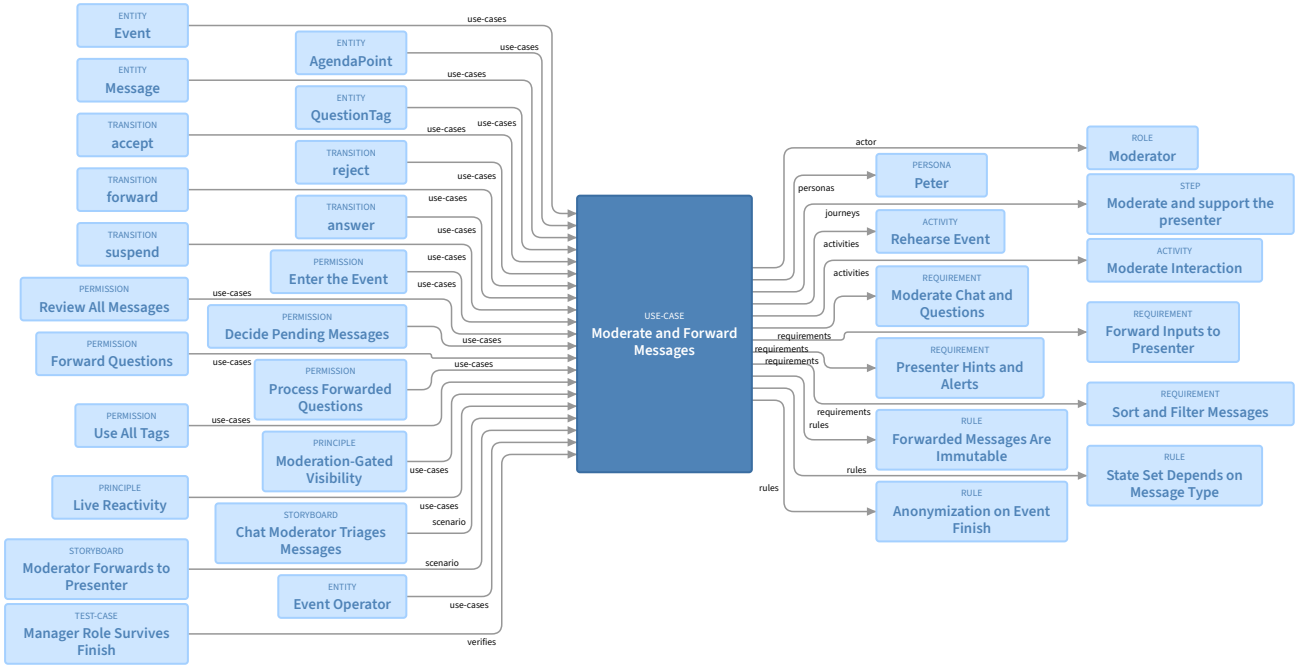
The attendee comments on the running event through chat messages, likes, replies, and poll answers, and corrects or removes their own contributions afterwards — **BECAUSE** live commentary turns the audience from silent viewers into an engaged community.

● SCENARIO	Properties & Description		
Post a Chat Message ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The attendee writes a chat message. 2. The attendee sends the message. 3. The system stores the message in state pending and accepts it at once, unless the event moderates chat messages and hence holds it for the moderator decision.	

● SCENARIO	Properties & Description		
		4. The system shows the message in the stream under the sender name in the form the event configures (full name, first name, or anonymous), exposing the email address on hover.	
Pending Message Stays Private ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	4
	○ OUTCOME	Description	
	The message is visible to its sender only until a moderator accepts it.	4. The system shows the pending message to its sender only, marked as awaiting the moderation decision. 5. The system shows the message to the other attendees once a moderator has accepted it.	
Like a Message ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The like count of the message reflects the current choice of the attendee.	1. The attendee likes a visible chat message or question of another attendee. 2. The system records the like and raises the like count of the message. 3. The attendee optionally undoes the like, and the system lowers the count again.	
Reply to a Chat Message ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The reply is stored like a chat message and shown threaded below the message it answers.	1. The attendee selects a visible chat message to reply to, replies being enabled for the event. 2. The attendee writes and sends the reply. 3. The system stores the reply like a chat message and shows it threaded below the message it answers.	
Edit an Own Message ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The corrected message is visible and marked as edited, moderated again where the event requires it.	1. The attendee opens one of their own chat messages or questions for editing. 2. The attendee changes the text and saves it. 3. The system marks the message as edited for the other readers. 4. The system resubmits the edited message to the moderation where the event requires it, hiding it from the audience until it is accepted anew.	
Delete an Own Message ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The message text is gone while its position in the stream is kept by a placeholder.	1. The attendee deletes one of their own chat messages or questions. 2. The system retains the position of the message in the stream and shows a “This message was deleted” placeholder instead of its text.	
Forwarded Question Locked ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Exceptional	Failure	1

● SCENARIO	Properties & Description		
	○ OUTCOME	Description	
	The forwarded question remains unchanged and the attendee is informed why.	1. The attendee tries to edit or delete one of their own questions which a moderator has already forwarded to the presenter. 2. The system refuses the change and informs the attendee that a forwarded question is immutable.	
Show Own Contributions Only ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	4
	○ OUTCOME	Description	
	The attendee sees only their own messages and questions until they choose to see those of the others again.	4. The attendee chooses to see only their own messages and questions. 5. The system hides the messages and questions of the other attendees from the view of the attendee.	
Answer a Multiple-Choice Poll ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The answer of the attendee is counted in the live result of the poll.	1. The system presents a multiple-choice question raised during the event. 2. The attendee picks one of the offered choices. 3. The system records the answer and updates the live result of the poll.	

10.05 ● USE-CASE: Moderate and Forward Messages ⚓



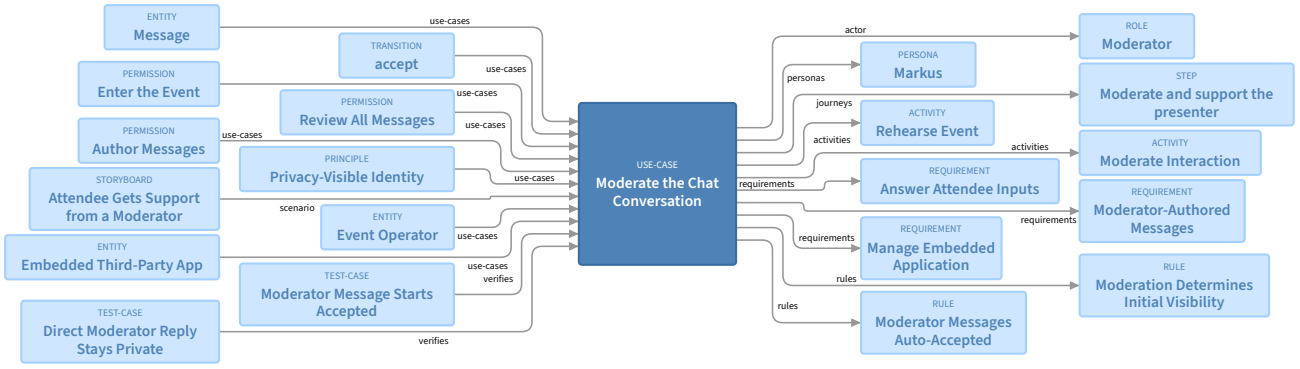
○ ACTOR	● ROLE: Moderator ⚭
○ PERSONAS	● PERSONA: Peter ⚭
○ JOURNEYS	● STEP: Moderate and support the presenter ⚭
○ ACTIVITIES	● ACTIVITY: Rehearse Event ⚭, ● ACTIVITY: Moderate Interaction ⚭

○ REQUIREMENTS	● REQUIREMENT: Moderate Chat and Questions ⚭, ● REQUIREMENT: Forward Inputs to Presenter ⚭, ● REQUIREMENT: Sort and Filter Messages ⚭, ● REQUIREMENT: Presenter Hints and Alerts ⚭
○ RULES	● RULE: State Set Depends on Message Type ⚭, ● RULE: Forwarded Messages Are Immutable ⚭, ● RULE: Anonymization on Event Finish ⚭
○ PRE-CONDITION	The event is running and the moderator has the Moderator role.
○ TRIGGER	An attendee message arrives in state pending for moderation.
○ POST-CONDITION	Messages are accepted, rejected, forwarded with optional hints, or settled on behalf of the presenter.

The moderator reviews the pending messages by state, approves or rejects them, forwards selected approved questions to the presenter in a chosen order, and may attach hints or raise a presenter alert — **BECAUSE** the presenter can only handle a curated, ordered selection of the audience input while on stage.

● SCENARIO	Properties & Description		
Approve and Forward Question ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The moderator narrows the messages down to the pending questions. 2. The moderator approves a relevant question, setting it accepted. 3. The moderator forwards the accepted question to the presenter. 4. The moderator optionally attaches a hint for the presenter. 5. The system places the forwarded question in the presenter's work basket and locks it from editing.	
Reject Improper Message ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The message is in state rejected and hidden from the audience, without reaching the presenter.	2. The moderator rejects an improper message. 3. The system sets the message to rejected and hides it from the audience.	
Settle a Question for the Presenter ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	4
	○ OUTCOME	Description	
	The forwarded question is marked answered or suspended without the presenter touching it.	4. The moderator marks the forwarded question as answered or suspended on behalf of the presenter, e.g. once the presenter addressed it on stage without marking it. 5. The system records the decision and removes the question from the presenter's work basket.	
Access Lost After Finish ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Exceptional	Failure	1
	○ OUTCOME	Description	
	The moderator cannot enter the finished event, and no message is moderated anymore.	1. The moderator opens the event after the manager has finished it. 2. The system refuses the access, as the anonymization has deleted the Moderator role of the event.	

10.06 ● USE-CASE: Moderate the Chat Conversation ⚓



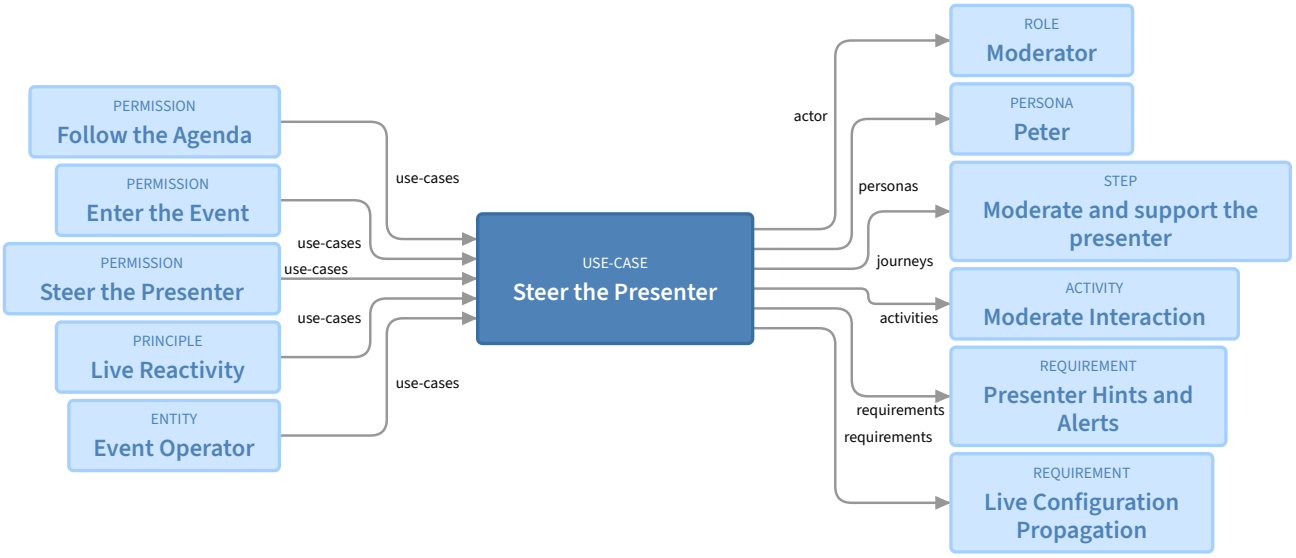
○ ACTOR	● ROLE: Moderator ⚭
○ PERSONAS	● PERSONA: Markus ⚭
○ JOURNEYS	● STEP: Moderate and support the presenter ⚭
○ ACTIVITIES	● ACTIVITY: Rehearse Event ⚭, ● ACTIVITY: Moderate Interaction ⚭
○ REQUIREMENTS	● REQUIREMENT: Answer Attendee Inputs ⚭, ● REQUIREMENT: Moderator-Authored Messages ⚭, ● REQUIREMENT: Manage Embedded Application ⚭
○ RULES	● RULE: Moderator Messages Auto-Accepted ⚭, ● RULE: Moderation Determines Initial Visibility ⚭
○ PRE-CONDITION	The event runs with chat enabled and the moderator has the Moderator role.
○ TRIGGER	Attendee chat messages arrive which need an answer, a steer, or a seed from the moderator.
○ POST-CONDITION	Attendees are answered, moderator-authored messages are visible, and the conversation stays on-topic.

The moderator keeps the chat conversation civil and on-topic by answering attendees publicly or privately, seeding the conversation with messages of their own, and controlling an embedded application — **BECAUSE** an unattended chat drowns in spam and off-topic noise.

● SCENARIO	Properties & Description		
Answer an Attendee Directly ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The moderator reviews the incoming chat messages. 2. The moderator selects a chat message which needs a clarification or support. 3. The moderator writes a reply to the attendee. 4. The moderator chooses whether the reply is visible to all or a direct message to the attendee only. 5. The system stores the reply in state pending, accepts it at once as a moderator-authored message, and delivers it accordingly.	
Seed the Conversation as Moderator ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The moderator-authored message is visible to the audience under the chosen sender name.	1. The moderator authors a chat message or a question for the audience, e.g. to seed a Q&A round. 2. The moderator keeps the sender name “Moderator” or overrides it for this message.	

● SCENARIO	Properties & Description		
		3. The system stores the message in state pending, accepts it at once as a moderator-authored message, and shows it to the audience.	
Administer the Embedded Application ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The embedded application is controlled by the moderator while the event continues.	1. The moderator opens the administration view an embedded third-party application exposes. 2. The moderator controls the application from this view. 3. The system passes the control on to the application without interrupting the stream.	

10.07 ● USE-CASE: Steer the Presenter ⚓



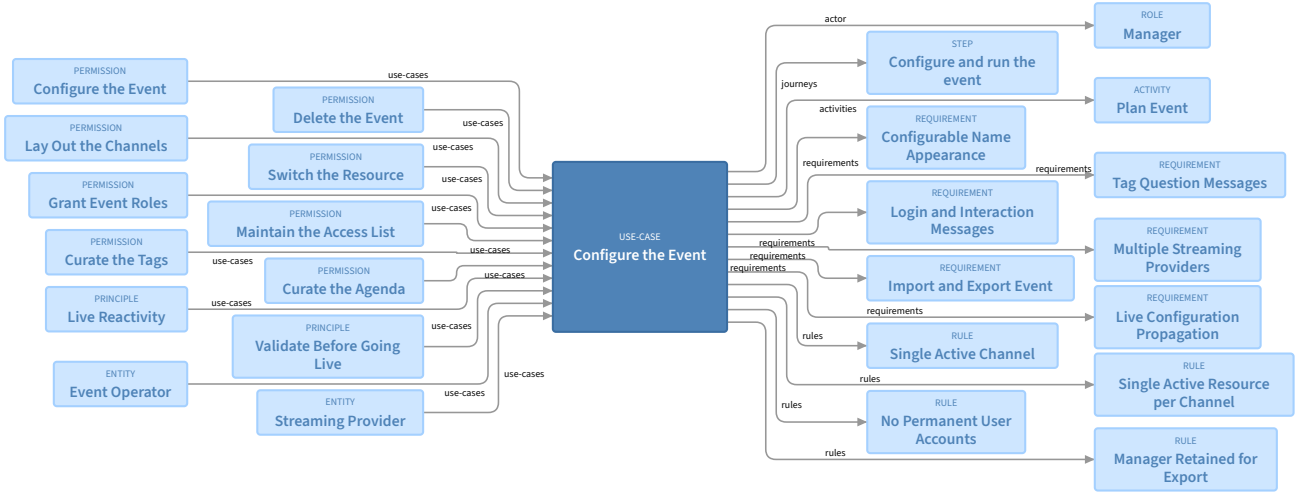
○ ACTOR	● ROLE: Moderator ⚭
○ PERSONAS	● PERSONA: Peter ⚭
○ JOURNEYS	● STEP: Moderate and support the presenter ⚭
○ ACTIVITIES	● ACTIVITY: Moderate Interaction ⚭
○ REQUIREMENTS	● REQUIREMENT: Presenter Hints and Alerts ⚭, ● REQUIREMENT: Live Configuration Propagation ⚭
○ PRE-CONDITION	The event is running and the moderator has the Moderator role.
○ TRIGGER	The presenter needs timing or routing guidance during the live event.
○ POST-CONDITION	The presenter alert is raised or the active agenda point is advanced, and the presenter sees it.

The moderator raises a textual alert on the stage view of the presenter and advances the active agenda point of the event — **BECAUSE** the presenter on stage cannot watch the clock and the audience input at the same time.

● SCENARIO	Properties & Description		
Raise a Presenter Alert ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅

● SCENARIO	Properties & Description		
	○ OUTCOME	Description	
	∅	1. The moderator writes an alert for the presenter, e.g. a timing hint. 2. The moderator raises the alert. 3. The system shows the alert on the stage view of the presenter until the presenter confirms it.	
Advance the Agenda ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	1
	○ OUTCOME	Description	
	The next agenda point is the active one of the event.	1. The moderator marks the next agenda point as the active one. 2. The system propagates the active agenda point to the presenter and the attendees.	

10.08 ● USE-CASE: Configure the Event ⚓

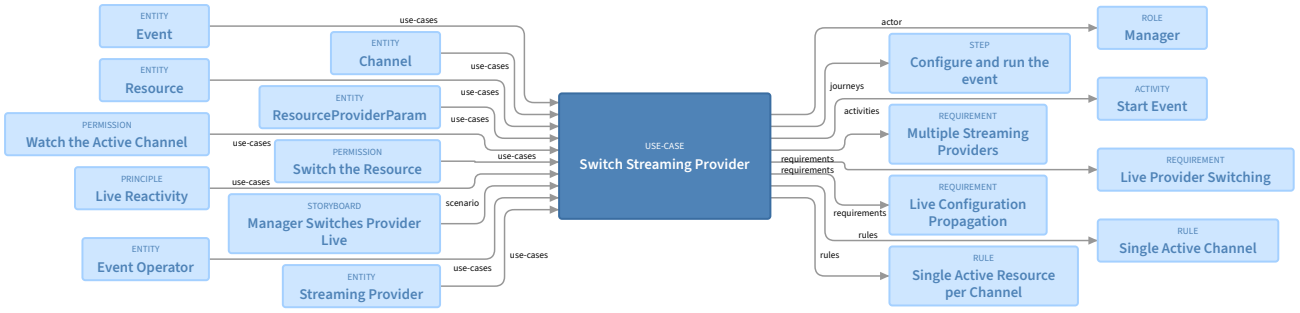


○ ACTOR	● ROLE: Manager ⚭
○ JOURNEYS	● STEP: Configure and run the event ⚭
○ ACTIVITIES	● ACTIVITY: Plan Event ⚭
○ REQUIREMENTS	● REQUIREMENT: Multiple Streaming Providers ⚭, ● REQUIREMENT: Configurable Name Appearance ⚭, ● REQUIREMENT: Tag Question Messages ⚭, ● REQUIREMENT: Login and Interaction Messages ⚭, ● REQUIREMENT: Live Configuration Propagation ⚭, ● REQUIREMENT: Import and Export Event ⚭
○ RULES	● RULE: Single Active Channel ⚭, ● RULE: Single Active Resource per Channel ⚭, ● RULE: No Permanent User Accounts ⚭, ● RULE: Manager Retained for Export ⚭
○ PRE-CONDITION	The event is provisioned by the administrator and the manager holds its Manager role.
○ TRIGGER	The manager receives the Manager role of an upcoming event.
○ POST-CONDITION	The event carries its channels, agenda, tags, roles, access list, and interaction settings, ready for the rehearsal.

The manager lays out the channels and their fallback resources, curates the agenda and the question tags, grants the Moderator and Presenter roles, maintains the access list, and sets the interaction options of the event — **BECAUSE** a broadcast event needs its stage, its staff, and its audience set up before anyone can rehearse it.

● SCENARIO	Properties & Description		
Lay Out the Channels ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The manager opens the configuration of the event. 2. The manager defines the channels of the event and marks one of them as active. 3. The manager selects the active resource of each channel among its provisioned streaming resources. 4. The manager sets the interaction options, the name appearance, and the login and interaction messages of the event. 5. The system validates the settings and propagates them to the connected clients.	
Curate the Agenda ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The event carries an ordered agenda the moderator advances during the live event.	2. The manager defines the ordered agenda points of the event and the tags corresponding to them. 3. The system stores the agenda for the presenter and the moderators.	
Curate the Tags ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The event carries a tag vocabulary for the questions, partly reserved for the moderators.	2. The manager defines the tag vocabulary of the questions, reserving some tags for the moderators. 3. The system offers the public tags to the attendees and every tag to the moderators.	
Grant the Event Roles ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The moderators and the presenter can enter the unpublished event for the rehearsal.	2. The manager grants the Attendee, Moderator, and Presenter roles to the users of the event by email address, or revokes them again. 3. The system creates the users of the roles without permanent accounts and admits them to the unpublished event.	
Maintain the Access List ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The access list names exactly the invited attendees.	2. The manager adds attendees to the access list by email address, or removes them again. 3. The system deletes a removed user without any role entirely.	
Delete the Event ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The event no longer exists.	2. The manager deletes the event entirely. 3. The system deletes the event with its messages, statistics, and remaining Manager roles.	

10.09 ● USE-CASE: Switch Streaming Provider

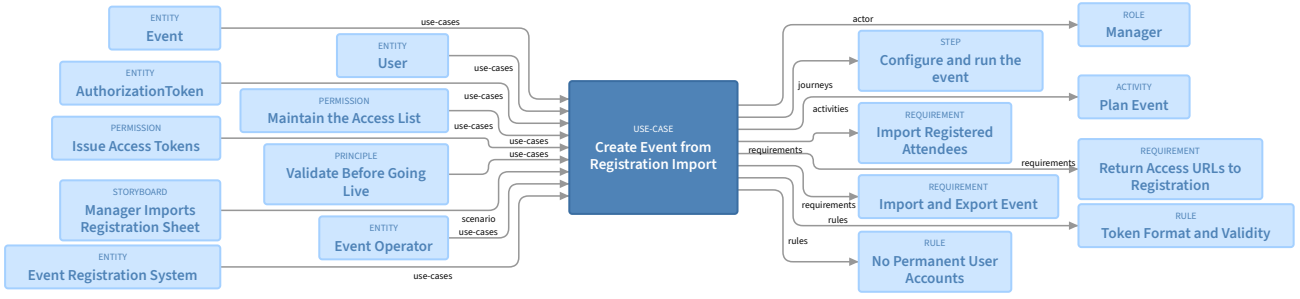


○ ACTOR	● ROLE: Manager
○ JOURNEYS	● STEP: Configure and run the event
○ ACTIVITIES	● ACTIVITY: Start Event
○ REQUIREMENTS	● REQUIREMENT: Multiple Streaming Providers, ● REQUIREMENT: Live Provider Switching, ● REQUIREMENT: Live Configuration Propagation
○ RULES	● RULE: Single Active Channel, ● RULE: Single Active Resource per Channel
○ PRE-CONDITION	The event runs and the channel has multiple configured resources.
○ TRIGGER	The active streaming resource shows problems during a running event.
○ POST-CONDITION	A new resource is active and all clients follow it automatically.

When a provider has problems, the manager activates a fallback resource on the channel and the system propagates the change so attendee clients switch streams without any user interaction — **BECAUSE** a live event cannot be paused while a streaming provider recovers.

● SCENARIO	○ TYPE	Description
Live Failover	Main	1. The manager observes streaming problems with the active resource. 2. The manager activates a different configured resource on the channel. 3. The system marks the new resource active and the old one inactive. 4. The system propagates the change to all connected clients. 5. The attendee clients switch to the new stream without user interaction.

10.10 ● USE-CASE: Create Event from Registration Import



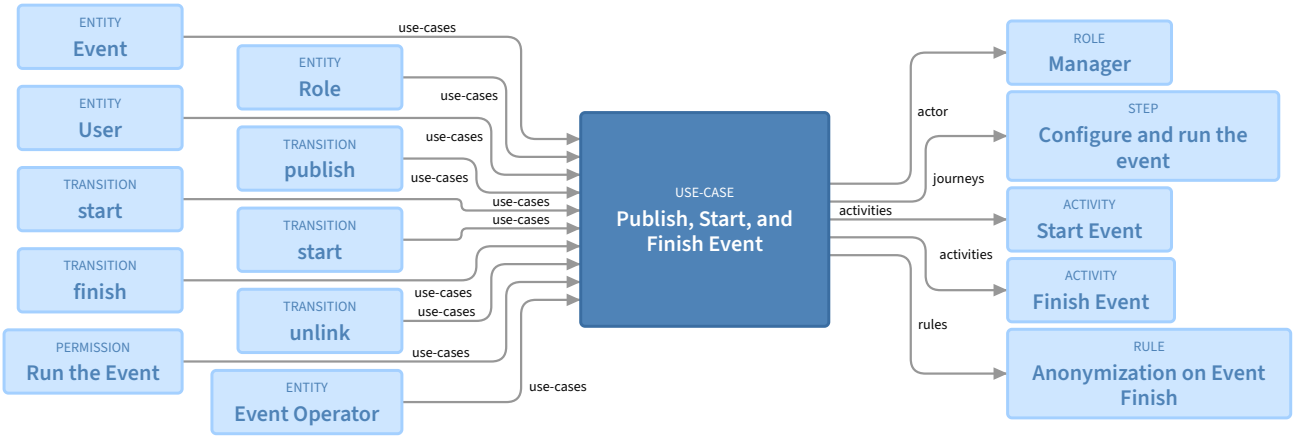
○ ACTOR	● ROLE: Manager

○ JOURNEYS	● STEP: Configure and run the event
○ ACTIVITIES	● ACTIVITY: Plan Event
○ REQUIREMENTS	● REQUIREMENT: Import Registered Attendees, ● REQUIREMENT: Return Access URLs to Registration, ● REQUIREMENT: Import and Export Event
○ RULES	● RULE: Token Format and Validity, ● RULE: No Permanent User Accounts
○ PRE-CONDITION	The manager has an Excel sheet of the Event Registration System and an event to populate.
○ TRIGGER	The manager receives the Excel sheet of the attendees of an upcoming event, exported by the Event Registration System.
○ POST-CONDITION	The access list and tokens are created and the URL sheet is ready for the manager to hand back to the Event Registration System.

The manager imports the Excel sheet of the Event Registration System to fill the event access list and generate authorization tokens, then exports an Excel sheet of personal access URLs back to the Event Registration System, avoiding duplicate invitations on repeated imports — **BECAUSE** provisioning hundreds of attendees by hand is error-prone and does not scale.

● SCENARIO	○ TYPE	Description
Import and Return URLs	Main	1. The manager uploads the Excel sheet of the Event Registration System to the event. 2. The system creates access-list users for new emails and skips existing ones. 3. The system generates a “NNN-NNN” authorization token per user in state issued. 4. The system composes each user’s personal access URL with event, user, and token. 5. The system returns an Excel sheet with the URL column filled, which the manager hands back to the Event Registration System.

10.11 ● USE-CASE: Publish, Start, and Finish Event



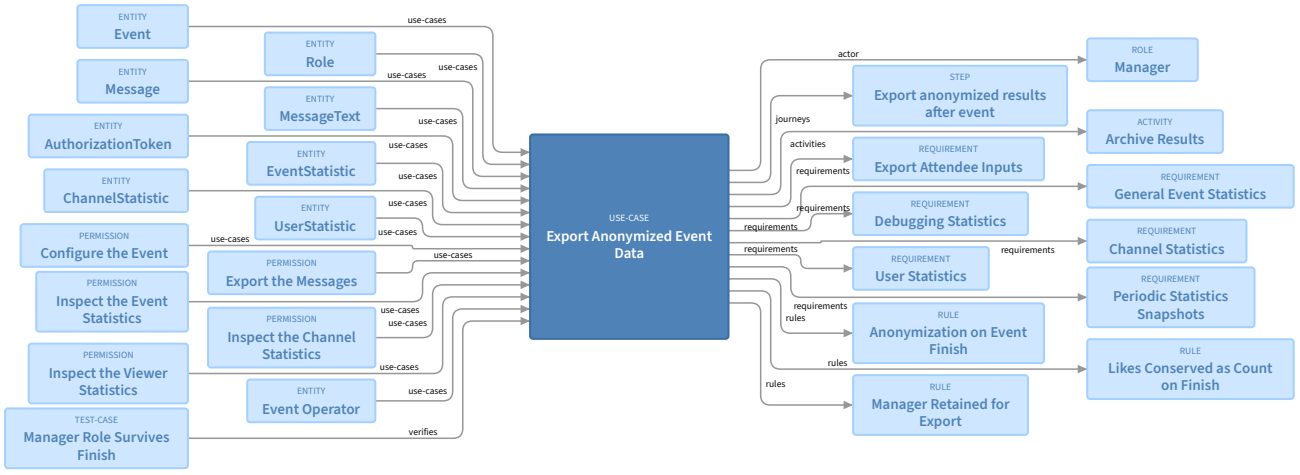
○ ACTOR	● ROLE: Manager
○ JOURNEYS	● STEP: Configure and run the event
○ ACTIVITIES	● ACTIVITY: Start Event, ● ACTIVITY: Finish Event
○ RULES	● RULE: Anonymization on Event Finish
○ PRE-CONDITION	The event is configured and its access list is populated.

○ TRIGGER	The scheduled date of the event approaches.
○ POST-CONDITION	The event is finished, access is closed, and the personal data is anonymized.

The manager publishes the configured event to make it visible to the invited attendees, starts it when the live stream goes on air, and finishes it afterwards — **BECAUSE** the visibility, the live interaction, and the anonymization of an event are deliberate decisions of the manager, not side effects of the clock.

● SCENARIO	Properties & Description		
Publish, Start, and Finish ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The manager publishes the configured event. 2. The system makes the event visible to the invited attendees. 3. The manager starts the event when the live stream goes on air. 4. The system opens the stream and the interaction channels for the attendees. 5. The manager finishes the event after the live stream has ended. 6. The system closes the access and anonymizes the personal data of the event.	
Start Without Publishing ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Resume	1
	○ OUTCOME	Description	
	The event is running without having been visible to the attendees beforehand, and the flow resumes at step 5.	1. The manager starts the still unpublished event directly. 2. The system opens the stream and the interaction channels for the attendees.	

10.12 ● USE-CASE: Export Anonymized Event Data ⚓



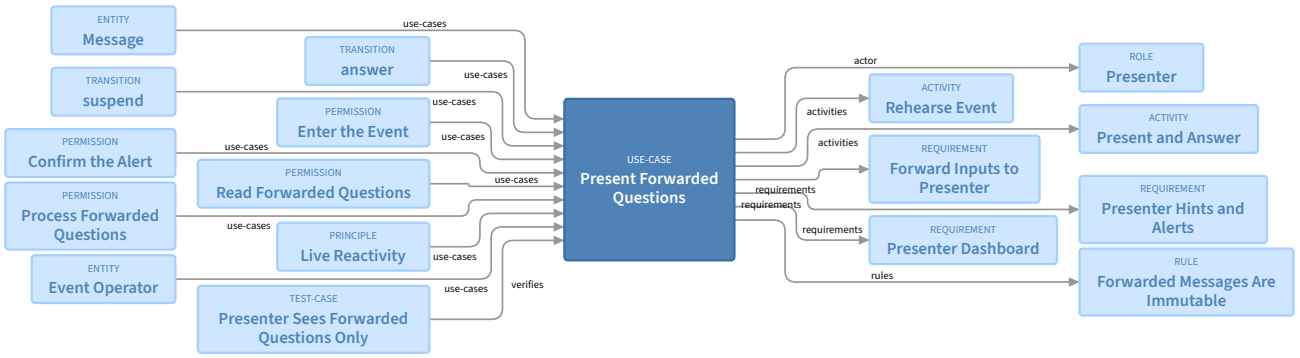
○ ACTOR	● ROLE: Manager ⚭
○ JOURNEYS	● STEP: Export anonymized results after event ⚭
○ ACTIVITIES	● ACTIVITY: Archive Results ⚭
○ REQUIREMENTS	● REQUIREMENT: Export Attendee Inputs ⚭, ● REQUIREMENT: General Event Statistics ⚭, ● REQUIREMENT: Channel Statistics ⚭, ● REQUIREMENT: User Statistics ⚭, ● REQUIREMENT: Debugging Statistics ⚭, ● REQUIREMENT: Periodic Statistics Snapshots ⚭

○ RULES	● RULE: Anonymization on Event Finish, ● RULE: Likes Conserved as Count on Finish, ● RULE: Manager Retained for Export
○ PRE-CONDITION	The event has finished and the manager retains the Manager role.
○ TRIGGER	The manager is asked to hand over the recorded interaction of a finished event.
○ POST-CONDITION	An export file of anonymized messages and statistics is produced.

After the event finishes and personal data is anonymized, the manager exports all attendee inputs with timestamps, states, like counts, and texts, together with the event statistics — **BECAUSE** the organizer evaluates the event and answers open questions afterwards from this record.

● SCENARIO	Properties & Description		
Export After Finish	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The manager selects the finished event. 2. The manager triggers the export of attendee inputs. 3. The system produces a file containing timestamp, state, likes, and text per message. 4. The manager downloads the export file.	
Inspect the Event Statistics	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The manager has evaluated the audience size, composition, channel popularity, and authentication flow without producing an export file.	2. The manager opens the statistics of the event instead of triggering the export. 3. The system shows the curve of logged-in attendees over time and the number of viewers per channel. 4. The system shows the recorded viewer statistics by country, browser, device, and viewport. 5. The system shows the login-challenge statistics of issued, sent, and used authorization tokens.	

10.13 ● USE-CASE: Present Forwarded Questions



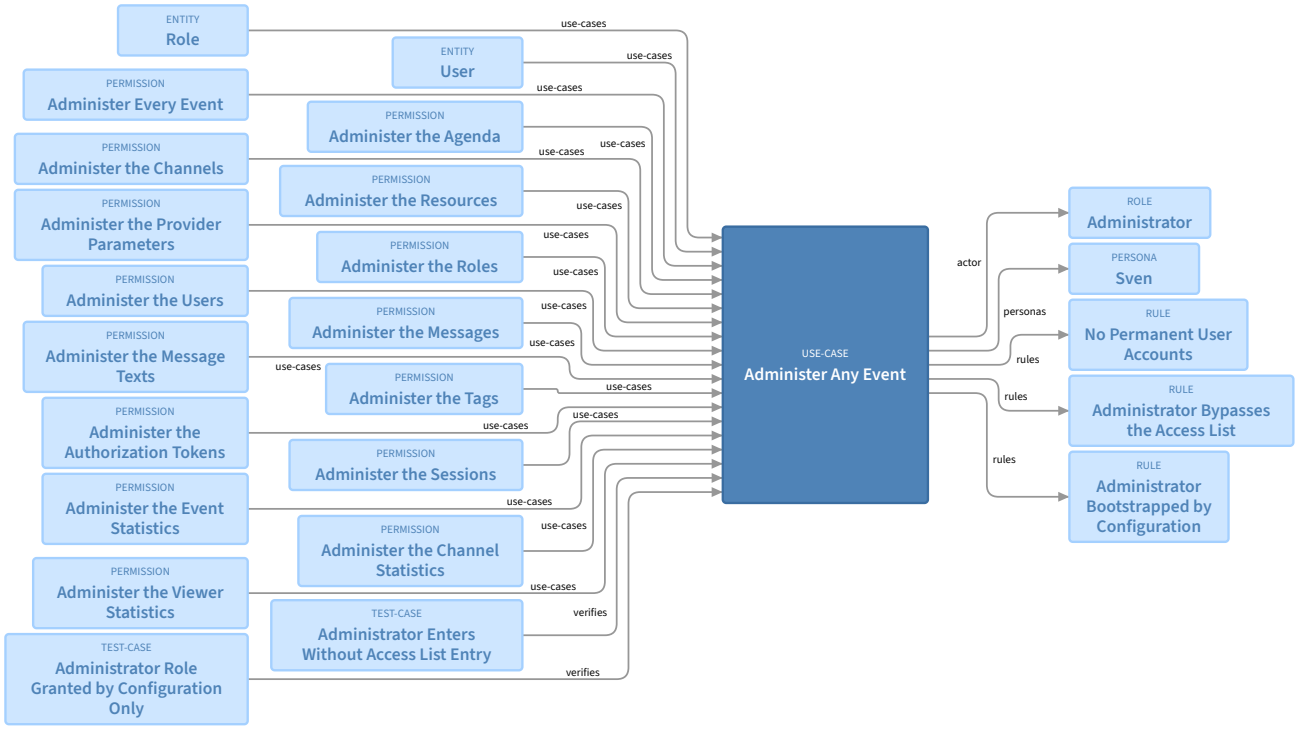
○ ACTOR	● ROLE: Presenter
○ ACTIVITIES	● ACTIVITY: Rehearse Event, ● ACTIVITY: Present and Answer
○ REQUIREMENTS	● REQUIREMENT: Forward Inputs to Presenter, ● REQUIREMENT: Presenter Dashboard, ● REQUIREMENT: Presenter Hints and Alerts
○ RULES	● RULE: Forwarded Messages Are Immutable

○ PRE-CONDITION	The event is running and questions have been forwarded.
○ TRIGGER	The moderator forwards a question to the presenter.
○ POST-CONDITION	Processed questions are marked answered or suspended.

The presenter views the forwarded questions in their intended order with any moderator hints, addresses them on stage, and marks each as answered or suspended — **BECAUSE** the presenter needs to keep track of what was already addressed while staying focused on the audience.

● SCENARIO	Properties & Description		
Answer Forwarded Question ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Main	∅	∅
	○ OUTCOME	Description	
	∅	1. The presenter views the ordered list of forwarded questions and hints. 2. The presenter addresses a question live on stage. 3. The presenter marks the question as answered. 4. The system records the answered timestamp and removes it from the active basket.	
Suspend a Forwarded Question ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Success	2
	○ OUTCOME	Description	
	The question is marked suspended and leaves the active basket without having been addressed on stage.	2. The presenter decides not to address a forwarded question on stage. 3. The presenter marks the question as suspended. 4. The system removes the question from the active basket.	
Forwarded Questions Only ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Resume	1
	○ OUTCOME	Description	
	The presenter works on the curated forwarded questions alone, and the flow resumes at step 2.	1. The presenter views the forwarded questions only, while the pending, accepted, and rejected messages of the event stay invisible to them.	
Confirm a Raised Alert ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Resume	1
	○ OUTCOME	Description	
	The alert is taken off the stage view, and the flow resumes at step 1.	1. The presenter sees an alert a moderator has raised on the stage view. 2. The presenter confirms the alert, which takes it off the stage view.	

10.14 ● USE-CASE: Administer Any Event ⚓



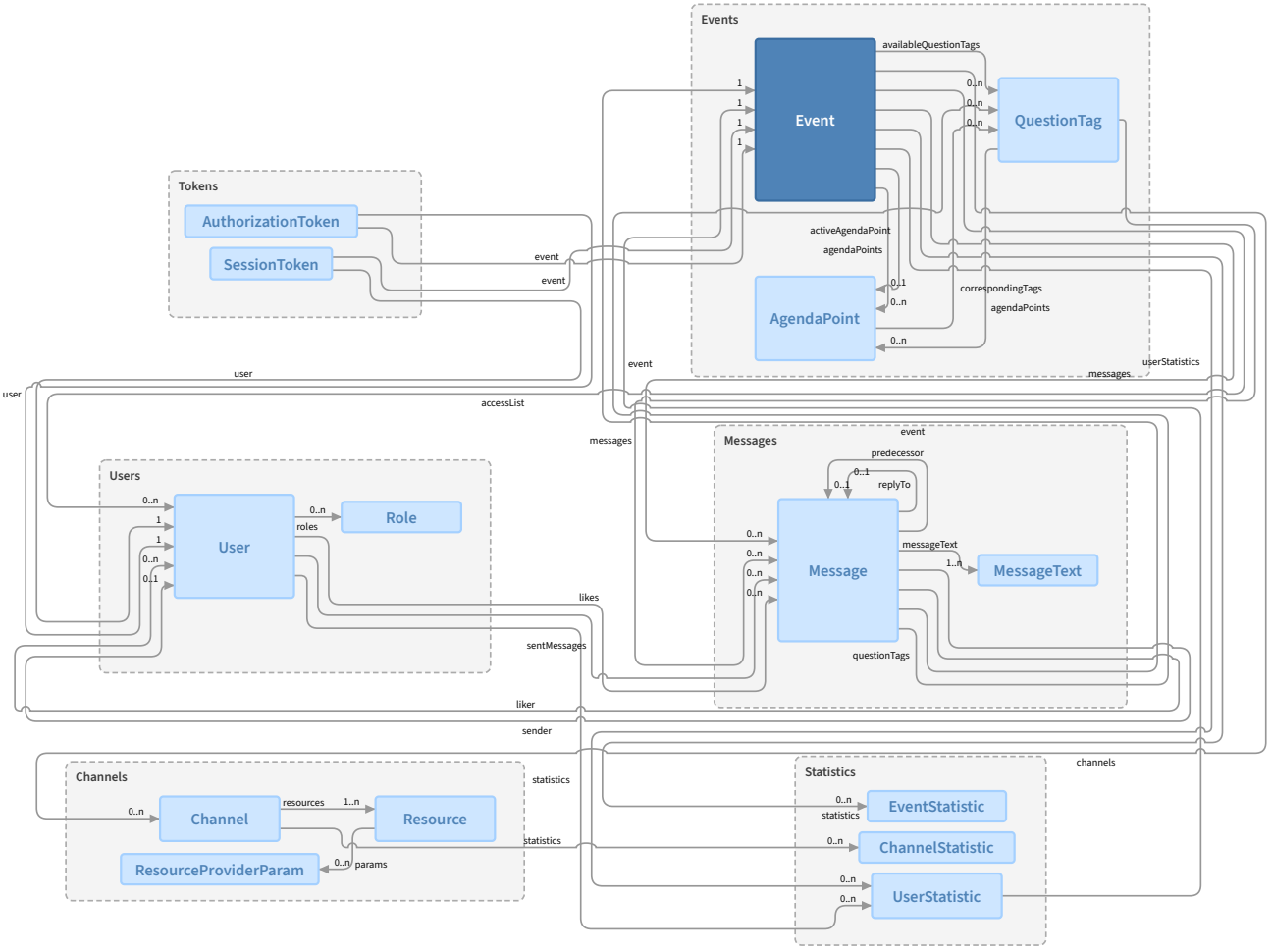
○ **ACTOR**	● **ROLE:** Administrator ⚭
○ **PERSONAS**	● **PERSONA:** Sven ⚭
○ **RULES**	● **RULE:** Administrator Bypasses the Access List ⚭, ● **RULE:** Administrator Bootstrapped by Configuration ⚭, ● **RULE:** No Permanent User Accounts ⚭
○ **PRE-CONDITION**	The Administrator role is granted to the email address of the administrator through the configuration.
○ **TRIGGER**	The administrator opens the URL of an event they are not enrolled in.
○ **POST-CONDITION**	The event carries the changes of the administrator.

The administrator enters any event without an access list entry and edits every setting, message, and user of it — **BECAUSE** the organizers need a last resort which depends on no event-specific grant when they are locked out or stuck.

● **SCENARIO**	Properties & Description		
Enter and Edit Any Event ⚓	○ **TYPE**	○ **RESULT**	○ **AT-MAIN-STEP**
	Main	∅	∅
	○ **OUTCOME**	Description	
	∅	1. The administrator opens the event URL and requests the login challenge for the configured administrator email address. 2. The system issues and sends the one-time token without consulting the access list of the event. 3. The administrator returns the token. 4. The system admits the administrator to the event in every state with every operation. 5. The administrator edits a setting, a message, or a user of the event. 6. The system applies the change and propagates it like the change of the respective event role.	

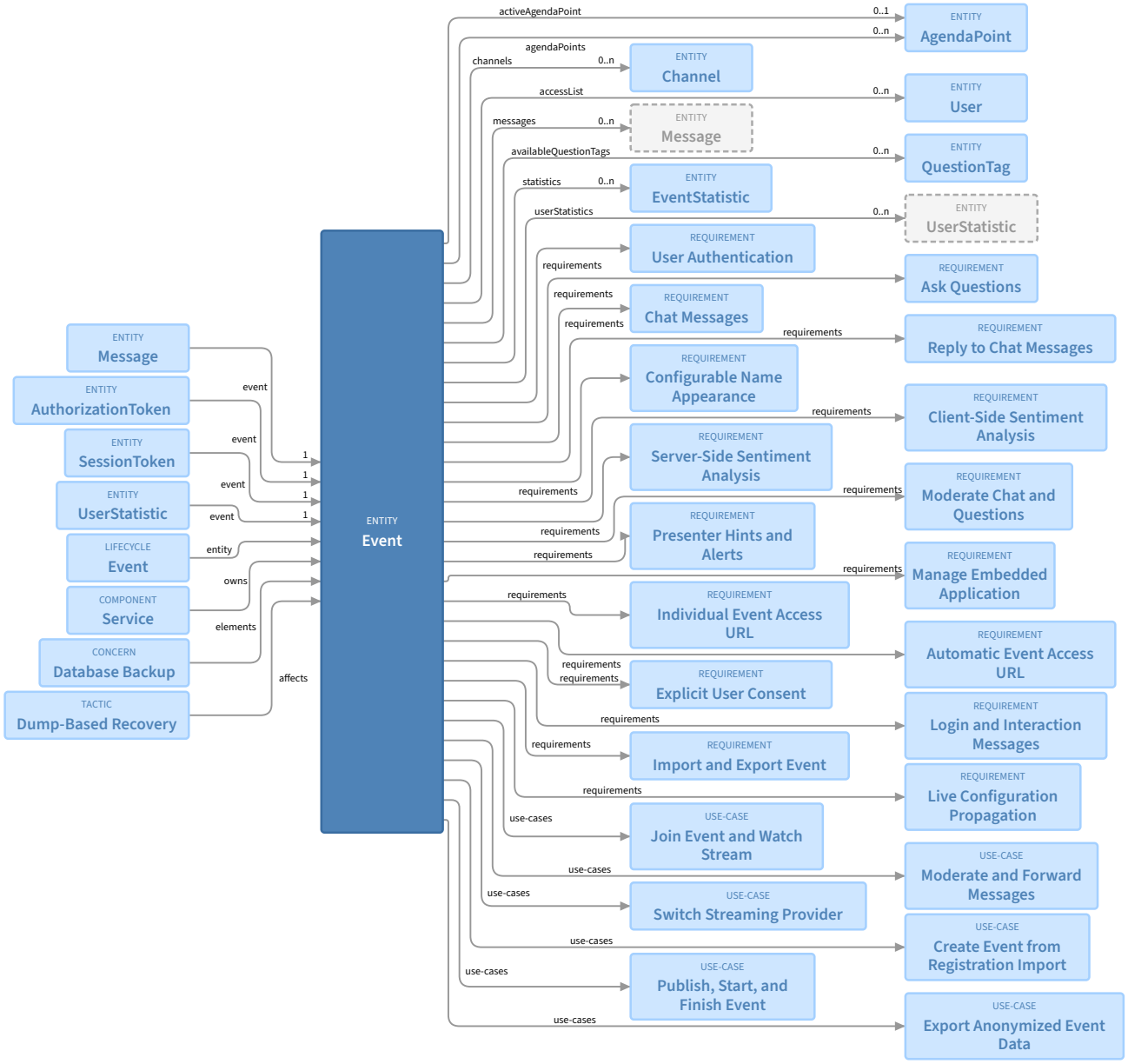
● SCENARIO	Properties & Description		
Reach an Event via the Overview ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Alternative	Resume	1
	○ OUTCOME	Description	
	The administrator has reached the event through the overview instead of its URL, and the flow resumes at step 1.	1. The administrator opens the overview of all events instead of the URL of a particular event. 2. The system lists every event in every state. 3. The administrator selects an event, which opens its URL.	
Administrator Role Stays Configured ⚓	○ TYPE	○ RESULT	○ AT-MAIN-STEP
	Exceptional	Failure	5
	○ OUTCOME	Description	
	The set of administrators stays as configured.	5. The administrator attempts to grant or revoke an Administrator role in the user interface. 6. The system refuses the change, as the Administrator role is granted through the configuration only.	

11 ● DATA: Data Model (DM) ⚓



● GROUP	Description
Events ⚓	An event together with its agenda points and its question tags — **BECAUSE** both are the event-specific configuration of its phases and questions.
Users ⚓	The user as the sole identity entity together with the roles granted to it — **BECAUSE** the user is the one entity nearly every other one refers to and the roles are the rights it holds.
Channels ⚓	A channel together with its provider resources and their provider parameters — **BECAUSE** the resources and parameters are the owned parts delivering the channel.
Messages ⚓	A message together with its language-specific texts — **BECAUSE** the texts are owned parts of the message and never exist without it.
Tokens ⚓	The authorization and session tokens of the login — **BECAUSE** both prove the access of a user to an event, before and after the login.
Statistics ⚓	The periodic snapshot entities of the event, its channels, and its users — **BECAUSE** all of them are recorded over time and exported together for trend visualization.

11.01 ● ENTITY: Event ⌘ ⚓



○ REQUIREMENTS	● REQUIREMENT: Individual Event Access URL, ● REQUIREMENT: User Authentication, ● REQUIREMENT: Automatic Event Access URL, ● REQUIREMENT: Explicit User Consent, ● REQUIREMENT: Login and Interaction Messages, ● REQUIREMENT: Configurable Name Appearance, ● REQUIREMENT: Chat Messages, ● REQUIREMENT: Reply to Chat Messages, ● REQUIREMENT: Ask Questions, ● REQUIREMENT: Moderate Chat and Questions, ● REQUIREMENT: Client-Side Sentiment Analysis, ● REQUIREMENT: Server-Side Sentiment Analysis, ● REQUIREMENT: Presenter Hints and Alerts, ● REQUIREMENT: Manage Embedded Application, ● REQUIREMENT: Live Configuration Propagation, ● REQUIREMENT: Import and Export Event
○ USE-CASES	● USE-CASE: Create Event from Registration Import, ● USE-CASE: Publish, Start, and Finish Event, ● USE-CASE: Join Event and Watch Stream, ● USE-CASE: Switch Streaming Provider, ● USE-CASE: Moderate and Forward Messages, ● USE-CASE: Export Anonymized Event Data
○ TERMS	● TERM: Event
○ GROUP	● GROUP: Events

The master entity describing a single live broadcast event and all of its configuration — **BECAUSE** the entire data model is event-centric and every other entity hangs off an event.

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	Description
eventId ⚓	`key uuid`	`uuid()`	∅	Unique identifier of the event used in the access URL — **BECAUSE** attendees reach a specific event by an unguessable link
title ⌘ ⚓	`string?`	∅	Public	Display title of the event such as “Townhall 1/23” — **BECAUSE** attendees and managers need a human-readable label
description ⌘ ⚓	`string?`	∅	∅	Free-text description of the event — **BECAUSE** organizers describe the purpose and content of the event
begin ⌘ ⚓	`datetime?`	∅	∅	Planned start date and time of the event — **BECAUSE** attendees and operators must know when the event starts
end ⌘ ⚓	`datetime?`	∅	∅	Planned end date and time of the event — **BECAUSE** scheduling and the finish procedure depend on the planned end
language ⚓	`string`	en	∅	Targeted mother language of the event — **BECAUSE** it sets the default language despite on-the-fly translation of interaction
state ⚓	`enum(planning,published,running,finished)`	`planning`	∅	Lifecycle state controlling attendee visibility and access — **BECAUSE** an event progresses from private setup to live to archived
loginInfo ⚓	`string`	`""`	∅	HTML markup shown to attendees at login — **BECAUSE** managers convey event information and links at entry

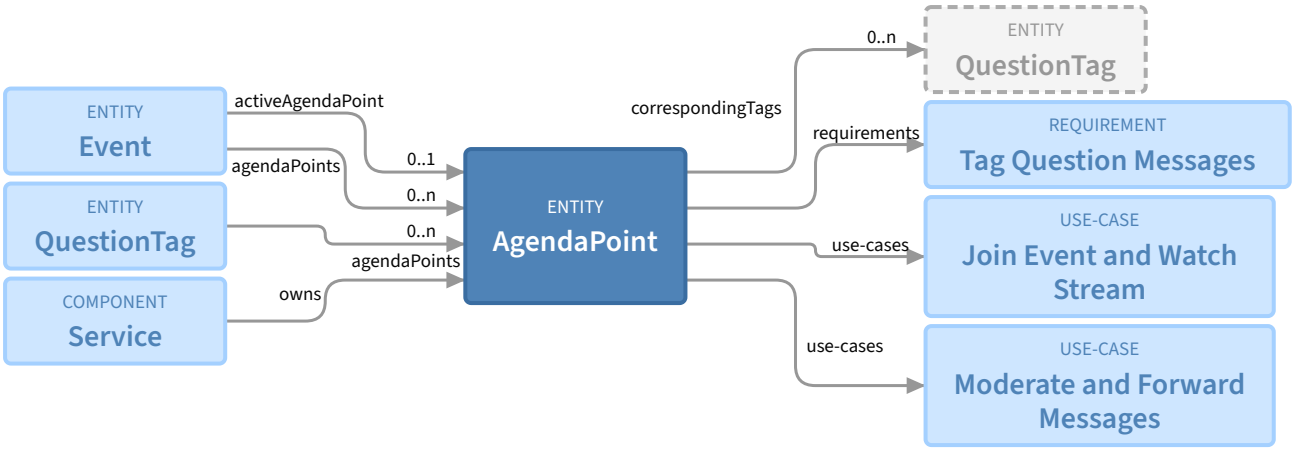
● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	Description
loginInfoToBeAccepted ⚓	`boolean`	`false`	∅	Whether the login information must be explicitly accepted — **BECAUSE** some events require acknowledged consent before joining
interactionInfo ⚓	`string`	`""`	∅	HTML markup shown when interaction is enabled — **BECAUSE** conduct rules must be presented before chat or questions
interactionInfoToBeAccepted ⚓	`boolean`	`false`	∅	Whether the interaction information must be explicitly accepted — **BECAUSE** some events require acknowledged conduct rules before interacting
allowAccessAnonymous ⚓	`boolean`	`false`	∅	Whether anonymous attendees may access the event without authorization — **BECAUSE** public events accept unauthenticated viewers
accessEmailPattern ⚓	`string`	`""`	∅	Pattern that grants access to any matching email beyond the access list — **BECAUSE** events can admit whole email domains without enumerating users
chatEnabled ⚓	`boolean`	`false`	∅	Whether attendees may send chat messages — **BECAUSE** chat is an optional interaction channel per event
chatAllowAnonymous ⚓	`boolean`	`false`	∅	Whether chat messages may be sent as “Anonymous” — **BECAUSE** events differ in how much chat identity is revealed
chatName ⚓	`enum(full,firstname,anonymous)`	`full`	∅	How the attendee name appears on chat messages — **BECAUSE** the event controls the displayed chat identity
chatReply ⚓	`boolean`	`false`	∅	Whether chat messages may be replied to — **BECAUSE** threaded replies are appropriate only for some events
chatThrottling ⚓	`integer`	`1`	∅	Maximum chat messages per user per minute — **BECAUSE** rate limiting prevents denial-of-service abuse
chatModerator ⚓	`boolean`	`false`	∅	Whether chat messages are moderated before becoming visible — **BECAUSE** organizers may require approval of chat content
supportEnabled ⚓	`boolean`	`false`	∅	Whether attendees may chat with support for technical problems — **BECAUSE** events may offer a dedicated support channel
appEnabled ⚓	`boolean`	`false`	∅	Whether a third-party application is integrated — **BECAUSE** events may

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	Description
				embed an interactive app
appTitle ⚓	`string`	`""`	∅	Display title of the embedded third-party application — **BECAUSE** the embedded app needs a label for attendees
appURL ⚓	`string`	`""`	∅	URL of the embedded third-party application — **BECAUSE** the client must load the app from a defined location
appAdminURL ⚓	`string`	`""`	∅	Administration URL of the embedded third-party application — **BECAUSE** moderators control the app through a separate admin view
presenterAlert ⚓	`string`	`""`	∅	Alert text a moderator can raise for the presenter — **BECAUSE** the moderator must signal urgent information to the presenter
presenterAlertState ⚓	`enum(raised,unraised)`	`unraised`	∅	Whether the presenter alert is currently visible to the presenter — **BECAUSE** the alert is toggled on by the moderator and confirmed off
questionsEnabled ⚓	`boolean`	`false`	∅	Whether attendees may ask questions — **BECAUSE** questions are an optional interaction channel per event
questionsAllowAnonymous ⚓	`boolean`	`false`	∅	Whether questions may be sent as “Anonymous” — **BECAUSE** events differ in how much question identity is revealed
questionsName ⚓	`enum(full,firstname,anonymous)`	`full`	∅	How the attendee name appears on questions — **BECAUSE** the event controls the displayed question identity
questionsThrottling ⚓	`integer`	`1`	∅	Maximum questions per user per time window — **BECAUSE** rate limiting prevents abuse of the question channel
questionsPrivate ⚓	`boolean`	`false`	∅	Whether questions are hidden from other attendees and seen only by moderators — **BECAUSE** some events keep questions private to the moderation team
questionsModerator ⚓	`boolean`	`false`	∅	Whether questions are moderated before becoming visible — **BECAUSE** organizers may require approval of questions
expireAuthTokenOnFirstUse ⚓	`boolean`	`true`	∅	Whether pre-generated authorization tokens expire after first use — **BECAUSE** events choose between one-time and reusable pre-generated tokens

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	Description
sentimentSenderAnalysis ⚓	boolean	false	∅	Whether lightweight client-side sentiment analysis is enabled — **BECAUSE** the event can check input at the source before sending
sentimentSenderAutoPrevent ⚓	boolean	false	∅	Whether improper input is prevented from submission on the client — **BECAUSE** the event can block misconduct before it reaches the server
sentimentModeratorAnalysis ⚓	boolean	false	∅	Whether full server-side sentiment analysis is enabled — **BECAUSE** the event can check input before it is stored
sentimentModeratorAutoAccept ⚓	boolean	false	∅	Whether proper input is auto-accepted on the server — **BECAUSE** moderators can be relieved of approving positive input
sentimentModeratorAutoReject ⚓	boolean	false	∅	Whether improper input is auto-rejected on the server — **BECAUSE** moderators can be relieved of rejecting negative input

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
channels ⚓	● **ENTITY:** Channel ⚭	Composition	0..n	Language-specific content distributors of the event — **BECAUSE** an event delivers content through one or more logical channels
accessList ⚓	● **ENTITY:** User ⚭	Composition	0..n	Invited attendees identified by email — **BECAUSE** access is granted to an explicit list of users
messages ⚓	● **ENTITY:** Message ⚭	Composition	0..n	Messages written during the event — **BECAUSE** all chat, question, and support input belongs to the event
statistics ⚓	● **ENTITY:** EventStatistic ⚭	Composition	0..n	Periodic cumulative statistics snapshots — **BECAUSE** trend visualization requires periodic counts
userStatistics ⚓	● **ENTITY:** UserStatistic ⚭	Composition	0..n	Viewer statistics recorded for the users of the event — **BECAUSE** the statistics stay attributable to the event after their users are deleted and are dropped with it
availableQuestionTags ⚓	● **ENTITY:** QuestionTag ⚭	Composition	0..n	Tags available for use on questions — **BECAUSE** the event defines the vocabulary for tagging questions
activeAgendaPoint ⚓	● **ENTITY:** AgendaPoint ⚭	Association	0..1	The currently active agenda point — **BECAUSE** attendees see which phase of the event is current
agendaPoints ⚓	● **ENTITY:** AgendaPoint ⚭	Composition	0..n	All agenda points of the event — **BECAUSE** the event has an ordered agenda of phases

11.02 ● ENTITY: AgendaPoint ⚓



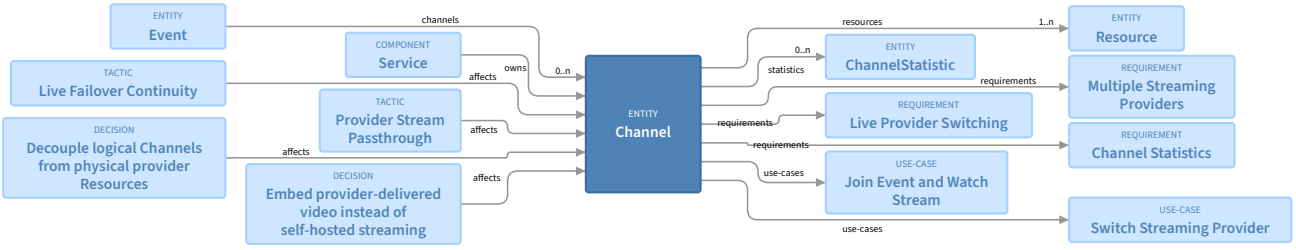
○ REQUIREMENTS	● REQUIREMENT: Tag Question Messages ⚭
○ USE-CASES	● USE-CASE: Join Event and Watch Stream ⚭, ● USE-CASE: Moderate and Forward Messages ⚭
○ GROUP	● GROUP: Events ⚭

The textual description of a phase in an event — **BECAUSE** attendees and moderators track which part of the event is currently active.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
agendaPointId ⚓	`key uuid`	`uuid()`	Unique identifier of the agenda point — **BECAUSE** it is referenced as a foreign key
text ⌘ ⚓	`string?`	∅	Description of the current phase of the event — **BECAUSE** the phase must be presented in human-readable form
orderPosition ⌘ ⚓	`integer?`	∅	Ordering position of the phase — **BECAUSE** agenda points have a defined sequence

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
correspondingTags ⚓	● ENTITY: QuestionTag ⚭	Association	`0..n`	Question tags corresponding to this agenda point — **BECAUSE** questions can be associated with the agenda phase they relate to

11.03 ● ENTITY: Channel ⚓



○ REQUIREMENTS	● REQUIREMENT: Multiple Streaming Providers ⚭, ● REQUIREMENT: Live Provider Switching ⚭, ● REQUIREMENT: Channel Statistics ⚭
○ USE-CASES	● USE-CASE: Join Event and Watch Stream ⚭, ● USE-CASE: Switch Streaming Provider ⚭

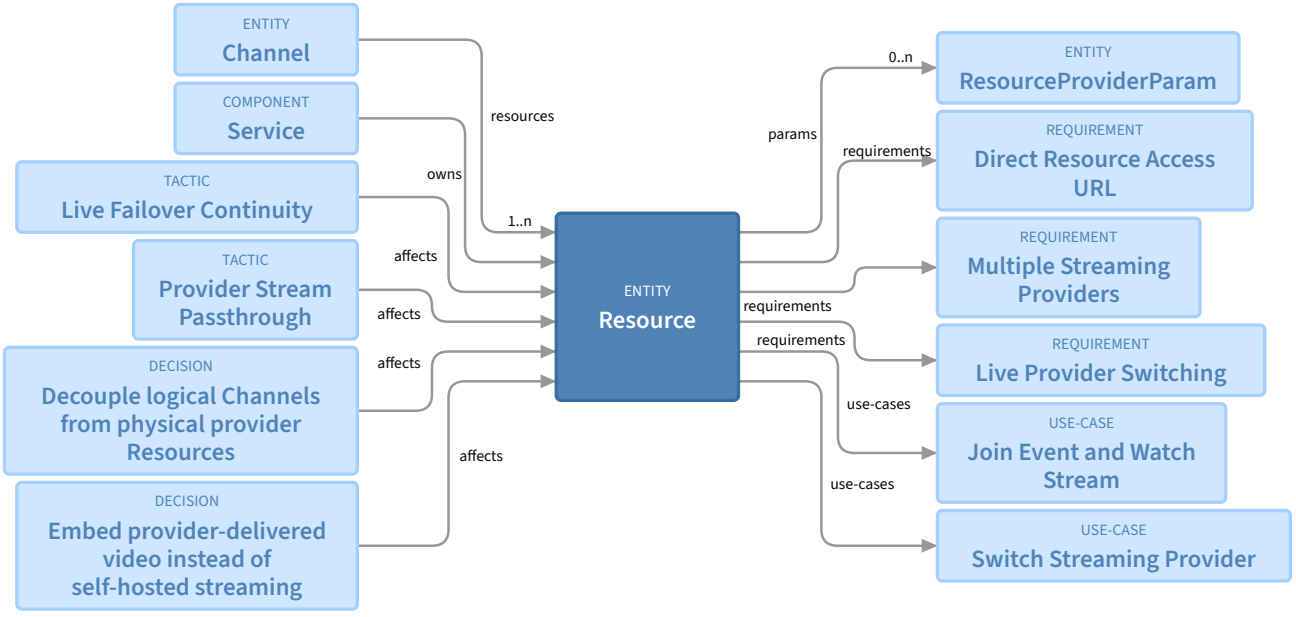
○ TERMS	● TERM: Channel
○ GROUP	● GROUP: Channels

A logical content delivery stream linking video streams to an event — **BECAUSE** an event groups its streams by language and resolution into channels.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
channelId	`key uuid`	`uuid()`	Unique identifier of the channel — **BECAUSE** it is referenced as a foreign key
name	`string?`	∅	Display name of the channel such as “Digital Townhall” — **BECAUSE** attendees choose between named channels
active	`boolean`	`false`	Whether the channel is the currently active one, derived to true for the first channel of an event (● **Initial Channel Becomes Active**) — **BECAUSE** only one channel of an event is active at once (● **Single Active Channel**)
default	`boolean`	`false`	Whether this channel is activated by default on entering an event — **BECAUSE** attendees need a defined initial channel

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
resources	● ENTITY: Resource	Composition	`1..n`	Physical resources backing the channel — **BECAUSE** a channel is delivered by one or more provider resources
statistics	● ENTITY: ChannelStatistic	Composition	`0..n`	Periodic viewer statistics of the channel — **BECAUSE** organizers track viewers per channel over time

11.04 ● ENTITY: Resource



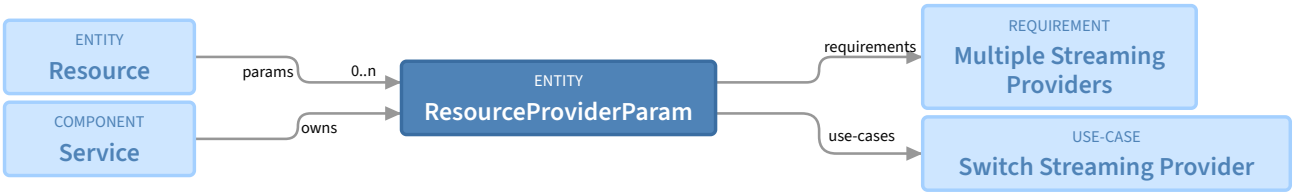
○ REQUIREMENTS	● REQUIREMENT: Multiple Streaming Providers, ● REQUIREMENT: Live Provider Switching, ● REQUIREMENT: Direct Resource Access URL
○ USE-CASES	● USE-CASE: Join Event and Watch Stream, ● USE-CASE: Switch Streaming Provider
○ TERMS	● TERM: Resource, ● TERM: Streaming Provider
○ GROUP	● GROUP: Channels

A physical content delivery resource such as a provider stream or static website linked to a channel — **BECAUSE** a channel must map to concrete provider endpoints to be playable.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
resourceId	`key uuid`	`uuid()`	Unique identifier of the resource — **BECAUSE** it is referenced as a foreign key and in the access URL
providerId ⌘	`string?`	∅	Provider identifier from the event configuration file — **BECAUSE** a resource binds to a specific configured streaming provider
active	`boolean`	`false`	Whether this resource is the active resource of the channel, derived to true for the first resource of a channel (● **Initial Resource Becomes Active**) — **BECAUSE** only one resource of a channel is active at once for provider switching (● **Single Active Resource per Channel**)

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
params	● ENTITY: ResourceProviderParam	Composition	`0..n`	Provider key-value parameters assigned to this resource — **BECAUSE** each provider needs configured parameters to address its stream

11.05 ● ENTITY: ResourceProviderParam



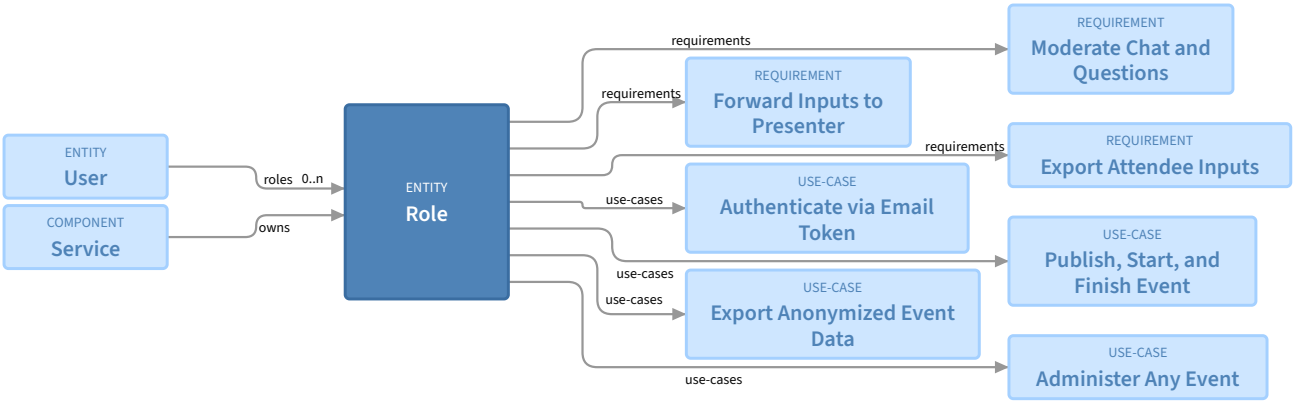
○ REQUIREMENTS	● REQUIREMENT: Multiple Streaming Providers
○ USE-CASES	● USE-CASE: Switch Streaming Provider
○ TERMS	● TERM: Streaming Provider
○ GROUP	● GROUP: Channels

A key-value parameter belonging to exactly one resource and provider, defined in the event configuration file — **BECAUSE** provider endpoints are parameterized by values an administrator supplies.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
resourceId	`key uuid`	`uuid()`	Identifier of the owning resource — **BECAUSE** the parameter belongs to exactly one resource

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
providerId ⌘ ⚓	`key string`	∅	Provider identifier from the configuration file — **BECAUSE** the parameter is scoped to one provider
key ⌘ ⚓	`key string`	∅	Parameter key defined in the configuration file — **BECAUSE** each provider parameter is identified by its key
value ⌘ ⚓	`string?`	∅	Value the administrator entered for the key — **BECAUSE** the concrete endpoint requires the supplied value

11.06 ● ENTITY: Role ⚓

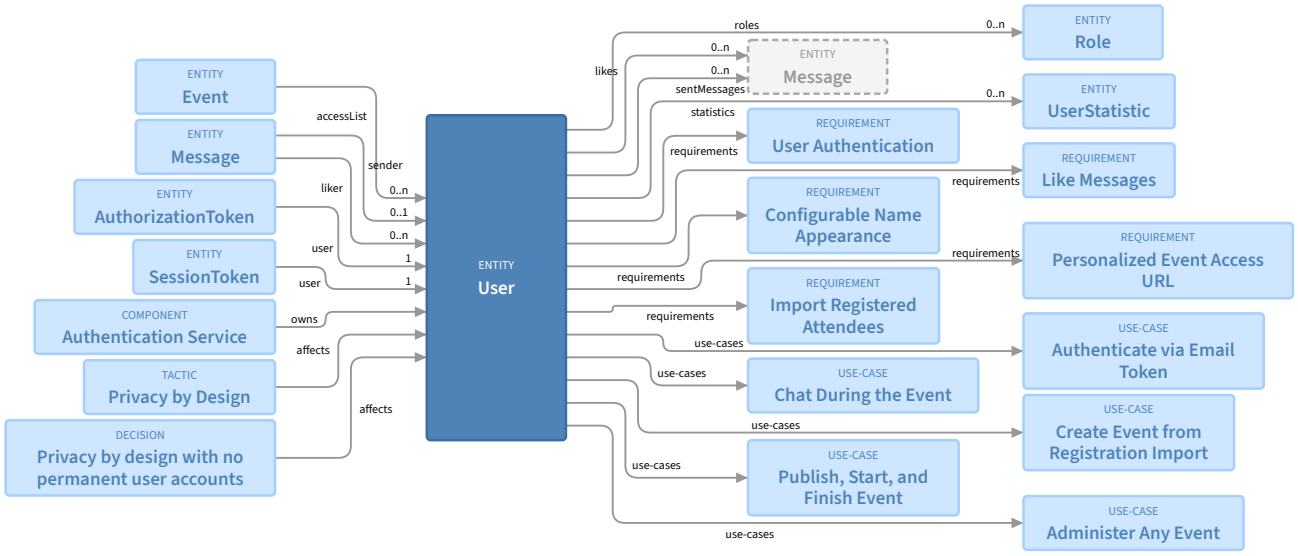


○ **REQUIREMENTS**	● **REQUIREMENT:** Moderate Chat and Questions ⚭, ● **REQUIREMENT:** Forward Inputs to Presenter ⚭, ● **REQUIREMENT:** Export Attendee Inputs ⚭
○ **USE-CASES**	● **USE-CASE:** Authenticate via Email Token ⚭, ● **USE-CASE:** Publish, Start, and Finish Event ⚭, ● **USE-CASE:** Export Anonymized Event Data ⚭, ● **USE-CASE:** Administer Any Event ⚭
○ **TERMS**	● **TERM:** Role ⚭, ● **TERM:** Attendee ⚭, ● **TERM:** Manager ⚭, ● **TERM:** Moderator ⚭, ● **TERM:** Presenter ⚭, ● **TERM:** Administrator ⚭
○ **GROUP**	● **GROUP:** Users ⚭

A grant of rights to a user, who exists within a specific event or, as the holder of the Administrator role, permanently and outside any event — **BECAUSE** the application is role-based and rights are granted through roles.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
roleId ⚓	`key uuid`	`uuid()`	Unique identifier of the role — **BECAUSE** it is referenced as a foreign key
type ⚓	`enum(Manager,Moderator,Presenter,Attendee,Administrator)`	`Attendee`	The role granted to the user, where Attendee is the plain audience membership and Administrator the permanent, event-independent role granted through the configuration only — **BECAUSE** each role carries a distinct set of rights

11.07 ● ENTITY: User ⚓



○ REQUIREMENTS	● REQUIREMENT: User Authentication ⚭, ● REQUIREMENT: Configurable Name Appearance ⚭, ● REQUIREMENT: Like Messages ⚭, ● REQUIREMENT: Personalized Event Access URL ⚭, ● REQUIREMENT: Import Registered Attendees ⚭
○ USE-CASES	● USE-CASE: Authenticate via Email Token ⚭, ● USE-CASE: Create Event from Registration Import ⚭, ● USE-CASE: Chat During the Event ⚭, ● USE-CASE: Publish, Start, and Finish Event ⚭, ● USE-CASE: Administer Any Event ⚭
○ TERMS	● TERM: User ⚭, ● TERM: Access List ⚭
○ GROUP	● GROUP: Users ⚭

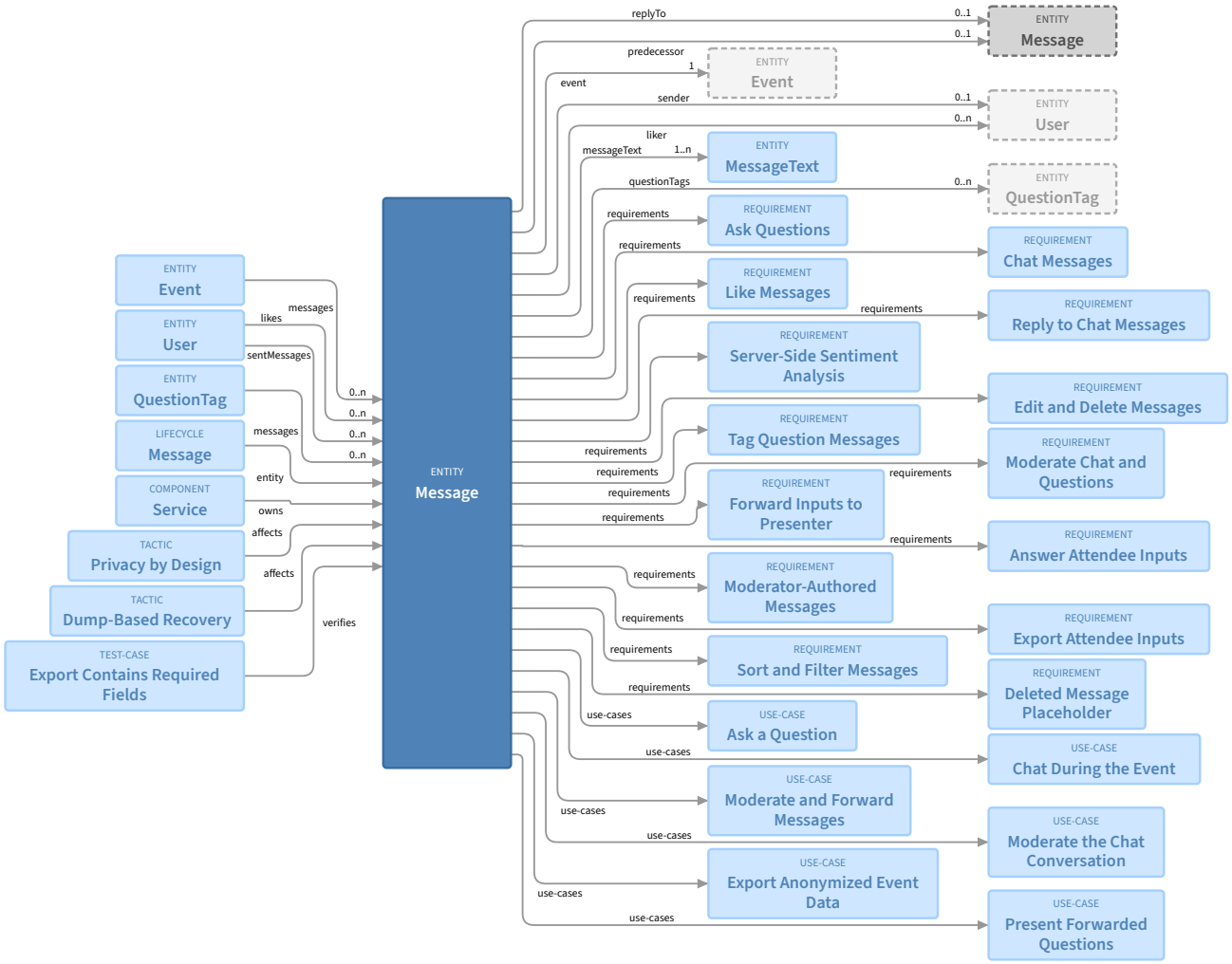
A helper entity identifying a person within an event for event-based logins and for the roles granted to them, or permanently and outside any event as the holder of the Administrator role — **BECAUSE** the system holds no permanent accounts except the administrator yet must identify attendees and role holders per event.

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	○ RETENTION	Description
userId ⚓	`key uuid`	`uuid()`	∅	∅	Unique identifier of the user — **BECAUSE** it is referenced as a foreign key
email ⌘ ⚓	`string?`	∅	Personal	until event finish, until event deletion for a holder of the Manager role, permanently for the holder of the Administrator role	Concrete email address of the user — **BECAUSE** authorization tokens are sent to this address at login and roles are granted to it
firstname ⚓	`string`	`""`	Personal	until event finish	Optional first name of the user — **BECAUSE** it is displayed on the user's chat and question messages

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	○ RETENTION	Description
lastname	`string`	`""`	Personal	until event finish	Optional last name of the user — **BECAUSE** it is displayed on the user's chat and question messages

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
roles	● **ENTITY: Role**	Composition	`0..n`	Roles granted to the user within its event — **BECAUSE** rights are held by the identified user rather than by a bare email address
likes	● **ENTITY: Message**	Association	`0..n`	Messages the user marked as liked — **BECAUSE** likes are tracked per user until anonymization
sentMessages	● **ENTITY: Message**	Association	`0..n`	Messages the user has sent — **BECAUSE** authorship links a message to its sending user
statistics	● **ENTITY: UserStatistic**	Aggregation	`0..n`	Periodic statistics about the user — **BECAUSE** viewer information is captured per user over time

11.08 ● ENTITY: Message ⚓



○ REQUIREMENTS	● REQUIREMENT: Chat Messages ⚭, ● REQUIREMENT: Ask Questions ⚭, ● REQUIREMENT: Like Messages ⚭, ● REQUIREMENT: Reply to Chat Messages ⚭, ● REQUIREMENT: Moderate Chat and Questions ⚭, ● REQUIREMENT: Forward Inputs to Presenter ⚭, ● REQUIREMENT: Answer Attendee Inputs ⚭, ● REQUIREMENT: Moderator-Authored Messages ⚭, ● REQUIREMENT: Edit and Delete Messages ⚭, ● REQUIREMENT: Deleted Message Placeholder ⚭, ● REQUIREMENT: Server-Side Sentiment Analysis ⚭, ● REQUIREMENT: Sort and Filter Messages ⚭, ● REQUIREMENT: Tag Question Messages ⚭, ● REQUIREMENT: Export Attendee Inputs ⚭
○ USE-CASES	● USE-CASE: Ask a Question ⚭, ● USE-CASE: Chat During the Event ⚭, ● USE-CASE: Moderate and Forward Messages ⚭, ● USE-CASE: Moderate the Chat Conversation ⚭, ● USE-CASE: Present Forwarded Questions ⚭, ● USE-CASE: Export Anonymized Event Data ⚭
○ TERMS	● TERM: Message ⚭, ● TERM: Chat ⚭, ● TERM: Question ⚭, ● TERM: Support ⚭, ● TERM: Like ⚭, ● TERM: Sentiment Analysis ⚭
○ GROUP	● GROUP: Messages ⚭

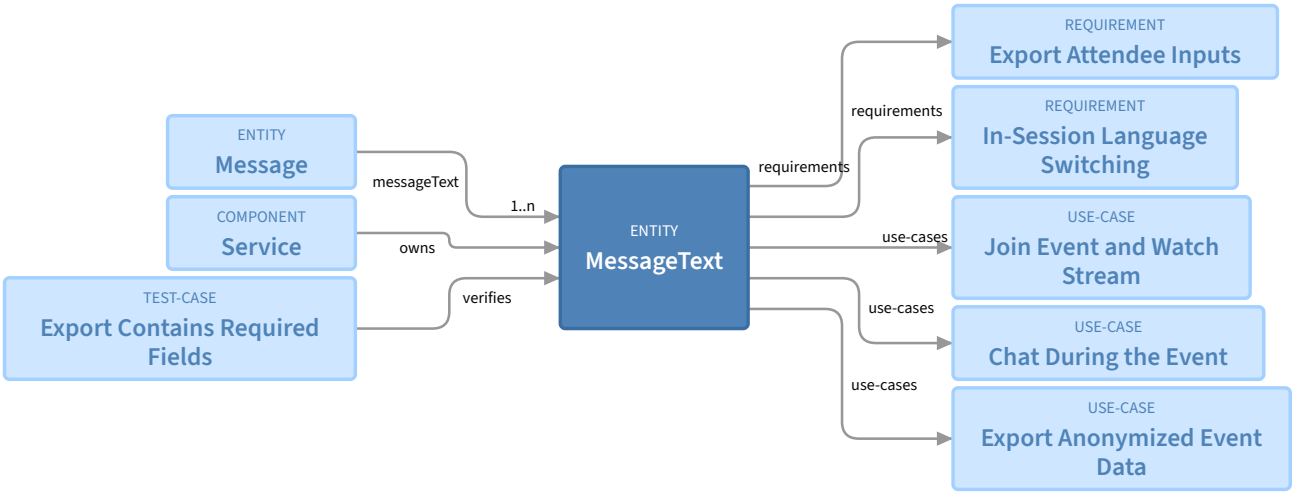
A single chat, support, or question item tracked for attendees and moderators — **BECAUSE** all event interaction is represented uniformly as messages with language-specific texts.

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	○ RETENTION	Description
messageId ⚓	key uuid	uuid()	∅	∅	Unique identifier of the message — **BECAUSE** it is the

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	○ RETENTION	Description
					foreign key for the translated message texts
type ⚓	`enum(Chat,Support,Question)`	`Chat`	∅	∅	The kind of message — **BECAUSE** each type has a distinct lifecycle and visibility
timestamp ⚓	`datetime`	`Date.now()`	∅	∅	Time the sender created the message — **BECAUSE** ordering and export require the creation time
timestampAnswered ⚓	`datetime`	`Date.now()`	∅	∅	Time the message was answered — **BECAUSE** the presenter records when a question was answered
state ⚓	`enum(pending,accepted,rejected,forwarded,answered,suspended)`	`pending`	∅	∅	Moderation and processing state of the message — **BECAUSE** the message moves through a defined moderation and presentation lifecycle
originalLanguage ⚓	`string?`	∅	∅	∅	Language the sender originally wrote the message in — **BECAUSE** the difference between human-written and AI-translated text must always be visible
senderName ⚓	`string?`	∅	Personal	until event finish, then replaced by “Anonymous”	Display name shown to others for the sender — **BECAUSE** the visible name depends on event naming and anonymity options
presenterAnnotation ⚓	`string?`	∅	∅	∅	Hint a moderator attaches for the presenter on forwarding — **BECAUSE** the presenter benefits from routing guidance on a forwarded message
likes ⚓	`integer`	`0`	∅	∅	Computed number of likes conserved on event finish — **BECAUSE** the like total must survive removal of liker relations for GDPR
sentimentScore ⚓	`float?`	∅	∅	∅	Server-side sentiment score between -1 and 1 — **BECAUSE** the analysis result is stored for display and auto-moderation
edited ⚓	`enum(none,insignificant,significant,deleted)?`	∅	∅	∅	Whether and how the message was changed or deleted — **BECAUSE** edits must be marked for others and edits stop once forwarded

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
sender	● ENTITY: User	Association	0..1	The authoring attendee of the message — **BECAUSE** a message has an author until the sender is removed on finish
liker	● ENTITY: User	Association	0..n	Attendees who liked the message — **BECAUSE** likes are tracked per liking user before anonymization
event	● ENTITY: Event	Association	1	The event the message belongs to — **BECAUSE** the event link must persist even after senders are deleted
replyTo	● ENTITY: Message	Association	0..1	The message this message replies to — **BECAUSE** chat replies and moderator answers chain messages together
predecessor	● ENTITY: Message	Association	0..1	The preceding message in a manual ordering — **BECAUSE** moderators sort forwarded messages for the presenter
questionTags	● ENTITY: QuestionTag	Association	0..n	Tags attached to a question message — **BECAUSE** questions can be tagged with zero or more tags for context
messageText	● ENTITY: MessageText	Composition	1..n	The message texts of this message — **BECAUSE** each message text can be translated to multiple languages

11.09 ● ENTITY: MessageText

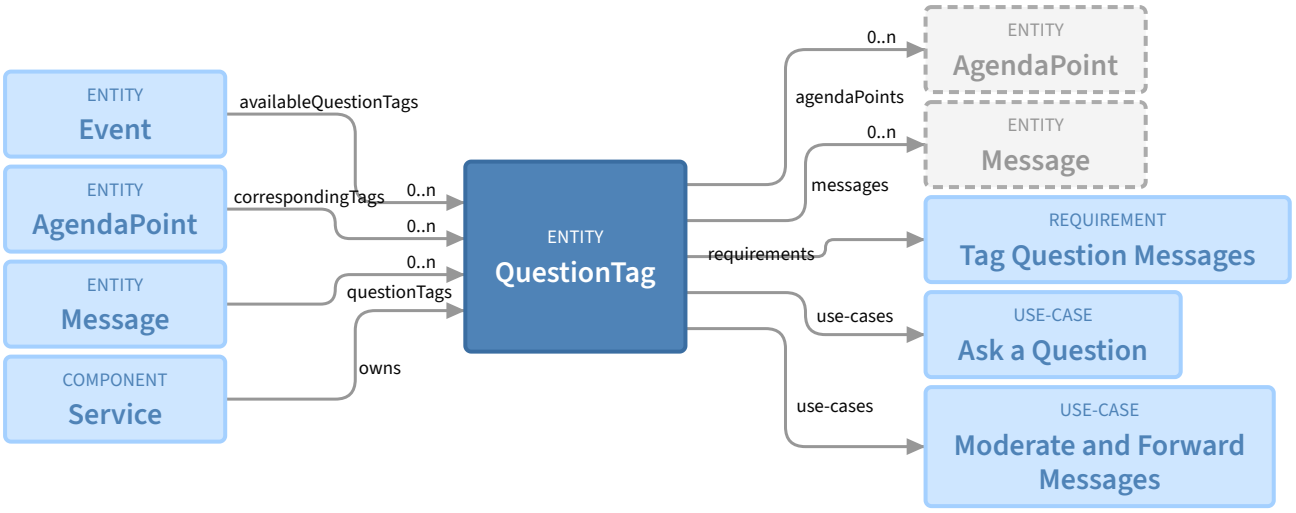


○ REQUIREMENTS	● REQUIREMENT: In-Session Language Switching, ● REQUIREMENT: Export Attendee Inputs
○ USE-CASES	● USE-CASE: Join Event and Watch Stream, ● USE-CASE: Chat During the Event, ● USE-CASE: Export Anonymized Event Data
○ TERMS	● TERM: Message
○ GROUP	● GROUP: Messages

A language-specific text of a message — **BECAUSE** a message is translated into multiple languages while retaining one original.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
messageTextId ⚓	`key uuid`	`uuid()`	Unique identifier of the message text — **BECAUSE** it is referenced as a foreign key
language ⚓	`string?`	∅	Language the text is written in — **BECAUSE** each text variant is identified by its language
text ⚓	`string?`	∅	The message text in the stated language — **BECAUSE** the displayed content depends on the chosen language

11.10 ● ENTITY: QuestionTag ⚓



○ REQUIREMENTS	● **REQUIREMENT:** Tag Question Messages ⚭
○ USE-CASES	● **USE-CASE:** Ask a Question ⚭, ● **USE-CASE:** Moderate and Forward Messages ⚭
○ GROUP	● **GROUP:** Events ⚭

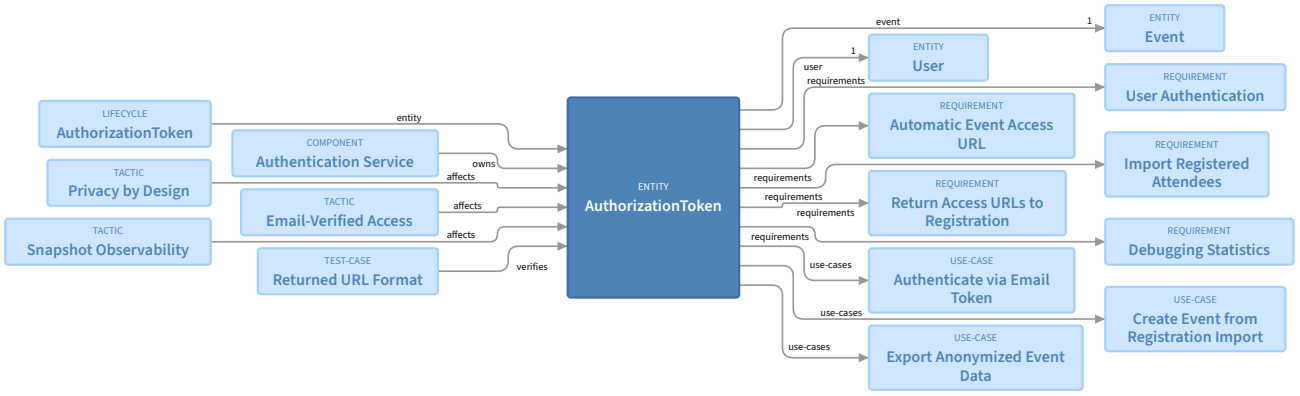
A named tag attachable to question messages — **BECAUSE** questions are categorized by topic or addressed person for routing and grouping.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
questionTagId ⚓	`key uuid`	`uuid()`	Unique identifier of the question tag — **BECAUSE** it is referenced as a foreign key
text ⚓	`unique string`	∅	Unique tag name — **BECAUSE** tags are identified and displayed by their name
moderatorOnly ⚓	`boolean?`	∅	Whether only a Moderator or Manager may use the tag — **BECAUSE** some tags are reserved for the moderation team
group ⚓	`string?`	∅	Logical group the tag belongs to such as a topic or person — **BECAUSE** tags are organized into meaningful groups

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
messages ⚓	● **ENTITY:** Message ⚭	Association	`0..n`	The question messages the tag is attached to — **BECAUSE** a tag categorizes many questions and its

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
				removal has to detach it from all of them
agendaPoints	● ENTITY: AgendaPoint	Association	`0..n`	The agenda points the tag corresponds to — **BECAUSE** a tag can relate to several phases of the event

11.11 ● ENTITY: AuthorizationToken



○ REQUIREMENTS	● REQUIREMENT: User Authentication, ● REQUIREMENT: Automatic Event Access URL, ● REQUIREMENT: Import Registered Attendees, ● REQUIREMENT: Return Access URLs to Registration, ● REQUIREMENT: Debugging Statistics
○ USE-CASES	● USE-CASE: Authenticate via Email Token, ● USE-CASE: Create Event from Registration Import, ● USE-CASE: Export Anonymized Event Data
○ TERMS	● TERM: Authorization Token
○ GROUP	● GROUP: Tokens

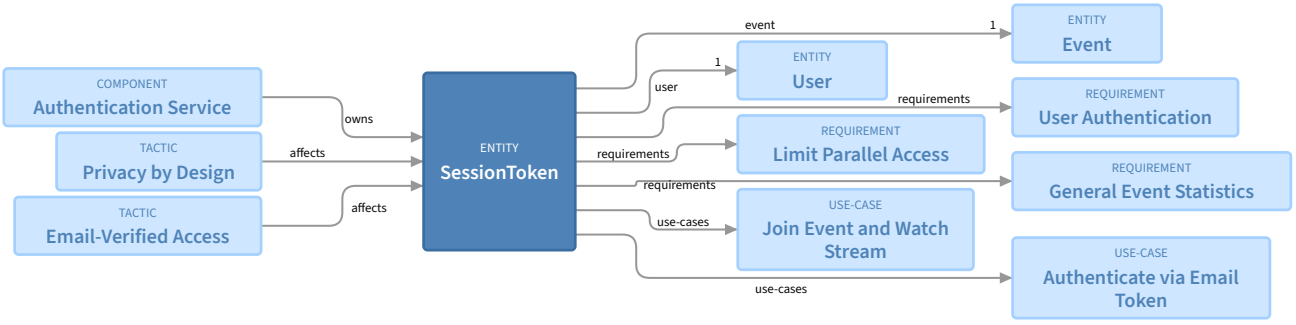
A one-time second factor proving an attendee controls the email address used as the first factor — **BECAUSE** email-verified access is the core mechanism limiting the audience.

● ATTRIBUTE	Properties & Description				
token	○ TYPE	○ DEFAULT	○ CONSTRAINT	○ CLASSIFICATION	○ RETENTION
	`string?`	∅	`six digits as NNN-NNN`	Confidential	until validity expiry, at the latest until event finish
	Description				
	The generated one-time token for the next login attempt — **BECAUSE** the attendee proves control of the email by returning this token				
validUntil	○ TYPE	○ DEFAULT	○ CONSTRAINT	○ CLASSIFICATION	○ RETENTION
	`datetime`	`Date.now() + 1d`	∅	∅	∅
	Description				
	Expiry time of the token, unset for pre-generated tokens — **BECAUSE** ordinary tokens expire within minutes while pre-generated ones last until event end				
state	○ TYPE	○ DEFAULT	○ CONSTRAINT	○ CLASSIFICATION	○ RETENTION
	`enum(issued,sent,used)`	`issued`	∅	∅	∅

● ATTRIBUTE	Properties & Description Description
	Lifecycle state of the token — **BECAUSE** debugging statistics and anonymized sums need the token state

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
user ⚓	● **ENTITY: User**	Association	1	The user the token was issued for — **BECAUSE** a token authorizes exactly one user
event ⚓	● **ENTITY: Event**	Association	1	The event the token was issued for — **BECAUSE** a token grants access to exactly one event

11.12 ● ENTITY: SessionToken ⚓



○ **REQUIREMENTS**	● **REQUIREMENT: User Authentication**, ● **REQUIREMENT: Limit Parallel Access**, ● **REQUIREMENT: General Event Statistics**
○ **USE-CASES**	● **USE-CASE: Authenticate via Email Token**, ● **USE-CASE: Join Event and Watch Stream**
○ **TERMS**	● **TERM: Session Token**
○ **GROUP**	● **GROUP: Tokens**

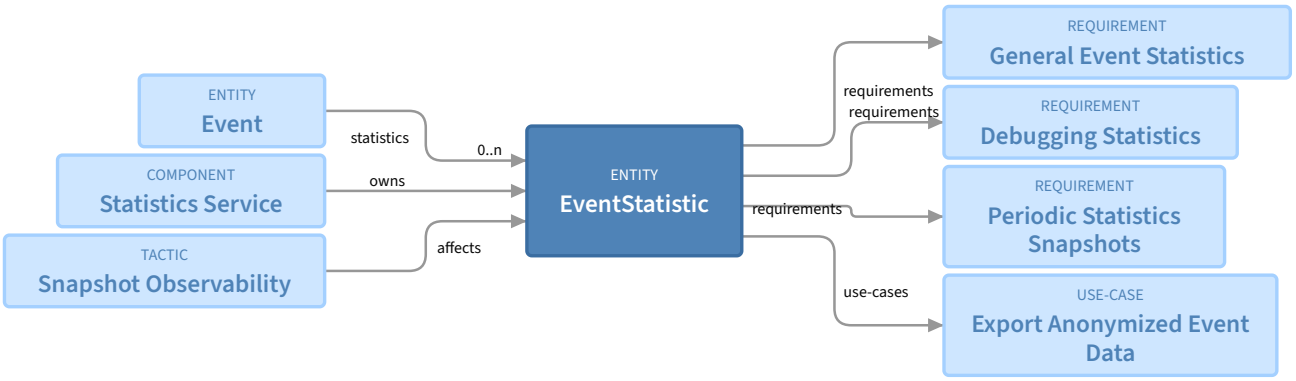
The result of a successful login of a user to an event — **BECAUSE** an active session must be tracked to enforce single concurrent access.

● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	○ RETENTION	Description
sessionId ⚓	key uuid	`uuid()`	Confidential	until event finish	Unique identifier of the session — **BECAUSE** the active session of a user for an event must be addressable
issuedAt ⚓	`datetime`	`Date.now()`	∅	∅	Time the user successfully entered the event — **BECAUSE** the session start time is recorded for tracking

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
user ⚓	● **ENTITY: User**	Association	1	The user the session was issued for — **BECAUSE** a session belongs to exactly one user

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
event ⚓	● ENTITY: Event ⚭	Association	1	The event the session was issued for — **BECAUSE** a session grants access to exactly one event

11.13 ● ENTITY: EventStatistic ⚓

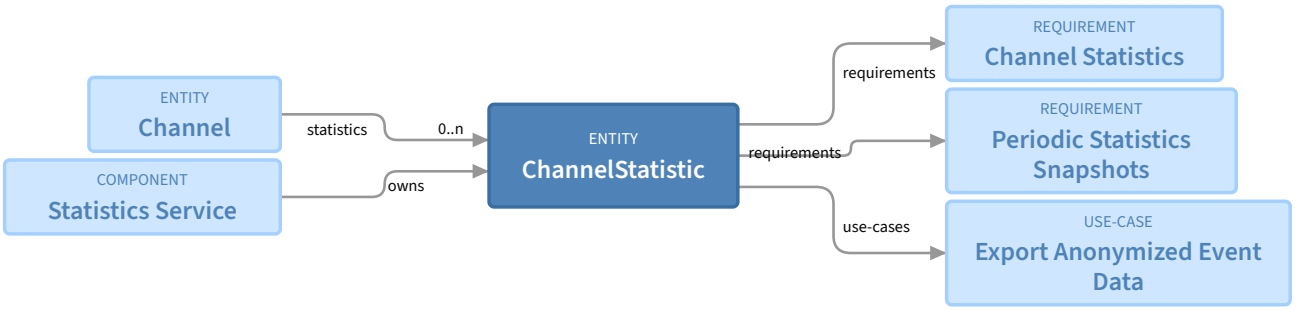


○ REQUIREMENTS	● REQUIREMENT: General Event Statistics ⚭, ● REQUIREMENT: Debugging Statistics ⚭, ● REQUIREMENT: Periodic Statistics Snapshots ⚭
○ USE-CASES	● USE-CASE: Export Anonymized Event Data ⚭
○ GROUP	● GROUP: Statistics ⚭

A periodic cumulative snapshot of event-wide counts — **BECAUSE** trend visualization of audience size and authentication flow requires regular snapshots.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
eventStatisticId ⚓	`key uuid`	`uuid()`	Unique identifier of the event statistic — **BECAUSE** it is referenced as a foreign key
timestamp ⚓	`datetime`	`Date.now()`	Time the snapshot was created — **BECAUSE** statistics are plotted over time
numberOfIssuedAuthTokens ⌘ ⚓	`integer?`	∅	Count of issued authorization tokens at the timestamp — **BECAUSE** debugging statistics track issued tokens
numberOfSentAuthTokens ⌘ ⚓	`integer?`	∅	Count of sent authorization tokens at the timestamp — **BECAUSE** debugging statistics track sent tokens
numberOfUsedAuthTokens ⌘ ⚓	`integer?`	∅	Count of used authorization tokens at the timestamp — **BECAUSE** debugging statistics track used tokens
numberOfSessionTokens ⌘ ⚓	`integer?`	∅	Count of session tokens at the timestamp — **BECAUSE** logged-in users are derived from session tokens
numberOfConnections ⌘ ⚓	`integer?`	∅	Count of active MQTT connections at the timestamp — **BECAUSE** active viewers differ from sessions once an attendee leaves

11.14 ● ENTITY: ChannelStatistic ⚓

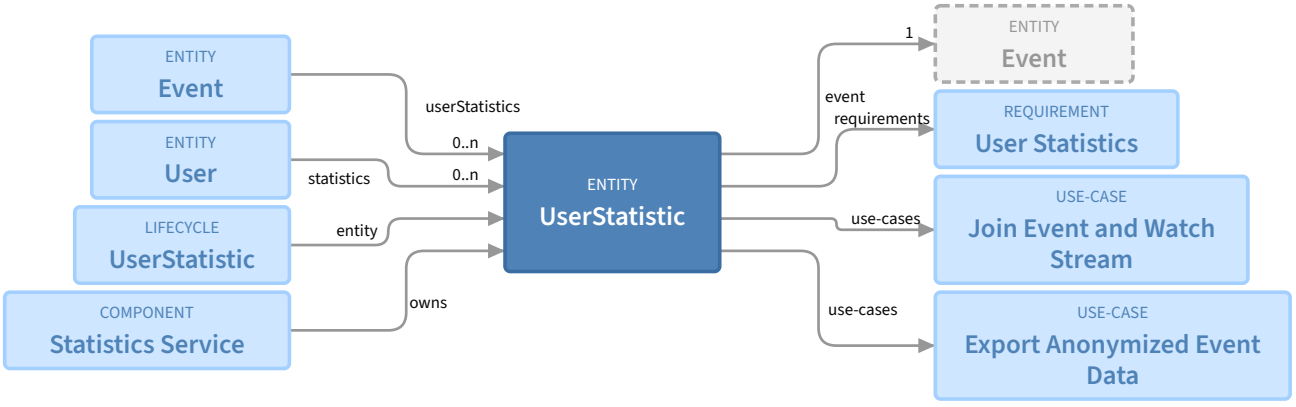


○ REQUIREMENTS	● REQUIREMENT: Channel Statistics ⚭, ● REQUIREMENT: Periodic Statistics Snapshots ⚭
○ USE-CASES	● USE-CASE: Export Anonymized Event Data ⚭
○ GROUP	● GROUP: Statistics ⚭

A periodic count of viewers for a channel — **BECAUSE** organizers need per-channel popularity over time.

● ATTRIBUTE	○ TYPE	○ DEFAULT	Description
channelStatisticId ⚓	`key uuid`	`uuid()`	Unique identifier of the channel statistic — **BECAUSE** it is referenced as a foreign key
timestamp ⚓	`datetime`	`Date.now()`	Time the snapshot was created — **BECAUSE** channel statistics are plotted over time
numberOfViewers ⌘ ⚓	`integer?`	∅	Count of viewers of the channel at the timestamp — **BECAUSE** the per-channel viewer count is the tracked metric

11.15 ● ENTITY: UserStatistic ⚓



○ REQUIREMENTS	● REQUIREMENT: User Statistics ⚭
○ USE-CASES	● USE-CASE: Join Event and Watch Stream ⚭, ● USE-CASE: Export Anonymized Event Data ⚭
○ GROUP	● GROUP: Statistics ⚭

Tracked viewer information about a user — **BECAUSE** audience composition informs reporting and default localization.

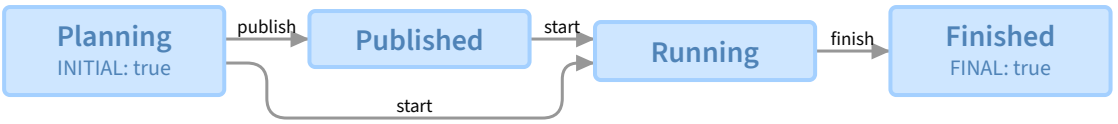
● ATTRIBUTE	○ TYPE	○ DEFAULT	○ CLASSIFICATION	○ RETENTION	Description
userStatisticId ⚓	`key uuid`	`uuid()`	∅	∅	Unique identifier of the user statistic — **BECAUSE** it is referenced as a foreign key
timestamp ⚓	`datetime`	`Date.now()`	∅	∅	Time the snapshot was created — **BECAUSE** user statistics are recorded over time
country ⌘ ⚓	`string?`	∅	Personal	linked to the user until event finish, retained unlinked afterwards	ISO country code from GeoIP tracking — **BECAUSE** country selects the default application language on first use
browserType ⌘ ⚓	`string?`	∅	Personal	linked to the user until event finish, retained unlinked afterwards	Type of browser used — **BECAUSE** browser distribution informs compatibility decisions
deviceType ⌘ ⚓	`string?`	∅	Personal	linked to the user until event finish, retained unlinked afterwards	Type of device used — **BECAUSE** device distribution informs responsive design priorities
viewportWidth ⌘ ⚓	`integer?`	∅	∅	∅	Width in pixels of the browser viewport — **BECAUSE** viewport sizing informs layout decisions
viewportHeight ⌘ ⚓	`integer?`	∅	∅	∅	Height in pixels of the browser viewport — **BECAUSE** viewport sizing informs layout decisions

● RELATION	○ TARGET	○ KIND	○ ARITY	Description
event ⚓	● **ENTITY: Event** ⚮	Association	1	The event the statistic was recorded in — **BECAUSE** the statistic must stay selectable per event once its user reference is dropped

12 ● DATA: State Model (SM) ⚓

Event	Message	AuthorizationToken	UserStatistic
ENTITY: Event	ENTITY: Message	ENTITY: AuthorizationToken	ENTITY: UserStatistic

12.01 ● LIFECYCLE: Event



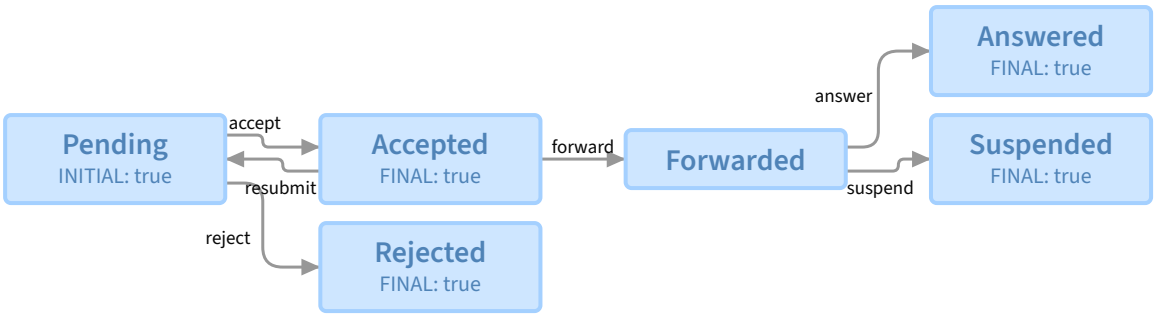
○ ENTITY	● ENTITY: Event

● STATE	○ INITIAL	○ FINAL	Description
Planning	true	∅	The event is created and configured but not visible to attendees
Published	∅	∅	The event is visible to attendees but not yet started
Running	∅	∅	The event is live and attendees can attend and interact
Finished	∅	true	The event has ended, access is closed, and data is anonymized

● TRANSITION	Properties & Description		
publish	○ FROM	○ TO	○ ACTOR
	● STATE: Planning	● STATE: Published	● ROLE: Manager
	○ GUARD	○ RULES	○ USE-CASES
	The event is fully configured.	∅	● SCENARIO: Publish, Start, and Finish
	Description		
	The event becomes visible to invited attendees		
start	○ FROM	○ TO	○ ACTOR
	● STATE: Published	● STATE: Running	● ROLE: Manager
	○ GUARD	○ RULES	○ USE-CASES
	∅	∅	● SCENARIO: Publish, Start, and Finish
	Description		
	The live stream and interaction channels open for attendees		
start	○ FROM	○ TO	○ ACTOR
	● STATE: Planning	● STATE: Running	● ROLE: Manager
	○ GUARD	○ RULES	○ USE-CASES
	∅	∅	● SCENARIO: Start Without Publishing
	Description		
	The event goes live directly from planning, without ever having been visible beforehand		
finish	○ FROM	○ TO	○ ACTOR
	● STATE: Running	● STATE: Finished	● ROLE: Manager
	○ GUARD	○ RULES	○ USE-CASES
	∅	● RULE: Anonymization on Event Finish	● SCENARIO: Publish, Start, and Finish

● TRANSITION	Properties & Description
	Description
	The anonymization procedure runs and access is closed

12.02 ● LIFECYCLE: Message



○ ENTITY	● ENTITY: Message

● STATE	○ INITIAL	○ FINAL	Description
Pending	true	∅	The attendee has submitted the message and it awaits moderation
Accepted	∅	true	The message is approved and visible to the audience if configured
Rejected	∅	true	The message is declined and will be deleted entirely on event finish
Forwarded	∅	∅	The accepted message is handed to the presenter as a work item
Answered	∅	true	The presenter has processed and answered the message in the live event
Suspended	∅	true	The presenter will not process the message in the live event

● TRANSITION	Properties & Description		
accept	○ FROM	○ TO	○ ACTOR
	● STATE: Pending	● STATE: Accepted	● ROLE: Moderator
	○ GUARD	○ RULES	○ USE-CASES
	∅	● RULE: Moderation Determines Initial Visibility	● SCENARIO: Approve and Forward Question
	Description		
	The message becomes visible to the audience		
accept	○ FROM	○ TO	○ ACTOR
	● STATE: Pending	● STATE: Accepted	System
	○ GUARD	○ RULES	○ USE-CASES
	The server-side sentiment analysis scores the message at or above the threshold and auto-accept is enabled.	● RULE: Moderation Determines Initial Visibility, ● RULE: Sentiment Auto-Decision Threshold	● SCENARIO: Auto-Accepted by Sentiment

● TRANSITION	Properties & Description		
	Description		
	The message becomes visible to the audience without a moderator decision		
accept ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Pending	● STATE: Accepted	System
	○ GUARD	○ RULES	○ USE-CASES
	The event does not moderate messages of this type.	● RULE: Moderation Determines Initial Visibility	● SCENARIO: Post a Chat Message, ● SCENARIO: Reply to a Chat Message
	Description		
	The message becomes visible to the audience at once, right after its creation		
accept ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Pending	● STATE: Accepted	System
	○ GUARD	○ RULES	○ USE-CASES
	The message is authored by a moderator.	● RULE: Moderator Messages Auto-Accepted	● SCENARIO: Answer an Attendee Directly, ● SCENARIO: Seed the Conversation as Moderator
	Description		
	The moderator-authored message becomes visible at once, right after its creation		
reject ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Pending	● STATE: Rejected	● ROLE: Moderator
	○ GUARD	○ RULES	○ USE-CASES
	∅	● RULE: Moderation Determines Initial Visibility	● SCENARIO: Reject Improper Message
	Description		
	The message is hidden and marked for deletion		
reject ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Pending	● STATE: Rejected	System
	○ GUARD	○ RULES	○ USE-CASES
	The server-side sentiment analysis scores the message below the threshold and auto-reject is enabled.	● RULE: Moderation Determines Initial Visibility, ● RULE: Sentiment Auto-Decision Threshold	● SCENARIO: Auto-Rejected by Sentiment
	Description		
	The message is hidden and marked for deletion without a moderator decision		
resubmit ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Accepted	● STATE: Pending	● ROLE: Attendee
	○ GUARD	○ RULES	○ USE-CASES
	The event moderates messages of this type.	● RULE: Moderation Determines Initial Visibility, ● RULE: Forwarded Messages Are Immutable	● SCENARIO: Edit an Own Message
	Description		
	The edited message is hidden from the audience again until it passes the moderation anew		

● TRANSITION	Properties & Description		
forward	○ FROM	○ TO	○ ACTOR
	● STATE: Accepted	● STATE: Forwarded	● ROLE: Moderator
	○ GUARD	○ RULES	○ USE-CASES
	The message is a question.	● RULE: State Set Depends on Message Type, ● RULE: Forwarded Messages Are Immutable	● SCENARIO: Approve and Forward Question
	Description		
	The message enters the presenter's work basket and becomes immutable		
answer	○ FROM	○ TO	○ ACTOR
	● STATE: Forwarded	● STATE: Answered	● ROLE: Presenter, ● ROLE: Moderator
	○ GUARD	○ RULES	○ USE-CASES
	∅	∅	● SCENARIO: Answer Forwarded Question, ● SCENARIO: Settle a Question for the Presenter
	Description		
	The answered timestamp is recorded		
suspend	○ FROM	○ TO	○ ACTOR
	● STATE: Forwarded	● STATE: Suspended	● ROLE: Presenter, ● ROLE: Moderator
	○ GUARD	○ RULES	○ USE-CASES
	∅	∅	● SCENARIO: Suspend a Forwarded Question, ● SCENARIO: Settle a Question for the Presenter
	Description		
	The message is set aside for the live event		

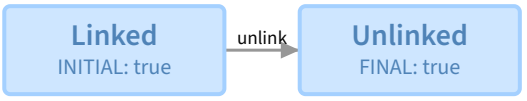
12.03 ● LIFECYCLE: AuthorizationToken

○ ENTITY	● ENTITY: AuthorizationToken

● STATE	○ INITIAL	○ FINAL	Description
Issued	true	∅	The token has been generated, typically at event creation
Sent	∅	∅	The token has been delivered to the user by email
Used	∅	true	The token has been consumed in a login attempt

● TRANSITION	Properties & Description		
send ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Issued ⚭	● STATE: Sent ⚭	● ROLE: Attendee ⚭
	○ GUARD	○ RULES	○ USE-CASES
	The email of the attendee is on the access list or matches the access pattern.	● RULE: Access Requires List or Pattern Match ⚭	● SCENARIO: Login with Emailed Token ⚭
	Description		
	The token is emailed to the attendee requesting a login challenge		
consume ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Sent ⚭	● STATE: Used ⚭	● ROLE: Attendee ⚭
	○ GUARD	○ RULES	○ USE-CASES
	The token has not expired.	● RULE: Token Format and Validity ⚭, ● RULE: Single Active Session per Event ⚭	● SCENARIO: Login with Emailed Token ⚭
	Description		
	The token is marked spent by a successful or unsuccessful login attempt		
consume ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Issued ⚭	● STATE: Used ⚭	● ROLE: Attendee ⚭
	○ GUARD	○ RULES	○ USE-CASES
	The event allows automatic-access URLs.	● RULE: Token Format and Validity ⚭	● SCENARIO: Automatic Token in URL ⚭
	Description		
	The pre-generated token is marked spent by the use of an automatic-access URL carrying it		

12.04 ● LIFECYCLE: UserStatistic ⚓

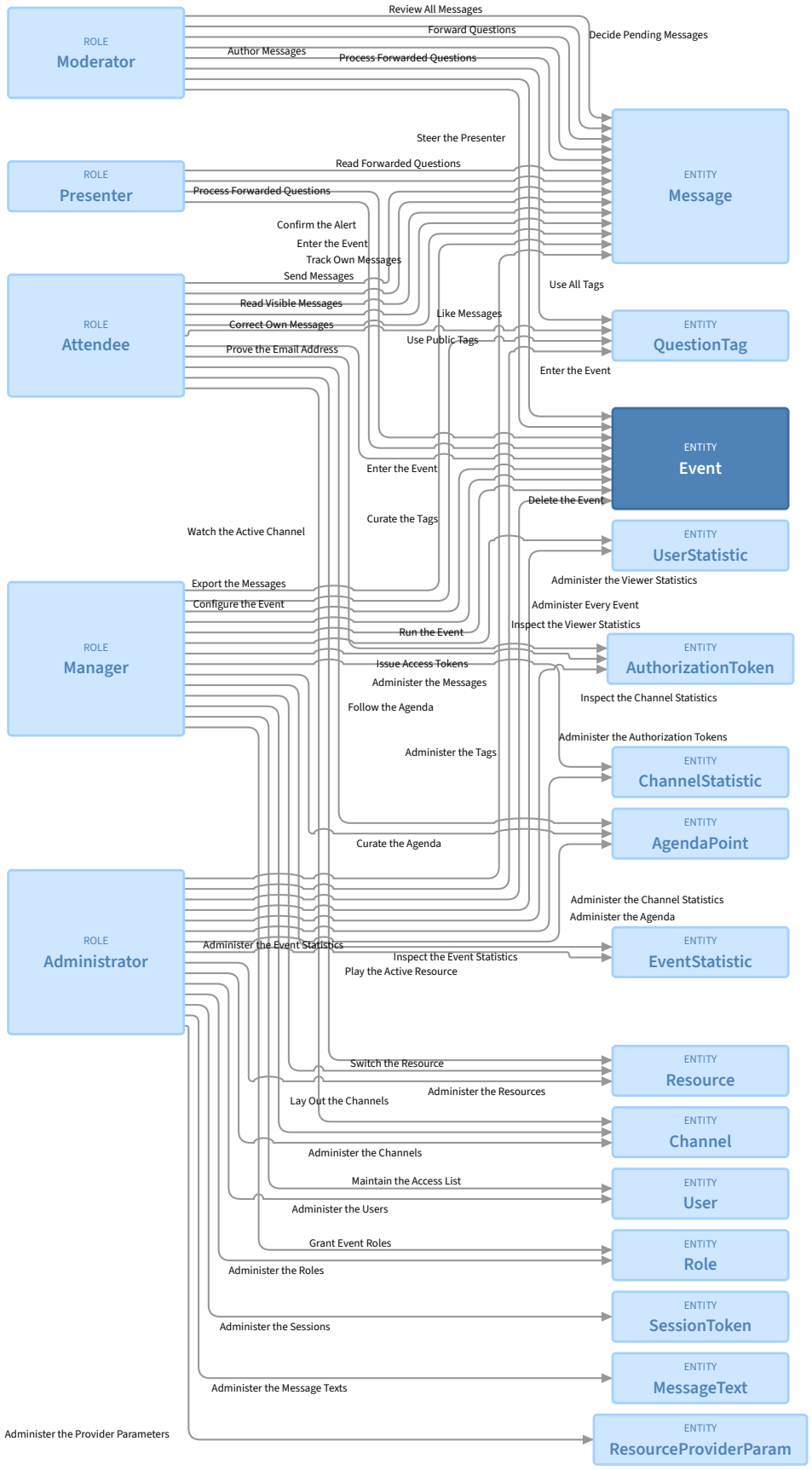


○ ENTITY	● ENTITY: UserStatistic ⚭

● STATE	○ INITIAL	○ FINAL	Description
Linked ⚓	true	∅	The statistic is attributed to the user it was recorded for
Unlinked ⚓	∅	true	The statistic is retained without any reference to a user

● TRANSITION	Properties & Description		
unlink ⚓	○ FROM	○ TO	○ ACTOR
	● STATE: Linked ⚭	● STATE: Unlinked ⚭	System
	○ RULES	○ USE-CASES	Description
	● RULE: Anonymization on Event Finish ⚭	● SCENARIO: Publish, Start, and Finish ⚭	The anonymization drops the user reference of the statistic on event finish

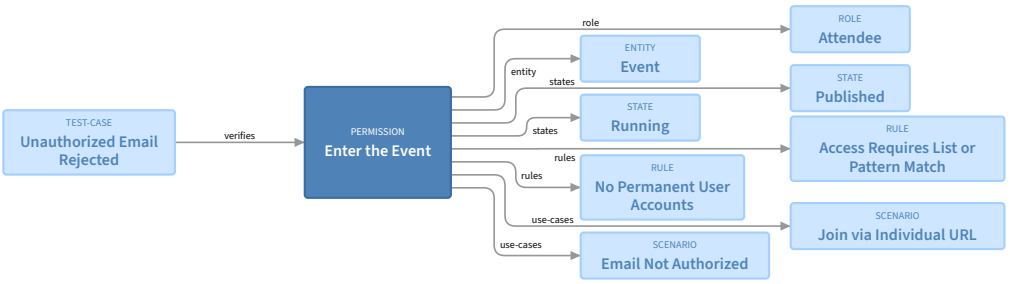
13 ● DATA: Authorization Model (AM) ⚓



● PERMISSION	Properties & Description

Enter the Event

Diagram



Properties

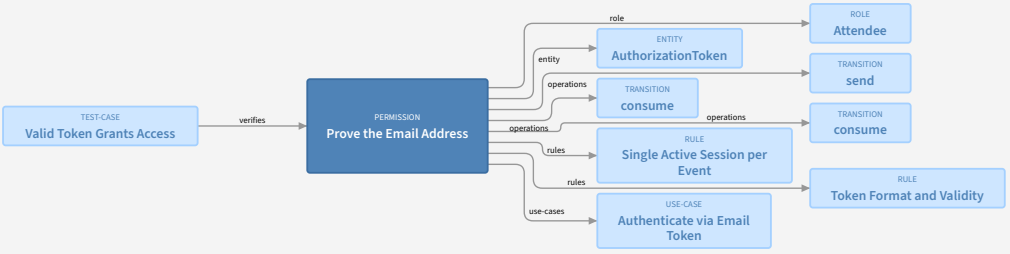
○ **ROLE**	● **ROLE:** Attendee
○ **ENTITY**	● **ENTITY:** Event
○ **OPERATIONS**	Read
○ **STATES**	● **STATE:** Published, ● **STATE:** Running
○ **CONDITION**	The email of the attendee is on the access list or matches the access email pattern of the event, or the event allows anonymous access.
○ **RULES**	● **RULE:** Access Requires List or Pattern Match, ● **RULE:** No Permanent User Accounts
○ **USE-CASES**	● **SCENARIO:** Join via Individual URL, ● **SCENARIO:** Email Not Authorized

Description

The attendee sees the event with its login and interaction information, but none of its configuration

Prove the Email Address

Diagram



Properties

○ **ROLE**	● **ROLE:** Attendee
○ **ENTITY**	● **ENTITY:** AuthorizationToken
○ **OPERATIONS**	● **TRANSITION:** send, ● **TRANSITION:** consume, ● **TRANSITION:** consume
○ **CONDITION**	The token was issued for the email address of the attendee.
○ **RULES**	● **RULE:** Token Format and Validity, ● **RULE:** Single Active Session per Event
○ **USE-CASES**	● **USE-CASE:** Authenticate via Email Token

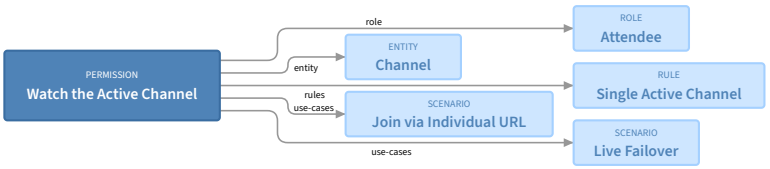
Description

The attendee requests, receives, and returns the one-time token of their own login challenge

● PERMISSION	Properties & Description

Watch the Active Channel

Diagram



Properties

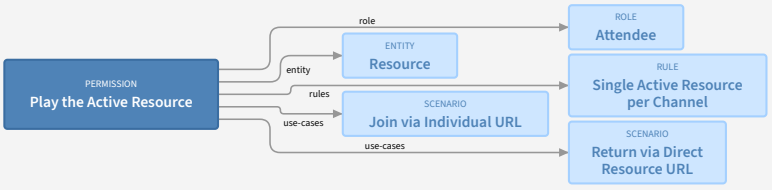
○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: Channel
○ OPERATIONS	Read
○ CONDITION	The channel belongs to the event the attendee entered.
○ RULES	● RULE: Single Active Channel
○ USE-CASES	● SCENARIO: Join via Individual URL, ● SCENARIO: Live Failover

Description

The attendee sees the channels of the event and follows the active one

Play the Active Resource

Diagram



Properties

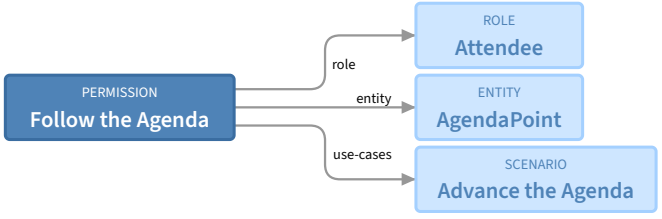
○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: Resource
○ OPERATIONS	Read
○ CONDITION	The resource backs a channel of the event the attendee entered.
○ RULES	● RULE: Single Active Resource per Channel
○ USE-CASES	● SCENARIO: Join via Individual URL, ● SCENARIO: Return via Direct Resource URL

Description

The attendee plays the active resource, or the resource named by the event URL, without seeing its provider parameters

Follow the Agenda

Diagram



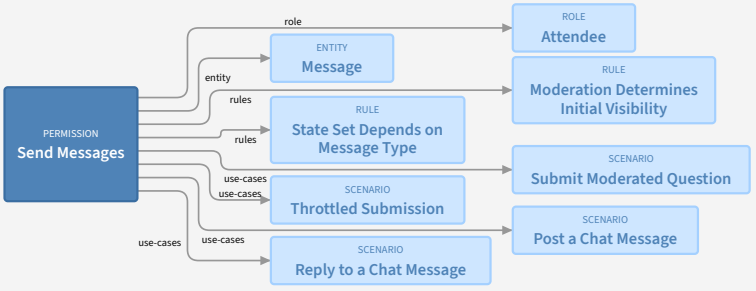
Properties

○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: AgendaPoint
○ OPERATIONS	Read

● PERMISSION	Properties & Description	
	○ CONDITION	The agenda point belongs to the event the attendee entered.
	○ USE-CASES	● SCENARIO: Advance the Agenda
	Description	
	The attendee sees the agenda of the event and which agenda point is currently active	

Send Messages

Diagram



Properties

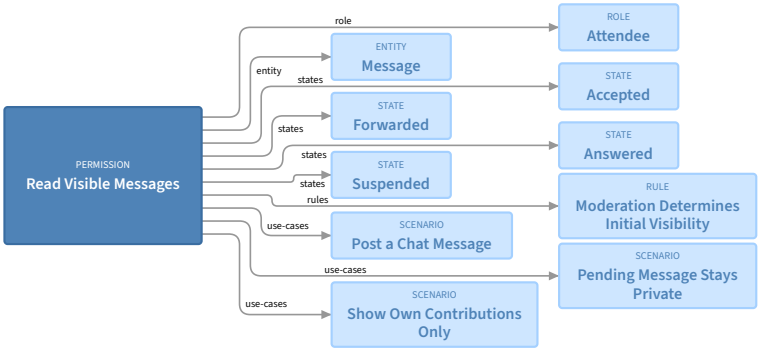
○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Create
○ CONDITION	The event is running, the type of the message is enabled for the event, and the throttling limit is not exceeded.
○ RULES	● RULE: Moderation Determines Initial Visibility, ● RULE: State Set Depends on Message Type
○ USE-CASES	● SCENARIO: Submit Moderated Question, ● SCENARIO: Throttled Submission, ● SCENARIO: Post a Chat Message, ● SCENARIO: Reply to a Chat Message

Description

The attendee sends chat, question, and support messages of their own, which await the moderation decision or are accepted at once, as the moderation setting of the event determines

Read Visible Messages

Diagram



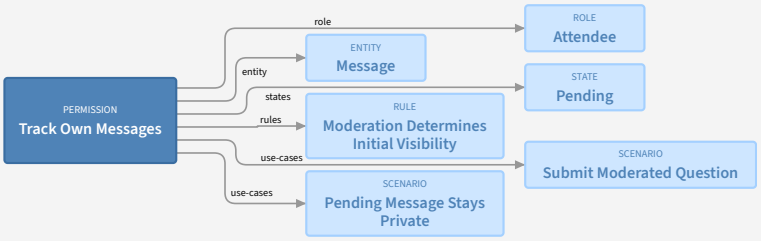
Properties

○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Read
○ STATES	● STATE: Accepted, ● STATE: Forwarded, ● STATE: Answered, ● STATE: Suspended
○ CONDITION	The message is a chat, a question the event does not keep private, or a support message of the attendee's own conversation.

● PERMISSION	Properties & Description
	○ RULES: ● RULE: Moderation Determines Initial Visibility
	○ USE-CASES: ● SCENARIO: Post a Chat Message, ● SCENARIO: Show Own Contributions Only, ● SCENARIO: Pending Message Stays Private
	Description
	The attendee reads the messages which passed the moderation, with their texts and like counts, under the sender name the event configures

Track Own Messages

Diagram



Properties

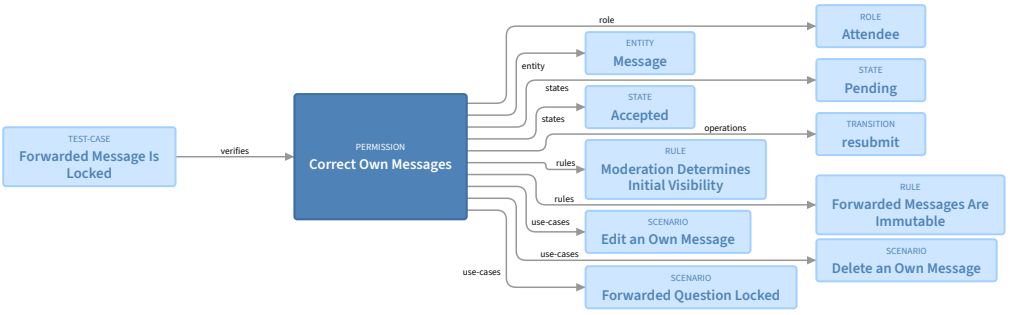
○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Read
○ STATES	● STATE: Pending
○ CONDITION	The attendee is the sender of the message.
○ RULES	● RULE: Moderation Determines Initial Visibility
○ USE-CASES	● SCENARIO: Submit Moderated Question, ● SCENARIO: Pending Message Stays Private

Description

The attendee sees their own messages while they still await the moderation decision

Correct Own Messages

Diagram



Properties

○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Update, Delete, ● TRANSITION: resubmit
○ STATES	● STATE: Pending, ● STATE: Accepted
○ CONDITION	The attendee is the sender of the message.
○ RULES	● RULE: Forwarded Messages Are Immutable, ● RULE: Moderation Determines Initial Visibility
○ USE-CASES	● SCENARIO: Edit an Own Message, ● SCENARIO: Delete an Own Message, ● SCENARIO: Forwarded Question Locked

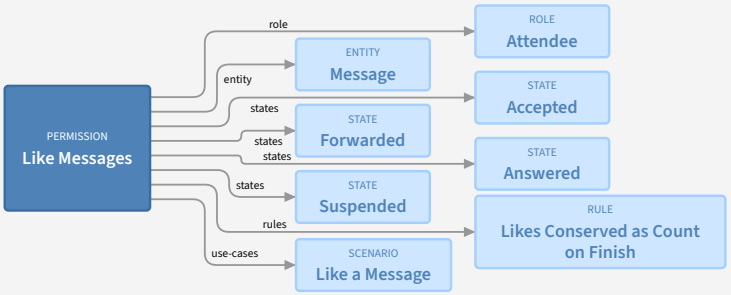
● PERMISSION	Properties & Description

Description

The attendee edits or deletes their own chat messages and questions until a question is forwarded, where an edit is marked and resubmitted to the moderation as the event requires

Like Messages

Diagram



Properties

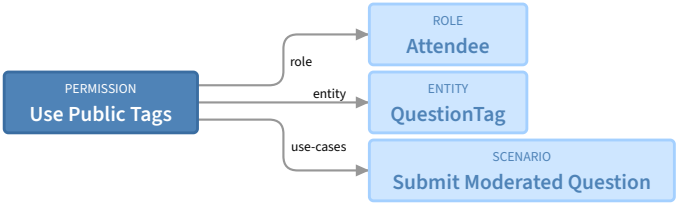
○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Update
○ STATES	● STATE: Accepted, ● STATE: Forwarded, ● STATE: Answered, ● STATE: Suspended
○ CONDITION	The update is confined to adding or revoking the attendee's own like on a chat or question of another attendee.
○ RULES	● RULE: Likes Conserved as Count on Finish
○ USE-CASES	● SCENARIO: Like a Message

Description

The attendee likes the visible messages of others and revokes the like again until the event finishes

Use Public Tags

Diagram



Properties

○ ROLE	● ROLE: Attendee
○ ENTITY	● ENTITY: QuestionTag
○ OPERATIONS	Read
○ CONDITION	The tag is not reserved for moderators.
○ USE-CASES	● SCENARIO: Submit Moderated Question

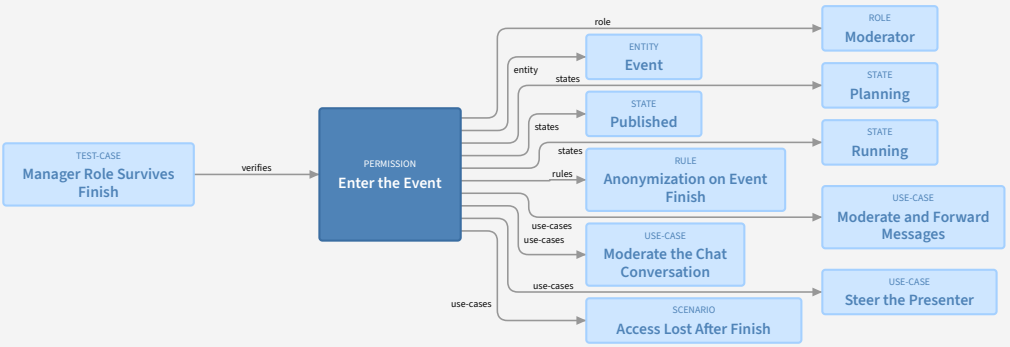
Description

The attendee sees the available tags of the event and attaches them to their own questions

● PERMISSION	Properties & Description

Enter the Event

Diagram



Properties

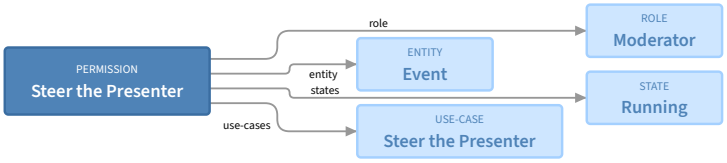
○ **ROLE**	● **ROLE:** Moderator
○ **ENTITY**	● **ENTITY:** Event
○ **OPERATIONS**	Read
○ **STATES**	● **STATE:** Planning, ● **STATE:** Published, ● **STATE:** Running
○ **CONDITION**	The moderator holds the Moderator role of the event.
○ **RULES**	● **RULE:** Anonymization on Event Finish
○ **USE-CASES**	● **USE-CASE:** Moderate and Forward Messages, ● **USE-CASE:** Moderate the Chat Conversation, ● **USE-CASE:** Steer the Presenter, ● **SCENARIO:** Access Lost After Finish

Description

The moderator sees the event with its interaction settings, also before it is published for the rehearsal, but loses the access when the event finishes, as the anonymization deletes the Moderator roles

Steer the Presenter

Diagram



Properties

○ **ROLE**	● **ROLE:** Moderator
○ **ENTITY**	● **ENTITY:** Event
○ **OPERATIONS**	Update
○ **STATES**	● **STATE:** Running
○ **CONDITION**	The update is confined to the presenter alert and the active agenda point of the event.
○ **USE-CASES**	● **USE-CASE:** Steer the Presenter

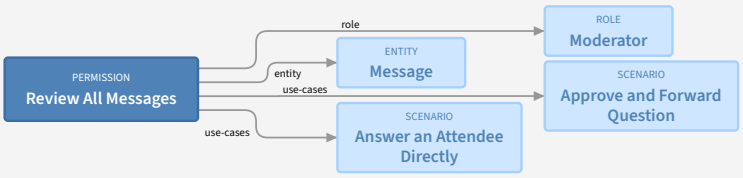
Description

The moderator raises the presenter alert and advances the active agenda point, but changes no other setting of the event

● PERMISSION	Properties & Description

Review All Messages

Diagram



Properties

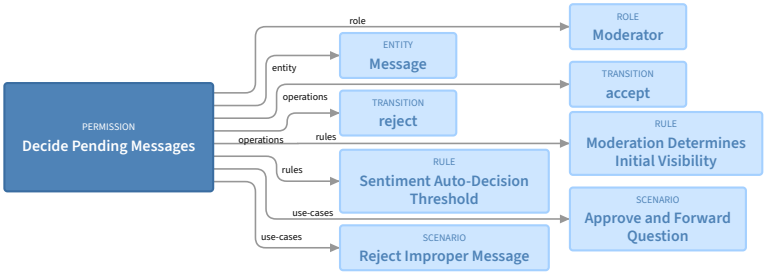
○ ROLE	● ROLE: Moderator
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Read
○ CONDITION	The message belongs to the event the moderator holds the Moderator role of.
○ USE-CASES	● SCENARIO: Approve and Forward Question, ● SCENARIO: Answer an Attendee Directly

Description

The moderator reads every message of the event in every state, with its sender, sentiment score, and like count

Decide Pending Messages

Diagram



Properties

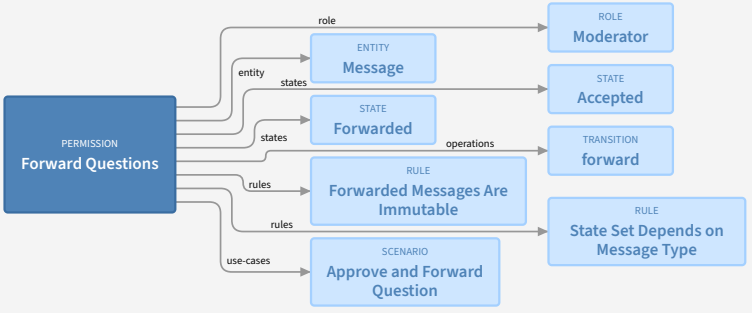
○ ROLE	● ROLE: Moderator
○ ENTITY	● ENTITY: Message
○ OPERATIONS	● TRANSITION: accept, ● TRANSITION: reject
○ RULES	● RULE: Moderation Determines Initial Visibility, ● RULE: Sentiment Auto-Decision Threshold
○ USE-CASES	● SCENARIO: Approve and Forward Question, ● SCENARIO: Reject Improper Message

Description

The moderator accepts or rejects the pending chats and questions, where the server-side sentiment analysis decides in their place if the event so configures

Forward Questions

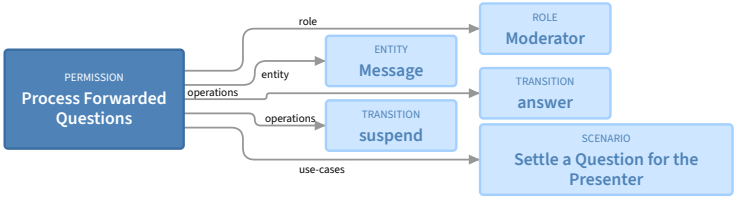
Diagram



● PERMISSION	Properties & Description
	Properties
	○ ROLE: ● ROLE: Moderator
	○ ENTITY: ● ENTITY: Message
	○ OPERATIONS: ● TRANSITION: forward, Update
	○ STATES: ● STATE: Accepted, ● STATE: Forwarded
	○ CONDITION: The message is a question, and the update is confined to the presenter annotation, the ordering, and the tags of the message.
	○ RULES: ● RULE: State Set Depends on Message Type, ● RULE: Forwarded Messages Are Immutable
	○ USE-CASES: ● SCENARIO: Approve and Forward Question
	Description
	The moderator forwards accepted questions to the presenter in a chosen order, attaching hints and tags, while the text of the question stays untouched

Process Forwarded Questions

Diagram



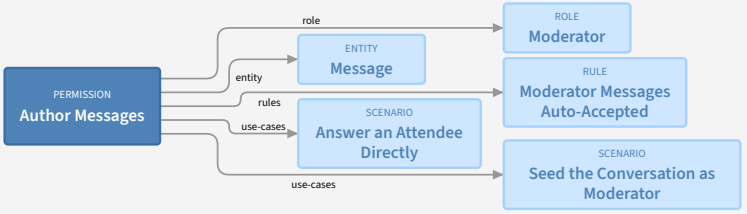
Properties	
○ ROLE	● ROLE: Moderator
○ ENTITY	● ENTITY: Message
○ OPERATIONS	● TRANSITION: answer, ● TRANSITION: suspend
○ USE-CASES	● SCENARIO: Settle a Question for the Presenter

Description

The moderator marks a forwarded question answered or suspended on behalf of the presenter

Author Messages

Diagram



Properties	
○ ROLE	● ROLE: Moderator
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Create
○ CONDITION	The event is running.
○ RULES	● RULE: Moderator Messages Auto-Accepted
○ USE-CASES	● SCENARIO: Answer an Attendee Directly, ● SCENARIO: Seed the Conversation as Moderator

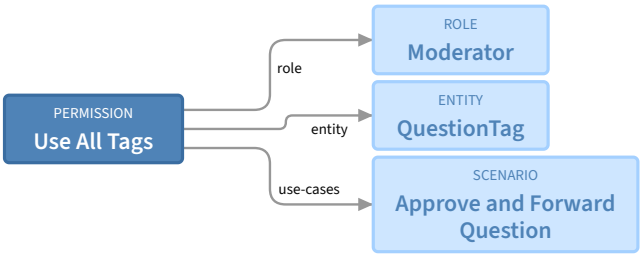
● PERMISSION	Properties & Description

Description

The moderator answers attendees publicly or privately and seeds the conversation, where the messages are accepted at once upon creation under the sender name “Moderator” unless overridden

Use All Tags

Diagram



Properties

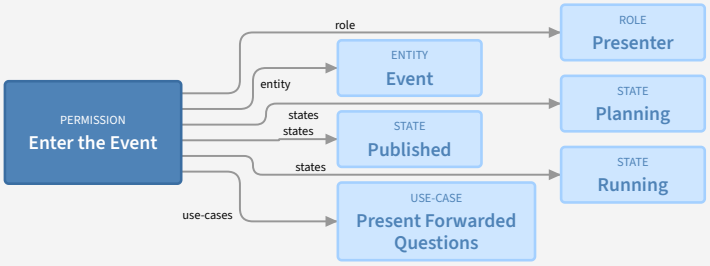
○ ROLE	● ROLE: Moderator
○ ENTITY	● ENTITY: QuestionTag
○ OPERATIONS	Read
○ USE-CASES	● SCENARIO: Approve and Forward Question

Description

The moderator uses every tag of the event on questions, including the ones reserved for moderators

Enter the Event

Diagram



Properties

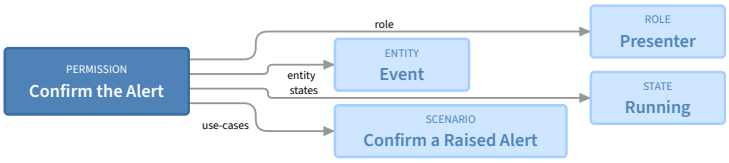
○ ROLE	● ROLE: Presenter
○ ENTITY	● ENTITY: Event
○ OPERATIONS	Read
○ STATES	● STATE: Planning, ● STATE: Published, ● STATE: Running
○ CONDITION	The presenter holds the Presenter role of the event.
○ USE-CASES	● USE-CASE: Present Forwarded Questions

Description

The presenter sees the event with its agenda and presenter alert, also before it is published for the rehearsal

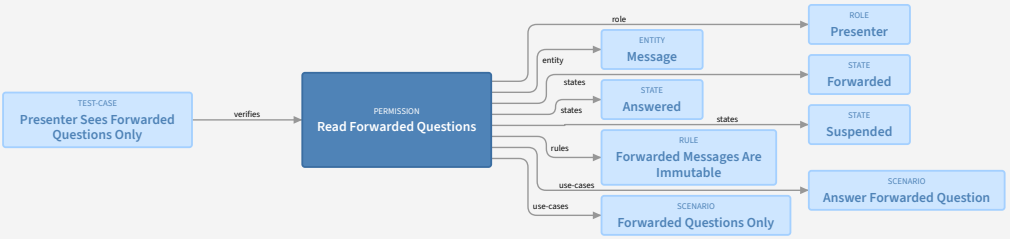
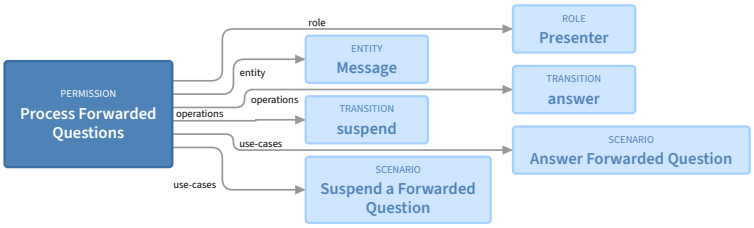
Confirm the Alert

Diagram



Properties

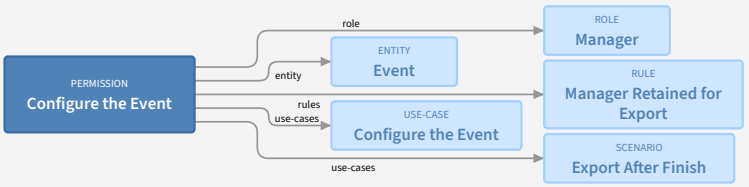
○ ROLE	● ROLE: Presenter

● PERMISSION	Properties & Description	
	○ ENTITY	● ENTITY: Event
	○ OPERATIONS	Update
	○ STATES	● STATE: Running
	○ CONDITION	The update is confined to unraising the presenter alert of the event.
	○ USE-CASES	● SCENARIO: Confirm a Raised Alert
	Description	
	The presenter confirms a raised alert, which takes it off the stage view again	
Read Forwarded Questions	Diagram	
		
	Properties	
	○ ROLE	● ROLE: Presenter
	○ ENTITY	● ENTITY: Message
	○ OPERATIONS	Read
	○ STATES	● STATE: Forwarded, ● STATE: Answered, ● STATE: Suspended
	○ RULES	● RULE: Forwarded Messages Are Immutable
	○ USE-CASES	● SCENARIO: Answer Forwarded Question, ● SCENARIO: Forwarded Questions Only
	Description	
	The presenter sees the forwarded questions in their order with the moderator hints, and never the pending, accepted, or rejected messages	
Process Forwarded Questions	Diagram	
		
	Properties	
	○ ROLE	● ROLE: Presenter
	○ ENTITY	● ENTITY: Message
	○ OPERATIONS	● TRANSITION: answer, ● TRANSITION: suspend
	○ USE-CASES	● SCENARIO: Answer Forwarded Question, ● SCENARIO: Suspend a Forwarded Question
	Description	
	The presenter marks a forwarded question answered once addressed on stage, or suspended when it will not be addressed	

● PERMISSION	Properties & Description

Configure the Event

Diagram



Properties

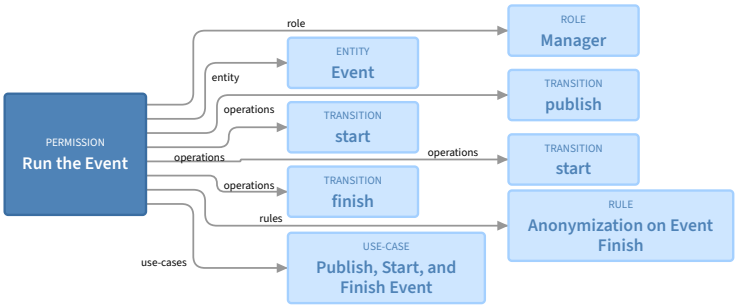
○ ROLE	● ROLE: Manager
○ ENTITY	● ENTITY: Event
○ OPERATIONS	Read, Update
○ CONDITION	The manager holds the Manager role of the event.
○ RULES	● RULE: Manager Retained for Export
○ USE-CASES	● USE-CASE: Configure the Event, ● SCENARIO: Export After Finish

Description

The manager reads and edits every setting of the event in every state, keeping the access after the event finishes for the export

Run the Event

Diagram



Properties

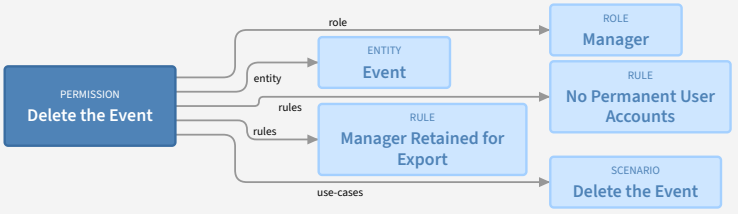
○ ROLE	● ROLE: Manager
○ ENTITY	● ENTITY: Event
○ OPERATIONS	● TRANSITION: publish, ● TRANSITION: start, ● TRANSITION: start, ● TRANSITION: finish
○ RULES	● RULE: Anonymization on Event Finish
○ USE-CASES	● USE-CASE: Publish, Start, and Finish Event

Description

The manager publishes, starts, and finishes the event, where finishing triggers the anonymization

Delete the Event

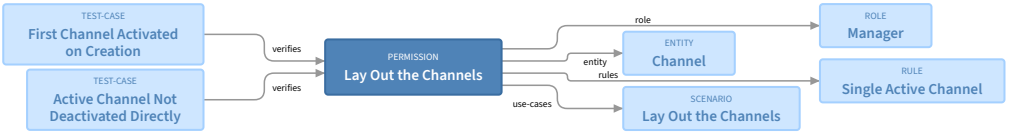
Diagram



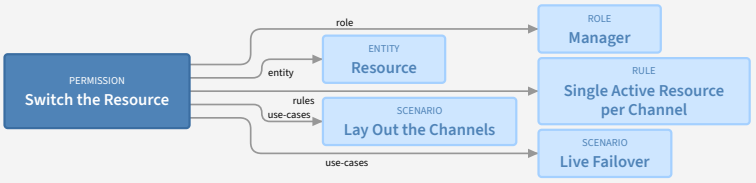
Properties

○ ROLE	● ROLE: Manager
○ ENTITY	● ENTITY: Event

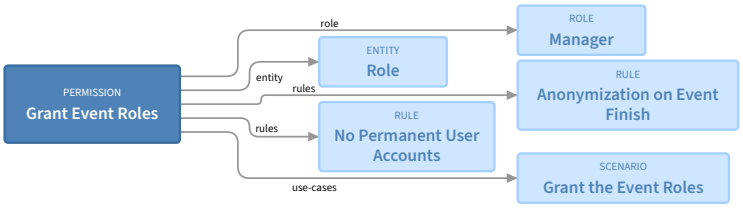
● PERMISSION	Properties & Description	
	○ OPERATIONS	Delete
	○ CONDITION	The manager holds the Manager role of the event.
	○ RULES	● RULE: Manager Retained for Export, ● RULE: No Permanent User Accounts
	○ USE-CASES	● SCENARIO: Delete the Event
	Description	
	The manager deletes the event entirely, which deletes the remaining Manager roles with it	
Lay Out the Channels	Diagram	



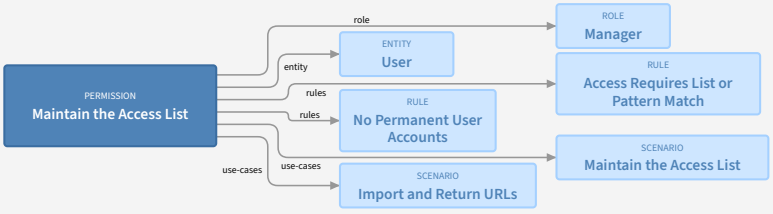
	Properties	
	○ ROLE	● ROLE: Manager
	○ ENTITY	● ENTITY: Channel
	○ OPERATIONS	Create, Read, Update, Delete
	○ CONDITION	The channel belongs to the event the manager holds the Manager role of.
	○ RULES	● RULE: Single Active Channel
	○ USE-CASES	● SCENARIO: Lay Out the Channels
	Description	
	The manager defines the channels of the event and switches the active one	
Switch the Resource	Diagram	



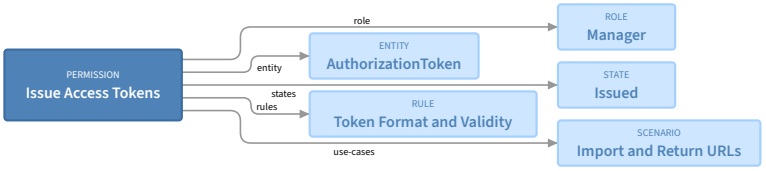
	Properties	
	○ ROLE	● ROLE: Manager
	○ ENTITY	● ENTITY: Resource
	○ OPERATIONS	Read, Update
	○ CONDITION	The update is confined to activating a resource of a channel of the event.
	○ RULES	● RULE: Single Active Resource per Channel
	○ USE-CASES	● SCENARIO: Lay Out the Channels, ● SCENARIO: Live Failover
	Description	
	The manager switches the active resource of a channel to a fallback streaming provider, without changing the provider parameters	
Grant Event Roles	Diagram	



● PERMISSION	Properties & Description
	Properties
	○ ROLE: ● ROLE: Manager
	○ ENTITY: ● ENTITY: Role
	○ OPERATIONS: Create, Read, Delete
	○ CONDITION: The role is held by a user of the event the manager holds the Manager role of.
	○ RULES: ● RULE: No Permanent User Accounts, ● RULE: Anonymization on Event Finish
	○ USE-CASES: ● SCENARIO: Grant the Event Roles
	Description
	The manager grants and revokes the Attendee, Manager, Moderator, and Presenter roles of the users of the event, identified by email
Maintain the Access List	Diagram

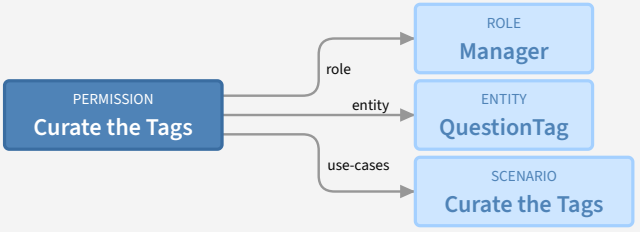


	Properties
	○ ROLE: ● ROLE: Manager
	○ ENTITY: ● ENTITY: User
	○ OPERATIONS: Create, Read, Update, Delete
	○ CONDITION: The user is on the access list of the event the manager holds the Manager role of.
	○ RULES: ● RULE: Access Requires List or Pattern Match, ● RULE: No Permanent User Accounts
	○ USE-CASES: ● SCENARIO: Maintain the Access List, ● SCENARIO: Import and Return URLs
	Description
	The manager invites attendees by email, individually or by importing the sheet of the Event Registration System, and removes them again, where a removed user without a role ceases to exist
Issue Access Tokens	Diagram

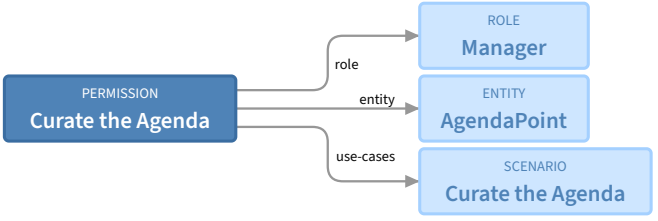


	Properties
	○ ROLE: ● ROLE: Manager
	○ ENTITY: ● ENTITY: AuthorizationToken
	○ OPERATIONS: Create, Read
	○ STATES: ● STATE: Issued
	○ CONDITION: The token belongs to the event the manager holds the Manager role of.
	○ RULES: ● RULE: Token Format and Validity

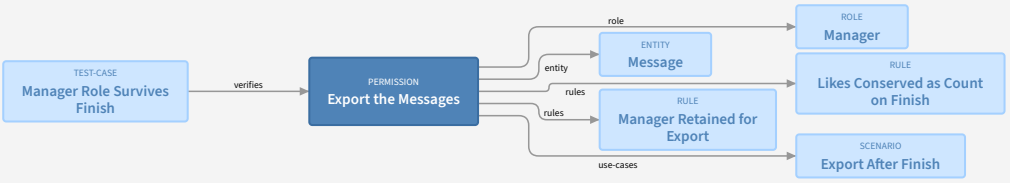
● PERMISSION	Properties & Description	
	○ USE-CASES	● SCENARIO: Import and Return URLs
	Description	
	The manager pre-generates the tokens of the imported attendees and exports their personal access URLs, seeing the tokens in state issued only	
Curate the Tags	Diagram	



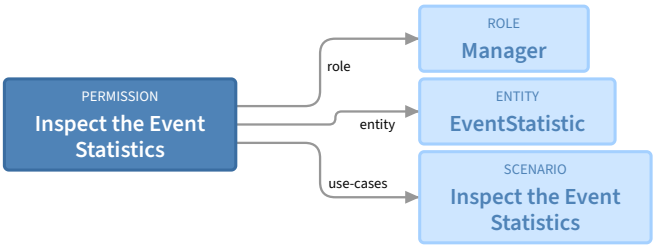
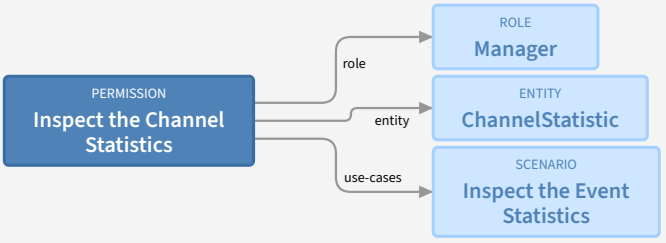
	Properties	
	○ ROLE	● ROLE: Manager
	○ ENTITY	● ENTITY: QuestionTag
	○ OPERATIONS	Create, Read, Update, Delete
	○ CONDITION	The tag belongs to the event the manager holds the Manager role of.
	○ USE-CASES	● SCENARIO: Curate the Tags
	Description	
	The manager defines the tag vocabulary of the event, including the tags reserved for moderators	
Curate the Agenda	Diagram	



	Properties	
	○ ROLE	● ROLE: Manager
	○ ENTITY	● ENTITY: AgendaPoint
	○ OPERATIONS	Create, Read, Update, Delete
	○ CONDITION	The agenda point belongs to the event the manager holds the Manager role of.
	○ USE-CASES	● SCENARIO: Curate the Agenda
	Description	
	The manager defines the ordered agenda of the event and the tags corresponding to its points	
Export the Messages	Diagram	



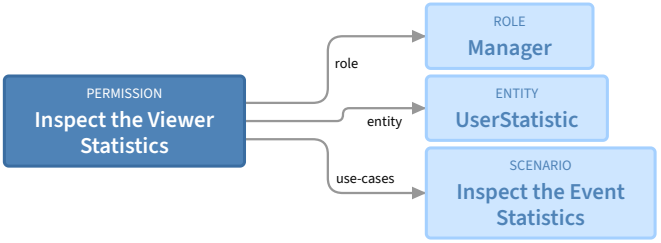
	Properties	
	○ ROLE	● ROLE: Manager
	○ ENTITY	● ENTITY: Message

● PERMISSION	Properties & Description	
	○ OPERATIONS	Read
	○ CONDITION	The message belongs to the event the manager holds the Manager role of.
	○ RULES	● RULE: Likes Conserved as Count on Finish, ● RULE: Manager Retained for Export
	○ USE-CASES	● SCENARIO: Export After Finish
	Description	
	The manager reads every message of the event with its texts, timestamps, state, and like count, and exports them after the event has finished	
Inspect the Event Statistics	Diagram	
		
	Properties	
	○ ROLE	● ROLE: Manager
	○ ENTITY	● ENTITY: EventStatistic
	○ OPERATIONS	Read
	○ CONDITION	The statistic belongs to the event the manager holds the Manager role of.
	○ USE-CASES	● SCENARIO: Inspect the Event Statistics
	Description	
	The manager inspects the curve of logged-in attendees and the login challenge counts over time	
Inspect the Channel Statistics	Diagram	
		
	Properties	
	○ ROLE	● ROLE: Manager
	○ ENTITY	● ENTITY: ChannelStatistic
	○ OPERATIONS	Read
	○ CONDITION	The statistic belongs to a channel of the event the manager holds the Manager role of.
	○ USE-CASES	● SCENARIO: Inspect the Event Statistics
	Description	
	The manager inspects the number of viewers per channel over time	

● PERMISSION	Properties & Description

Inspect the Viewer Statistics

Diagram



Properties

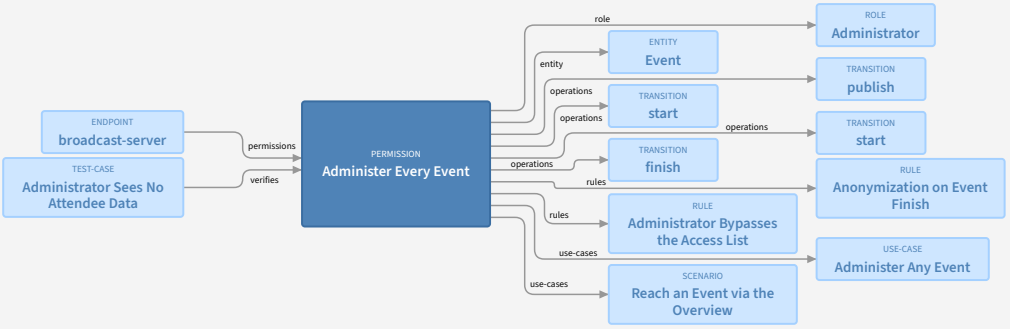
○ ROLE	● ROLE: Manager
○ ENTITY	● ENTITY: UserStatistic
○ OPERATIONS	Read
○ CONDITION	The statistic belongs to a user of the event the manager holds the Manager role of.
○ USE-CASES	● SCENARIO: Inspect the Event Statistics

Description

The manager inspects the audience composition by country, browser, device, and viewport

Administer Every Event

Diagram



Properties

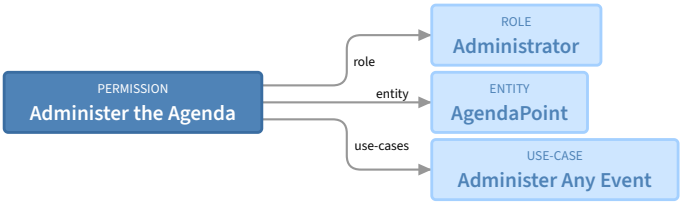
○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: Event
○ OPERATIONS	Create, Read, Update, Delete, ● TRANSITION: publish, ● TRANSITION: start, ● TRANSITION: start, ● TRANSITION: finish
○ RULES	● RULE: Administrator Bypasses the Access List, ● RULE: Anonymization on Event Finish
○ USE-CASES	● USE-CASE: Administer Any Event, ● SCENARIO: Reach an Event via the Overview

Description

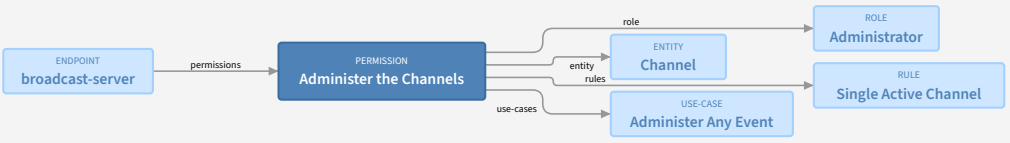
The administrator provisions the events through the configuration and lists, enters, configures, runs, and deletes every event in every state through the user interface, without any access list entry

Administer the Agenda

Diagram



● PERMISSION	Properties & Description
	Properties
	○ ROLE — ● ROLE: Administrator
	○ ENTITY — ● ENTITY: AgendaPoint
	○ OPERATIONS — Create, Read, Update, Delete
	○ USE-CASES — ● USE-CASE: Administer Any Event
	Description
	The administrator curates the agenda of every event and advances its active agenda point
Administer the Channels ⚓	Diagram

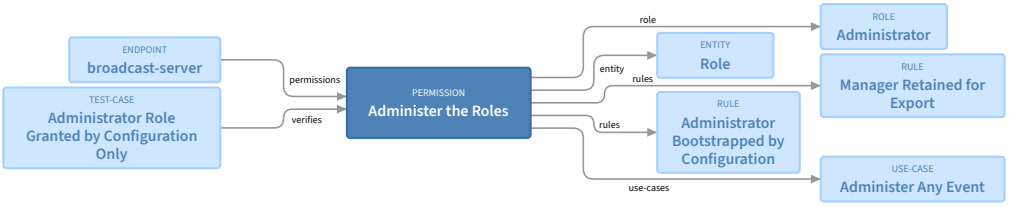


	Properties
	○ ROLE — ● ROLE: Administrator
	○ ENTITY — ● ENTITY: Channel
	○ OPERATIONS — Create, Read, Update, Delete
	○ RULES — ● RULE: Single Active Channel
	○ USE-CASES — ● USE-CASE: Administer Any Event
	Description
	The administrator provisions the initial channels through the configuration and lays out the channels of every event, including the active one
Administer the Resources ⚓	Diagram

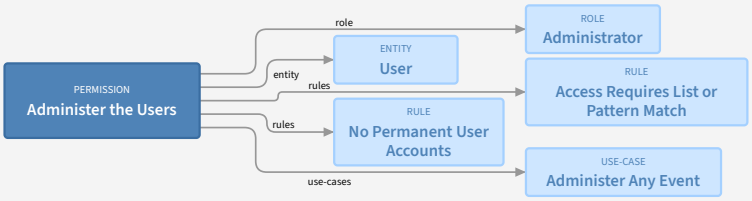
	Properties
	○ ROLE — ● ROLE: Administrator
	○ ENTITY — ● ENTITY: Resource
	○ OPERATIONS — Create, Read, Update, Delete
	○ RULES — ● RULE: Single Active Resource per Channel
	○ USE-CASES — ● USE-CASE: Administer Any Event
	Description
	The administrator binds the resources of every channel to the configured streaming providers and switches the active resource
Administer the Provider Parameters ⚓	Diagram

	Properties
	○ ROLE — ● ROLE: Administrator
	○ ENTITY — ● ENTITY: ResourceProviderParam

● PERMISSION	Properties & Description	
	○ OPERATIONS	Create, Read, Update, Delete
	○ USE-CASES	● USE-CASE: Administer Any Event
	Description	
	The administrator supplies and changes the key-value parameters each streaming provider needs to address its stream	
Administer the Roles	Diagram	



Properties	
○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: Role
○ OPERATIONS	Create, Read, Delete
○ CONDITION	The role is not an Administrator role, which the configuration alone grants and revokes.
○ RULES	● RULE: Administrator Bootstrapped by Configuration, ● RULE: Manager Retained for Export
○ USE-CASES	● USE-CASE: Administer Any Event
Description	
The administrator grants the initial Manager role of a provisioned event through the configuration and grants and revokes every Attendee, Manager, Moderator, and Presenter role of every event	
Administer the Users	Diagram

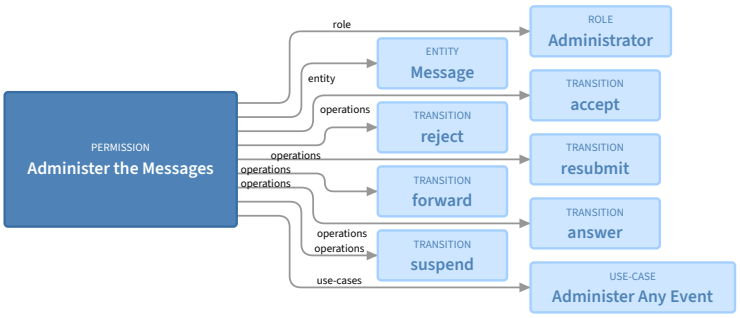


Properties	
○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: User
○ OPERATIONS	Create, Read, Update, Delete
○ RULES	● RULE: Access Requires List or Pattern Match, ● RULE: No Permanent User Accounts
○ USE-CASES	● USE-CASE: Administer Any Event
Description	
The administrator maintains the access list and the users of every event, personal data included	

● PERMISSION	Properties & Description

Administer the Messages

Diagram



Properties

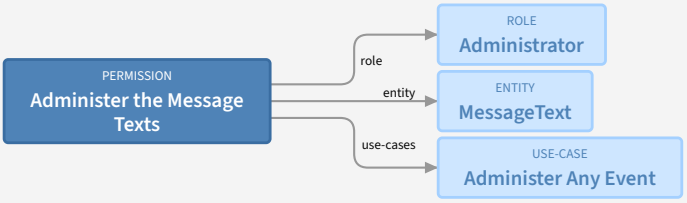
○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: Message
○ OPERATIONS	Create, Read, Update, Delete, ● TRANSITION: accept, ● TRANSITION: reject, ● TRANSITION: forward, ● TRANSITION: answer, ● TRANSITION: suspend, ● TRANSITION: resubmit
○ USE-CASES	● USE-CASE: Administer Any Event

Description

The administrator reads, moderates, forwards, settles, edits, and deletes every message of every event in every state

Administer the Message Texts

Diagram



Properties

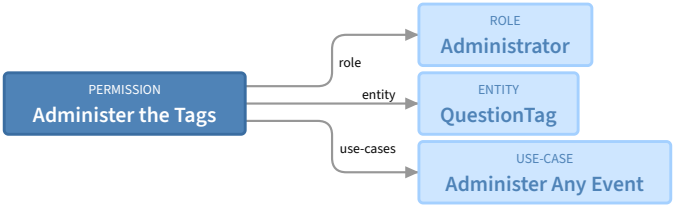
○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: MessageText
○ OPERATIONS	Create, Read, Update, Delete
○ USE-CASES	● USE-CASE: Administer Any Event

Description

The administrator reads and corrects the language-specific texts of every message

Administer the Tags

Diagram



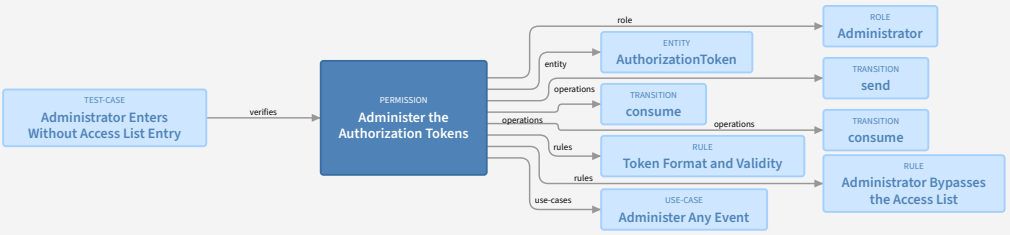
Properties

○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: QuestionTag
○ OPERATIONS	Create, Read, Update, Delete

● PERMISSION	Properties & Description	
	○ USE-CASES	● USE-CASE: Administer Any Event
	Description	
	The administrator curates the tag vocabulary of every event, including the tags reserved for moderators	

Administer the Authorization Tokens

Diagram



Properties

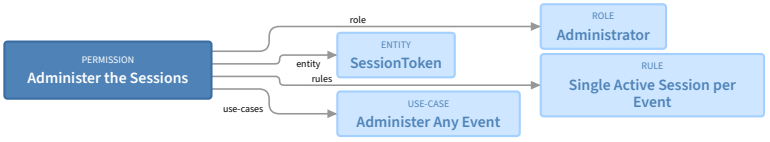
○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: AuthorizationToken
○ OPERATIONS	Create, Read, Update, Delete, ● TRANSITION: send, ● TRANSITION: consume, ● TRANSITION: consume
○ RULES	● RULE: Administrator Bypasses the Access List, ● RULE: Token Format and Validity
○ USE-CASES	● USE-CASE: Administer Any Event

Description

The administrator passes the login challenge of the configured administrator email address without any access list check, and issues, inspects, and revokes the tokens of every event

Administer the Sessions

Diagram



Properties

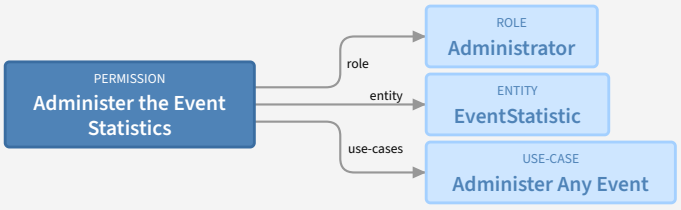
○ ROLE	● ROLE: Administrator
○ ENTITY	● ENTITY: SessionToken
○ OPERATIONS	Create, Read, Update, Delete
○ RULES	● RULE: Single Active Session per Event
○ USE-CASES	● USE-CASE: Administer Any Event

Description

The administrator inspects and terminates the sessions of every event

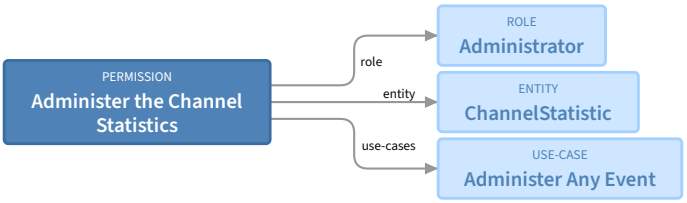
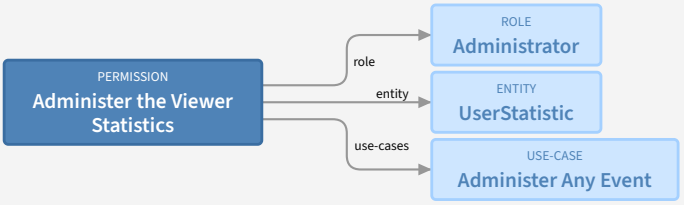
Administer the Event Statistics

Diagram



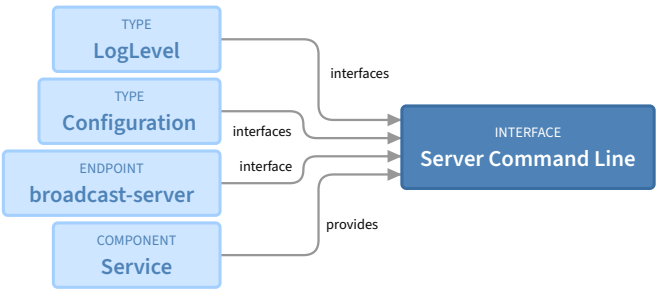
Properties

○ ROLE	● ROLE: Administrator

● PERMISSION	Properties & Description	
	○ ENTITY	● ENTITY: EventStatistic
	○ OPERATIONS	Create, Read, Update, Delete
	○ USE-CASES	● USE-CASE: Administer Any Event
	Description	
	The administrator inspects and prunes the event statistics of every event	
Administer the Channel Statistics	Diagram	
		
	Properties	
	○ ROLE	● ROLE: Administrator
	○ ENTITY	● ENTITY: ChannelStatistic
	○ OPERATIONS	Create, Read, Update, Delete
	○ USE-CASES	● USE-CASE: Administer Any Event
	Description	
	The administrator inspects and prunes the channel statistics of every event	
Administer the Viewer Statistics	Diagram	
		
	Properties	
	○ ROLE	● ROLE: Administrator
	○ ENTITY	● ENTITY: UserStatistic
	○ OPERATIONS	Create, Read, Update, Delete
	○ USE-CASES	● USE-CASE: Administer Any Event
	Description	
	The administrator inspects and prunes the viewer statistics of every event	

14 ● APIS: Interface Model (IM)

● INTERFACE	Properties & Description
Server Command Line	Diagram

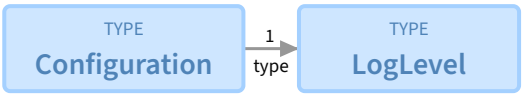


○ KIND	○ LOCATION	○ CONSUMERS
CLI	`broadcast-server`	● PERSONA: Sven

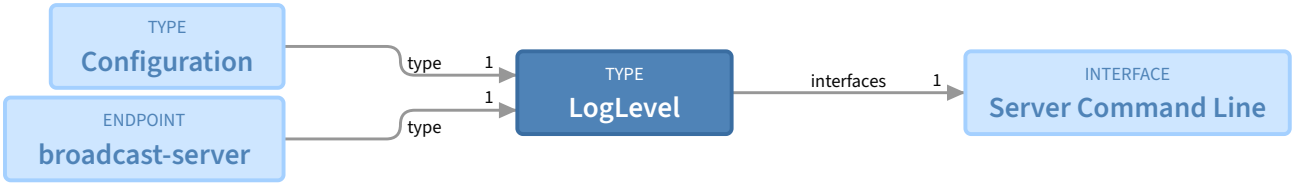
Description

The command-line program starting the server process of the service layer, which resolves its runtime parameters from the layered configuration of built-in defaults, an optional YAML configuration file, BROADCAST_* environment variables, and the command-line options (in increasing precedence) — **BECAUSE** the administrator and the deployment automation have to start the very same server with different parameters per environment without changing any file.

15 ● APIS: Interface Datatypes (ID)



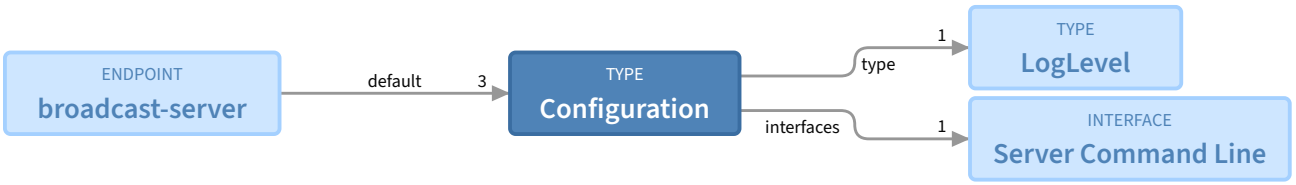
15.01 ● TYPE: LogLevel



○ INTERFACES	● INTERFACE: Server Command Line
○ BASE	`enum(error,warning,info,debug)`

The verbosity level of the server log, from errors only up to debugging details — **BECAUSE** the amount of log output has to be tuned per environment.

15.02 ● TYPE: Configuration ⚓



○ INTERFACES	● INTERFACE: Server Command Line ⚭

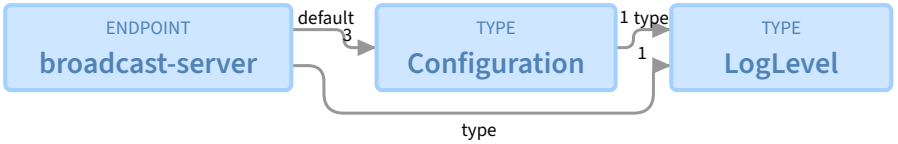
The structure of the YAML configuration file, whose keys are also the surface of the recognized `BROADCAST_*` environment variables (`BROADCAST_ADMIN_USERNAME` maps onto `adminUsername`) and of the corresponding command-line options, where an unrecognized key is a fatal startup error — **BECAUSE** one schema has to define all three configuration layers consistently.

● FIELD	○ TYPE	○ DEFAULT	○ CONSTRAINT	Description
adminUsername ⚓	`string`	∅	`non-empty`	The user name of the administrator account — **BECAUSE** the administrator has to be able to log in without any event-specific token
adminPassword ⚓	`string`	∅	`non-empty`	The password of the administrator account — **BECAUSE** the administrator account has to be protected
mqttUrl ⚓	`string`	● CONSTANT: DEFAULT_MQTT_URL ⚭	∅	The URL of the MQTT broker the server connects to, including the credentials and the topic prefix — **BECAUSE** the server exchanges all live data through the broker
dbUrl ⚓	`string`	● CONSTANT: DEFAULT_DB_URL ⚭	∅	The URL of the PostgreSQL database the server persists into — **BECAUSE** the authoritative state has to be stored in the database tier
directory ⚓	`string`	● CONSTANT: DEFAULT_DIRECTORY ⚭	∅	The directory of the static client content the server delivers — **BECAUSE** the client bundle is distributed alongside the server
logLevel ⚓	● TYPE: LogLevel ⚭	`info`	∅	The verbosity level of the server log — **BECAUSE** the amount of log output has to be tuned per environment
workers ⚓	`integer`	1	`1..n`	The number of worker processes the primary process pre-forks — **BECAUSE** the load-balanced traffic of the broker has to be spread over several processes

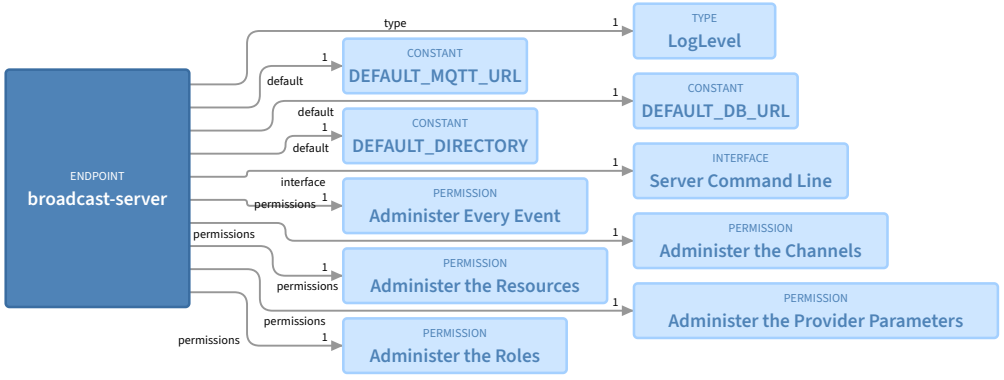
● CONSTANT	○ TYPE	○ VALUE	Description
ENV_PREFIX ⚓	`string`	BROADCAST_	The prefix of all environment variables overriding configuration parameters, where a double underscore separates the nesting path segments and a single underscore the words of one camel-cased segment — **BECAUSE** the server has to ignore the unrelated variables of the ambient environment

● CONSTANT	○ TYPE	○ VALUE	Description
DEFAULT_MQTT_URL ⚓	`string`	`wss://example:example@127.0.0.1:1883/pr/api/server/?topic=broadcast`	The built-in default of the MQTT broker URL — **BECAUSE** the server has to start against a local broker without any configuration
DEFAULT_DB_URL ⚓	`string`	`postgres://127.0.0.1:5432/broadcast`	The built-in default of the PostgreSQL database URL — **BECAUSE** the server has to start against a local database without any configuration
DEFAULT_DIRECTORY ⚓	`string`	`client/dst`	The built-in default of the static content directory, resolved relative to the installation of the server — **BECAUSE** the client bundle is built next to the server by default
RESPAWN_DELAY ⚓	`duration`	`1000 ms`	The backoff delay before an unexpectedly dead worker process is respawned — **BECAUSE** a tight crash-loop would hammer the broker and the database

16 ● APIS: Interface Endpoints (IE) ⚓



16.01 ● ENDPOINT: broadcast-server ⚓



○ INTERFACE	● **INTERFACE:** Server Command Line ⚭
○ PERMISSIONS	● **PERMISSION:** Administer Every Event ⚭, ● **PERMISSION:** Administer the Channels ⚭, ● **PERMISSION:** Administer the Resources ⚭, ● **PERMISSION:** Administer the Provider Parameters ⚭, ● **PERMISSION:** Administer the Roles ⚭
○ OPERATION	command
○ LOCATOR	`broadcast-server [options]`

The sole command of the program, which loads the layered configuration, brings the database schema up to date, pre-forks the configured number of worker processes, supervises them (respawning a dead one after the ● **RESPAWN_DELAY**), and runs until it receives a `SIGINT` or `SIGTERM` signal, which it forwards to all workers before terminating gracefully — **BECAUSE** the server is a long-running daemon whose process model has to stay uniform even for a single worker.

● OPTION	○ TYPE	○ ALIAS	○ DEFAULT	○ CONSTRAINT	Description
—version ⚓	`boolean`	`-V`	`false`	∅	Prints the name and version of the program and exits — **BECAUSE** the deployed version has to be identifiable
—help ⚓	`boolean`	`-h`	`false`	∅	Prints the usage of the program and exits — **BECAUSE** the options have to be discoverable on the command line
—log-level ⚓	● **TYPE: LogLevel** ⚭	`-l`	`info`	∅	Overrides the `logLevel` configuration parameter — **BECAUSE** the verbosity has to be raised ad-hoc for troubleshooting
—config ⚓	`string`	`-c`	∅	`readable YAML file`	The configuration file of the structure ● **Configuration**, merged over the built-in defaults — **BECAUSE** a deployment has to keep its parameters in one versioned file
—env ⚓	`string`	`-e`	∅	`readable dotenv file`	The environment file loaded in addition to, and overriding, the `.env` file of the current working directory — **BECAUSE** secrets have to be injected without appearing in the configuration file or the command line
—admin-username ⚓	`string`	`-u`	∅	∅	Overrides the `adminUsername` configuration parameter — **BECAUSE** the administrator account has to be settable per environment
—admin-password ⚓	`string`	`-p`	∅	∅	Overrides the `adminPassword` configuration parameter — **BECAUSE** the administrator account has to be settable per environment
—mqtt-url ⚓	`string`	`-m`	● **CONSTANT: DEFAULT_MQTT_URL** ⚭	∅	Overrides the `mqttUrl` configuration parameter — **BECAUSE** the broker differs per environment
—db-url ⚓	`string`	`-D`	● **CONSTANT: DEFAULT_DB_URL** ⚭	∅	Overrides the `dbUrl` configuration parameter — **BECAUSE** the database differs per environment
—directory ⚓	`string`	`-d`	● **CONSTANT: DEFAULT_DIRECTORY** ⚭	`existing directory`	Overrides the `directory` configuration parameter — **BECAUSE** the client bundle may be deployed apart from the server

● OPTION	○ TYPE	○ ALIAS	○ DEFAULT	○ CONSTRAINT	Description
—workers ⚓	`integer`	`-w`	`1`	`1..n`	Overrides the `workers` configuration parameter, rejecting a non-positive or non-integer value as a usage error — **BECAUSE** the process count has to match the CPU resources of the environment

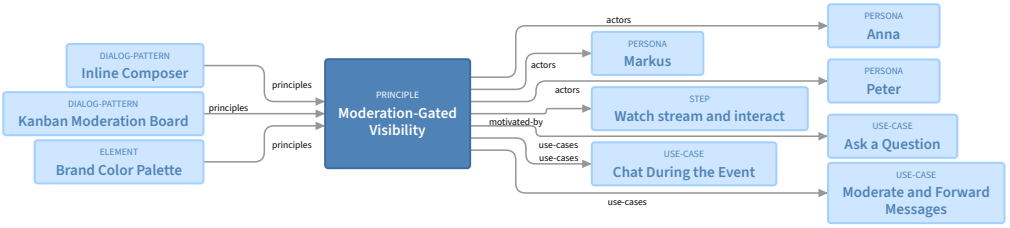
● RESULT	○ TYPE	○ ARITY	○ CONSTRAINT	Description
log ⚓	`text`	`*`	∅	The log lines written to the standard output, each carrying the timestamp, the level, and the originating primary or worker process, with the credentials of all URL-valued parameters redacted — **BECAUSE** the operator has to trace the server without ever seeing a secret
exit-code ⚓	`integer`	∅	`0 or 1`	The process exit code, `0` after a graceful termination, after `--help`, and after `--version`, and `1` after a usage error, an invalid configuration, a failed database migration, or a failed shutdown — **BECAUSE** the process supervisor has to tell a clean stop from a failure

17 ● UXUI: Interaction Concept (IC) ⚓

- **Moderation-Gated Visibility** — CATEGORY: Expectation-Conformity
- **Privacy-Visible Identity** — CATEGORY: Self-Descriptiveness
- **Frictionless Joining** — CATEGORY: Task-Suitability
- **Live Reactivity** — CATEGORY: Self-Descriptiveness
- **Validate Before Going Live** — CATEGORY: Error-Robustness
- **Never Leave the Stream** — CATEGORY: Controllability
- **Correct over Confirm** — CATEGORY: Error-Robustness
- **Familiar Conventions** — CATEGORY: Learnability
- **One-Tap Contribution** — CATEGORY: Engagement

● PRINCIPLE	Properties & Description
Moderation-Gated Visibility ⚓	

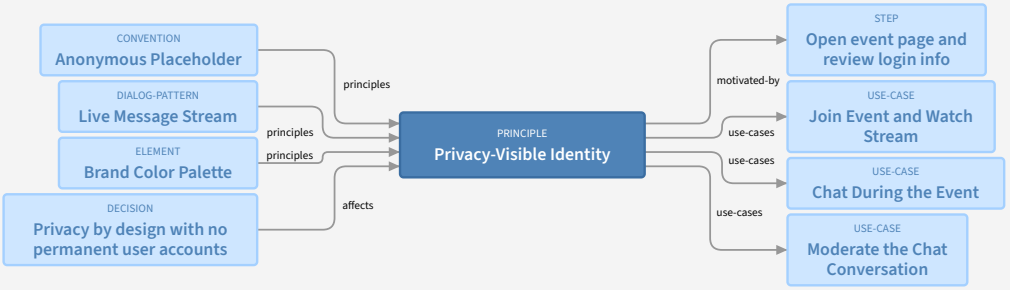
Diagram



Properties

○ CATEGORY	Expectation-Conformity

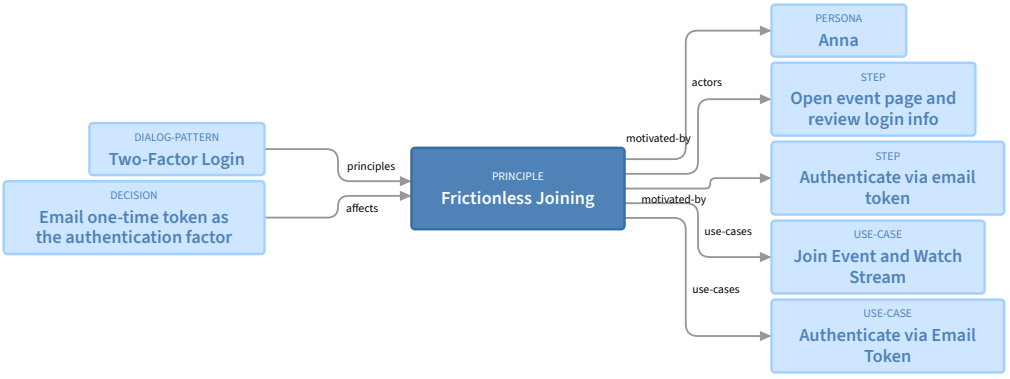
● PRINCIPLE	Properties & Description	
	○ ACTORS	● PERSONA: Anna, ● PERSONA: Markus, ● PERSONA: Peter
	○ MOTIVATED-BY	● STEP: Watch stream and interact
	○ USE-CASES	● USE-CASE: Ask a Question, ● SCENARIO: Pending Message Stays Private, ● SCENARIO: Edit an Own Message, ● USE-CASE: Moderate and Forward Messages
	○ TRADE-OFF	The liveliness of an unmoderated stream in which every input is public the moment it is sent
	Description	
	Attendee input is acknowledged immediately as submitted but becomes publicly visible only after moderation when enabled, with its state clearly indicated to the author — **BECAUSE** authors must trust their input arrived while organizers retain control	
Privacy-Visible Identity	Diagram	



Properties	
○ CATEGORY	Self-Descriptiveness
○ MOTIVATED-BY	● STEP: Open event page and review login info
○ USE-CASES	● SCENARIO: Post a Chat Message, ● SCENARIO: Switch the Language, ● SCENARIO: Seed the Conversation as Moderator
○ TRADE-OFF	Full anonymity of all participants, which would spare the anonymity setting and the hover reveal
Description	
Names are shown according to the event's anonymity setting while the email remains discoverable on hover, and AI-translated text is always marked distinct from the human original — **BECAUSE** participants must always know who said what and in which language	

Frictionless Joining

Diagram



Properties	
○ CATEGORY	Task-Suitability
○ ACTORS	● PERSONA: Anna

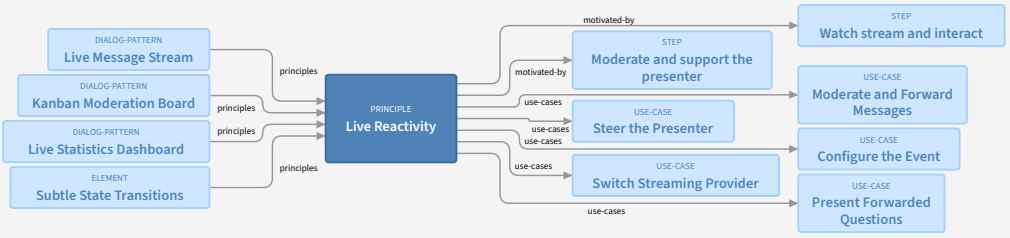
● PRINCIPLE	Properties & Description
	○ MOTIVATED-BY: ● STEP: Open event page and review login info, ● STEP: Authenticate via email token
	○ USE-CASES: ● USE-CASE: Join Event and Watch Stream, ● USE-CASE: Authenticate via Email Token
	○ TRADE-OFF: An explicit registration with a permanent account, which would give organizers richer attendee profiles

Description

The attendee is carried into the event with the least possible effort, with email and token pre-filled from the URL where available and automatic access when a valid token is embedded — **BECAUSE** large audiences must enter quickly without a support burden

Live Reactivity

Diagram



Properties

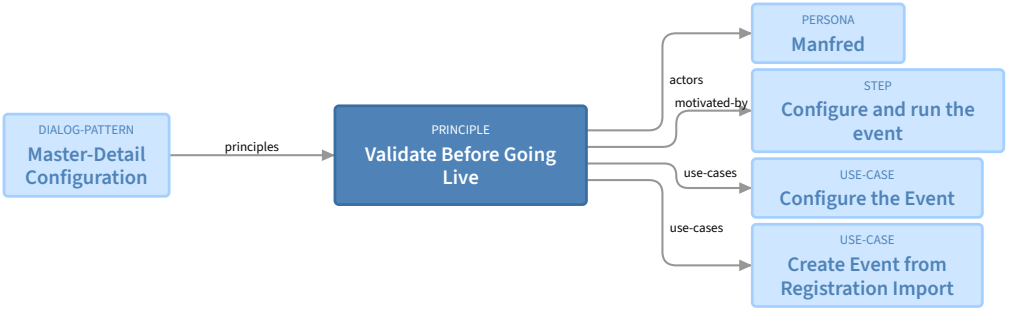
○ CATEGORY	Self-Descriptiveness
○ MOTIVATED-BY	● STEP: Watch stream and interact, ● STEP: Moderate and support the presenter
○ USE-CASES	● SCENARIO: Lay Out the Channels, ● USE-CASE: Switch Streaming Provider, ● USE-CASE: Steer the Presenter, ● SCENARIO: Approve and Forward Question, ● USE-CASE: Present Forwarded Questions
○ TRADE-OFF	Lower server load and simpler clients through polling or a manual reload

Description

Every configuration change, provider switch, and moderation outcome is pushed to connected clients over WebSockets so the UI updates without a reload — **BECAUSE** a live event demands that all participants see a single, current state at all times

Validate Before Going Live

Diagram



Properties

○ CATEGORY	Error-Robustness
○ ACTORS	● PERSONA: Manfred
○ MOTIVATED-BY	● STEP: Configure and run the event
○ USE-CASES	● SCENARIO: Lay Out the Channels, ● USE-CASE: Create Event from Registration Import
○ TRADE-OFF	The freedom to save any partial or inconsistent configuration unchecked

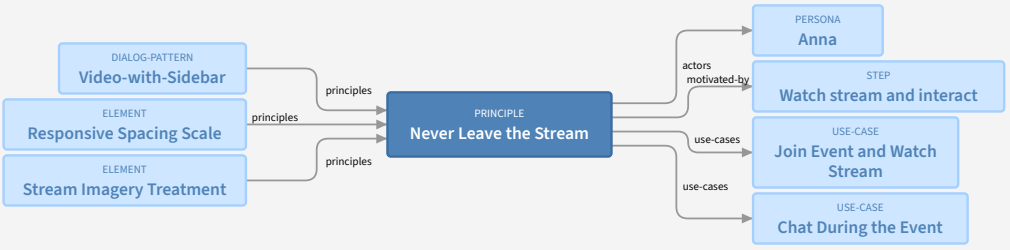
● PRINCIPLE	Properties & Description

Description

Imported attendee lists and streaming settings are validated at entry and problems are reported in place before the event starts — **BECAUSE** a configuration mistake surfacing during the broadcast hits the whole audience at once

Never Leave the Stream

Diagram



Properties

○ **CATEGORY**	Controllability
○ **ACTORS**	● **PERSONA:** Anna
○ **MOTIVATED-BY**	● **STEP:** Watch stream and interact
○ **USE-CASES**	● **SCENARIO:** Switch the Language, ● **SCENARIO:** Toggle the Theme, ● **SCENARIO:** Show Own Contributions Only
○ **TRADE-OFF**	A dedicated settings page with room for more options than the header can hold

Description

Every adjustment available to the attendee, like language, theme, or the interaction tab, is made in place from the event screen, applies immediately, and can be reverted at any time — **BECAUSE** navigating away from a live stream means missing part of the event

Correct over Confirm

Diagram

Properties

○ **CATEGORY**	Error-Robustness
○ **ACTORS**	● **PERSONA:** Anna
○ **MOTIVATED-BY**	● **STEP:** Watch stream and interact
○ **USE-CASES**	● **SCENARIO:** Post a Chat Message, ● **SCENARIO:** Like a Message, ● **SCENARIO:** Edit an Own Message, ● **SCENARIO:** Delete an Own Message
○ **TRADE-OFF**	A confirmation step before every submission, which would reduce mistaken posts at the cost of interrupting the flow

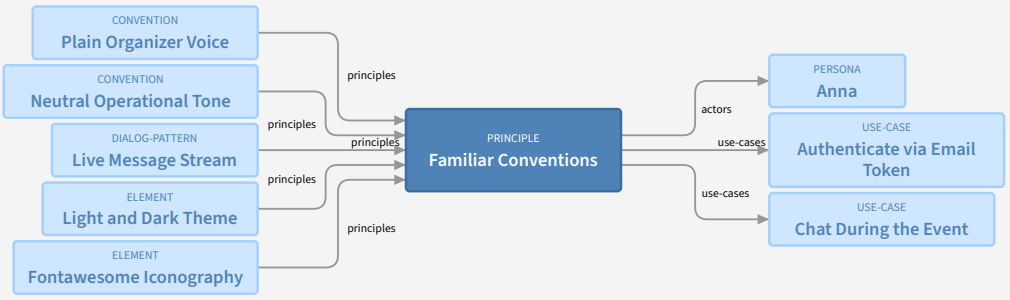
Description

Input is sent without a confirmation step and corrected afterwards instead, as the attendee can edit and delete their own messages while a deleted message leaves a placeholder — **BECAUSE** a fast conversation tolerates a repaired mistake better than a delay before every message

● PRINCIPLE	Properties & Description

Familiar Conventions

Diagram



Properties

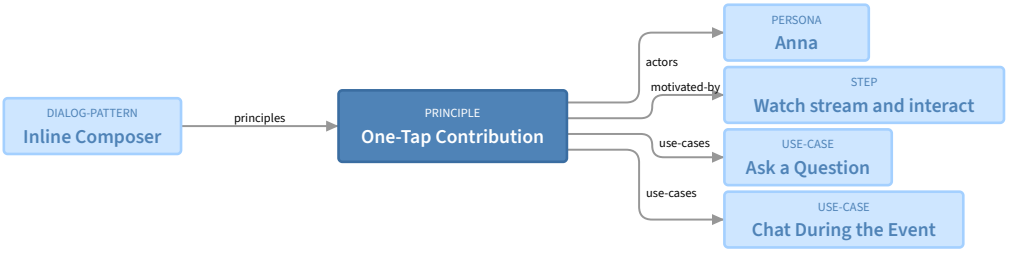
○ **CATEGORY**	Learnability
○ **ACTORS**	● **PERSONA:** Anna
○ **USE-CASES**	● **USE-CASE:** Chat During the Event, ● **SCENARIO:** Login with Emailed Token
○ **TRADE-OFF**	Novel interaction forms tailored to the event format, which would be more expressive but need explaining

Description

Chat, questions, and likes look and behave like the consumer messengers the attendee already knows, so that the login message is the only instruction an attendee ever reads — **BECAUSE** a one-time audience has neither the time nor the motivation to learn a new tool

One-Tap Contribution

Diagram



Properties

○ **CATEGORY**	Engagement
○ **ACTORS**	● **PERSONA:** Anna
○ **MOTIVATED-BY**	● **STEP:** Watch stream and interact
○ **USE-CASES**	● **SCENARIO:** Submit Moderated Question, ● **SCENARIO:** Post a Chat Message, ● **SCENARIO:** Like a Message, ● **SCENARIO:** Answer a Multiple-Choice Poll
○ **TRADE-OFF**	Richer contributions like threads, attachments, or formatting, which would raise the bar for taking part

Description

A like is a single tap and a message is a single field plus send, with the emoji picker as the only enrichment — **BECAUSE** the more attendees contribute during the event, the livelier it feels for everyone watching

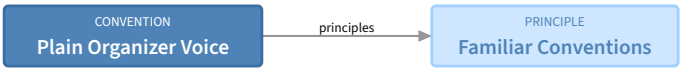
18 ● UXUI: Language Conventions (LC)

Plain Organizer Voice CATEGORY: Voice	Neutral Operational Tone CATEGORY: Tone	Second-Person Address CATEGORY: Grammar	Product Name Spelling CATEGORY: Capitalization
Anonymous Placeholder CATEGORY: Vocabulary	Token Display Format CATEGORY: Formatting		

● CONVENTION	Properties & Description

Plain Organizer Voice

Diagram



Properties

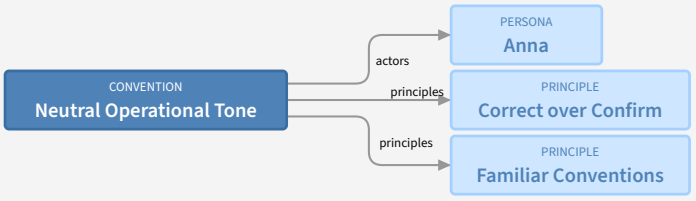
○ CATEGORY	Voice
○ PRINCIPLES	● PRINCIPLE: Familiar Conventions
○ EXAMPLE	“The event starts at 10:00.”, not “Get ready, the show is about to begin!”

Description

The solution speaks with the plain, factual voice of the event organizer, never with a playful or marketing persona of its own — **BECAUSE** attendees join for the event, not for the tool, and the tool must not compete with the presenter for attention

Neutral Operational Tone

Diagram



Properties

○ CATEGORY	Tone
○ ACTORS	● PERSONA: Anna
○ PRINCIPLES	● PRINCIPLE: Correct over Confirm, ● PRINCIPLE: Familiar Conventions
○ EXAMPLE	“You are not authorized for this event.”, not “Access forbidden!”

Description

User-facing system messages use a calm, neutral, and concise tone that informs without alarming, especially around access denial and connection changes — **BECAUSE** a one-time audience has no way to recover from a message that sounds like an accusation

Second-Person Address

Properties

○ CATEGORY	Grammar
○ EXAMPLE	“Your message was sent.”, not “The user’s message has been sent.”

Description

The solution addresses the user as “you” and never refers to itself as “I” or “we” — **BECAUSE** direct address is the shortest wording and avoids the false impression of a person behind the interface

Product Name Spelling

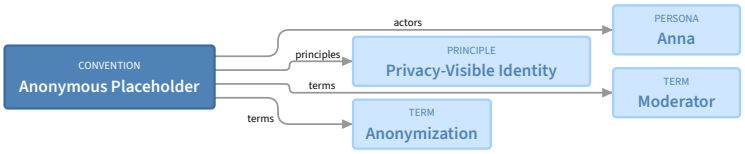
Properties

○ CATEGORY	Capitalization
○ EXAMPLE	“Broadcast”, not “broadcast” or “BROADCAST”

● CONVENTION	Properties & Description
	Description
	The product is always written “Broadcast” — **BECAUSE** a single spelling keeps the product recognizable and distinct from the generic verb

Anonymous Placeholder ⚓

Diagram



Properties

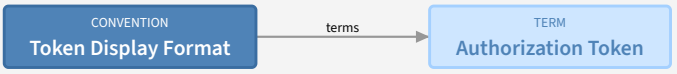
○ **CATEGORY**	Vocabulary
○ **ACTORS**	● **PERSONA:** Anna
○ **PRINCIPLES**	● **PRINCIPLE:** Privacy-Visible Identity
○ **TERMS**	● **TERM:** Anonymization, ● **TERM:** Moderator
○ **EXAMPLE**	“Anonymous”, not “anonymous user” or “N/A”

Description

When identity is hidden, the displayed sender name is exactly “Anonymous”, and moderator-authored messages default to the sender name “Moderator” — **BECAUSE** a fixed placeholder reads as a deliberate choice while an ad-hoc wording reads as missing data

Token Display Format ⚓

Diagram



Properties

○ **CATEGORY**	Formatting
○ **TERMS**	● **TERM:** Authorization Token
○ **EXAMPLE**	“123-456”, not “123456”

Description

Authorization tokens are always presented in the grouped six-digit form “NNN-NNN” with a hyphen separator, in emails and on screen alike — **BECAUSE** the grouping makes a token easy to read aloud and to type from a second device

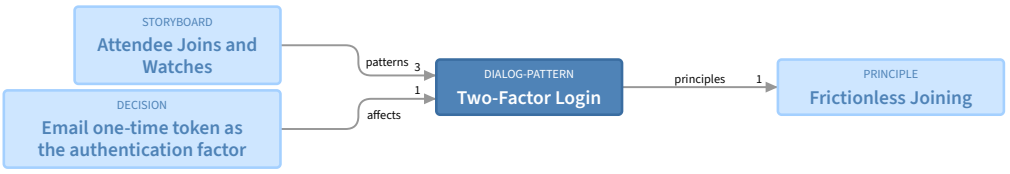
19 ● UXUI: Dialog Patterns (DP) ⚓

Two-Factor Login CATEGORY: Form	Consent Gate CATEGORY: Action	Video-with-Sidebar CATEGORY: Layout	Live Message Stream CATEGORY: List
Inline Composer CATEGORY: Form	Kanban Moderation Board CATEGORY: List	Master-Detail Configuration CATEGORY: Layout	Live Statistics Dashboard CATEGORY: Data

● DIALOG-PATTERN	Properties & Description

Two-Factor Login

Diagram



○ CATEGORY	○ ACTORS	○ CONTEXT
Form	● **PERSONA:** Anna	An attendee accessing an event that requires email-verified access.
○ PROBLEM	**○ PRINCIPLES**	**○ USES**
The system must confirm the attendee controls an authorized email without permanent accounts.	● **PRINCIPLE:** Frictionless Joining	∅
○ TRADE-OFF	Description	
A second round-trip through the attendee's mailbox before every access without an embedded token.	The dialog collects the email, sends a one-time token to it, and then collects that token in a second step, optionally pre-filling fields from the URL — **BECAUSE** a challenge-response over email proves address control with no stored credential.	

Consent Gate

Diagram

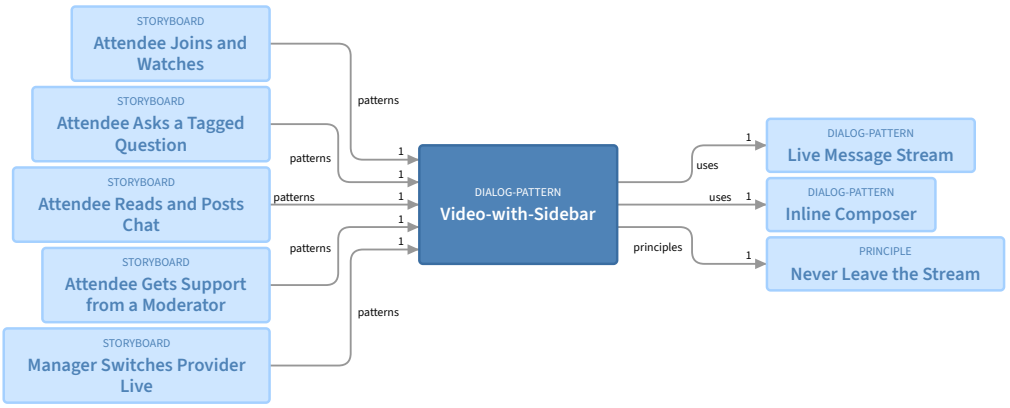
○ CATEGORY	○ ACTORS	○ CONTEXT
Action	● **PERSONA:** Anna	Entry to an event or interaction channel that requires acknowledged terms.
○ PROBLEM	**○ PRINCIPLES**	**○ USES**
Participation must be blocked until the attendee explicitly accepts the stated conditions.	∅	∅
○ TRADE-OFF	Description	
An interruption of the entry flow the attendee cannot skip, even on repeated visits.	A modal presents the configured login or interaction information with required-to-accept controls and only releases the attendee onward once accepted — **BECAUSE** explicit acknowledgement is both a legal and psychological safeguard.	

● **DIALOG-PATTERN**

Video-with-Sidebar

Properties & Description

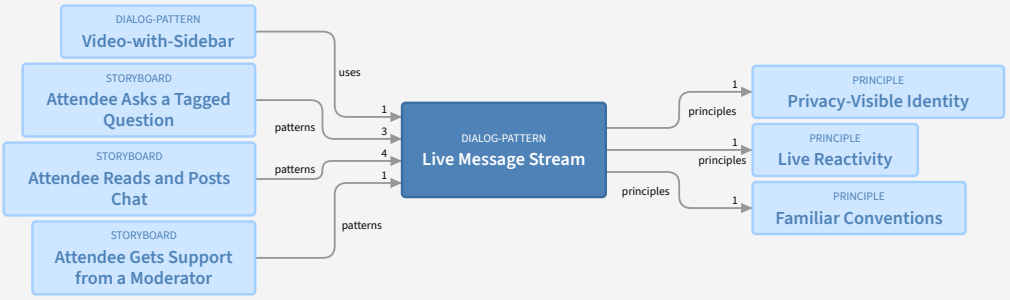
Diagram



○ CATEGORY	○ ACTORS	○ CONTEXT
Layout	● **PERSONA:** Anna	The main attendee screen during a live event with interaction enabled.
○ PROBLEM	**○ PRINCIPLES**	**○ USES**
The attendee must watch the stream and interact without either crowding the other.	● **PRINCIPLE:** Never Leave the Stream	● **DIALOG-PATTERN:** Live Message Stream, ● **DIALOG-PATTERN:** Inline Composer
○ TRADE-OFF	Description	
A smaller video area on wide screens and a sidebar hidden behind a toggle on small screens.	The video occupies the primary area while a tabbed sidebar hosts the Q&A, Hudspad, Chat, and Support tabs, collapsing responsively on small screens — **BECAUSE** co-locating stream and interaction keeps attention on the event while interaction stays reachable.	

Live Message Stream

Diagram



○ CATEGORY	○ ACTORS	○ CONTEXT
List	● **PERSONA:** Anna	The chat and Q&A tabs where attendees read and post a continuous flow of messages.
○ PROBLEM	**○ PRINCIPLES**	**○ USES**
Messages of mixed provenance and state arrive continuously and the attendee must follow them without losing their place.	● **PRINCIPLE:** Live Reactivity, ● **PRINCIPLE:** Privacy-Visible Identity, ● **PRINCIPLE:** Familiar Conventions	∅

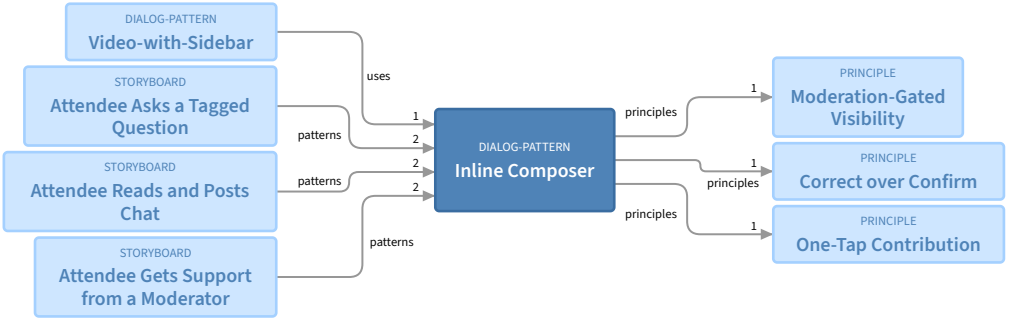
● **DIALOG-PATTERN**

Properties & Description

○ TRADE-OFF	Description
A dense visual vocabulary of alignments, tints, and indicators the attendee has to learn.	Messages render as a scrolling list of bubbles distinguished by provenance and state, with own messages right-aligned, others left-aligned, moderator messages tinted, quoted originals linking back to their source, deleted messages shown as a placeholder, per-message like counts and reply affordances, and floating "new messages" jump indicators above and below — **BECAUSE** clear per-message styling and unread cues let attendees follow a fast, mixed stream without confusion.

Inline Composer ⚓

Diagram



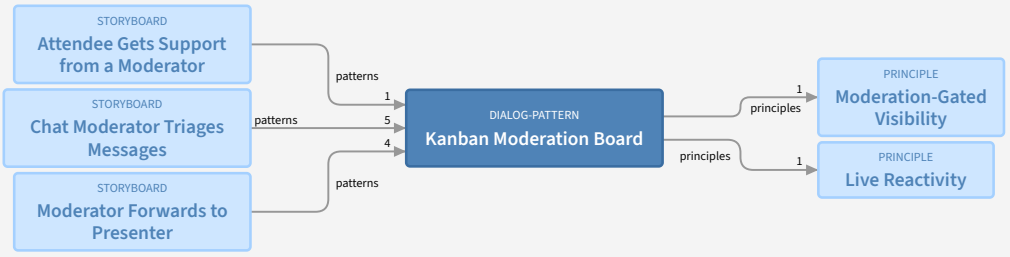
○ CATEGORY	○ ACTORS	○ CONTEXT
Form	● **PERSONA:** Anna ⚭	The bottom of the chat and Q&A tabs where the attendee writes input.

○ PROBLEM	○ PRINCIPLES	○ USES
The attendee must compose, send, and see the moderation outcome of their own input within the same stream.	● **PRINCIPLE:** Moderation-Gated Visibility ⚭, ● **PRINCIPLE:** Correct over Confirm ⚭, ● **PRINCIPLE:** One-Tap Contribution ⚭	∅

○ TRADE-OFF	Description
A permanently occupied strip of the sidebar, even while the attendee is only reading.	A persistent input field with an emoji picker and send control sits below the message list, and the attendee's just-sent message appears in the stream marked "pending… / under review" until moderation resolves it — **BECAUSE** keeping the composer and the input's own state in one place makes posting and its consequence immediately legible.

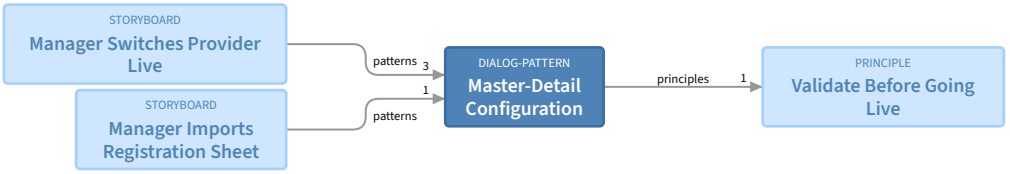
Kanban Moderation Board ⚓

Diagram



○ CATEGORY	○ ACTORS	○ CONTEXT
List	● **PERSONA:** Markus ⚭, ● **PERSONA:** Peter ⚭	A moderator triaging a stream of incoming chat and question messages.

● DIALOG-PATTERN	Properties & Description		
	○ PROBLEM	○ PRINCIPLES	○ USES
	Many messages must be classified and routed quickly with a clear sense of their state.	● PRINCIPLE: Live Reactivity, ● PRINCIPLE: Moderation-Gated Visibility	∅
	○ TRADE-OFF	Description	
	A wide multi-lane board that needs a desktop-sized screen and drag-capable pointer input.	Messages appear as cards in state lanes (pending, accepted, forwarded, answered) that the moderator advances by action, with filters and manual reordering — **BECAUSE** spatial state and drag actions make high-throughput triage fast and unambiguous.	
Master-Detail Configuration	Diagram		



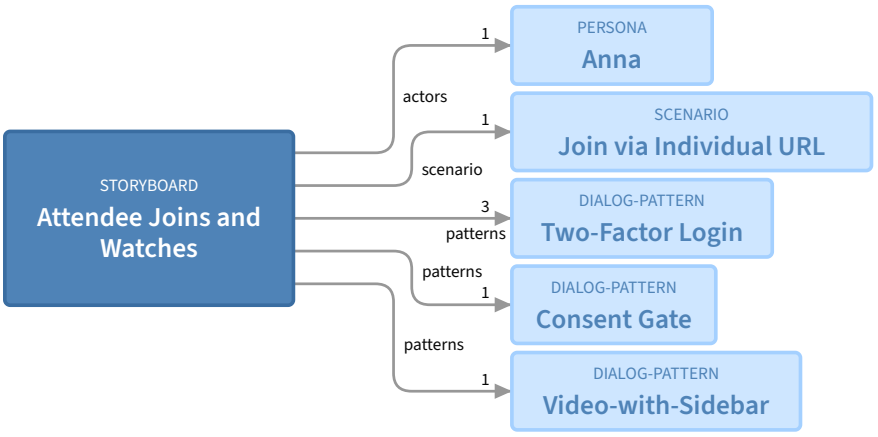
	○ CATEGORY	○ ACTORS	○ CONTEXT
	Layout	● PERSONA: Manfred	A manager configuring an event's channels, resources, and options.
	○ PROBLEM	○ PRINCIPLES	○ USES
	A large nested configuration must be navigated and edited without losing orientation.	● PRINCIPLE: Validate Before Going Live	∅
	○ TRADE-OFF	Description	
	The horizontal space of two panes, cramped on small screens, and only one item editable at a time.	A master list of channels and resources on one side drives a detail editor on the other, so selecting an item reveals its parameters for editing — **BECAUSE** master-detail keeps a large hierarchical configuration navigable and focused.	
Live Statistics Dashboard	Diagram		

	○ CATEGORY	○ ACTORS	○ CONTEXT
	Data	● PERSONA: Manfred, ● PERSONA: Petra	A manager or studio screen observing event health during the broadcast.
	○ PROBLEM	○ PRINCIPLES	○ USES
	Operators need an at-a-glance, continuously updating view of audience and channels.	● PRINCIPLE: Live Reactivity	∅
	○ TRADE-OFF	Description	
	A periodic snapshot cadence, so the charts lag the live state by up to one interval.	A dashboard composes a time-series attendee curve with per-channel pie charts that refresh from periodic statistics snapshots — **BECAUSE** combined trend and distribution views convey event health without manual querying.	

20 ● UXUI: Dialog Storyboard (DS)

Attendee Joins and Watches ACTORS: Anna	Attendee Asks a Tagged Question ACTORS: Anna	Attendee Reads and Posts Chat ACTORS: Anna	Attendee Gets Support from a Moderator ACTORS: Anna, Markus
Chat Moderator Triages Messages ACTORS: Markus	Moderator Forwards to Presenter ACTORS: Peter, Petra	Manager Switches Provider Live ACTORS: Manfred, Anna	Manager Imports Registration Sheet ACTORS: Manfred

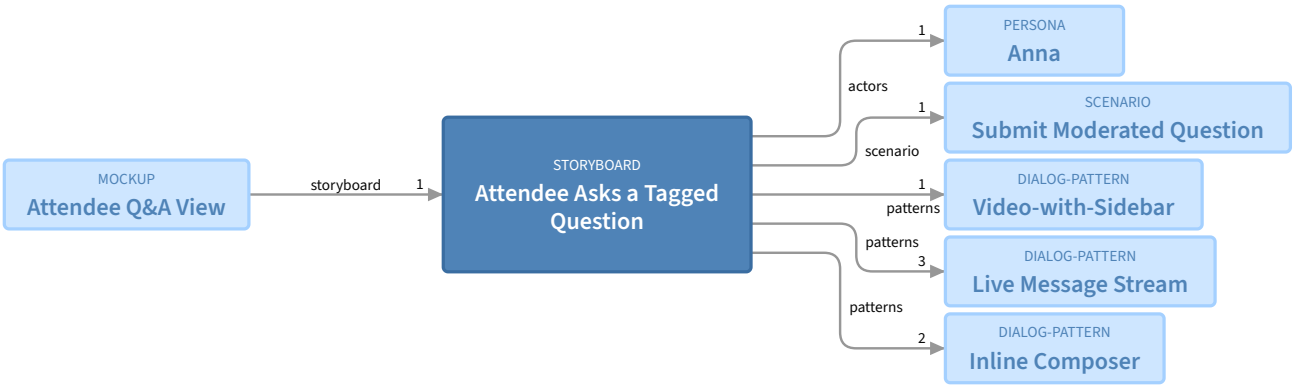
20.01 ● STORYBOARD: Attendee Joins and Watches



○ ACTORS	● PERSONA: Anna
○ SCENARIO	● SCENARIO: Join via Individual URL

● FRAME	○ SCENARIO-STEP	○ PATTERNS	Description
Event Landing	1	● DIALOG-PATTERN: Two-Factor Login	The attendee opens the event URL and sees the event title, login message, and an email field
Email Entry	2	● DIALOG-PATTERN: Two-Factor Login	The attendee enters their email; the system confirms a token has been sent to that address
Token Challenge	2	● DIALOG-PATTERN: Two-Factor Login	A token field appears with the email read-only; the attendee types the six-digit token
Consent Gate	2	● DIALOG-PATTERN: Consent Gate	The required login information is shown and the attendee accepts the conditions
Main Screen	3	● DIALOG-PATTERN: Video-with-Sidebar	The live video plays on the left with the interaction sidebar on the right

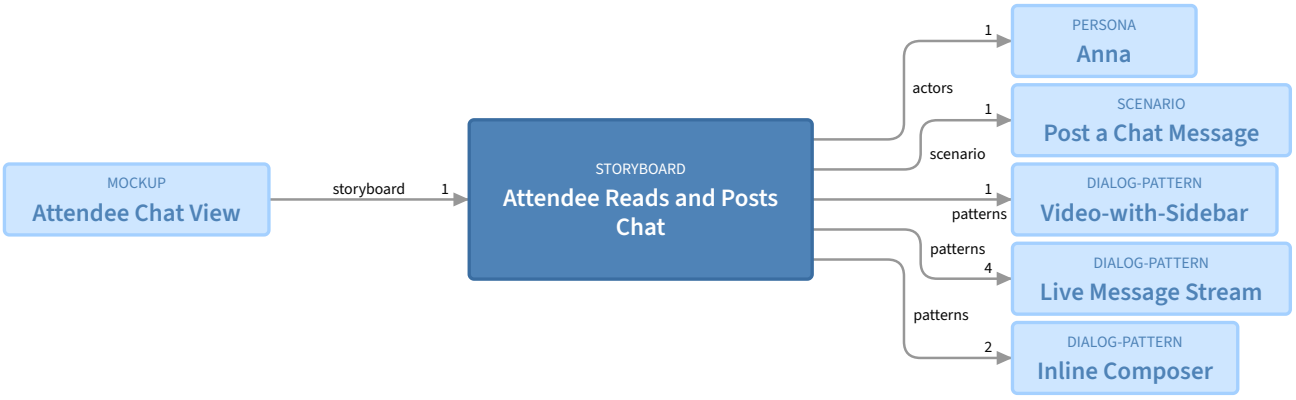
20.02 ● STORYBOARD: Attendee Asks a Tagged Question



○ ACTORS	● PERSONA: Anna
○ SCENARIO	● SCENARIO: Submit Moderated Question

● FRAME	○ SCENARIO-STEP	○ PATTERNS	Description
Questions Tab	1	● DIALOG-PATTERN: Video-with-Sidebar, ● DIALOG-PATTERN: Live Message Stream	The attendee selects the questions tab in the sidebar showing existing accepted questions
Compose Question	2	● DIALOG-PATTERN: Inline Composer	The attendee types a question and the available question tags are offered for selection
Tag and Submit	3	● DIALOG-PATTERN: Inline Composer	The attendee picks one or more tags and submits the question
Pending Acknowledgement	4	● DIALOG-PATTERN: Live Message Stream	The question appears in the attendee's own list marked as pending moderation, a rejected one tinted and visible to the author only
Becomes Visible	5	● DIALOG-PATTERN: Live Message Stream	After approval the question becomes visible and likeable for all attendees

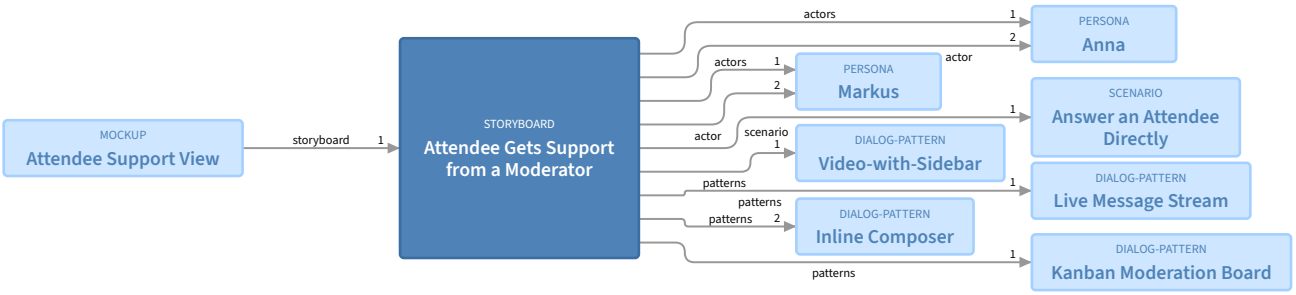
20.03 ● STORYBOARD: Attendee Reads and Posts Chat



○ ACTORS	● PERSONA: Anna
○ SCENARIO	● SCENARIO: Post a Chat Message

● FRAME	○ SCENARIO-STEP	○ PATTERNS	Description
Chat Tab	∅	● DIALOG-PATTERN: Video-with-Sidebar, ● DIALOG-PATTERN: Live Message Stream	The attendee opens the Chat tab and sees the scrolling stream with a “2 new messages” jump indicator
Read Mixed Stream	∅	● DIALOG-PATTERN: Live Message Stream	Others’ messages show name, time, and like count, an amber moderator message stands out, and a quoted original links back to its source
Like and Reply	∅	● DIALOG-PATTERN: Live Message Stream	The attendee likes a message, raising its count, and follows a quoted original link back to its source
Compose and Send	2	● DIALOG-PATTERN: Inline Composer	The attendee types in the bottom composer, optionally adds an emoji, and sends the message
Under Review	4	● DIALOG-PATTERN: Inline Composer, ● DIALOG-PATTERN: Live Message Stream	The attendee’s own message appears right-aligned marked “pending… / under review” until moderation resolves it

20.04 ● STORYBOARD: Attendee Gets Support from a Moderator



○ ACTORS	● PERSONA: Anna, ● PERSONA: Markus
○ SCENARIO	● SCENARIO: Answer an Attendee Directly

● FRAME	○ SCENARIO-STEP	○ ACTOR	○ PATTERNS	Description
Support Tab	∅	● PERSONA: Anna	● DIALOG-PATTERN: Video-with-Sidebar, ● DIALOG-PATTERN: Inline Composer	The attendee opens the Support tab, which shows only their own requests, and writes a request for help into the composer
Chat Moderation Board	1	● PERSONA: Markus	● DIALOG-PATTERN: Kanban Moderation Board	The moderator reviews the incoming chat messages in the rejected, pending, and accepted lanes beside the stream, the request among them
Direct Reply	4	● PERSONA: Markus	● DIALOG-PATTERN: Inline Composer	The moderator writes a reply in the board’s composer and marks it as a

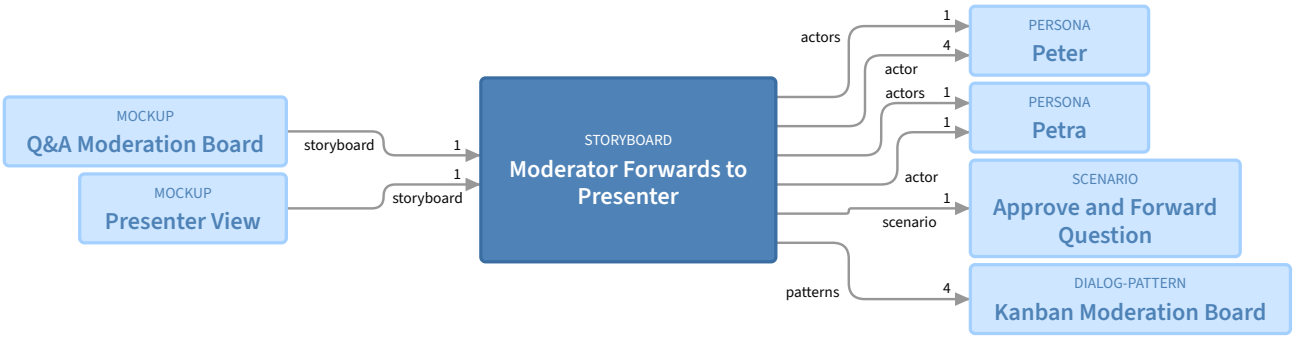
● FRAME	○ SCENARIO-STEP	○ ACTOR	○ PATTERNS	Description
				direct message to the attendee instead of a public one
Reply Received	5	● PERSONA: Anna	● DIALOG-PATTERN: Live Message Stream	The attendee sees the amber-tinted moderator reply below their own request in the Support tab, without any further approval step

20.05 ● STORYBOARD: Chat Moderator Triages Messages

○ ACTORS	● PERSONA: Markus
○ SCENARIO	● SCENARIO: Reject Improper Message

● FRAME	○ SCENARIO-STEP	○ PATTERNS	Description
Board Overview	∅	● DIALOG-PATTERN: Kanban Moderation Board	The moderator opens the chat moderation board with the stream preview beside the rejected, pending, and accepted lanes, each lane heading counting its messages
Narrow Lane	∅	● DIALOG-PATTERN: Kanban Moderation Board	The moderator narrows a lane through the timestamp and tag label filter chips at its head, collapsing the messages outside the filter
Assess Pending	∅	● DIALOG-PATTERN: Kanban Moderation Board	Each pending card shows the sentiment pre-assessment chip (safe, iffy, risk) beside sender and time, above the outlined reject and filled accept buttons
Reject Improper	2	● DIALOG-PATTERN: Kanban Moderation Board	The moderator clicks reject on an improper message, which leaves the pending lane immediately
Hidden from Audience	3	● DIALOG-PATTERN: Kanban Moderation Board	The card reappears at the top of the rejected lane marked rejected, while the accepted lane the audience sees stays unchanged

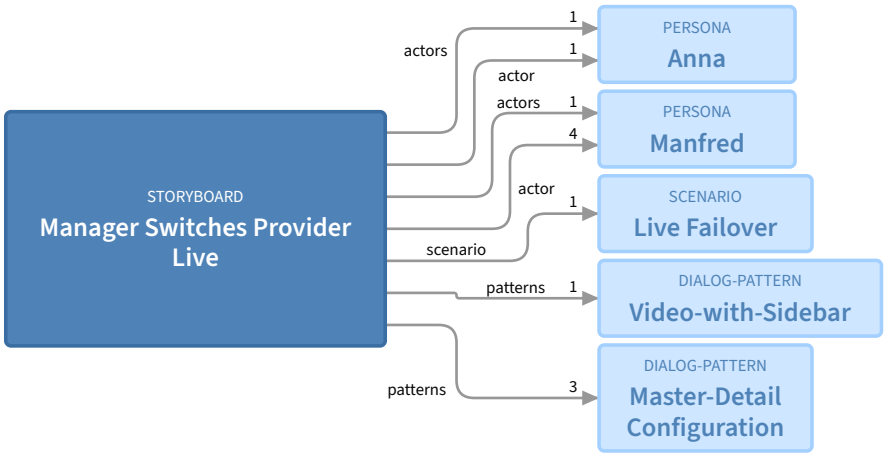
20.06 ● STORYBOARD: Moderator Forwards to Presenter



○ ACTORS	● PERSONA: Peter, ● PERSONA: Petra
○ SCENARIO	● SCENARIO: Approve and Forward Question

● FRAME	○ SCENARIO-STEP	○ ACTOR	○ PATTERNS	Description
Board Overview	∅	● PERSONA: Peter	● DIALOG-PATTERN: Kanban Moderation Board	The moderator sees question cards in the rejected, pending, accepted, suspended, forwarded, and answered lanes
Filter Pending	1	● PERSONA: Peter	● DIALOG-PATTERN: Kanban Moderation Board	The moderator filters to show only pending questions to focus the triage
Approve Card	2	● PERSONA: Peter	● DIALOG-PATTERN: Kanban Moderation Board	The moderator approves a relevant question, moving it to the accepted lane
Forward with Hint	3	● PERSONA: Peter	● DIALOG-PATTERN: Kanban Moderation Board	The moderator forwards the card to the presenter and attaches a routing hint in the presenter alert
Presenter Basket	5	● PERSONA: Petra	∅	The card appears in the presenter's ordered work basket below the acknowledgeable hint and is locked from attendee edits

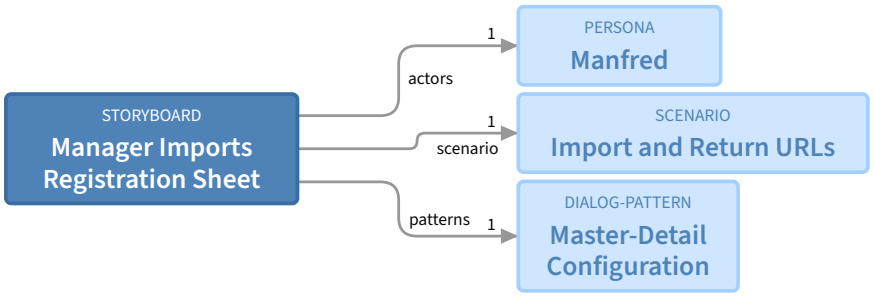
20.07 ● STORYBOARD: Manager Switches Provider Live



○ ACTORS	● PERSONA: Manfred, ● PERSONA: Anna
○ SCENARIO	● SCENARIO: Live Failover

● FRAME	○ SCENARIO-STEP	○ ACTOR	○ PATTERNS	Description
Channel Master	1	● PERSONA: Manfred	● DIALOG-PATTERN: Master-Detail Configuration	The manager selects the affected channel from the master list of channels
Resource Detail	1	● PERSONA: Manfred	● DIALOG-PATTERN: Master-Detail Configuration	The channel's resources are shown with the currently active provider highlighted
Activate Fallback	2	● PERSONA: Manfred	● DIALOG-PATTERN: Master-Detail Configuration	The manager activates a different resource as the new active one
Propagation Confirmed	4	● PERSONA: Manfred	∅	The system confirms the change has been pushed to all connected clients
Seamless Switch	5	● PERSONA: Anna	● DIALOG-PATTERN: Video-with-Sidebar	Attendee players transition to the new stream with no user action required

20.08 ● STORYBOARD: Manager Imports Registration Sheet



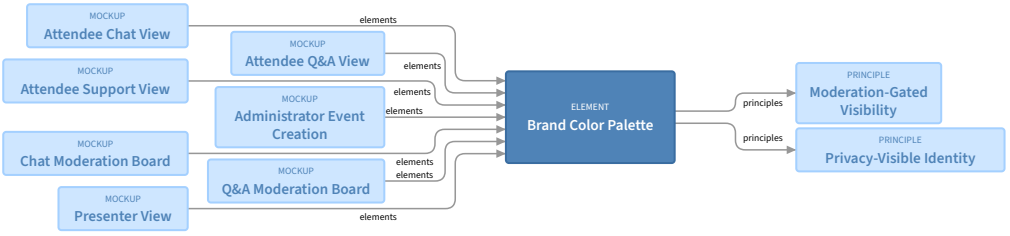
○ ACTORS	● PERSONA: Manfred
○ SCENARIO	● SCENARIO: Import and Return URLs

● FRAME	○ SCENARIO-STEP	○ PATTERNS	Description
Event Config	∅	● DIALOG-PATTERN: Master-Detail Configuration	The manager opens the event configuration, with its basics and event type profile, and selects the attendee import action
Upload Sheet	1	∅	The manager uploads the Excel sheet of the Event Registration System with attendee columns
Import Summary	3	∅	The system reports new users created and existing users skipped, with tokens generated
URL Preview	4	∅	The system shows the composed personal access URLs per attendee
Return Download	5	∅	The manager downloads the returned Excel sheet with the URL column filled for the Event Registration System

21 ● UXUI: Visual Design (VD)

Brand Color Palette CATEGORY: Color	Light and Dark Theme CATEGORY: Color	Broadcast Wordmark CATEGORY: Brand	Self-Hosted Typography CATEGORY: Typography
Fontawesome Iconography CATEGORY: Iconography	Responsive Spacing Scale CATEGORY: Spacing	Card and Bubble Shape CATEGORY: Shape	Stream Imagery Treatment CATEGORY: Imagery
Subtle State Transitions CATEGORY: Motion			

● ELEMENT	Properties & Description
Brand Color Palette	Diagram



● ELEMENT	Properties & Description

Properties

○ CATEGORY	Color
○ PRINCIPLES	● PRINCIPLE: Moderation-Gated Visibility, ● PRINCIPLE: Privacy-Visible Identity
○ SOURCE	Tailwind theme configuration of the frontend (`tailwind.config.ts`), color section
○ TRADE-OFF	A palette bound to the corporate identity, leaving no room for event-specific branding.

Description

The interface uses the msg corporate palette with a primary msg blue for actions, the active channel, and own outgoing messages, neutral surfaces for others' messages, a warm amber tint for moderator-authored messages and the "under review" state, a red tint for rejected input, and muted italic styling for deleted messages — **BECAUSE** color-coded message provenance and state give attendees and moderators instant visual feedback.

Light and Dark Theme

Diagram

Properties

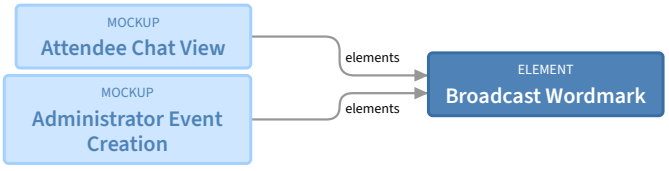
○ CATEGORY	Color
○ PRINCIPLES	● PRINCIPLE: Familiar Conventions
○ SOURCE	Tailwind theme configuration of the frontend (`tailwind.config.ts`), dark mode section
○ TRADE-OFF	Every semantic color role has to be defined and checked twice, once per theme.

Description

The application provides both a light and a dark theme, toggled from a sun/moon control in the header bar and applied consistently across video frame, sidebar, and message bubbles — **BECAUSE** attendees view events in varied lighting and expect to choose a comfortable appearance.

Broadcast Wordmark

Diagram



Properties

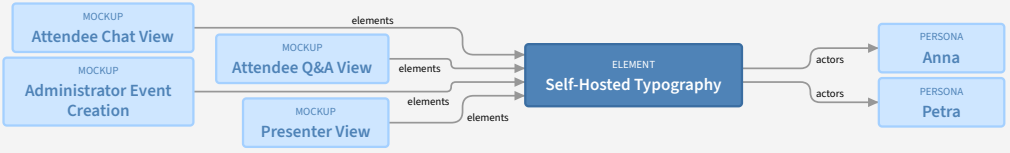
○ CATEGORY	Brand
○ TRADE-OFF	A reserved header strip on every screen, including the space-constrained mobile layouts.

Description

The product is represented by the "BROADCAST" wordmark with the circled "O" accent in msg blue, placed top-left in the header bar — **BECAUSE** a distinctive, consistently placed wordmark anchors the brand on every screen.

Self-Hosted Typography

Diagram



● ELEMENT	Properties & Description

Properties

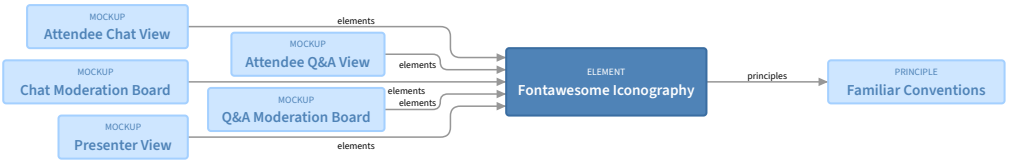
○ CATEGORY	Typography
○ ACTORS	● PERSONA: Anna, ● PERSONA: Petra
○ SOURCE	TypoPRO font package of the frontend (`@typopro/web-*`), subsetted at build time
○ TRADE-OFF	A larger initial download than a system font stack, as the font files ship with the application.

Description

Text is set in a self-hosted TypoPRO web font family with a clear type scale for headings, body, and captions, sized for legibility on the studio screen at 2-4 meters viewing distance — **BECAUSE** self-hosted fonts avoid third-party requests and the studio view must remain readable from afar.

Fontawesome Iconography

Diagram



Properties

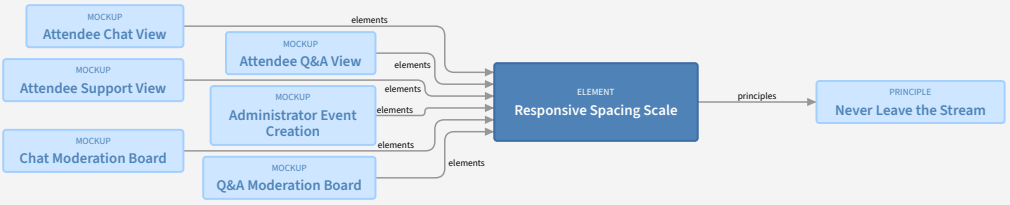
○ CATEGORY	Iconography
○ PRINCIPLES	● PRINCIPLE: Familiar Conventions
○ SOURCE	Fontawesome Free package of the frontend (`@fortawesome/fontawesome-free`)
○ TRADE-OFF	A vocabulary limited to the coverage of one icon set, so a missing metaphor is worked around with a text label.

Description

Actions and message states are reinforced with a consistent Fontawesome icon set, pairing icons with text labels for moderation controls — **BECAUSE** consistent icons speed recognition of repeated moderation actions.

Responsive Spacing Scale

Diagram



Properties

○ CATEGORY	Spacing
○ PRINCIPLES	● PRINCIPLE: Never Leave the Stream
○ SOURCE	Tailwind theme configuration of the frontend (`tailwind.config.ts`), spacing and screens sections
○ TRADE-OFF	Fixed breakpoints, so a layout jumps between its arrangements instead of scaling fluidly.

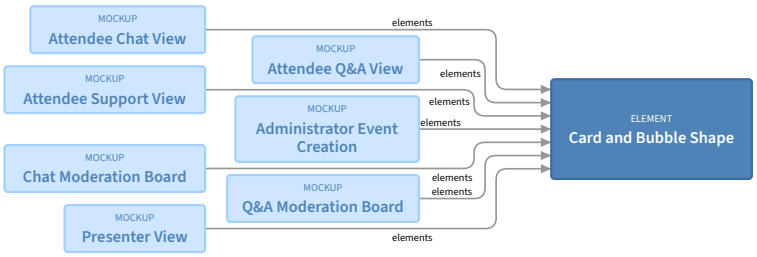
Description

Spacing follows the Tailwind 4px-based scale with a 16px base gutter, and the breakpoints `sm` (640px), `md` (768px), and `lg` (1024px) decide at which widths the video-with-sidebar arrangement collapses into the stacked single column — **BECAUSE** one shared scale keeps the density of video, sidebar, and boards consistent across the device classes attendees use.

● ELEMENT	Properties & Description

Card and Bubble Shape

Diagram



Properties

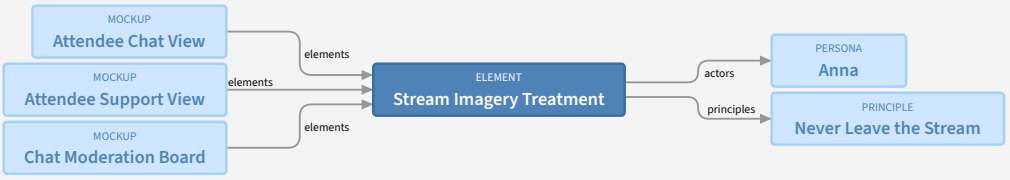
○ **CATEGORY**	Shape
○ **TRADE-OFF**	A soft look which relies on subtle tints and hairline borders, demanding a well-tuned contrast in both themes.

Description

Messages, questions, and form fields use rounded rectangles with a 6px corner radius and a hairline border, moderation cards get a subtle elevation shadow, and buttons are pill-shaped with the primary action filled and the secondary action outlined — **BECAUSE** a uniform shape language lets the color and iconography carry the meaning of an item.

Stream Imagery Treatment

Diagram



Properties

○ **CATEGORY**	Imagery
○ **ACTORS**	● **PERSONA:** Anna
○ **PRINCIPLES**	● **PRINCIPLE:** Never Leave the Stream
○ **TRADE-OFF**	Letterbox bars on screens whose aspect ratio differs from 16:9, wasting some display area.

Description

The live video occupies a prominent 16:9 frame with a neutral letterbox and a subtle channel and language badge overlay — **BECAUSE** the stream is the focal content and attendees must see which channel they are watching.

Subtle State Transitions

Diagram

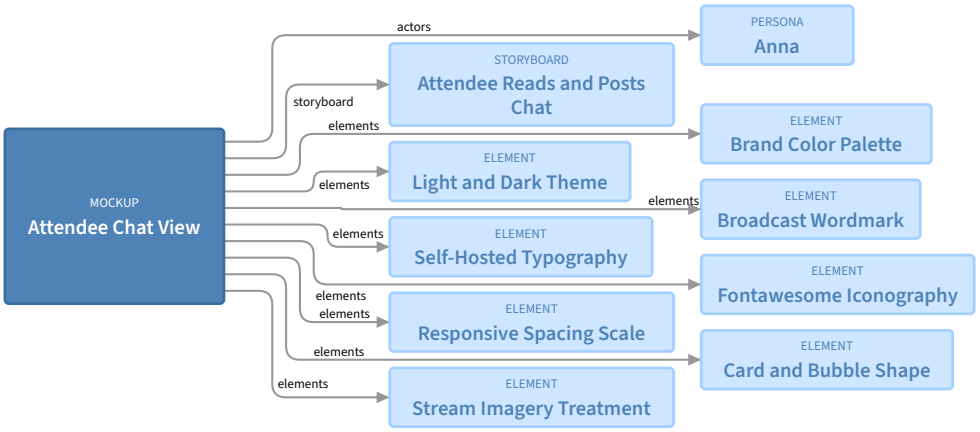
Properties

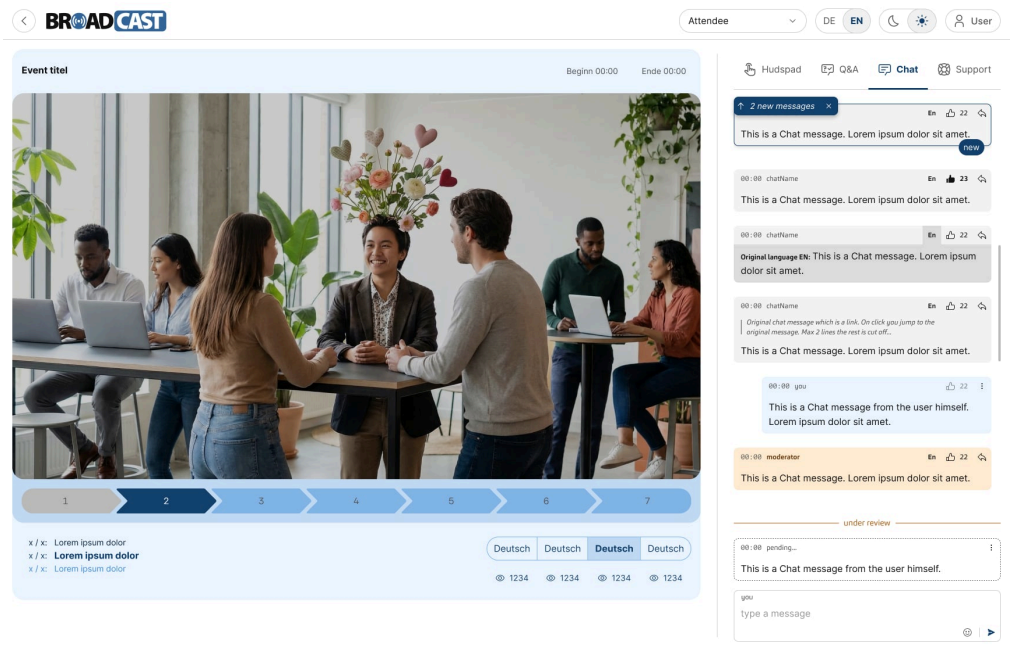
○ **CATEGORY**	Motion
○ **PRINCIPLES**	● **PRINCIPLE:** Live Reactivity
○ **TRADE-OFF**	Transitions short enough to be missed when the user is not looking at the changed region.

Description

State and configuration changes animate with short, subtle transitions under 200ms, including the seamless cross-fade on a stream provider switch — **BECAUSE** gentle motion signals live updates without distracting from the event.

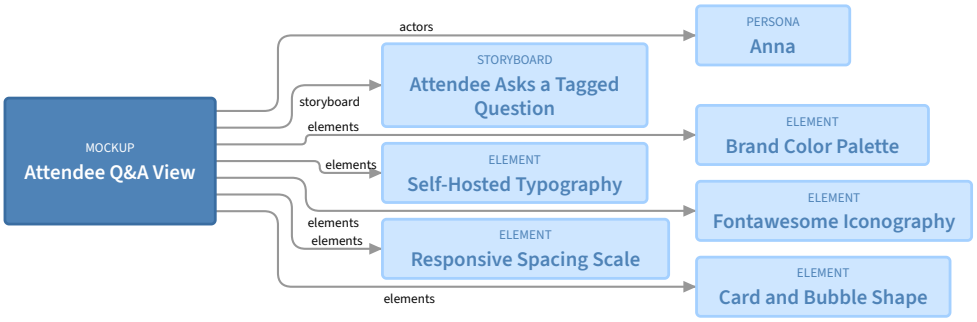
21.01 ● MOCKUP: Attendee Chat View ⚓

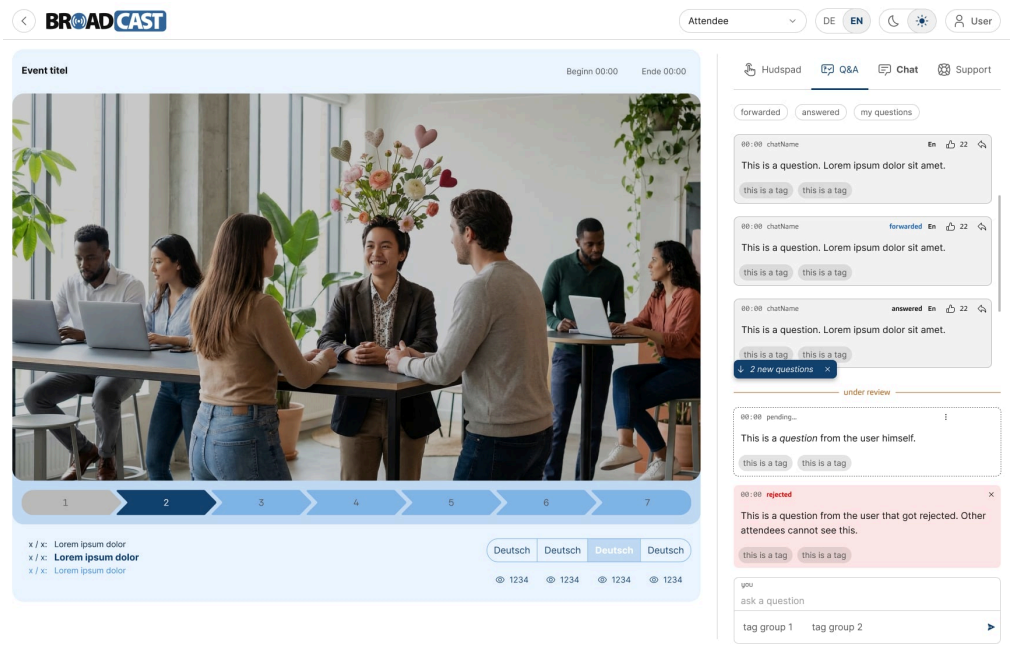


○ ACTORS	● PERSONA: Anna ⚭
○ STORYBOARD	● STORYBOARD: Attendee Reads and Posts Chat ⚭
○ ELEMENTS	● ELEMENT: Brand Color Palette ⚭, ● ELEMENT: Light and Dark Theme ⚭, ● ELEMENT: Broadcast Wordmark ⚭, ● ELEMENT: Self-Hosted Typography ⚭, ● ELEMENT: Fontawesome Iconography ⚭, ● ELEMENT: Responsive Spacing Scale ⚭, ● ELEMENT: Card and Bubble Shape ⚭, ● ELEMENT: Stream Imagery Treatment ⚭
○ IMAGE	

The main attendee screen with the Chat tab open, showing the blue own message, the amber moderator message, the “under review” divider, and the new-messages badge above the letterboxed stream.

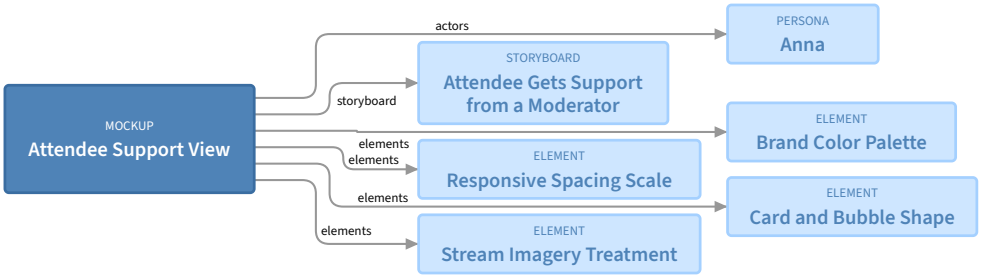
21.02 ● MOCKUP: Attendee Q&A View

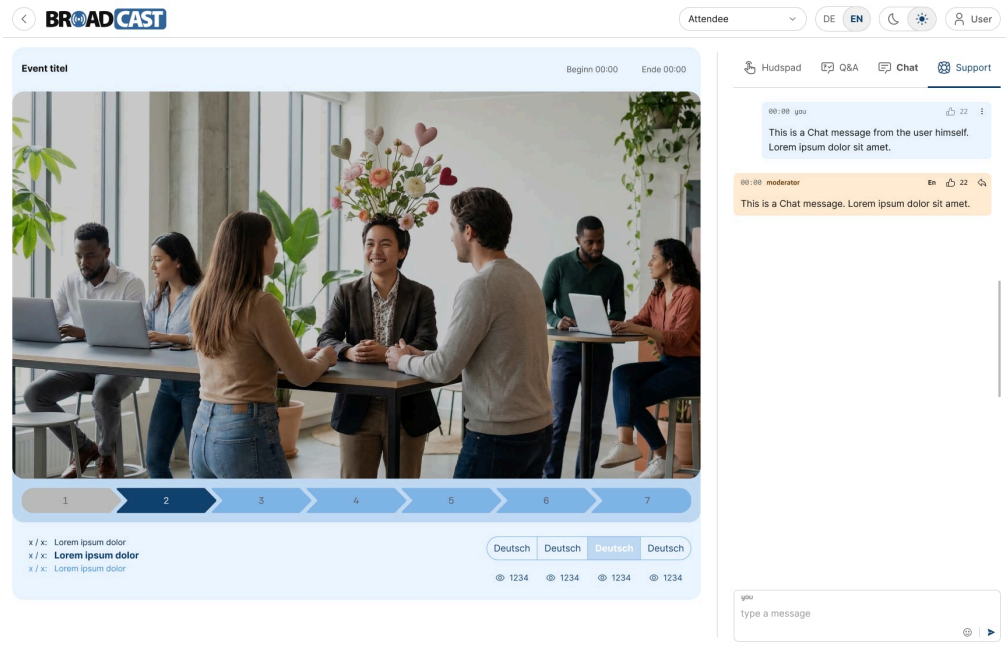


○ ACTORS	● PERSONA: Anna
○ STORYBOARD	● STORYBOARD: Attendee Asks a Tagged Question
○ ELEMENTS	● ELEMENT: Brand Color Palette, ● ELEMENT: Self-Hosted Typography, ● ELEMENT: Fontawesome Iconography, ● ELEMENT: Responsive Spacing Scale, ● ELEMENT: Card and Bubble Shape
○ IMAGE	

The main attendee screen with the Q&A tab open, showing the filter pills, the tag chips below each question, the pending own question under review, and the red-tinted rejected question.

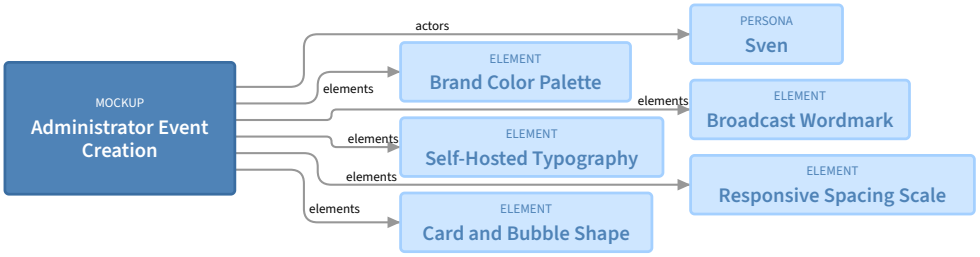
21.03 ● MOCKUP: Attendee Support View



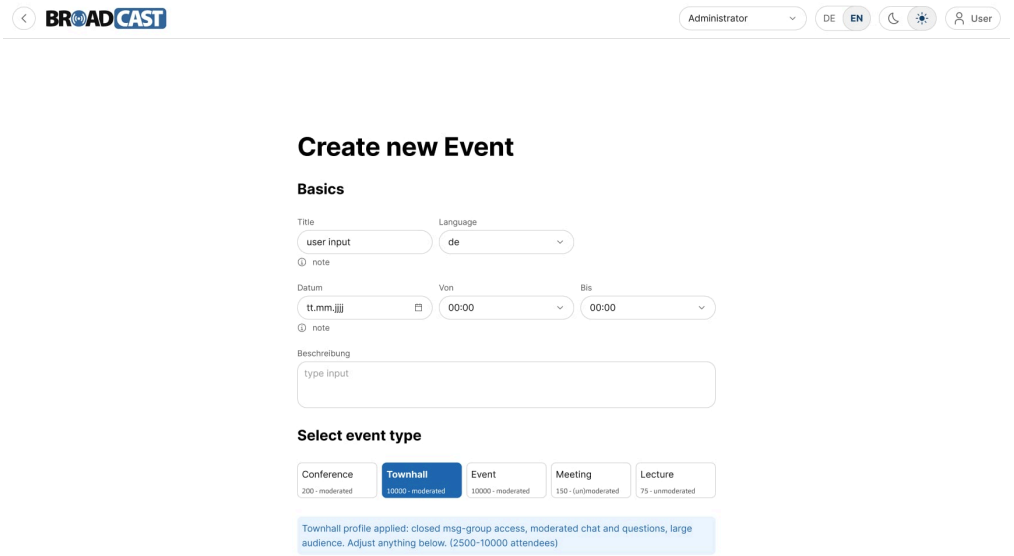
○ ACTORS	● PERSONA: Anna
○ STORYBOARD	● STORYBOARD: Attendee Gets Support from a Moderator
○ ELEMENTS	● ELEMENT: Brand Color Palette, ● ELEMENT: Responsive Spacing Scale, ● ELEMENT: Card and Bubble Shape, ● ELEMENT: Stream Imagery Treatment
○ IMAGE	

The main attendee screen with the Support tab open, showing the private exchange of a blue own message and an amber moderator reply above the composer.

21.04 ● MOCKUP: Administrator Event Creation

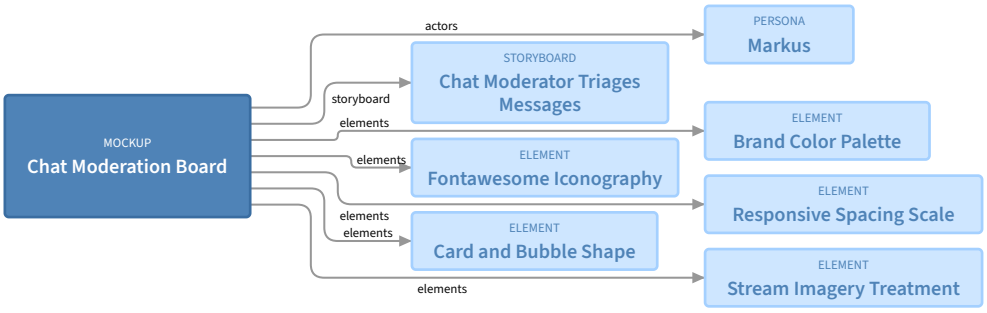


○ ACTORS	● PERSONA: Sven
○ ELEMENTS	● ELEMENT: Brand Color Palette, ● ELEMENT: Broadcast Wordmark, ● ELEMENT: Self-Hosted Typography, ● ELEMENT: Responsive Spacing Scale, ● ELEMENT: Card and Bubble Shape

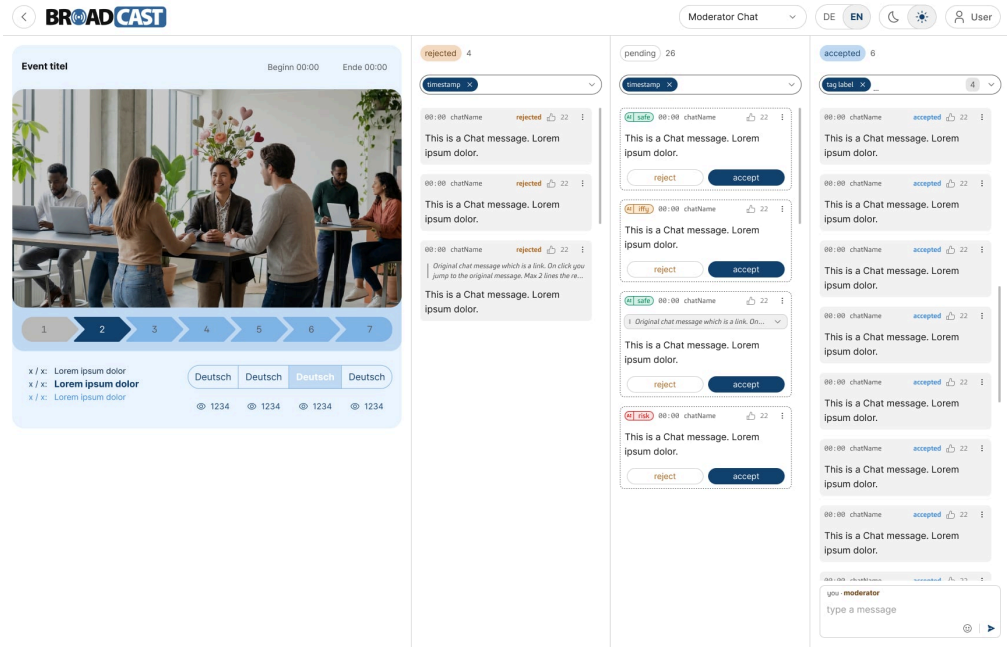
○ IMAGE	

The event creation form of the administrator, showing the type scale of the headings, the outlined form fields, the event type selector with the filled selected profile, and the blue-tinted profile summary.

21.05 ● MOCKUP: Chat Moderation Board ⚓

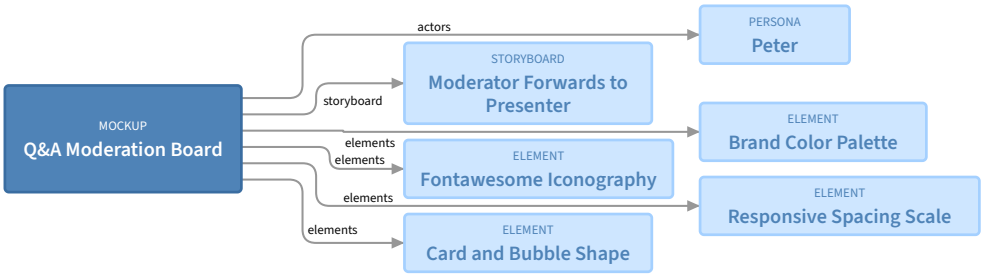


○ ACTORS	● PERSONA: Markus ⚭
○ STORYBOARD	● STORYBOARD: Chat Moderator Triages Messages ⚭
○ ELEMENTS	● ELEMENT: Brand Color Palette ⚭, ● ELEMENT: Fontawesome Iconography ⚭, ● ELEMENT: Responsive Spacing Scale ⚭, ● ELEMENT: Card and Bubble Shape ⚭, ● ELEMENT: Stream Imagery Treatment ⚭

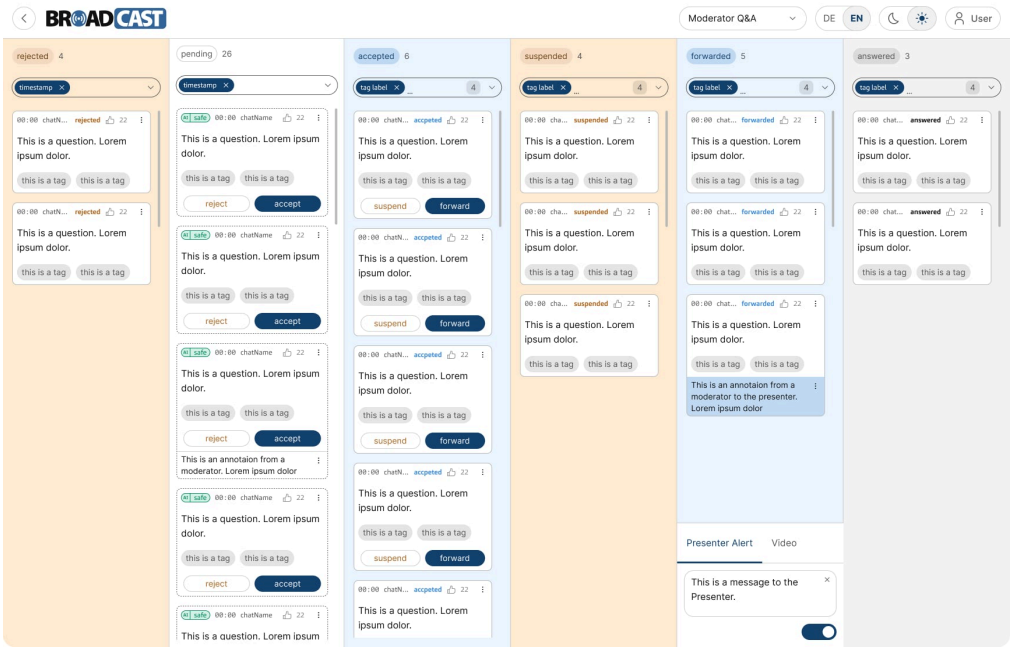
○ IMAGE	

The chat moderation board with the stream preview beside the rejected, pending, and accepted columns, showing the elevated pending cards with their outlined reject and filled accept buttons and the AI assessment chips.

21.06 ● MOCKUP: Q&A Moderation Board ⚓

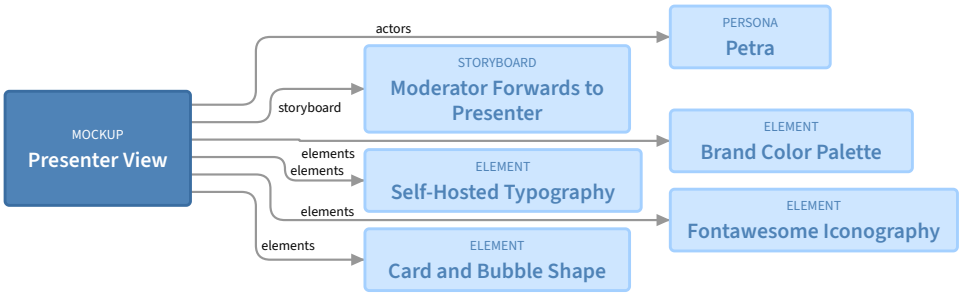


○ ACTORS	● PERSONA: Peter ⚭
○ STORYBOARD	● STORYBOARD: Moderator Forwards to Presenter ⚭
○ ELEMENTS	● ELEMENT: Brand Color Palette ⚭, ● ELEMENT: Fontawesome Iconography ⚭, ● ELEMENT: Responsive Spacing Scale ⚭, ● ELEMENT: Card and Bubble Shape ⚭

○ IMAGE	

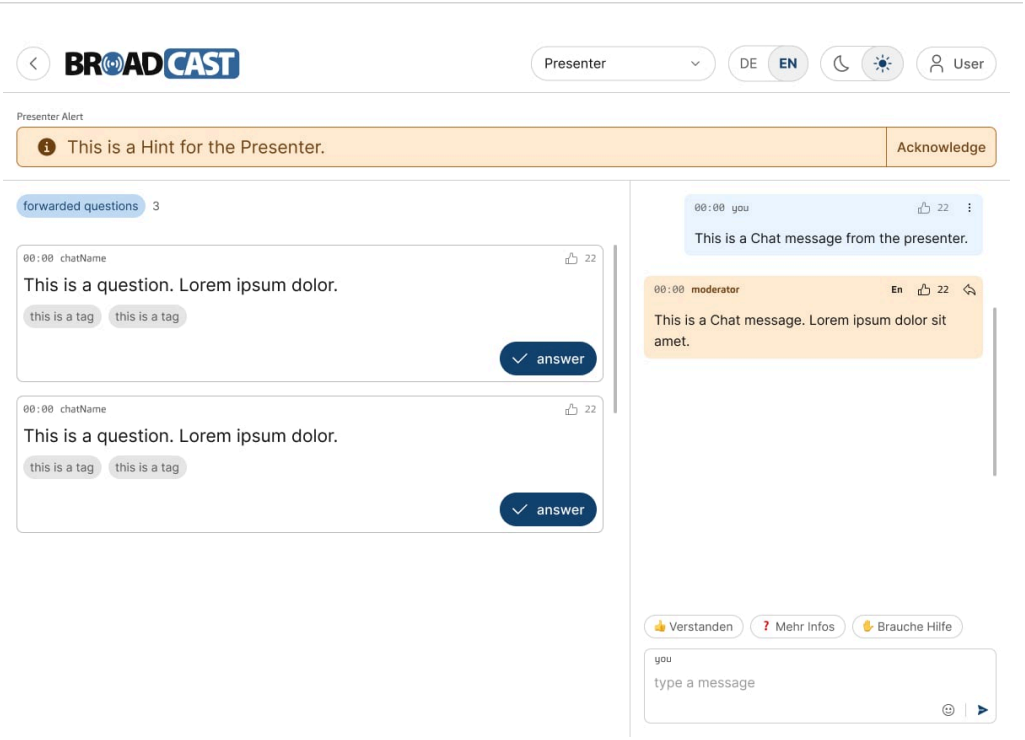
The all-in-one Q&A moderation board with the rejected, pending, accepted, suspended, forwarded, and answered columns, showing the amber moderator annotations and the presenter alert composer.

21.07 ● MOCKUP: Presenter View ⚓



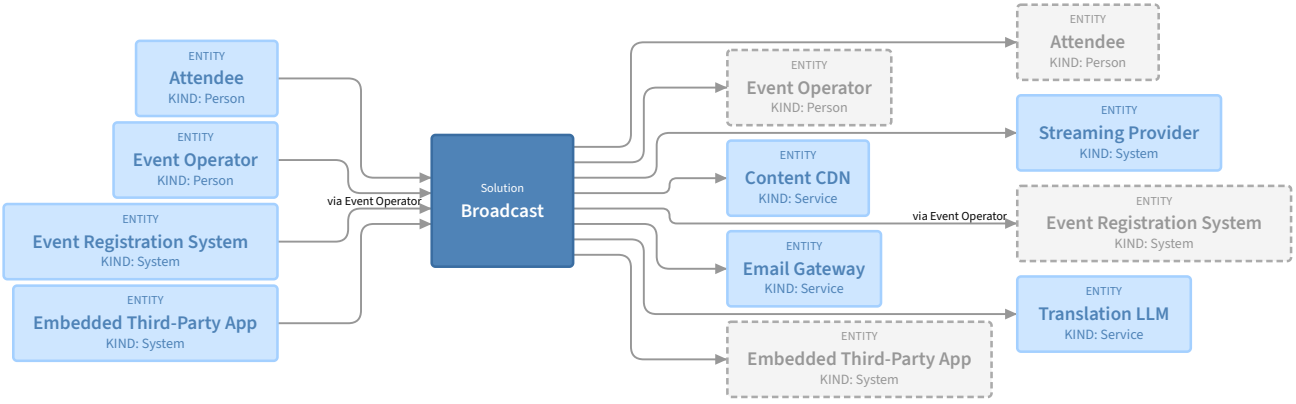
○ ACTORS	● PERSONA: Petra ⚭
○ STORYBOARD	● STORYBOARD: Moderator Forwards to Presenter ⚭
○ ELEMENTS	● ELEMENT: Brand Color Palette ⚭, ● ELEMENT: Self-Hosted Typography ⚭, ● ELEMENT: Fontawesome Iconography ⚭, ● ELEMENT: Card and Bubble Shape ⚭

○ IMAGE



The presenter view for the studio screen with the enlarged type scale, the amber presenter alert banner, the forwarded questions with their filled answer buttons, and the private moderator chat.

22 ● ARCH: Context View (CV) ⚓

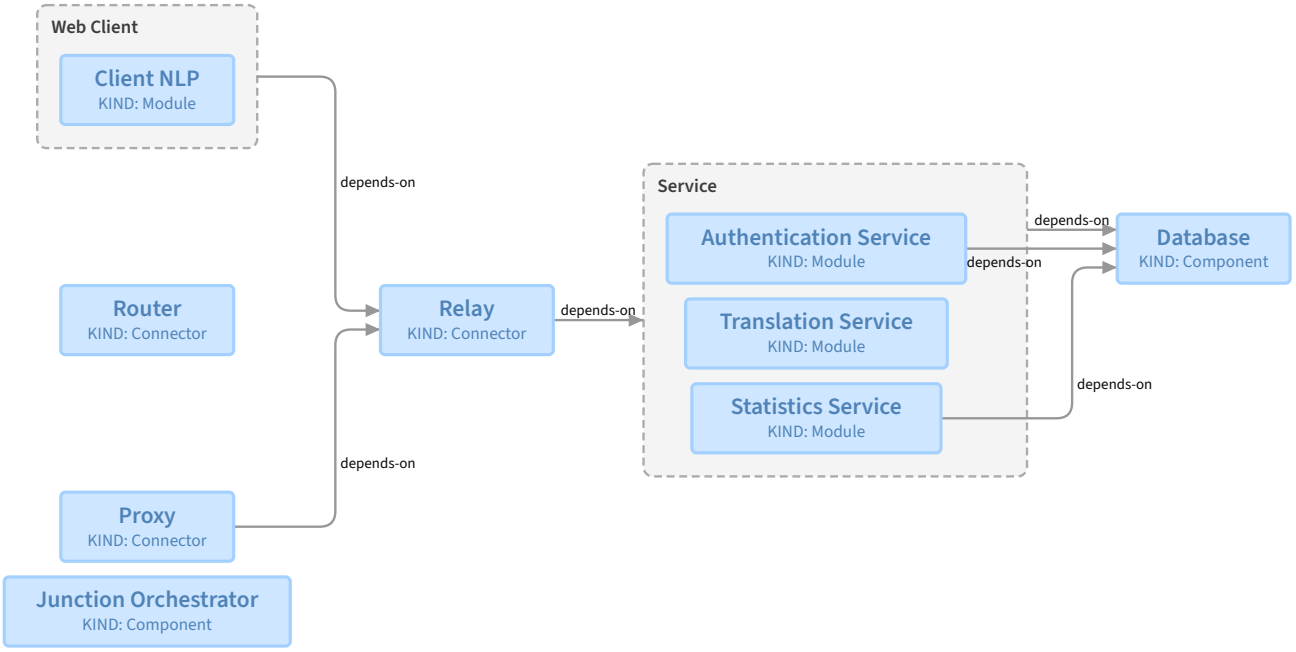


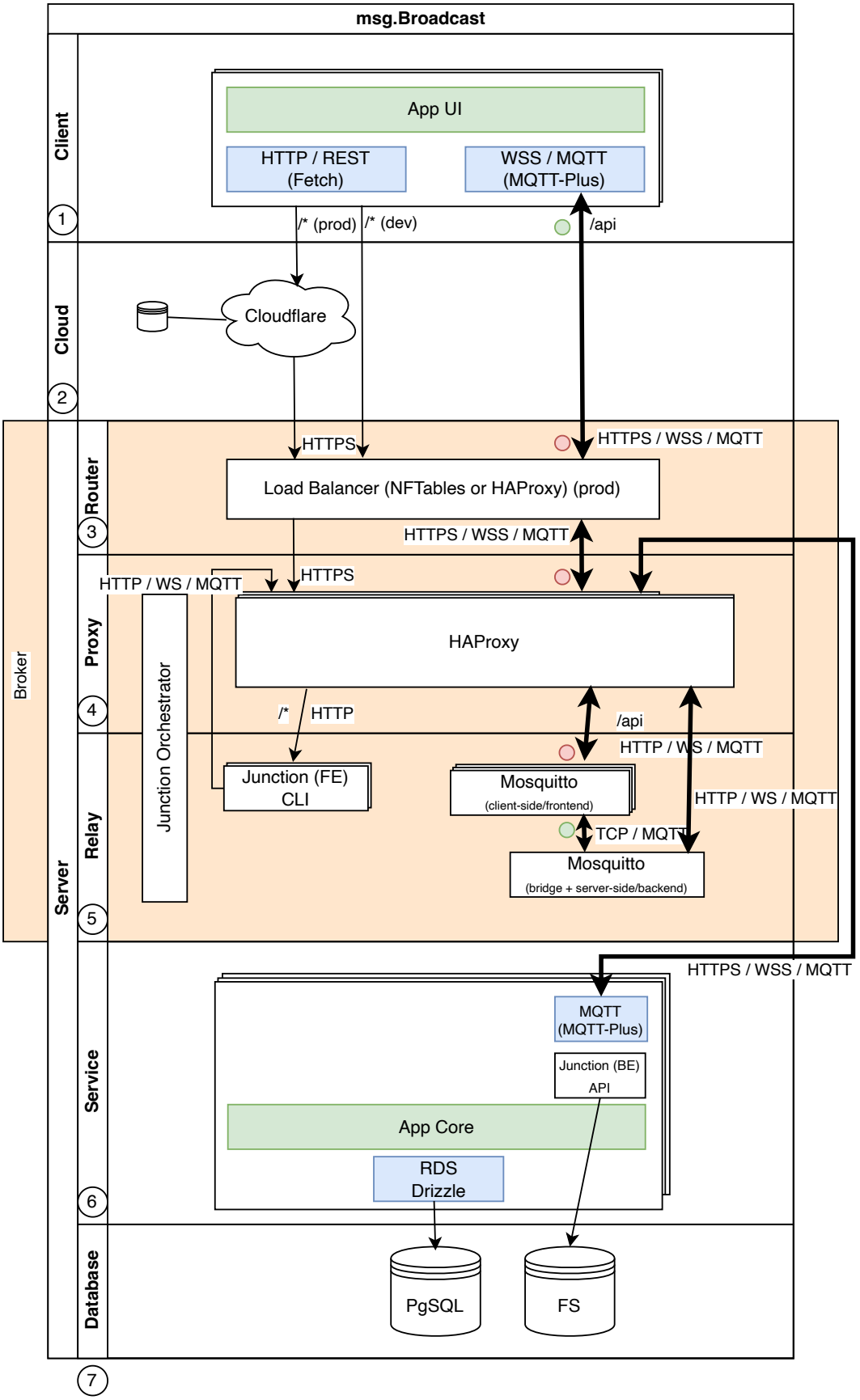
● ENTITY	Properties & Description
Attendee ⚓	Properties
○ KIND	Person
○ DIRECTION	Bidirectional
○ TERMS	● TERM: Attendee ⚭
○ ACTORS	● ROLE: Attendee ⚭
○ USE-CASES	● USE-CASE: Join Event and Watch Stream ⚭, ● USE-CASE: Authenticate via Email Token ⚭, ● USE-CASE: Ask a Question ⚭, ● USE-CASE: Chat During the Event ⚭

● ENTITY	Properties & Description	
	○ PREMISES	● PREMISE: Attendee Networks Pass WebSocket Traffic
	○ DATA	Login credentials, video playback, chat, questions, likes
	Description	
	The attendee is the person who authenticates, watches the stream, and interacts, through the browser-based web client of the solution — BECAUSE Broadcast is a browser-delivered solution serving thousands of concurrent attendees.	
Event Operator	Properties	
	○ KIND	Person
	○ DIRECTION	Bidirectional
	○ TERMS	● TERM: Manager, ● TERM: Moderator, ● TERM: Presenter
	○ ACTORS	● ROLE: Moderator, ● ROLE: Presenter, ● ROLE: Manager
	○ USE-CASES	● USE-CASE: Configure the Event, ● USE-CASE: Create Event from Registration Import, ● USE-CASE: Switch Streaming Provider, ● USE-CASE: Publish, Start, and Finish Event, ● USE-CASE: Export Anonymized Event Data, ● USE-CASE: Moderate and Forward Messages, ● USE-CASE: Moderate the Chat Conversation, ● USE-CASE: Steer the Presenter, ● USE-CASE: Present Forwarded Questions
	○ DATA	Event configuration, moderation actions, statistics, exports
	Description	
	The event operator is a manager, moderator, or presenter who configures events, moderates input, and presents forwarded questions, through the same browser-based web client as the attendees — BECAUSE all event operation happens through the same web application as attendance.	
Streaming Provider	Properties	
	○ KIND	System
	○ DIRECTION	Outbound
	○ TERMS	● TERM: Streaming Provider
	○ USE-CASES	● USE-CASE: Join Event and Watch Stream, ● USE-CASE: Configure the Event, ● USE-CASE: Switch Streaming Provider
	○ PREMISES	● PREMISE: Streaming Providers Deliver the Video
	○ DATA	Live video ingest and playback streams
	Description	
	External streaming providers such as msg Filmstudio, msg Broadcast, YouTube, Cloudflare, Twitch, and 3Q ingest and deliver the video — BECAUSE the solution is streaming-provider independent and consumes provider endpoints rather than hosting video.	
Content CDN	Properties	
	○ KIND	Service
	○ DIRECTION	Outbound
	○ USE-CASES	● USE-CASE: Join Event and Watch Stream
	○ DATA	Static client assets and static resources
	Description	
	A Cloudflare content delivery network fronts static content for fast, stable distribution — BECAUSE static assets must load quickly and reliably for a global, large-scale audience.	
Event Registration System	Properties	
	○ KIND	System

● ENTITY	Properties & Description	
	○ DIRECTION	Bidirectional
	○ VIA	● ENTITY: Event Operator
	○ TERMS	● TERM: Event Registration System
	○ USE-CASES	● USE-CASE: Create Event from Registration Import
	○ PREMISES	● PREMISE: Registered Audience Known in Advance, ● PREMISE: Registration System Exchanges Attendee Data
	○ DATA	Attendee registration data and generated access URLs
	Description	
	The Event Registration System supplies attendee registration data and receives generated access URLs without any direct technical interface: the manager carries both as Excel sheets, uploading the exported attendee list into the event and handing the returned URL sheet back — **BECAUSE** events are provisioned from the organization's existing registration system, which distributes the invitations itself.	
Email Gateway	Properties	
	○ KIND	Service
	○ DIRECTION	Outbound
	○ USE-CASES	● USE-CASE: Authenticate via Email Token
	○ PREMISES	● PREMISE: Email Gateway Delivers Tokens Within Minutes
	○ DATA	Authorization token emails
	Description	
	An external mail-sending service delivers one-time authorization tokens to attendees — **BECAUSE** email is the first-factor channel through which access is verified.	
Translation LLM	Properties	
	○ KIND	Service
	○ DIRECTION	Outbound
	○ USE-CASES	● USE-CASE: Join Event and Watch Stream, ● USE-CASE: Ask a Question, ● USE-CASE: Chat During the Event
	○ PREMISES	● PREMISE: Translation Service Available During Events, ● PREMISE: Message Texts Contain Personal Data
	○ DATA	Source and translated message texts
	Description	
	An external AI/LLM service translates message texts between the supported languages — **BECAUSE** chat and questions are made available in German and English on the fly.	
Embedded Third-Party App	Properties	
	○ KIND	System
	○ DIRECTION	Bidirectional
	○ USE-CASES	● USE-CASE: Moderate the Chat Conversation
	○ DATA	Interactive app content and administration
	Description	
	An optional third-party application is embedded into the event for additional interactivity, with a separate admin URL for moderators — **BECAUSE** some events integrate external interactive tooling.	

23 ● ARCH: Functionality View (FV) ⚓

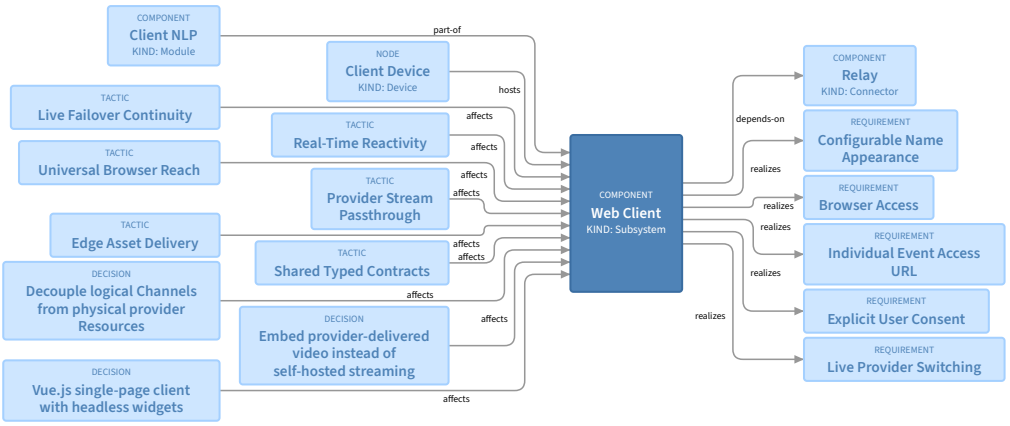



msg.Broadcast
Client
App UI
HTTP / REST
(Fetch)
WSS / MQTT
(MQTT-Plus)
1
/* (prod)
/* (dev)
/api
Cloud
Cloudflare
2
Broker
Server
Router
HTTPS
HTTPS / WSS / MQTT
Load Balancer (NFTables or HAProxy) (prod)
3
HTTPS / WSS / MQTT
Proxy
HTTP / WS / MQTT
HTTPS
HAProxy
4
/*
HTTP
/api
Junction Orchestrator
Relay
HTTP / WS / MQTT
Junction (FE)
CLI
Mosquitto
(client-side/frontend)
HTTP / WS / MQTT
TCP / MQTT
Mosquitto
(bridge + server-side/backend)
5
HTTPS / WSS / MQTT
Service
MQTT
(MQTT-Plus)
Junction (BE)
API
App Core
RDS
Drizzle
6
Database
PgSQL
FS
7

● COMPONENT	Properties & Description

Web Client

Diagram



Properties

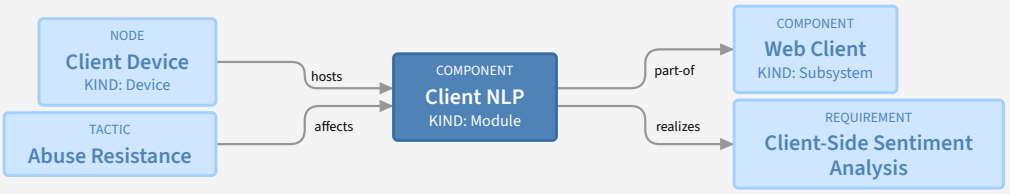
○ **KIND**	Subsystem
○ **REALIZES**	● **REQUIREMENT:** Configurable Name Appearance, ● **REQUIREMENT:** Browser Access, ● **REQUIREMENT:** Individual Event Access URL, ● **REQUIREMENT:** Explicit User Consent, ● **REQUIREMENT:** Live Provider Switching
○ **DEPENDS-ON**	● **COMPONENT:** Relay

Description

The Vue.js single-page application renders the attendee and operator UI and drives all user interaction: it presents the panel, attendee, studio, moderator, and manager screens, plays the video stream, and exchanges live data over MQTT — **BECAUSE** msg.Broadcast delivers its entire experience in the browser.

Client NLP

Diagram



Properties

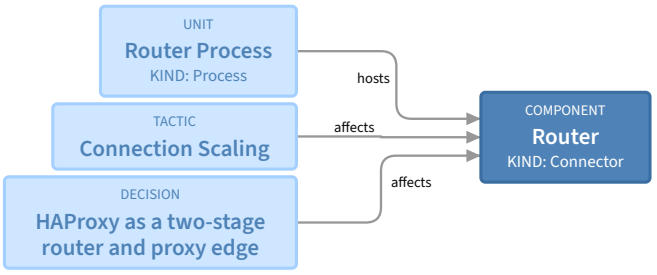
○ **KIND**	Module
○ **PART-OF**	● **COMPONENT:** Web Client
○ **REALIZES**	● **REQUIREMENT:** Client-Side Sentiment Analysis

Description

The client NLP module performs lightweight client-side sentiment analysis and language identification on attendee input before it is sent — **BECAUSE** filtering at the source reduces server load and can prevent improper submissions.

Router

Diagram



● COMPONENT	Properties & Description

Properties

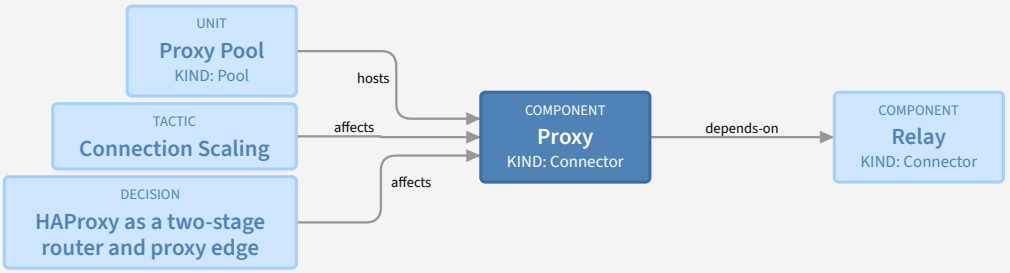
○ KIND	Connector

Description

The router (HAProxy with NFTables) routes incoming HTTP and WebSocket traffic across the proxy instances of an environment using round-robin and separates the dev, QA, and production environments — **BECAUSE** traffic must be balanced and environments isolated at the edge.

Proxy

Diagram



Properties

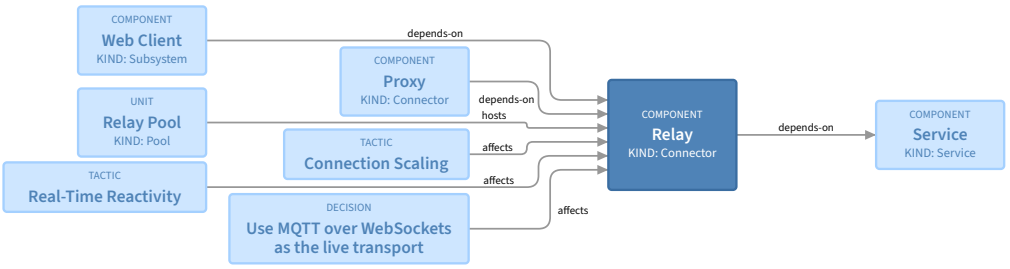
○ KIND	Connector
○ DEPENDS-ON	● COMPONENT: Relay

Description

The proxy layer (HAProxy) forwards the HTTP and WebSocket requests of a specific environment to the relay layer and scales horizontally per environment — **BECAUSE** request handling must scale independently of the messaging layer.

Relay

Diagram



Properties

○ KIND	Connector
○ DEPENDS-ON	● COMPONENT: Service

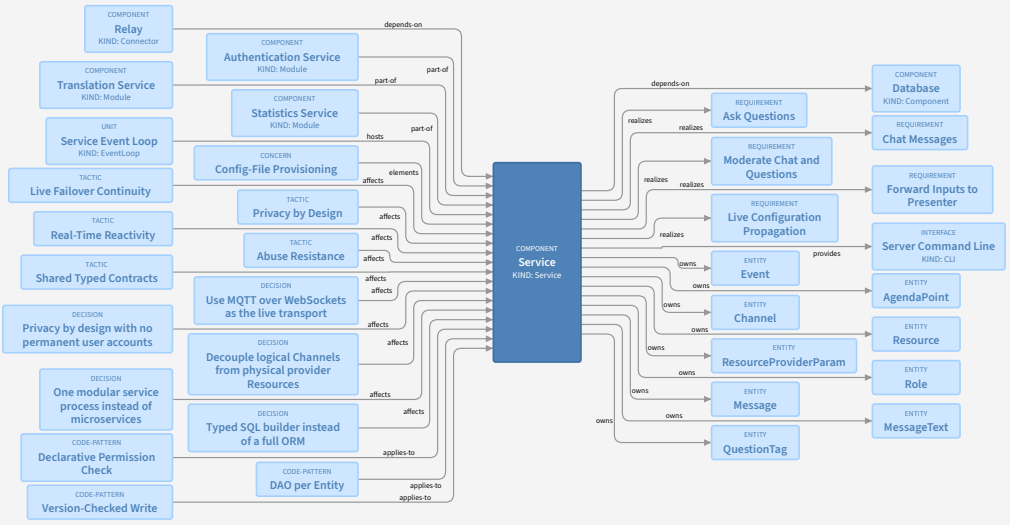
Description

The relay layer maintains thousands of bidirectional WebSocket/MQTT connections, brokering MQTT messages between clients and the service layer and scaling horizontally — **BECAUSE** real-time fan-out to up to 10000 attendees is the central performance challenge.

● COMPONENT	Properties & Description

Service ⚓

Diagram



Properties

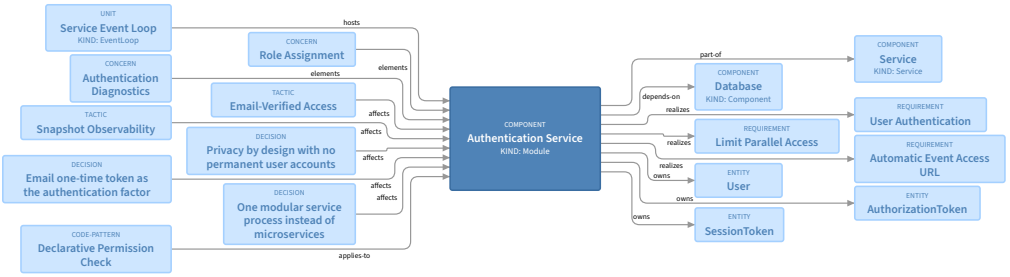
○ **KIND**	Service
○ **REALIZES**	● **REQUIREMENT:** Ask Questions ⚭, ● **REQUIREMENT:** Chat Messages ⚭, ● **REQUIREMENT:** Moderate Chat and Questions ⚭, ● **REQUIREMENT:** Forward Inputs to Presenter ⚭, ● **REQUIREMENT:** Live Configuration Propagation ⚭
○ **OWNS**	● **ENTITY:** Event ⚭, ● **ENTITY:** AgendaPoint ⚭, ● **ENTITY:** Channel ⚭, ● **ENTITY:** Resource ⚭, ● **ENTITY:** ResourceProviderParam ⚭, ● **ENTITY:** Role ⚭, ● **ENTITY:** Message ⚭, ● **ENTITY:** MessageText ⚭, ● **ENTITY:** QuestionTag ⚭
○ **PROVIDES**	● **INTERFACE:** Server Command Line ⚭
○ **DEPENDS-ON**	● **COMPONENT:** Database ⚭

Description

The service layer executes all business logic for events, messages, authentication, and statistics: as the main server it holds the event, moderation, access, and configuration logic, reacts to MQTT messages, and persists state — **BECAUSE** the authoritative business rules must live in a single server tier.

Authentication Service ⚓

Diagram



Properties

○ **KIND**	Module
○ **PART-OF**	● **COMPONENT:** Service ⚭
○ **REALIZES**	● **REQUIREMENT:** User Authentication ⚭, ● **REQUIREMENT:** Limit Parallel Access ⚭, ● **REQUIREMENT:** Automatic Event Access URL ⚭

● COMPONENT	Properties & Description
	○ OWNS: ● ENTITY: User, ● ENTITY: AuthorizationToken, ● ENTITY: SessionToken
	○ DEPENDS-ON: ● COMPONENT: Database

Description

The authentication module issues and validates the authorization and session tokens: it generates one-time tokens, sends them via the email gateway, validates logins, and enforces a single active session per user per event — **BECAUSE** email-verified, single-session access is the core security mechanism.

Translation Service

Diagram

Properties

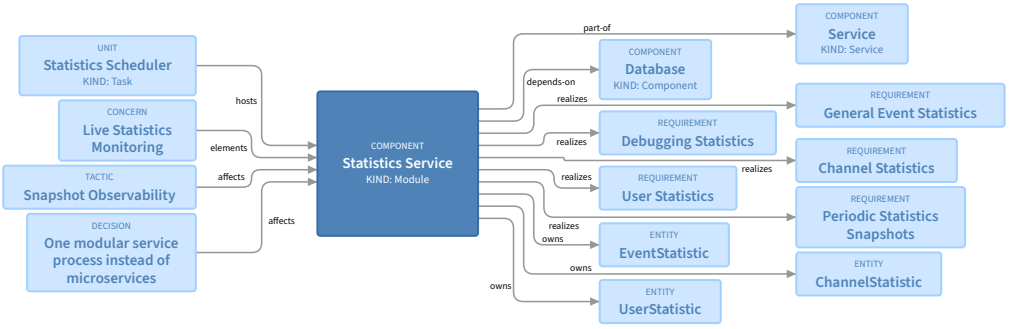
○ KIND	Module
○ PART-OF	● COMPONENT: Service
○ REALIZES	● REQUIREMENT: In-Session Language Switching

Description

The translation module produces language-specific message texts via an external LLM, translating them into the supported languages on the fly while preserving the original — **BECAUSE** chat and questions must be available in both German and English.

Statistics Service

Diagram



Properties

○ KIND	Module
○ PART-OF	● COMPONENT: Service
○ REALIZES	● REQUIREMENT: General Event Statistics, ● REQUIREMENT: Debugging Statistics, ● REQUIREMENT: Channel Statistics, ● REQUIREMENT: User Statistics, ● REQUIREMENT: Periodic Statistics Snapshots
○ OWNS	● ENTITY: EventStatistic, ● ENTITY: ChannelStatistic, ● ENTITY: UserStatistic
○ DEPENDS-ON	● COMPONENT: Database

Description

The statistics module generates periodic event, channel, and user statistics snapshots, capturing cumulative counts of tokens, sessions, connections, and viewers during a running event — **BECAUSE** trend dashboards require regular snapshots over the event's lifetime.

● COMPONENT	Properties & Description

Junction Orchestrator

Diagram

Properties

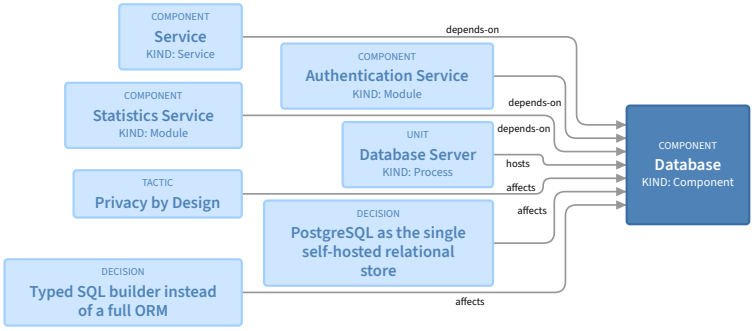
○ KIND	Component
○ REALIZES	● REQUIREMENT: Browser Access

Description

The Junction component serves the static client bundle and orchestrates its delivery to the relay over MQTT+ — **BECAUSE** the client must be distributed alongside the same messaging infrastructure used for live data.

Database

Diagram



Properties

○ KIND	Component

Description

The PostgreSQL database with the filesystem persists all business objects and static assets, storing all events, messages, tokens, and statistics accessed through the Drizzle persistence layer — **BECAUSE** the authoritative, durable state of every event must be persisted in one tier.

24 ● ARCH: Technology Stack (TS)

Common

- Language — PRODUCT: TypeScript
- Task Runner — PRODUCT: STX
- Linting — PRODUCT: ESLint
- Messaging — PRODUCT: MQTT-Plus
- HTTP Client — PRODUCT: OFetch
- Logging — PRODUCT: Pino
- Date Management — PRODUCT: Luxon
- Unique Identifier Generation — PRODUCT: nanoid

Client

- Build Tool — PRODUCT: Vite
- Styling Framework — PRODUCT: Tailwind CSS
- UI Framework — PRODUCT: Vue.js
- Widget Framework — PRODUCT: Reka UI
- Typography — PRODUCT: TypoPRO
- Iconography — PRODUCT: Fontawesome
- Tooltips — PRODUCT: vue-tippy + tippy.js
- Animations — PRODUCT: Anime.js
- Sentiment Analysis — PRODUCT: natural + @nlpjs/core + multilang-sentiment
- Language Identification — PRODUCT: tinyld + franc + lande

Middleware

- Reverse Proxy — PRODUCT: HAProxy
- Message Broker — PRODUCT: Mosquitto

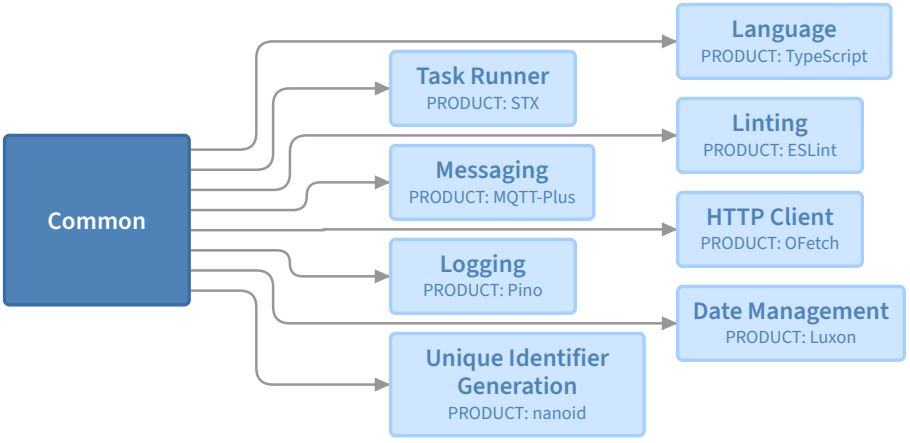
Server

- Runtime — PRODUCT: Node.js
- Persistence Layer — PRODUCT: Drizzle
- Static Content Server — PRODUCT: Junction
- Argument Parsing — PRODUCT: Commander.js
- Configuration Loading — PRODUCT: dotenvx + YAML
- Configuration Validation — PRODUCT: Valibot
- AI/LLM Connectivity — PRODUCT: @ai-sdk
- Container Orchestration — PRODUCT: Docker Compose

Database

- Database — PRODUCT: PostgreSQL
- Database Backup — PRODUCT: pg_dump

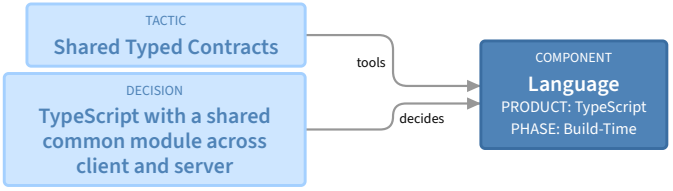
24.01 ● TIER: Common



● COMPONENT	Properties & Description

Language

Diagram



Properties

○ PRODUCT	TypeScript
○ LICENSE	Apache-2.0
○ PHASE	Build-Time
○ COVERAGE	Build, Standardization

Description

All client-side and server-side code, including the shared common module, is authored in strongly typed TypeScript and compiled to JavaScript — **BECAUSE** a single strongly typed language catches contract mismatches against the shared common module at compile time.

Task Runner

Properties

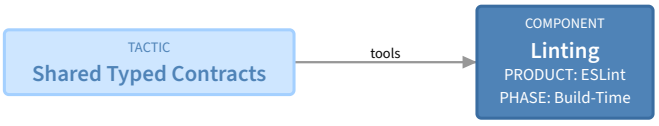
○ PRODUCT	STX
○ LICENSE	MIT
○ PHASE	Build-Time
○ COVERAGE	Tooling

Description

The STX task runner drives the build, lint, and run tasks of every module from a per-module `stx.conf` — **BECAUSE** a single task runner gives every module a consistent developer entry point.

Linting

Diagram



Properties

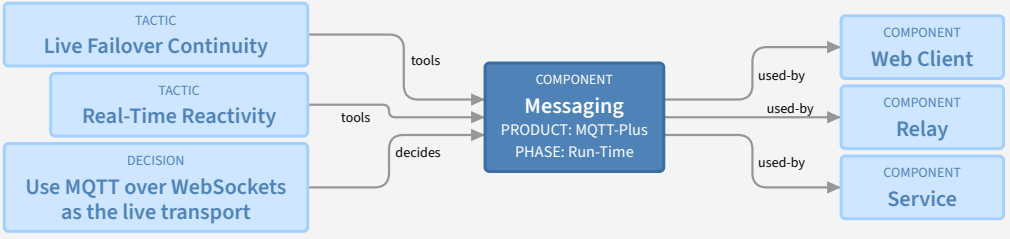
○ PRODUCT	ESLint
○ LICENSE	MIT
○ PHASE	Build-Time
○ COVERAGE	Standardization

Description

ESLint enforces code quality and style for the client and server TypeScript, complemented by the faster OxLint, HTMLHint for HTML, and StyleLint for Stylus — **BECAUSE** multiple focused linters keep each language layer clean and consistent.

Messaging

Diagram



● COMPONENT	Properties & Description

Properties

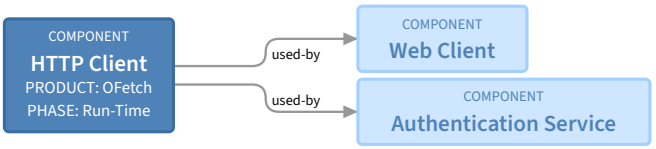
○ PRODUCT	MQTT-Plus
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Client Networking, Dialog Communication, Server Networking, Component Communication
○ USED-BY	● COMPONENT: Web Client, ● COMPONENT: Relay, ● COMPONENT: Service

Description

MQTT-Plus handles the MQTT-over-WebSocket communication of the client with the relay layer and the MQTT messaging of the service and relay layers on top of the base MQTT.js functionality — **BECAUSE** MQTT is the messaging protocol for the real-time relay layer and MQTT-Plus adds the needed request/response and event handling on all tiers alike.

HTTP Client

Diagram



Properties

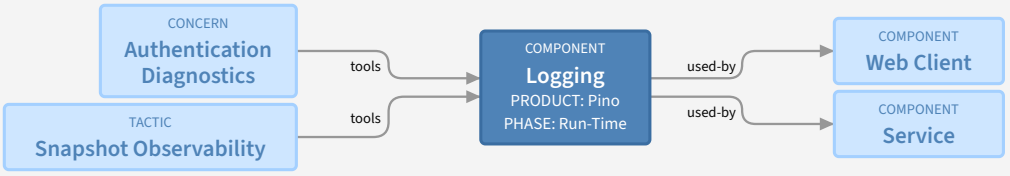
○ PRODUCT	OFetch
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Client Networking
○ USED-BY	● COMPONENT: Web Client, ● COMPONENT: Authentication Service

Description

OFetch provides the HTTP/REST request/response communication of the client and the call of the external GraphQL mail-sending API by the authentication service — **BECAUSE** REST calls complement the MQTT channel for request/response-style interactions on both tiers.

Logging

Diagram



Properties

○ PRODUCT	Pino
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Execution Tracing
○ USED-BY	● COMPONENT: Web Client, ● COMPONENT: Service

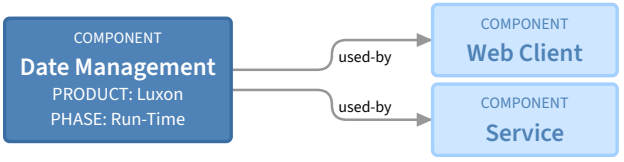
Description

Pino provides the leveled, structured logging facility, wrapping the browser console on the client and rendered human-readably by pino-pretty on the server during development — **BECAUSE** a single logging library shared by client and server keeps the tracing consistent.

● COMPONENT	Properties & Description

Date Management

Diagram



Properties

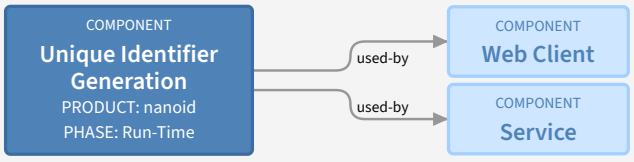
○ **PRODUCT**	Luxon
○ **LICENSE**	MIT
○ **PHASE**	Run-Time
○ **COVERAGE**	Data Conversion
○ **USED-BY**	● **COMPONENT:** Web Client, ● **COMPONENT:** Service

Description

Luxon handles date and time parsing, formatting, and localization across the client UI and the timestamps emitted by the server services — **BECAUSE** robust, locale-aware date handling is required for event schedules and timestamps in German and English on client and server alike.

Unique Identifier Generation

Diagram



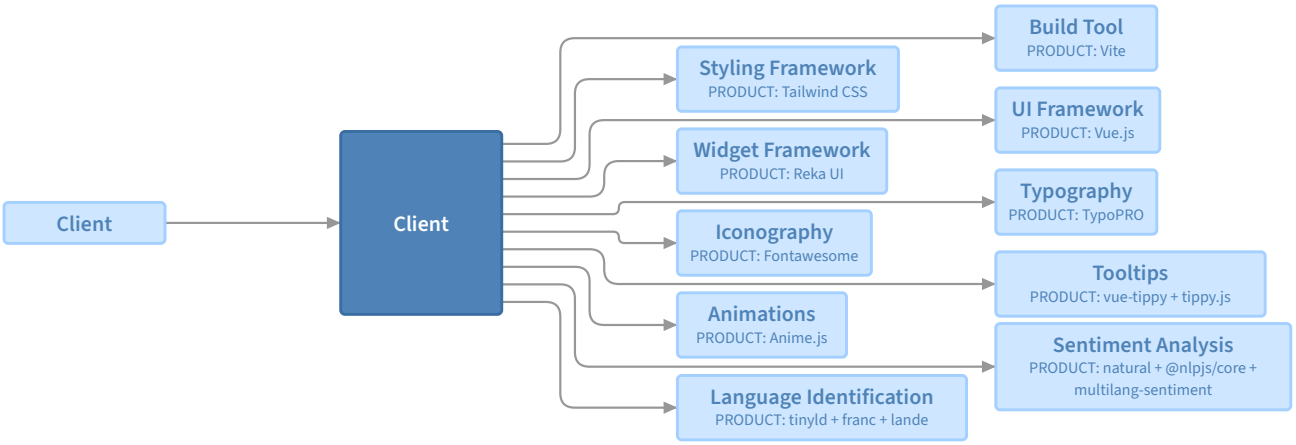
Properties

○ **PRODUCT**	nanoid
○ **LICENSE**	MIT
○ **PHASE**	Run-Time
○ **COVERAGE**	Business Model, Peer Information
○ **USED-BY**	● **COMPONENT:** Web Client, ● **COMPONENT:** Service

Description

nanoid generates the compact, collision-resistant unique identifiers for the messaging peers of client and server and for the entities the client creates — **BECAUSE** short URL-safe identifiers are needed on both tiers without the size and overhead of full UUIDs.

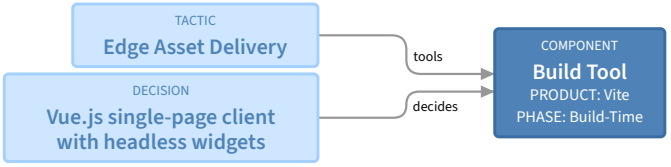
24.02 ● TIER: Client



● COMPONENT	Properties & Description

Build Tool

Diagram



Properties

○ PRODUCT	Vite
○ LICENSE	MIT
○ PHASE	Build-Time
○ COVERAGE	Build

Description

Vite serves the client in development with fast HMR and bundles the runtime stack into the optimized static client artifact — **BECAUSE** it provides a fast development server and an optimized production bundle for the Vue stack.

Styling Framework

Diagram

Properties

○ PRODUCT	Tailwind CSS
○ LICENSE	MIT
○ PHASE	Build-Time, Run-Time
○ COVERAGE	Build, Interface Theme, Interface Layouting, Interface Effects
○ USED-BY	● COMPONENT: Web Client

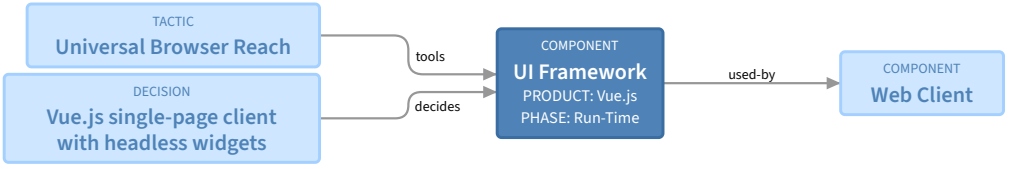
Description

Tailwind CSS generates the utility-first styling for theme, layout, and effects at build time, with Stylus as the preferred preprocessor for authoring custom CSS — **BECAUSE** utility-first styling speeds consistent UI work while Stylus covers bespoke styling.

● COMPONENT	Properties & Description

UI Framework

Diagram



Properties

○ PRODUCT	Vue.js
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Dialog Structure · Dialog Life-Cycle · Interface Mask · Data Binding
○ USED-BY	● COMPONENT: Web Client

Description

Vue.js provides the reactive, component-based UI framework rendering the panel, attendee, studio, moderator, and manager screens — **BECAUSE** a reactive component model is the natural fit for the browser-delivered, real-time experience.

Widget Framework

Diagram

Properties

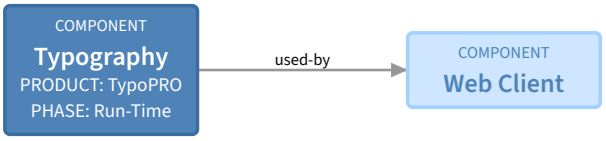
○ PRODUCT	Reka UI
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Interface Widgets · Interface States · Interface Interactions
○ USED-BY	● COMPONENT: Web Client

Description

Reka UI supplies the headless, unstyled widget primitives composed into the application's interface elements — **BECAUSE** headless widgets provide accessible interaction behavior while leaving the visual styling fully under design control.

Typography

Diagram



Properties

○ PRODUCT	TypoPRO
○ LICENSE	OFL-1.1, Apache-2.0, MIT, CC0-1.0 (per font)
○ PHASE	Run-Time
○ COVERAGE	Interface Theme
○ USED-BY	● COMPONENT: Web Client

Description

TypoPRO supplies the web typography and font assets for the client interface — **BECAUSE** consistent, self-hosted typography is needed across the branded UI.

● COMPONENT	Properties & Description

Iconography

Diagram

Properties

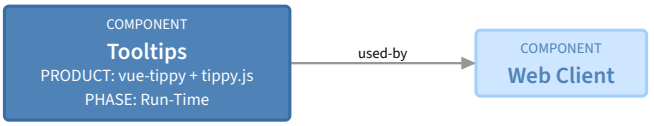
○ **PRODUCT**	Fontawesome
○ **LICENSE**	CC-BY-4.0, OFL-1.1, MIT (per asset)
○ **PHASE**	Run-Time
○ **COVERAGE**	Interface Widgets
○ **USED-BY**	● **COMPONENT:** **Web Client**

Description

Fontawesome provides the icon set used throughout the client interface — **BECAUSE** a comprehensive icon library covers the UI's iconography needs out of the box.

Tooltips

Diagram



Properties

○ **PRODUCT**	vue-tippy + tippy.js
○ **LICENSE**	MIT
○ **PHASE**	Run-Time
○ **COVERAGE**	Interface Widgets, Interface States
○ **USED-BY**	● **COMPONENT:** **Web Client**

Description

vue-tippy integrates the tippy.js tooltip and popover engine as a Vue.js directive and component for the floating hints on the interface elements of the client — **BECAUSE** a proven positioning engine with a Vue.js integration provides consistent tooltips without hand-written placement logic.

Animations

Diagram

Properties

○ **PRODUCT**	Anime.js
○ **LICENSE**	MIT
○ **PHASE**	Run-Time
○ **COVERAGE**	Interface Effects
○ **USED-BY**	● **COMPONENT:** **Web Client**

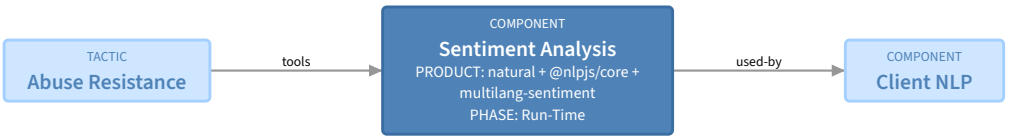
Description

Anime.js drives the scripted transitions and keyframe animations of the client interface elements beyond the declarative CSS transitions of the styling framework — **BECAUSE** timeline-based animations with easing need a JavaScript animation engine which CSS alone does not provide.

● COMPONENT	Properties & Description

Sentiment Analysis

Diagram



Properties

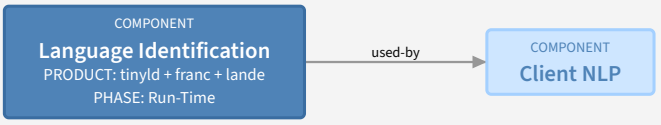
○ PRODUCT	natural + @nlpjs/core + multilang-sentiment
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Presentation Model
○ USED-BY	● COMPONENT: Client NLP

Description

The natural, @nlpjs/core, and multilang-sentiment libraries perform local sentiment analysis on attendee input in the browser — **BECAUSE** filtering sentiment at the source reduces server load and can prevent improper submissions.

Language Identification

Diagram



Properties

○ PRODUCT	tinyld + franc + lande
○ PHASE	Run-Time
○ COVERAGE	Presentation Model
○ USED-BY	● COMPONENT: Client NLP

Description

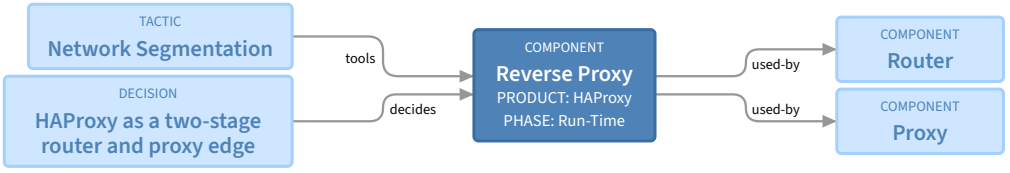
The tinyld, franc, and lande libraries perform local language identification on attendee input in the browser — **BECAUSE** detecting the message language client-side enables on-the-fly handling without an initial server round-trip.

24.03 ● TIER: Middleware

● COMPONENT	Properties & Description

Reverse Proxy

Diagram



● COMPONENT	Properties & Description

Properties

○ PRODUCT	HAProxy
○ LICENSE	GPL-2.0
○ PHASE	Run-Time
○ COVERAGE	Request Routing, Load Balancing, Transport Security
○ USED-BY	● COMPONENT: Router, ● COMPONENT: Proxy

Description

HAProxy terminates the inbound HTTP and WebSocket traffic at the edge router, firewalled by NFTables, and forwards it per environment from the proxy instances to the relay layer — **BECAUSE** a single mature proxy covers both the hardened entry point and the horizontally scaled per-environment forwarding.

Message Broker

Diagram

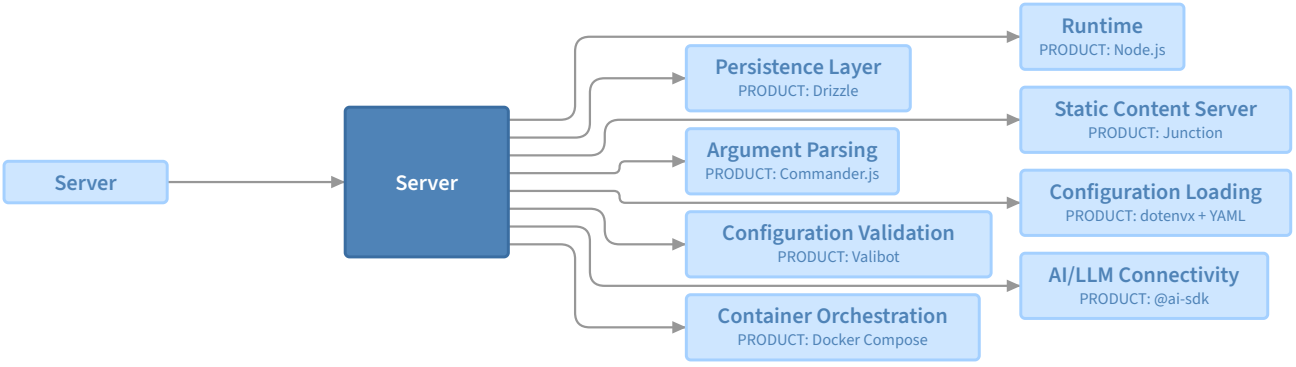
Properties

○ PRODUCT	Mosquitto
○ LICENSE	EPL-2.0
○ PHASE	Run-Time
○ COVERAGE	Message Brokering
○ USED-BY	● COMPONENT: Relay

Description

Mosquitto brokers the MQTT messages between the thousands of WebSocket connections of the clients and the service layer as a pool of instances per environment — **BECAUSE** a lightweight, proven MQTT broker sustains the connection count of a large event without business logic of its own.

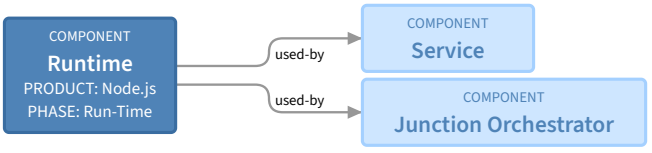
24.04 ● TIER: Server



● COMPONENT	Properties & Description

Runtime

Diagram



● COMPONENT	Properties & Description

Properties

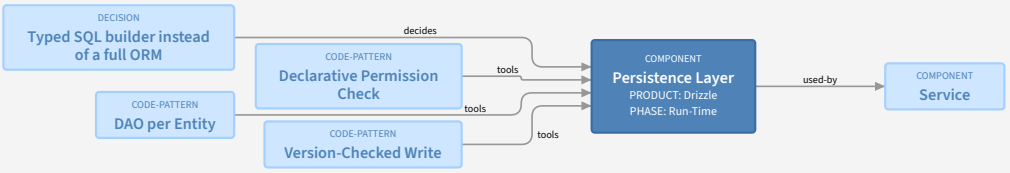
○ PRODUCT	Node.js
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Environment Detection, Process Management, Server Networking
○ USED-BY	● COMPONENT: Service, ● COMPONENT: Junction Orchestrator

Description

Node.js executes the business services and the Junction orchestrator as single-threaded event loops in the service containers — **BECAUSE** the message-driven service logic is I/O-bound and an event loop handles it with the least concurrency complexity.

Persistence Layer

Diagram



Properties

○ PRODUCT	Drizzle
○ ALTERNATIVES	Prisma, TypeORM
○ LICENSE	Apache-2.0
○ PHASE	Run-Time
○ COVERAGE	Database Access, Database Connectivity, Database Schema
○ USED-BY	● COMPONENT: Service

Description

Drizzle provides the type-safe SQL query builder and schema definition layer between the server's TypeScript code and PostgreSQL, with its schema generated spec-first from the data model — **BECAUSE** a thin, zero-overhead typed layer preserves end-to-end TypeScript type-safety and full PostgreSQL feature access without the runtime cost of a heavy ORM.

Static Content Server

Diagram

Properties

○ PRODUCT	Junction
○ PHASE	Run-Time
○ COVERAGE	Server Networking, Request Processing
○ USED-BY	● COMPONENT: Junction Orchestrator

Description

Junction serves the static client content over MQTT+ from the backend — **BECAUSE** the client must be distributed alongside the same messaging infrastructure used for live data.

● COMPONENT	Properties & Description

Argument Parsing

Diagram

Properties

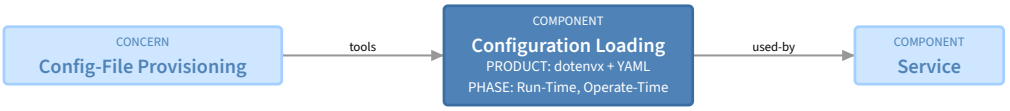
○ PRODUCT	Commander.js
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Argument Parsing
○ USED-BY	● COMPONENT: Service

Description

Commander.js parses the server's command-line options and arguments to bootstrap application parameters — **BECAUSE** a mature CLI parser covers option handling at server startup without custom code.

Configuration Loading

Diagram



Properties

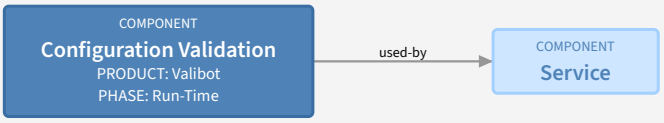
○ PRODUCT	dotenvx + YAML
○ LICENSE	BSD-3-Clause, ISC
○ PHASE	Run-Time, Operate-Time
○ COVERAGE	Configuration Parsing, Configuration
○ USED-BY	● COMPONENT: Service

Description

dotenvx loads optional environment files into the process environment and the YAML library parses the optional configuration file, together feeding the layered configuration resolved from defaults, file, environment, and command-line — **BECAUSE** the server's runtime parameters and secrets must be configurable from files and the `BROADCAST_*` environment without hardcoding them.

Configuration Validation

Diagram



Properties

○ PRODUCT	Valibot
○ LICENSE	MIT
○ PHASE	Run-Time
○ COVERAGE	Configuration Parsing
○ USED-BY	● COMPONENT: Service

Description

Valibot declares the canonical, strict configuration schema that validates and coerces the merged configuration at startup — **BECAUSE** a single schema as the source of truth gives fail-fast validation of secrets and a typed configuration surface that catches misconfiguration before the service runs.

● COMPONENT	Properties & Description

AI/LLM Connectivity ⚓

Diagram

Properties

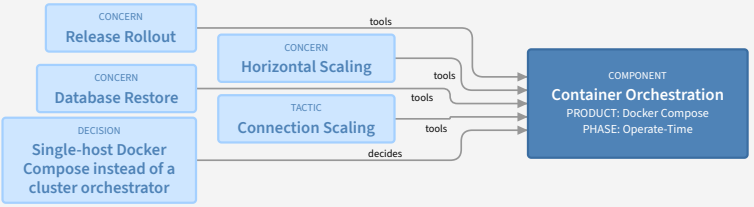
○ **PRODUCT**	@ai-sdk
○ **LICENSE**	Apache-2.0
○ **PHASE**	Run-Time
○ **COVERAGE**	Client Networking
○ **USED-BY**	● **COMPONENT:** Translation Service

Description

The @ai-sdk library connects the translation service to an external AI/LLM service for on-the-fly translation of message texts — **BECAUSE** chat and questions must be made available in both German and English.

Container Orchestration ⚓

Diagram



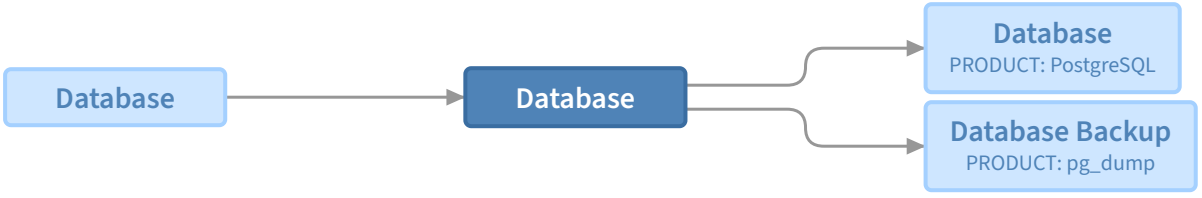
Properties

○ **PRODUCT**	Docker Compose
○ **ALTERNATIVES**	Kubernetes, Nomad
○ **LICENSE**	Apache-2.0
○ **PHASE**	Operate-Time
○ **COVERAGE**	Installation, Upgrade, Capacity

Description

Docker Compose runs and recreates the service containers of an environment from a per-environment compose file, with the proxy and relay instances added by the same file — **BECAUSE** a single-host orchestration is sufficient for the per-environment container count and avoids operating a cluster.

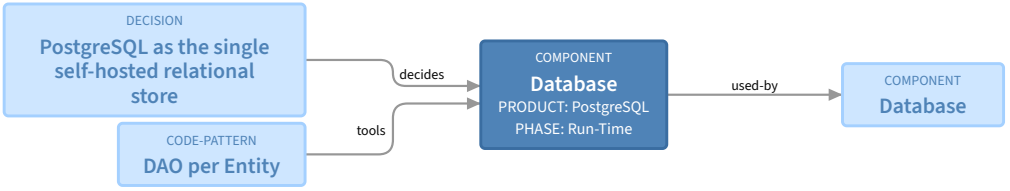
24.05 ● TIER: Database ⚓



● COMPONENT	Properties & Description

Database

Diagram



Properties

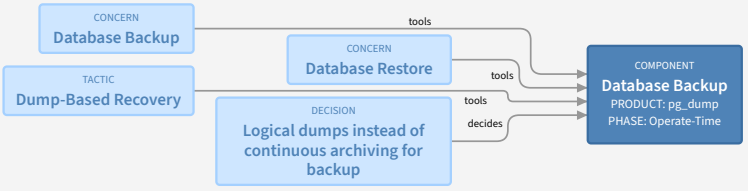
○ **PRODUCT**	PostgreSQL
○ **LICENSE**	PostgreSQL License
○ **PHASE**	Run-Time
○ **COVERAGE**	Storage Engine · Transaction Management · Query Language · Indexing
○ **USED-BY**	● **COMPONENT:** Database

Description

PostgreSQL stores the authoritative, durable state of every event together with its channels, messages, tokens, and statistics snapshots — **BECAUSE** the event-centric data model is inherently relational and demands transactional integrity, EU-resident self-hosting, and rich query support for moderation and reporting.

Database Backup

Diagram



Properties

○ **PRODUCT**	pg_dump
○ **ALTERNATIVES**	pgBackRest, Barman
○ **LICENSE**	PostgreSQL License
○ **PHASE**	Operate-Time
○ **COVERAGE**	Backup · Recovery

Description

The pg_dump utility of PostgreSQL produces the logical database dumps stored together with the filesystem assets on the backup storage and restored with pg_restore — **BECAUSE** a logical dump is portable across PostgreSQL versions and needs no additional backup server.

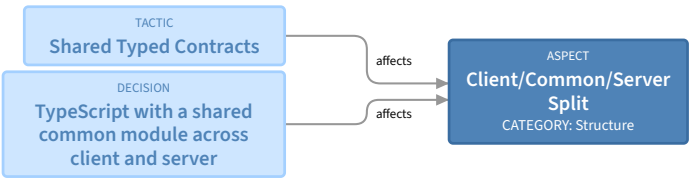
25 ● ARCH: Development Aspects (DA)

Client/Common/Server Split CATEGORY: Structure	TypeScript Everywhere CATEGORY: Standardization	STX Task Running CATEGORY: Tooling	Vite Client Build CATEGORY: Build
Layered Linting CATEGORY: Standardization	Dependency Layering CATEGORY: Dependency	Git Codeline CATEGORY: Codeline	Identifier Naming CATEGORY: Standardization

● ASPECT	Properties & Description

Client/Common/Server Split ⚓

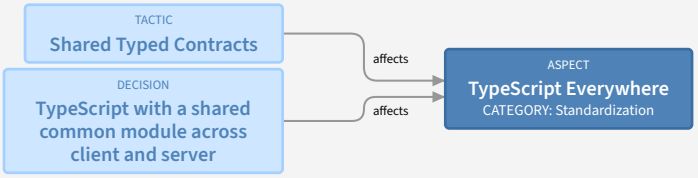
Diagram



○ CATEGORY	Description
Structure	The codebase is organized into `src/client`, `src/common`, and `src/server` modules, with shared types and topic definitions living in the common module imported by both sides — **BECAUSE** client and server must agree on contracts without duplicating them.

TypeScript Everywhere ⚓

Diagram



○ CATEGORY	Description
Standardization	All client-side and server-side code is written in strongly typed TypeScript against shared type definitions — **BECAUSE** a single typed language across the stack catches contract mismatches at compile time.

STX Task Running ⚓

○ CATEGORY	Description
Tooling	Build, lint, and run tasks are driven uniformly by the STX task runner with per-module `stx.conf` files — **BECAUSE** a single task runner gives every module a consistent developer entry point.

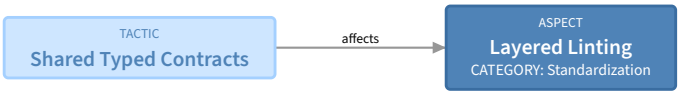
Vite Client Build ⚓

Diagram

○ CATEGORY	Description
Build	The client is built and served in development by Vite, bundling the Vue, Pinia, Reka UI, Tailwind, and Stylus stack into the static client artifact — **BECAUSE** Vite provides fast HMR development and an optimized production bundle.

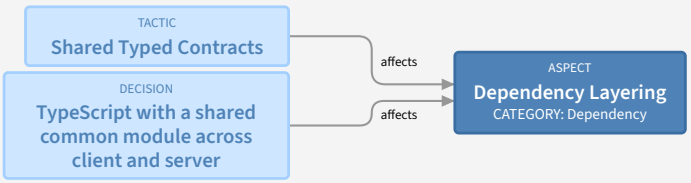
Layered Linting ⚓

Diagram



○ CATEGORY	Description
Standardization	Code quality is enforced with ESLint and the faster OxLint for TypeScript, HTMLHint for HTML, and StyleLint for Stylus — **BECAUSE** multiple focused linters keep each language layer clean and consistent.

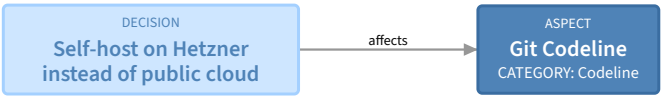
● ASPECT	Properties & Description
Dependency Layering ⚓	Diagram



○ CATEGORY	Description
Dependency	The common module must depend on neither client nor server, while both client and server may depend on common — **BECAUSE** a one-directional dependency keeps the shared contract free of platform-specific concerns.

Git Codeline ⚓

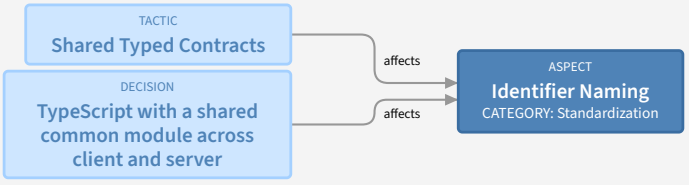
Diagram



○ CATEGORY	Description
Codeline	The project is versioned in a single Git repository hosted on the self-administered Gitea, with the Software Administrator granting repository access — **BECAUSE** self-hosted version control keeps source under full organizational control.

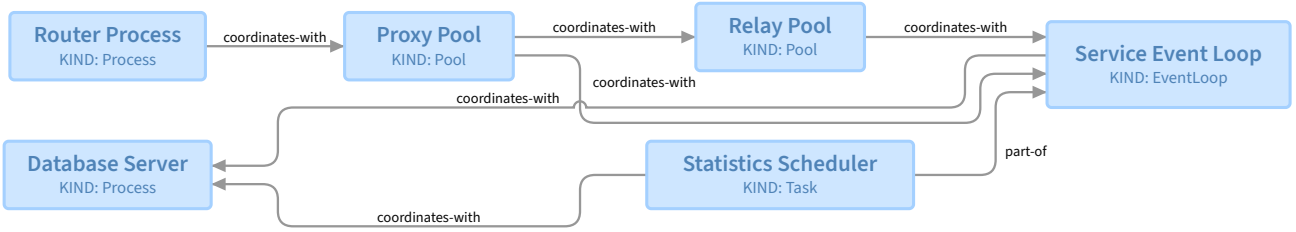
Identifier Naming ⚓

Diagram



○ CATEGORY	Description
Standardization	Data-model entities are named in PascalCase and their attributes in camelCase, and both are referred to by exactly these identifiers, set in code font, wherever code or documents mention them (e.g. `Event.accessList`, `Message.senderName`) — **BECAUSE** a single identifier form lets a reader grep from a document straight into the code.

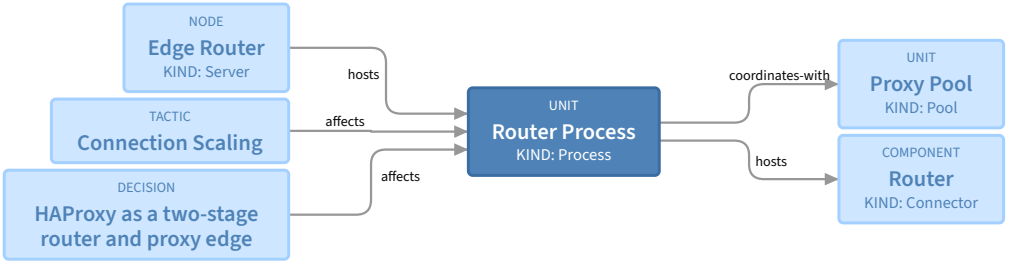
26 ● ARCH: Concurrency View (CO) ⚓



● UNIT	Properties & Description

Router Process

Diagram



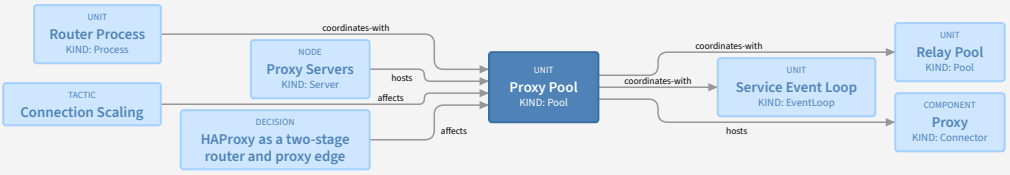
○ KIND	○ PART-OF	○ HOSTS
Process	∅	● **COMPONENT: Router**
○ MULTIPLICITY	**○ COORDINATES-WITH**	**○ COORDINATION**
1 per data center entry point	● **UNIT: Proxy Pool**	stateless round-robin distribution

Description

The router runs as a single front-door process per data center, distributing connections across environments and proxy instances — **BECAUSE** a stateless balancer can spread load without holding per-connection state.

Proxy Pool

Diagram



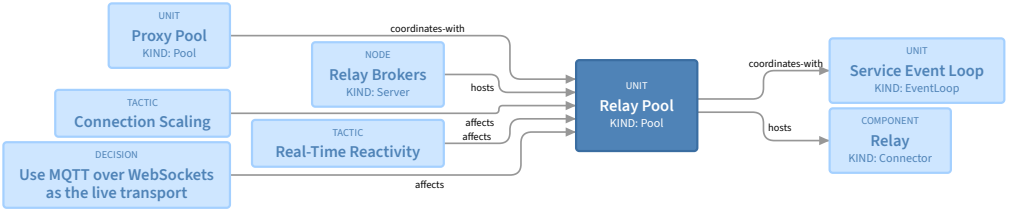
○ KIND	○ PART-OF	○ HOSTS
Pool	∅	● **COMPONENT: Proxy**
○ MULTIPLICITY	**○ COORDINATES-WITH**	**○ COORDINATION**
0..n instances per environment	● **UNIT: Relay Pool**, ● **UNIT: Service Event Loop**	shared-nothing behind the router

Description

The proxy layer runs as a pool of independent, stateless reverse-proxy instances per environment — **BECAUSE** request handling must scale out horizontally to absorb connection spikes.

Relay Pool

Diagram



○ KIND	○ PART-OF	○ HOSTS
Pool	∅	● **COMPONENT: Relay**
○ MULTIPLICITY	**○ COORDINATES-WITH**	**○ COORDINATION**
0..n MQTT broker instances per environment	● **UNIT: Service Event Loop**	MQTT topic subscriptions and broker bridging

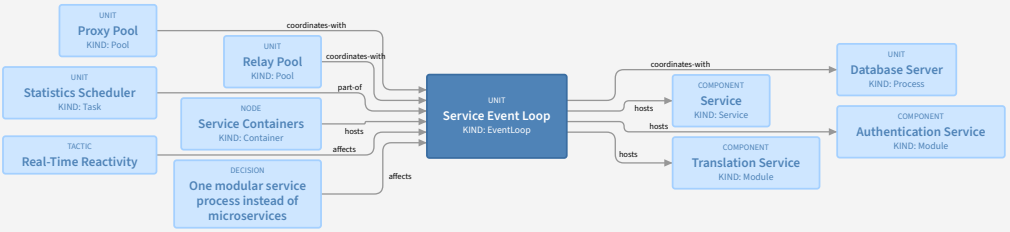
● UNIT	Properties & Description

Description

The relay layer runs as a pool of Mosquitto/MQTT-Plus broker instances, each maintaining thousands of WebSocket connections and fanning out per-event topics — **BECAUSE** sustaining up to 10000 concurrent connections requires horizontal broker scaling.

Service Event Loop

Diagram



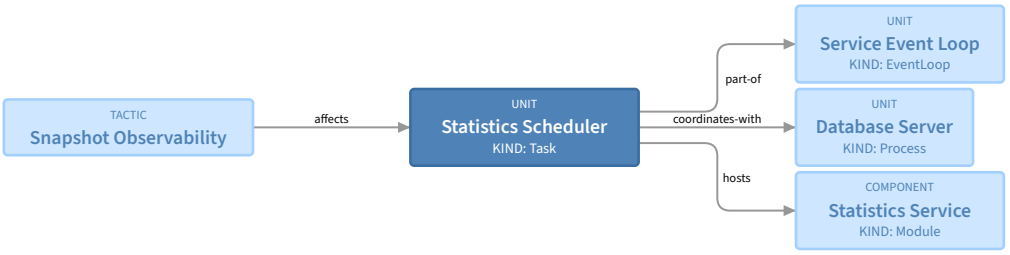
○ KIND	○ PART-OF	○ HOSTS
EventLoop	∅	● **COMPONENT:** Service ⚭, ● **COMPONENT:** Authentication Service ⚭, ● **COMPONENT:** Translation Service ⚭
○ MULTIPLICITY	**○ COORDINATES-WITH**	**○ COORDINATION**
0..n Node.js instances per environment	● **UNIT:** Database Server ⚭	MQTT message passing and database transactions

Description

The service runs as Node.js single-threaded event loops that react to MQTT messages and serialize state changes through database transactions — **BECAUSE** an event-loop model handles many concurrent I/O-bound interactions without shared-memory races.

Statistics Scheduler

Diagram



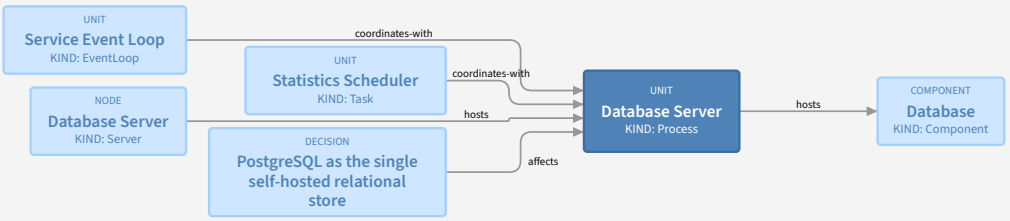
○ KIND	○ PART-OF	○ HOSTS
Task	● **UNIT:** Service Event Loop ⚭	● **COMPONENT:** Statistics Service ⚭
○ MULTIPLICITY	**○ COORDINATES-WITH**	**○ COORDINATION**
1 active per running event	● **UNIT:** Database Server ⚭	timer-driven, writes via database transactions

Description

A scheduled task on the service event loop captures a statistics snapshot every five minutes for each running event — **BECAUSE** periodic cumulative counts must be produced on a fixed cadence independent of user activity.

● UNIT	Properties & Description
Database Server ⚓	

Diagram

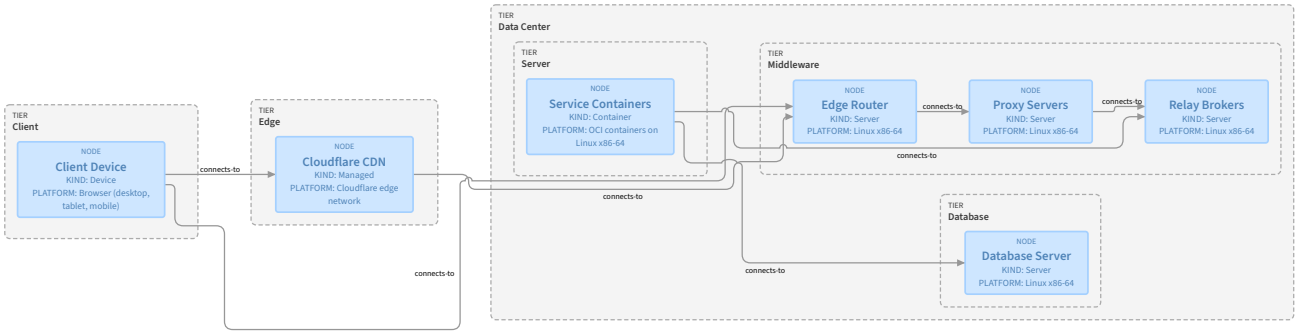


○ KIND	○ PART-OF	○ HOSTS
Process	∅	● **COMPONENT: Database** ⚭
○ MULTIPLICITY	**○ COORDINATES-WITH**	**○ COORDINATION**
1 primary per environment	∅	ACID transactions with optimistic concurrency via the ORM

Description

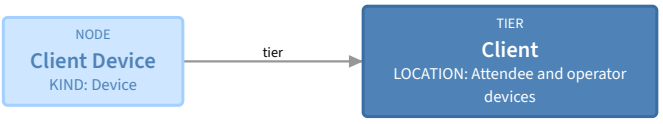
PostgreSQL runs as the single authoritative persistence process per environment, serializing concurrent writes through transactions — **BECAUSE** a single source of truth with ACID guarantees keeps event state consistent under concurrency.

27 ● ARCH: Deployment View (DV) ⚓



● TIER	Properties & Description
Client ⚓	

Diagram



○ PART-OF	○ LOCATION	○ STACK
∅	Attendee and operator devices	● **TIER: Client** ⚭

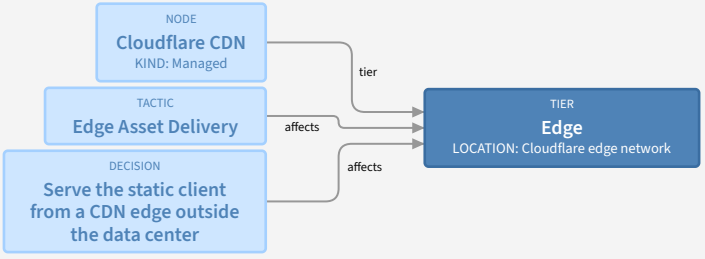
Description

The client tier is the web client running on the devices of the audience and the operators — **BECAUSE** the interactive part of the solution must run where the people are, without installation.

● TIER	Properties & Description

Edge

Diagram



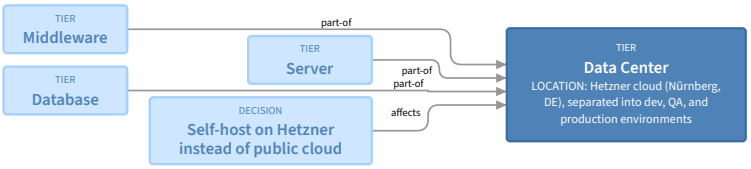
○ PART-OF	○ LOCATION	○ STACK
∅	Cloudflare edge network	∅

Description

The edge tier delivers the static client bundle from a global content delivery network — **BECAUSE** static content is best served close to a large, distributed audience instead of from the data center.

Data Center

Diagram



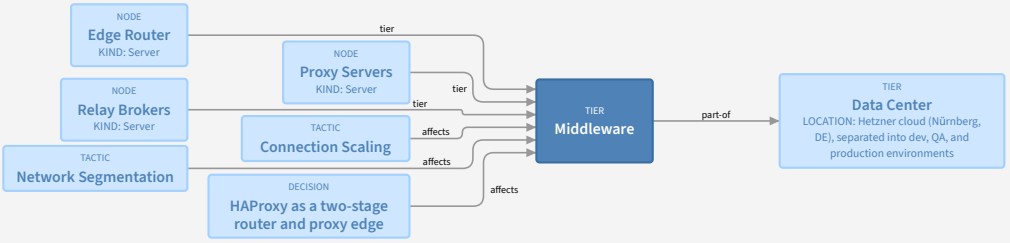
○ PART-OF	○ LOCATION	○ STACK
∅	Hetzner cloud (Nürnberg, DE), separated into dev, QA, and production environments	∅

Description

The data center tier groups all server-side tiers within one EU-resident hosting location — **BECAUSE** EU-resident self-hosting satisfies GDPR and minimizes per-event cost versus public cloud.

Middleware

Diagram



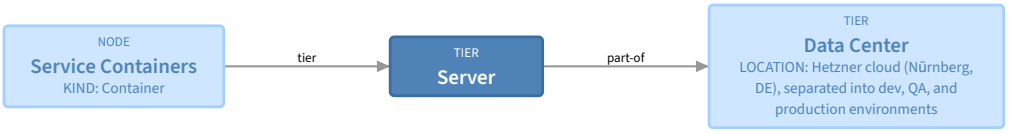
○ PART-OF	○ LOCATION	○ STACK
● **TIER:** Data Center	∅	● **TIER:** Middleware

Description

The middleware tier terminates, balances, and relays the client connections between the Internet and the business services — **BECAUSE** sustaining thousands of bidirectional connections is a concern separate from the business logic.

Server

Diagram



○ PART-OF	○ LOCATION	○ STACK
● **TIER:** Data Center	∅	● **TIER:** Server

● TIER

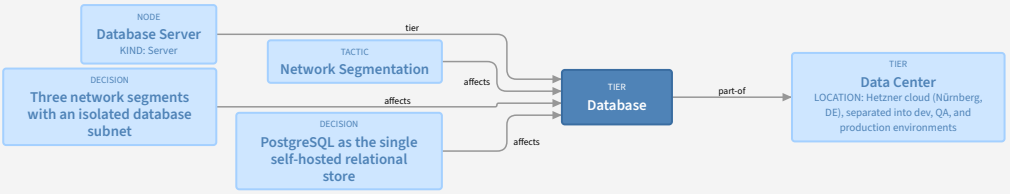
Properties & Description

Description

The server tier runs the business services and the orchestrator — **BECAUSE** the business logic must scale and change independently of the connection handling and the persistence.

Database

Diagram



○ PART-OF	○ LOCATION	○ STACK
● **TIER:** Data Center	∅	● **TIER:** Database

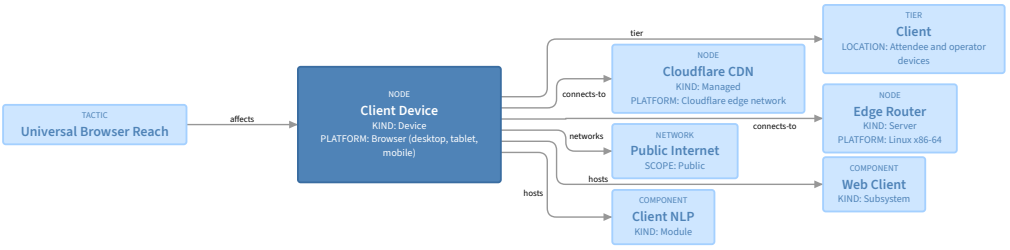
Description

The database tier holds the authoritative persistent state — **BECAUSE** durable state must reside on a protected tier reachable from the business services alone.

● NODE

Client Device

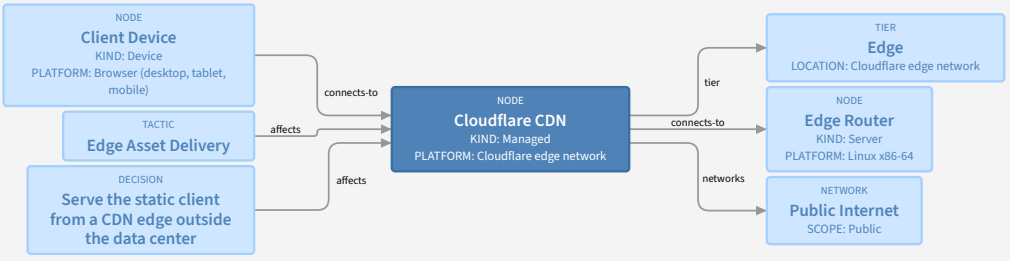
Diagram



○ KIND	○ TIER	○ PLATFORM
Device	● **TIER:** Client	Browser (desktop, tablet, mobile)
○ HOSTS	**○ INSTANCES**	**○ CONNECTS-TO**
● **COMPONENT:** Web Client, ● **COMPONENT:** Client NLP	1 per attendee or operator	● **NODE:** Cloudflare CDN, ● **NODE:** Edge Router
○ NETWORKS	Description	
● **NETWORK:** Public Internet	The attendee and operator devices run the web client and local NLP entirely in the browser — **BECAUSE** the solution requires no installation and must run on both managed and unmanaged devices.	

Cloudflare CDN

Diagram

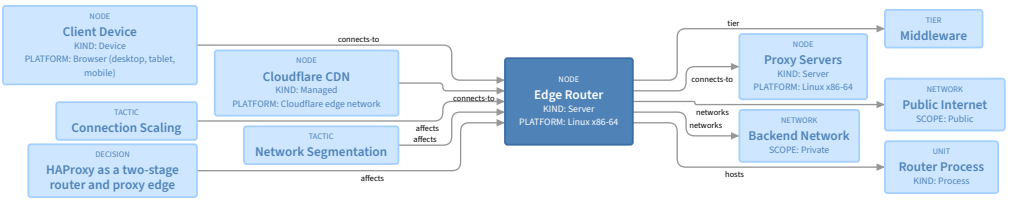


○ KIND	○ TIER	○ PLATFORM
Managed	● **TIER:** Edge	Cloudflare edge network

● NODE	Properties & Description		
	○ HOSTS	○ INSTANCES	○ CONNECTS-TO
	∅	∅	● NODE: Edge Router
	○ NETWORKS	Description	
	● NETWORK: Public Internet	A Cloudflare CDN edge distributes the static client bundle and static resources close to users — **BECAUSE** static content must be delivered quickly and stably to a large, distributed audience.	

Edge Router

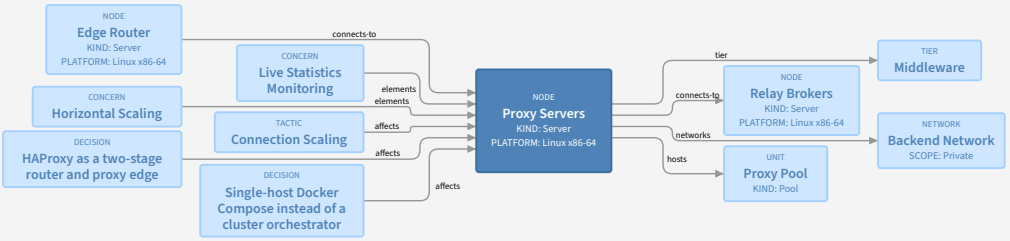
Diagram



○ KIND	○ TIER	○ PLATFORM
Server	● TIER: Middleware	Linux x86-64
○ HOSTS	○ INSTANCES	○ CONNECTS-TO
● UNIT: Router Process	1 per data center entry point	● NODE: Proxy Servers
○ NETWORKS	Description	
● NETWORK: Public Internet, ● NETWORK: Backend Network	The edge router terminates inbound HTTP and WebSocket traffic and round-robins it to per-environment proxies — **BECAUSE** a single hardened entry point must balance load and isolate environments.	

Proxy Servers

Diagram



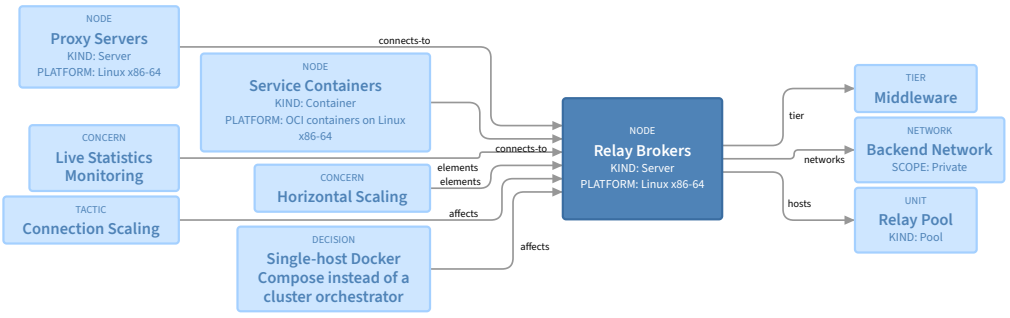
○ KIND	○ TIER	○ PLATFORM
Server	● TIER: Middleware	Linux x86-64
○ HOSTS	○ INSTANCES	○ CONNECTS-TO
● UNIT: Proxy Pool	0..n per environment	● NODE: Relay Brokers
○ NETWORKS	Description	
● NETWORK: Backend Network	The proxy servers run several reverse proxy instances per environment forwarding to the relay brokers — **BECAUSE** per-environment scaling of request handling needs multiple proxy instances.	

● NODE

Relay Brokers ⚓

Properties & Description

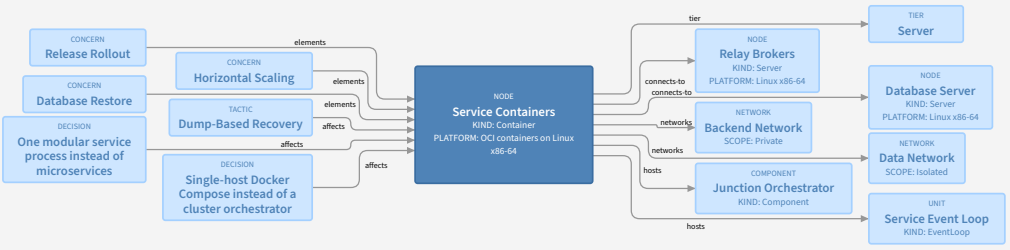
Diagram



○ KIND	○ TIER	○ PLATFORM
Server	● **TIER: Middleware** ⚭	Linux x86-64
○ HOSTS	**○ INSTANCES**	**○ CONNECTS-TO**
● **UNIT: Relay Pool** ⚭	0..n per environment	∅
○ NETWORKS	Description	
● **NETWORK: Backend Network** ⚭	The relay brokers run several MQTT broker instances per environment holding the live WebSocket connections — **BECAUSE** sustaining thousands of bidirectional connections requires a dedicated, scalable messaging node.	

Service Containers ⚓

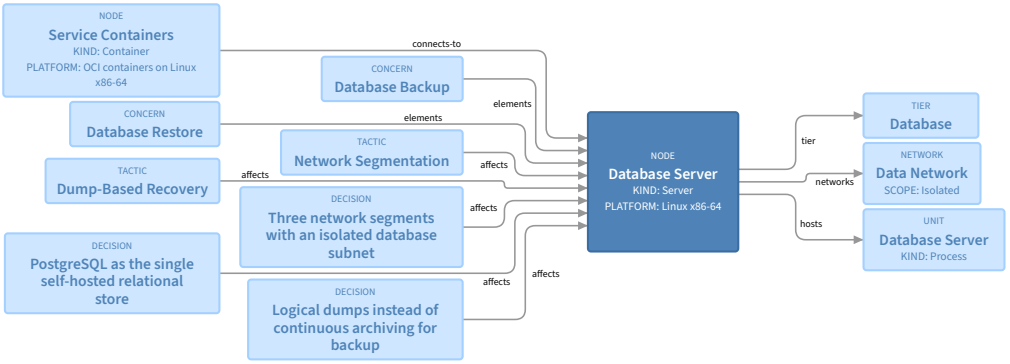
Diagram



○ KIND	○ TIER	○ PLATFORM
Container	● **TIER: Server** ⚭	OCI containers on Linux x86-64
○ HOSTS	**○ INSTANCES**	**○ CONNECTS-TO**
● **UNIT: Service Event Loop** ⚭, ● **COMPONENT: Junction Orchestrator** ⚭	0..n per environment	● **NODE: Relay Brokers** ⚭, ● **NODE: Database Server** ⚭
○ NETWORKS	Description	
● **NETWORK: Backend Network** ⚭, ● **NETWORK: Data Network** ⚭	The service containers run the Node.js business services and the Junction orchestrator per environment — **BECAUSE** containerized services give reproducible, independently scalable business logic.	

Database Server ⚓

Diagram



● NODE

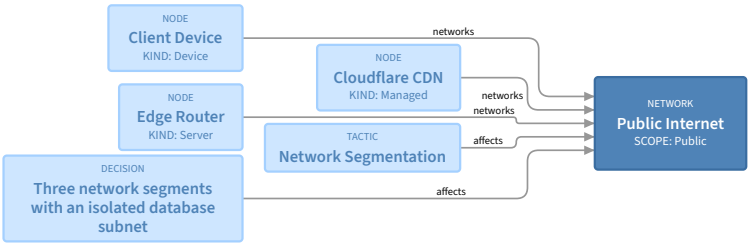
Properties & Description

○ KIND	○ TIER	○ PLATFORM
Server	● **TIER: Database**	Linux x86-64
○ HOSTS	**○ INSTANCES**	**○ CONNECTS-TO**
● **UNIT: Database Server**	1 primary per environment	∅
○ NETWORKS	Description	
● **NETWORK: Data Network**	The database server runs PostgreSQL with filesystem asset storage as the authoritative persistence node — **BECAUSE** durable state must reside on a protected, non-public node.	

● NETWORK

Public Internet

Diagram



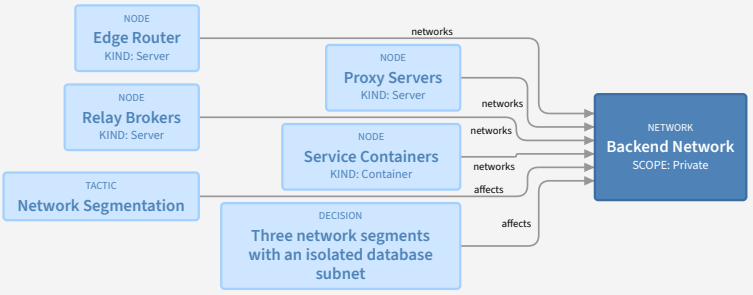
○ SCOPE	○ PROTOCOLS	○ ADDRESSING
Public	HTTPS 443, WSS 443	Public IPv4 and IPv6

Description

The public Internet carries all attendee and operator traffic to the CDN and the edge router over TLS only — **BECAUSE** the audience joins from arbitrary, untrusted networks.

Backend Network

Diagram



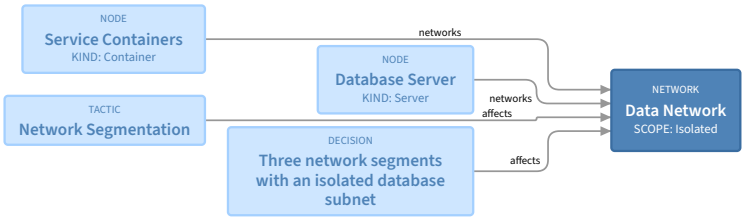
○ SCOPE	○ PROTOCOLS	○ ADDRESSING
Private	HTTP, WS, WSS	Hetzner private vSwitch, per-environment VLAN

Description

The backend network connects the router, the proxies, the relays, and the service containers within the data center — **BECAUSE** the internal hops must stay unreachable from the Internet while remaining cheap to scale per environment.

Data Network

Diagram

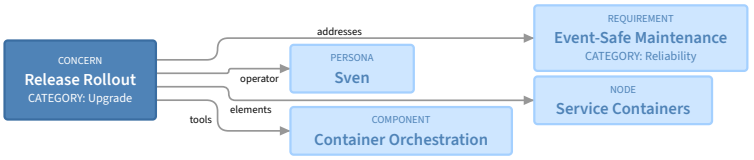


● NETWORK	Properties & Description		
	○ SCOPE	○ PROTOCOLS	○ ADDRESSING
	Isolated	TCP 5432	Hetzner private vSwitch, database subnet
	Description		
	The data network connects the service containers to the database server only — **BECAUSE** the persistence node must be reachable from the business services alone.		

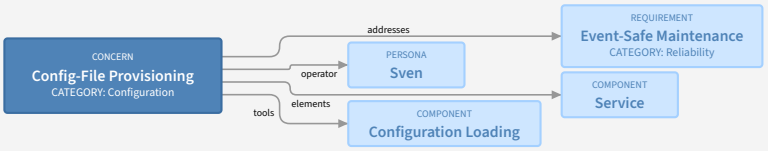
28 ● ARCH: Operations View (OV)

Release Rollout CATEGORY: Upgrade	Config-File Provisioning CATEGORY: Configuration	Role Assignment CATEGORY: Administration	Live Statistics Monitoring CATEGORY: Monitoring
Authentication Diagnostics CATEGORY: Incident	Horizontal Scaling CATEGORY: Capacity	Database Backup CATEGORY: Backup	Database Restore CATEGORY: Recovery

● CONCERN	Properties & Description		
Release Rollout	Diagram		



	○ CATEGORY	○ TRIGGER	○ OPERATOR
	Upgrade	on every release, outside running events	● **PERSONA:** Sven
	○ ADDRESSES	○ ELEMENTS	○ TOOLS
	● **REQUIREMENT:** Event-Safe Maintenance	● **NODE:** Service Containers	● **COMPONENT:** Container Orchestration
	Description		
	A release is rolled out per environment by pulling the new service container images and recreating the service containers between events — **BECAUSE** a running broadcast must never be interrupted by a rollout.		
Config-File Provisioning	Diagram		

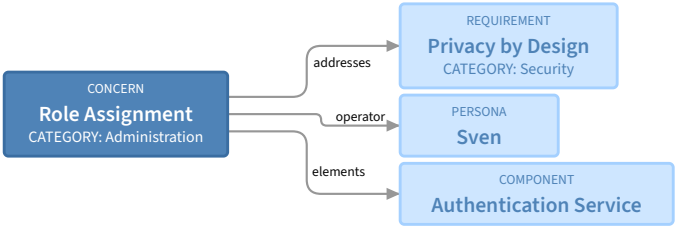


	○ CATEGORY	○ TRIGGER	○ OPERATOR
	Configuration	before each event	● **PERSONA:** Sven
	○ ADDRESSES	○ ELEMENTS	○ TOOLS
	● **REQUIREMENT:** Event-Safe Maintenance	● **COMPONENT:** Service	● **COMPONENT:** Configuration Loading
	Description		
	Streaming providers, their parameters, admins, and broker URLs are provisioned through a `broadcast.yaml` configuration file read by the service — **BECAUSE** provider endpoints and administrators are operational concerns kept out of the database.		

● CONCERN	Properties & Description

Role Assignment

Diagram



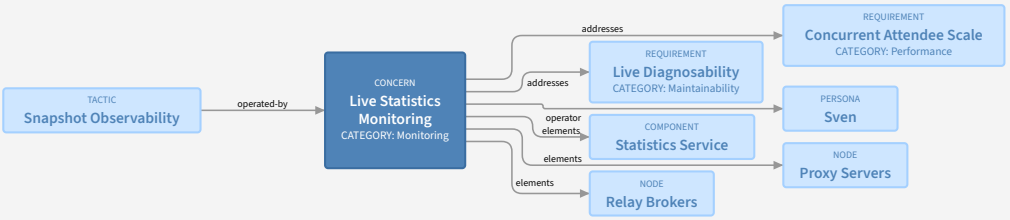
○ CATEGORY	○ TRIGGER	○ OPERATOR
Administration	on event creation	● **PERSONA:** Sven
○ ADDRESSES	**○ ELEMENTS**	**○ TOOLS**
● **REQUIREMENT:** Privacy by Design	● **COMPONENT:** Authentication Service	∅

Description

The first Manager of an event is assigned by the Software Administrator through configuration rather than a UI — **BECAUSE** permanent roles must be controlled administratively to uphold privacy by design.

Live Statistics Monitoring

Diagram



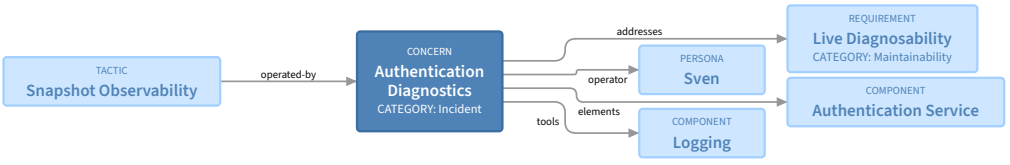
○ CATEGORY	○ TRIGGER	○ OPERATOR
Monitoring	during each event	● **PERSONA:** Sven
○ ADDRESSES	**○ ELEMENTS**	**○ TOOLS**
● **REQUIREMENT:** Concurrent Attendee Scale, ● **REQUIREMENT:** Live Diagnosability	● **COMPONENT:** Statistics Service, ● **NODE:** Proxy Servers, ● **NODE:** Relay Brokers	∅

Description

During a broadcast the administrator watches the statistics dashboard for the attendee curve, the channel distribution, and the authentication-flow counters — **BECAUSE** capacity and login problems must be detected while the audience is still online.

Authentication Diagnostics

Diagram



○ CATEGORY	○ TRIGGER	○ OPERATOR
Incident	on reported login failures	● **PERSONA:** Sven
○ ADDRESSES	**○ ELEMENTS**	**○ TOOLS**
● **REQUIREMENT:** Live Diagnosability	● **COMPONENT:** Authentication Service	● **COMPONENT:** Logging

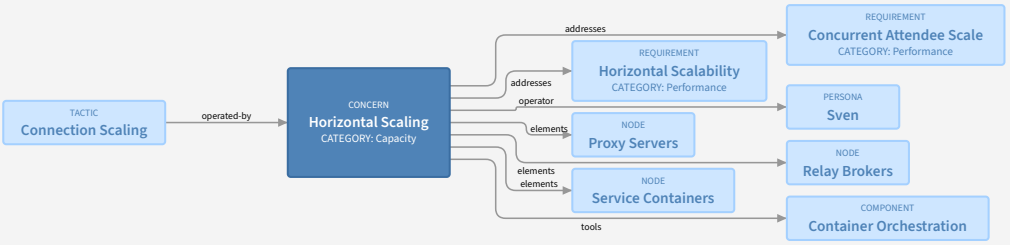
● CONCERN	Properties & Description

Description

Reported attendee login failures are diagnosed from the recorded token states (issued, sent, used) and the server log — **BECAUSE** operators must find out why an attendee fails to authenticate while the event runs.

Horizontal Scaling

Diagram



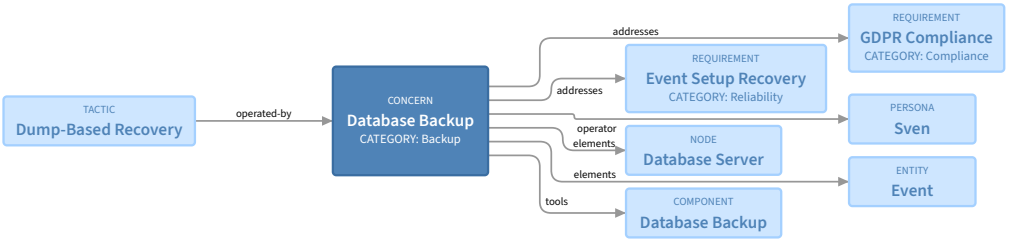
○ CATEGORY	○ TRIGGER	○ OPERATOR
Capacity	ahead of large events	● **PERSONA:** Sven
○ ADDRESSES	**○ ELEMENTS**	**○ TOOLS**
● **REQUIREMENT:** Horizontal Scalability, ● **REQUIREMENT:** Concurrent Attendee Scale	● **NODE:** Proxy Servers, ● **NODE:** Relay Brokers, ● **NODE:** Service Containers	● **COMPONENT:** Container Orchestration

Description

Operators scale an environment by adding proxy, relay, and service instances ahead of large events — **BECAUSE** townhall-scale audiences of up to 10000 require capacity provisioned per environment.

Database Backup

Diagram



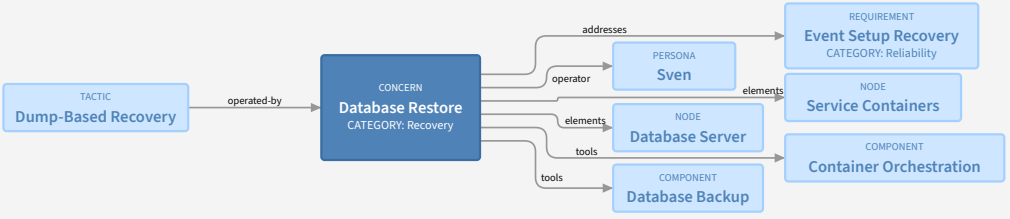
○ CATEGORY	○ TRIGGER	○ OPERATOR
Backup	nightly and before each event	● **PERSONA:** Sven
○ ADDRESSES	**○ ELEMENTS**	**○ TOOLS**
● **REQUIREMENT:** Event Setup Recovery, ● **REQUIREMENT:** GDPR Compliance	● **NODE:** Database Server, ● **ENTITY:** Event	● **COMPONENT:** Database Backup

Description

The PostgreSQL database and the filesystem assets are dumped nightly and before each event onto the backup storage of the same data center — **BECAUSE** event setups must survive a database loss without leaving the EU hosting boundary.

Database Restore

Diagram

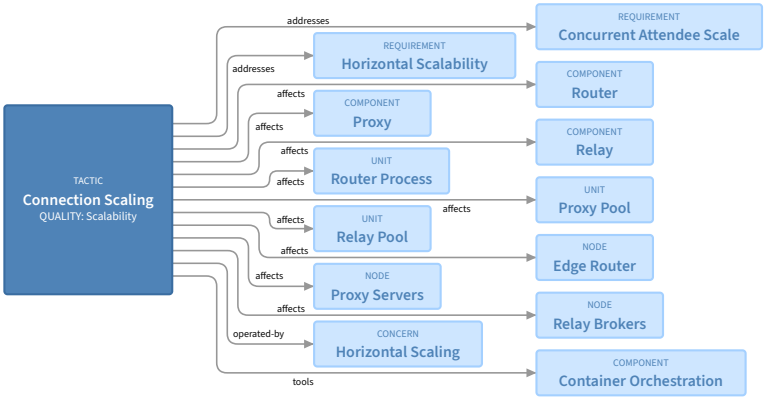


● CONCERN	Properties & Description		
	○ CATEGORY	○ TRIGGER	○ OPERATOR
	Recovery	on database loss	● PERSONA: Sven
	○ ADDRESSES	○ ELEMENTS	○ TOOLS
	● REQUIREMENT: Event Setup Recovery	● NODE: Database Server, ● NODE: Service Containers	● COMPONENT: Database Backup, ● COMPONENT: Container Orchestration
	Description		
	A lost database is rebuilt from the latest dump onto a fresh database server and the service containers are recreated against it — **BECAUSE** the recovery must be a rehearsed procedure rather than an improvisation during an event.		

29 ● ARCH: Quality Perspectives (QP)

Connection Scaling QUALITY: Scalability	Live Failover Continuity QUALITY: Availability	Privacy by Design QUALITY: Privacy	Email-Verified Access QUALITY: Security
Real-Time Reactivity QUALITY: Performance	Abuse Resistance QUALITY: Security	Universal Browser Reach QUALITY: Portability	Provider Stream Passthrough QUALITY: Performance
Network Segmentation QUALITY: Security	Dump-Based Recovery QUALITY: Reliability	Edge Asset Delivery QUALITY: Scalability	Shared Typed Contracts QUALITY: Maintainability
Snapshot Observability QUALITY: Maintainability			

● TACTIC	Properties & Description
Connection Scaling	Diagram

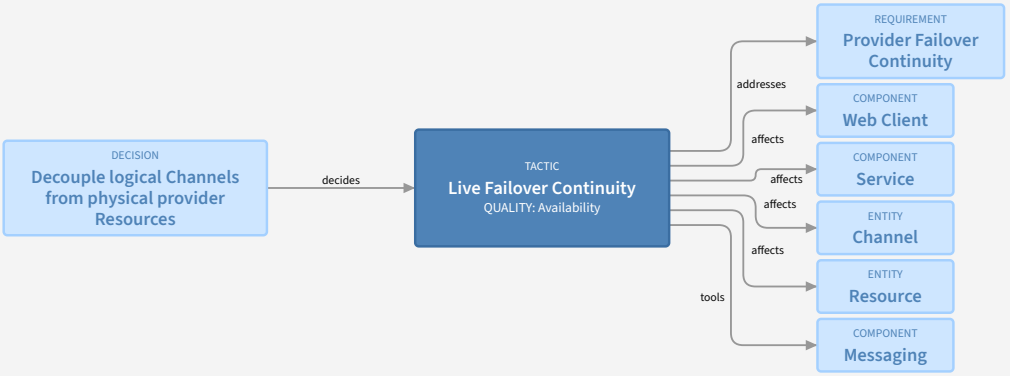


Properties	
○ QUALITY	Scalability
○ ADDRESSES	● REQUIREMENT: Concurrent Attendee Scale, ● REQUIREMENT: Horizontal Scalability
○ MECHANISM	stateless proxy and relay pools behind a round-robin router
○ TRADES-OFF	Maintainability
○ AFFECTS	● COMPONENT: Router, ● COMPONENT: Proxy, ● COMPONENT: Relay, ● UNIT: Router Process, ● UNIT: Proxy Pool, ● UNIT: Relay Pool, ● TIER: Middleware, ● NODE: Edge Router, ● NODE: Proxy Servers, ● NODE: Relay Brokers
○ OPERATED-BY	● CONCERN: Horizontal Scaling

● TACTIC	Properties & Description
	○ TOOLS: ● COMPONENT: Container Orchestration
	Description
	The proxy and relay tiers scale horizontally as stateless pools so connection load distributes across instances per environment — **BECAUSE** a single broker cannot hold the required attendee count while a sharded pool can.

Live Failover Continuity

Diagram



Properties

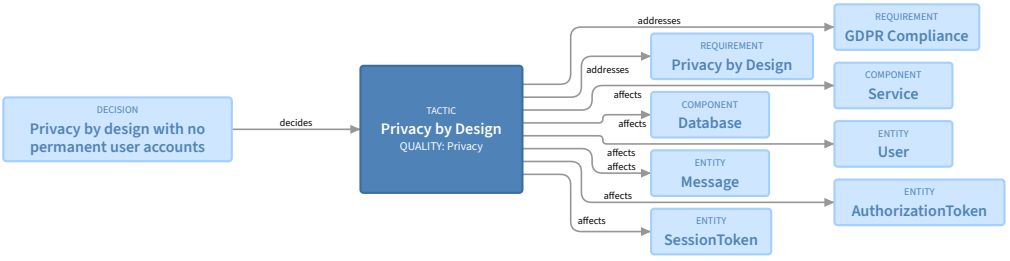
○ QUALITY	Availability
○ ADDRESSES	● REQUIREMENT: Provider Failover Continuity
○ MECHANISM	multi-provider resources with server-pushed active-resource switch
○ AFFECTS	● COMPONENT: Service, ● COMPONENT: Web Client, ● ENTITY: Channel, ● ENTITY: Resource
○ TOOLS	● COMPONENT: Messaging

Description

Each channel carries multiple configured resources and the service pushes the active-resource change over MQTT so clients follow automatically — **BECAUSE** provider outages must be survived without attendee action.

Privacy by Design

Diagram



Properties

○ QUALITY	Privacy
○ ADDRESSES	● REQUIREMENT: Privacy by Design, ● REQUIREMENT: GDPR Compliance
○ MECHANISM	automated anonymization on finish and no permanent accounts
○ TRADES-OFF	Usability
○ AFFECTS	● COMPONENT: Service, ● COMPONENT: Database, ● ENTITY: User, ● ENTITY: Message, ● ENTITY: AuthorizationToken, ● ENTITY: SessionToken

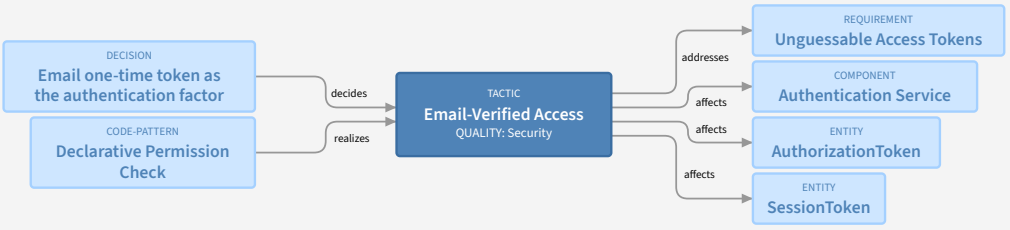
● TACTIC	Properties & Description

Description

Personal data exists only while an event runs and is automatically anonymized or deleted on finish, with no standing user accounts — **BECAUSE** minimizing retained personal data is the strongest guarantee against data-protection risk.

Email-Verified Access

Diagram



Properties

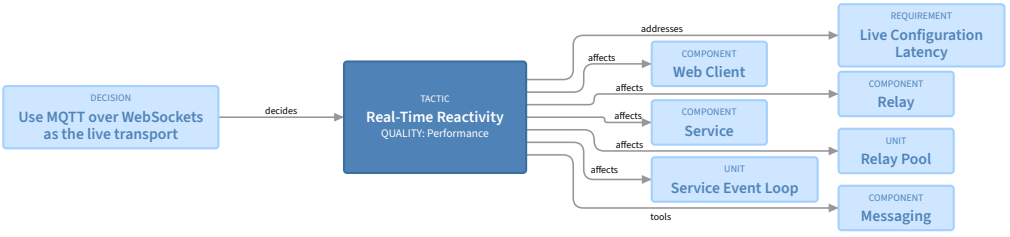
○ **QUALITY**	Security
○ **ADDRESSES**	● **REQUIREMENT:** Unguessable Access Tokens
○ **MECHANISM**	unguessable URLs plus time-limited one-time tokens and single sessions
○ **TRADES-OFF**	Usability
○ **AFFECTS**	● **COMPONENT:** Authentication Service, ● **ENTITY:** AuthorizationToken, ● **ENTITY:** SessionToken

Description

Access requires an unguessable URL and a short-lived one-time token, with only one active session enforced per user per event — **BECAUSE** layered, expiring secrets keep unauthorized viewers out without permanent credentials.

Real-Time Reactivity

Diagram



Properties

○ **QUALITY**	Performance
○ **ADDRESSES**	● **REQUIREMENT:** Live Configuration Latency
○ **MECHANISM**	MQTT publish/subscribe fan-out on per-event topics
○ **AFFECTS**	● **COMPONENT:** Relay, ● **COMPONENT:** Service, ● **COMPONENT:** Web Client, ● **UNIT:** Relay Pool, ● **UNIT:** Service Event Loop
○ **TOOLS**	● **COMPONENT:** Messaging

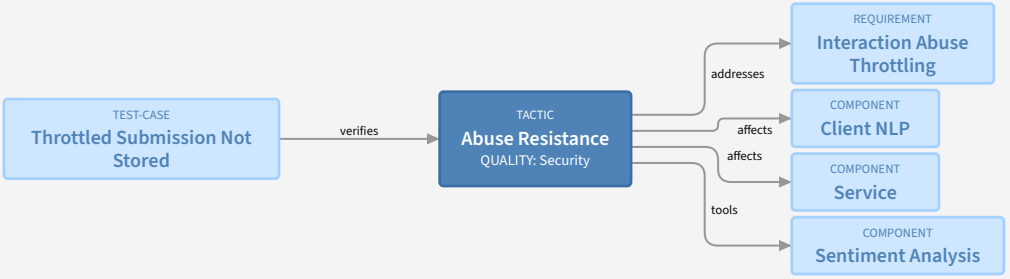
Description

Configuration and interaction changes are published on per-event MQTT topics and fanned out to subscribers — **BECAUSE** publish/subscribe propagates every change to all clients within the required time without polling.

● TACTIC	Properties & Description

Abuse Resistance

Diagram



Properties

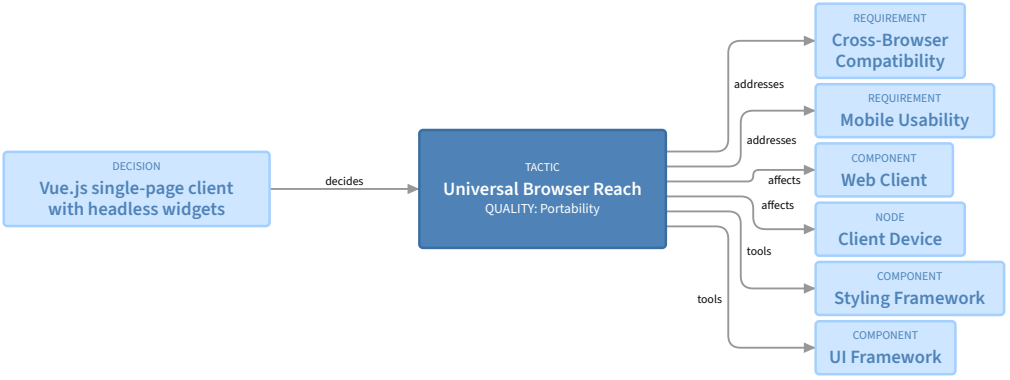
○ QUALITY	Security
○ ADDRESSES	● REQUIREMENT: Interaction Abuse Throttling
○ MECHANISM	per-user rate limiting plus client and server sentiment filtering
○ TRADES-OFF	Usability
○ AFFECTS	● COMPONENT: Service, ● COMPONENT: Client NLP
○ TOOLS	● COMPONENT: Sentiment Analysis

Description

Interaction is protected by configurable per-user throttling and optional client- and server-side sentiment checks — **BECAUSE** combining rate limits with content filtering defends the interaction channels against flooding and abuse.

Universal Browser Reach

Diagram



Properties

○ QUALITY	Portability
○ ADDRESSES	● REQUIREMENT: Cross-Browser Compatibility, ● REQUIREMENT: Mobile Usability
○ MECHANISM	standards-based responsive web client with no plugins
○ AFFECTS	● COMPONENT: Web Client, ● NODE: Client Device
○ TOOLS	● COMPONENT: UI Framework, ● COMPONENT: Styling Framework

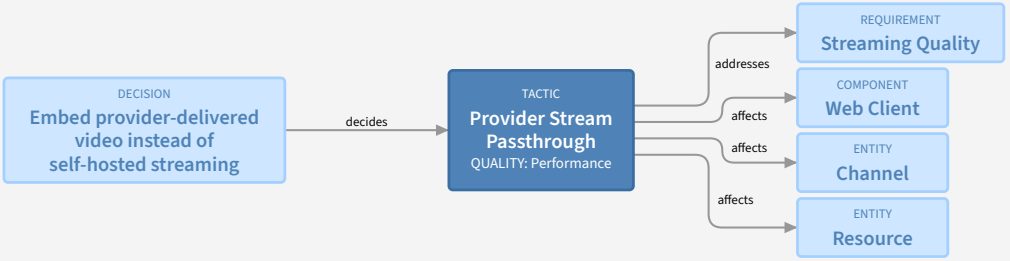
Description

The client is a standards-based responsive web application that adapts from desktop to mobile without plugins — **BECAUSE** a plugin-free web app reaches every recent browser on managed and unmanaged devices.

● TACTIC	Properties & Description

Provider Stream Passthrough

Diagram



Properties

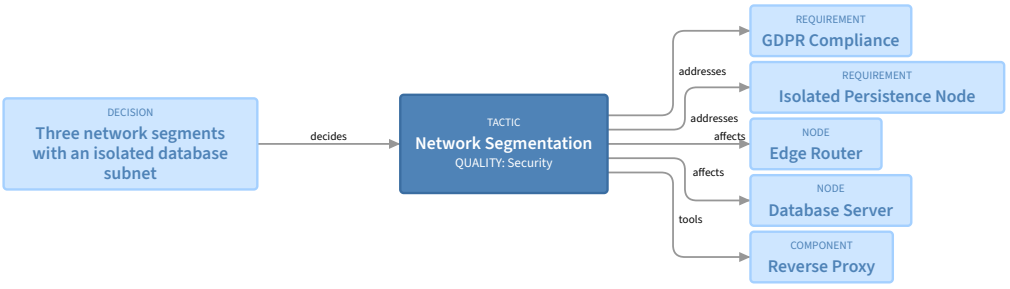
○ **QUALITY**	Performance
○ **ADDRESSES**	● **REQUIREMENT:** Streaming Quality
○ **MECHANISM**	provider-hosted encoding and delivery embedded by the client without transcoding
○ **AFFECTS**	● **COMPONENT:** Web Client, ● **ENTITY:** Channel, ● **ENTITY:** Resource

Description

The video is encoded and delivered by the streaming provider in the produced 1080p30 quality per language and embedded by the client as a resource of the channel, so the solution never touches the video itself — **BECAUSE** transcoding or relaying the video through the solution would cap the quality at the solution's own capacity.

Network Segmentation

Diagram



Properties

○ **QUALITY**	Security
○ **ADDRESSES**	● **REQUIREMENT:** Isolated Persistence Node, ● **REQUIREMENT:** GDPR Compliance
○ **MECHANISM**	TLS-only public ingress, private backend VLAN, and an isolated database subnet
○ **TRADES-OFF**	Maintainability
○ **AFFECTS**	● **TIER:** Middleware, ● **TIER:** Database, ● **NODE:** Edge Router, ● **NODE:** Database Server, ● **NETWORK:** Public Internet, ● **NETWORK:** Backend Network, ● **NETWORK:** Data Network
○ **TOOLS**	● **COMPONENT:** Reverse Proxy

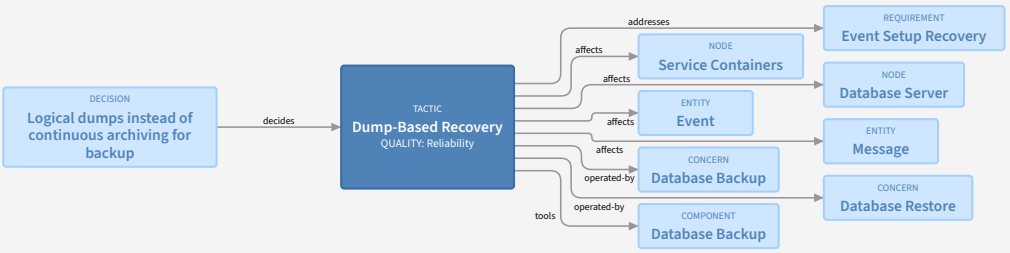
Description

Traffic passes three segments, the TLS-only public Internet up to the hardened router, the private backend VLAN between the connection-handling tiers and the services, and the isolated data subnet reaching the database from the service containers alone — **BECAUSE** every hop an attacker has to cross before reaching the persisted personal data is one more chance to stop them.

● TACTIC	Properties & Description

Dump-Based Recovery

Diagram



Properties

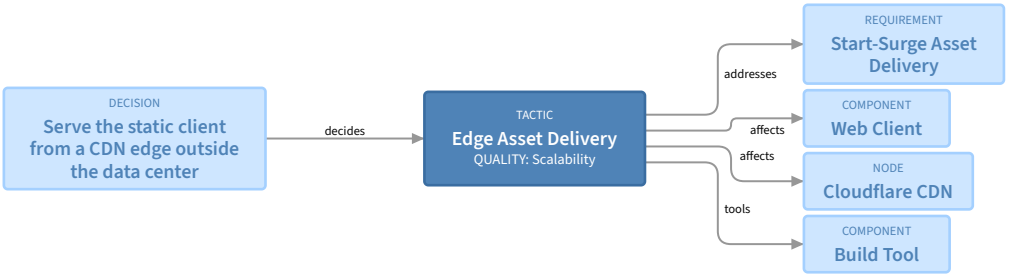
○ **QUALITY**	Reliability
○ **ADDRESSES**	● **REQUIREMENT:** Event Setup Recovery
○ **MECHANISM**	logical database dumps with a rehearsed restore onto a fresh server
○ **TRADES-OFF**	Availability
○ **AFFECTS**	● **NODE:** Database Server, ● **NODE:** Service Containers, ● **ENTITY:** Event, ● **ENTITY:** Message
○ **OPERATED-BY**	● **CONCERN:** Database Backup, ● **CONCERN:** Database Restore
○ **TOOLS**	● **COMPONENT:** Database Backup

Description

The database is dumped logically on a schedule tightened before each event and rebuilt from the latest dump onto a fresh server by a rehearsed procedure — **BECAUSE** a portable dump plus a drilled restore meets the recovery objectives without operating a standby database.

Edge Asset Delivery

Diagram



Properties

○ **QUALITY**	Scalability
○ **ADDRESSES**	● **REQUIREMENT:** Start-Surge Asset Delivery
○ **MECHANISM**	static client bundle served from a global CDN edge
○ **AFFECTS**	● **COMPONENT:** Web Client, ● **TIER:** Edge, ● **NODE:** Cloudflare CDN
○ **TOOLS**	● **COMPONENT:** Build Tool

Description

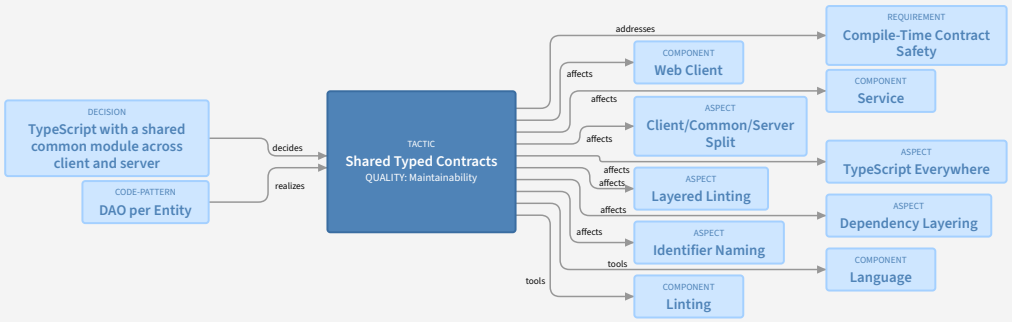
The client bundle is built into static files and served from the CDN edge close to the audience, so the start surge of an event downloads from the edge instead of from the data center — **BECAUSE** a CDN absorbs ten thousand simultaneous downloads which would saturate the single data center entry point.

● **TACTIC**

Shared Typed Contracts ⚓

Properties & Description

Diagram



Properties

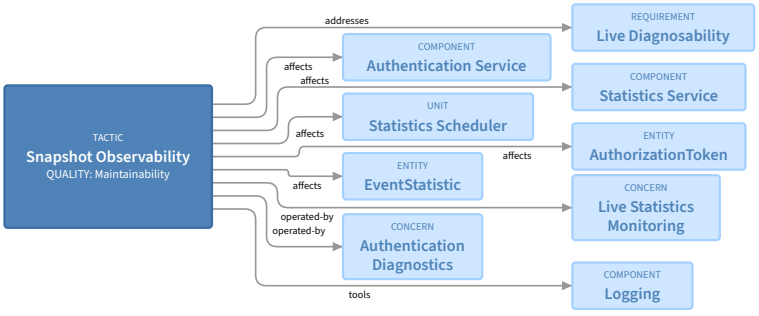
○ **QUALITY**	Maintainability
○ **ADDRESSES**	● **REQUIREMENT:** Compile-Time Contract Safety ⚭
○ **MECHANISM**	common module of shared types compiled by one TypeScript toolchain on both sides
○ **AFFECTS**	● **COMPONENT:** Web Client ⚭, ● **COMPONENT:** Service ⚭, ● **ASPECT:** Client/Common/Server Split ⚭, ● **ASPECT:** TypeScript Everywhere ⚭, ● **ASPECT:** Dependency Layering ⚭, ● **ASPECT:** Layered Linting ⚭, ● **ASPECT:** Identifier Naming ⚭
○ **TOOLS**	● **COMPONENT:** Language ⚭, ● **COMPONENT:** Linting ⚭

Description

The message topics and payload types live once in the common module and are compiled into both client and server by the same strongly typed toolchain, layered so that the module depends on neither side — **BECAUSE** a contract shared at the type level turns every mismatch into a build error.

Snapshot Observability ⚓

Diagram



Properties

○ **QUALITY**	Maintainability
○ **ADDRESSES**	● **REQUIREMENT:** Live Diagnosability ⚭
○ **MECHANISM**	periodic statistics snapshots plus recorded token states and structured logs
○ **AFFECTS**	● **COMPONENT:** Statistics Service ⚭, ● **COMPONENT:** Authentication Service ⚭, ● **UNIT:** Statistics Scheduler ⚭, ● **ENTITY:** EventStatistic ⚭, ● **ENTITY:** AuthorizationToken ⚭
○ **OPERATED-BY**	● **CONCERN:** Live Statistics Monitoring ⚭, ● **CONCERN:** Authentication Diagnostics ⚭
○ **TOOLS**	● **COMPONENT:** Logging ⚭

Description

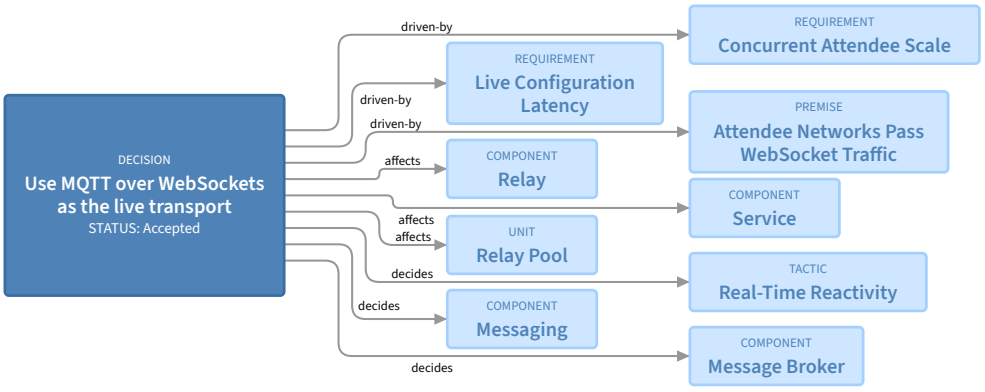
A scheduled task snapshots the event, channel, and user counters every five minutes and the authentication module records every token state next to its structured log — **BECAUSE** the running

● TACTIC	Properties & Description
	event can be observed and a failed login explained from data the solution collects anyway.

30 ● ARCH: Design Decisions (DD)

Use MQTT over WebSockets as the live transport STATUS: Accepted	Self-host on Hetzner instead of public cloud STATUS: Accepted	Privacy by design with no permanent user accounts STATUS: Accepted	Email one-time token as the authentication factor STATUS: Accepted
Decouple logical Channels from physical provider Resources STATUS: Accepted	Embed provider-delivered video instead of self-hosted streaming STATUS: Accepted	One modular service process instead of microservices STATUS: Accepted	TypeScript with a shared common module across client and server STATUS: Accepted
Vue.js single-page client with headless widgets STATUS: Accepted	HAProxy as a two-stage router and proxy edge STATUS: Accepted	Three network segments with an isolated database subnet STATUS: Accepted	Single-host Docker Compose instead of a cluster orchestrator STATUS: Accepted
PostgreSQL as the single self-hosted relational store STATUS: Accepted	Typed SQL builder instead of a full ORM STATUS: Accepted	Logical dumps instead of continuous archiving for backup STATUS: Accepted	Serve the static client from a CDN edge outside the data center STATUS: Accepted

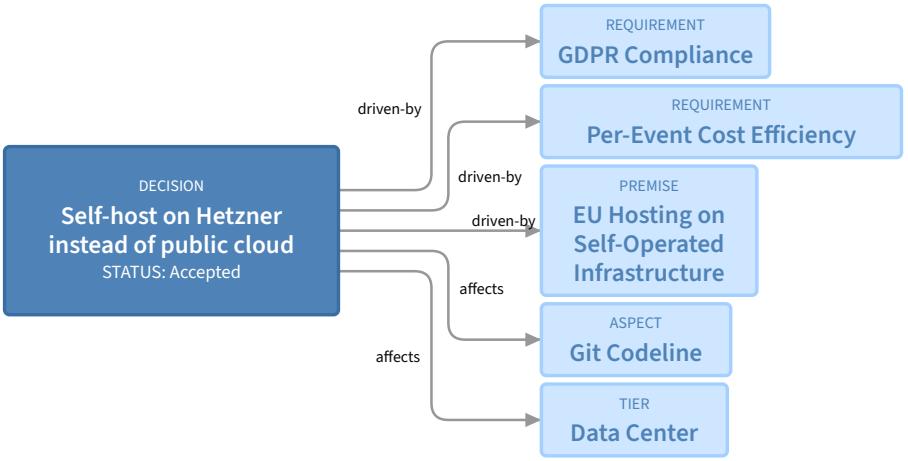
30.01 ● DECISION: Use MQTT over WebSockets as the live transport



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Concurrent Attendee Scale, ● REQUIREMENT: Live Configuration Latency, ● PREMISE: Attendee Networks Pass WebSocket Traffic
○ AFFECTS	● COMPONENT: Relay, ● COMPONENT: Service, ● UNIT: Relay Pool
○ DECIDES	● TACTIC: Real-Time Reactivity, ● COMPONENT: Messaging, ● COMPONENT: Message Broker
○ ALTERNATIVES	REST polling, bespoke WebSocket protocol
○ WHEN	A single event must push video state, configuration changes, chat, questions, and likes to between 2500 and 10000 concurrently connected browsers with sub-two-second latency, and the same channel must carry bidirectional interaction.
○ WHAT	We use MQTT over WebSockets as the live transport, brokered by Mosquitto and MQTT-Plus in a horizontally scaled relay tier, with per-event topics for fan-out and a thin Node.js service reacting to published messages.

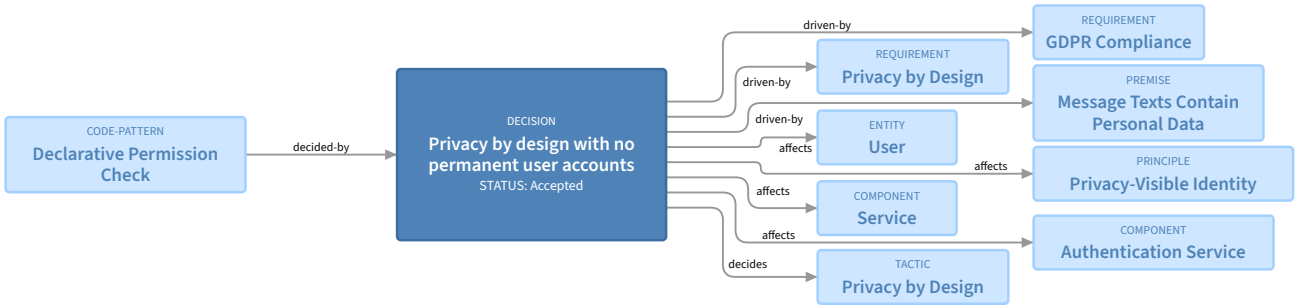
○ **WHY**	MQTT's publish/subscribe model fans a single publish out to thousands of subscribers far more cheaply than per-client REST polling or bespoke socket handling, and its mature broker ecosystem already solves connection scaling; a plain HTTP/REST design was rejected because it cannot deliver low-latency server-initiated fan-out at this connection count.
○ **CONSEQUENCES**	Every live capability has to be expressible as messages on per-event topics, and the relay tier becomes a scaling and operating unit of its own, while attendee networks blocking WebSocket traffic lock their users out of the live channel.

30.02 ● DECISION: Self-host on Hetzner instead of public cloud ⚓



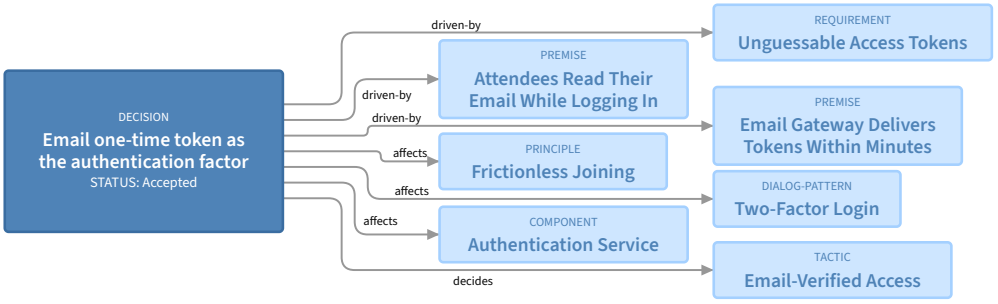
○ **STATUS**	Accepted
○ **DRIVEN-BY**	● **REQUIREMENT:** GDPR Compliance ⚭, ● **REQUIREMENT:** Per-Event Cost Efficiency ⚭, ● **PREMISE:** EU Hosting on Self-Operated Infrastructure ⚭
○ **AFFECTS**	● **TIER:** Data Center ⚭, ● **ASPECT:** Git Codeline ⚭
○ **ALTERNATIVES**	Azure, AWS, GitHub
○ **WHEN**	The solution must be GDPR-compliant with EU data residency, and a primary economic goal is to minimize recurring cost per event, the solution having been built specifically to replace a costlier third-party platform.
○ **WHAT**	We operate all server tiers on self-managed Hetzner infrastructure in Nürnberg, Germany, separated into dev, QA, and production environments, rather than on Azure or AWS, and keep the source on a self-administered Gitea rather than on a hosted forge.
○ **WHY**	Hetzner delivers EU-resident hosting at a fraction of hyperscaler cost, satisfying both the data-residency and cost-minimization forces; public cloud was rejected because its per-event egress and compute pricing would undermine the cost goal that justified building the solution at all.
○ **CONSEQUENCES**	Provisioning, scaling, monitoring, and backup are operated by the team itself instead of being consumed as managed services, so the Operations View carries the procedures a hyperscaler would have provided.

30.03 ● DECISION: Privacy by design with no permanent user accounts ⚓



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Privacy by Design ⚭, ● REQUIREMENT: GDPR Compliance ⚭, ● PREMISE: Message Texts Contain Personal Data ⚭
○ AFFECTS	● COMPONENT: Authentication Service ⚭, ● COMPONENT: Service ⚭, ● ENTITY: User ⚭, ● PRINCIPLE: Privacy-Visible Identity ⚭
○ DECIDES	● TACTIC: Privacy by Design ⚭
○ ALTERNATIVES	persistent user accounts
○ WHEN	Events handle personal attendee data under GDPR, yet attendance is transient and the operator wants to carry as little personal-data liability as possible while still supporting per-event identity for chat and moderation.
○ WHAT	We hold no permanent user accounts: users exist only via an event access list, role grant, or pattern match, and on event finish an automated procedure anonymizes messages and deletes users, tokens, and Moderator roles.
○ WHY	Deleting personal data the moment it is no longer needed minimizes both breach impact and compliance burden, which outweighs the convenience of persistent accounts; a conventional account model was rejected because standing personal data is exactly the liability privacy by design exists to avoid.

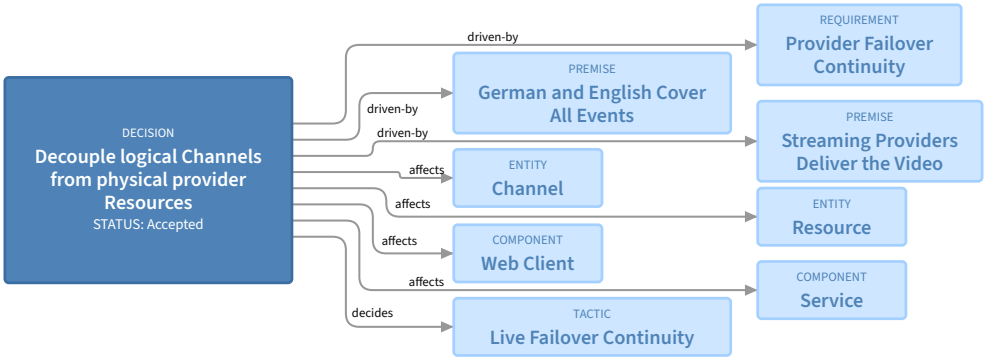
30.04 ● DECISION: Email one-time token as the authentication factor ⚓



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Unguessable Access Tokens ⚭, ● PREMISE: Attendees Read Their Email While Logging In ⚭, ● PREMISE: Email Gateway Delivers Tokens Within Minutes ⚭
○ AFFECTS	● COMPONENT: Authentication Service ⚭, ● DIALOG-PATTERN: Two-Factor Login ⚭, ● PRINCIPLE: Frictionless Joining ⚭
○ DECIDES	● TACTIC: Email-Verified Access ⚭
○ ALTERNATIVES	password accounts, external identity providers

○ **WHEN**	Access must be limited to invited attendees without permanent credentials, while also supporting frictionless and fully automated joining for large distributions provisioned from the Event Registration System.
○ **WHAT**	We authenticate via a one-time “NNN-NNN” token emailed to the attendee’s address, with optionally pre-generated tokens embeddable in the access URL for automatic entry, and we enforce a single active session per user per event.
○ **WHY**	An emailed token proves control of the authorized address with zero stored credentials and reuses email as a channel everyone already has, fitting the no-accounts stance; password accounts and external identity providers were rejected as too heavy and as reintroducing the persistent personal data the design forbids.

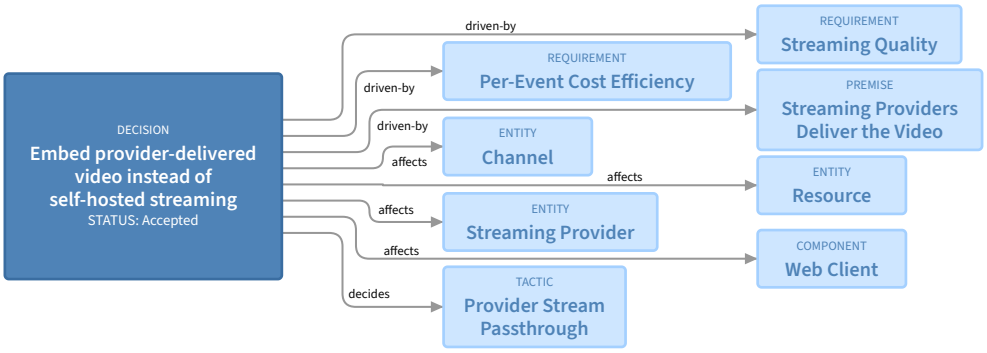
30.05 ● DECISION: Decouple logical Channels from physical provider Resources ⚓



○ **STATUS**	Accepted
○ **DRIVEN-BY**	● **REQUIREMENT:** Provider Failover Continuity ⚭, ● **PREMISE:** Streaming Providers Deliver the Video ⚭, ● **PREMISE:** German and English Cover All Events ⚭
○ **AFFECTS**	● **COMPONENT:** Service ⚭, ● **COMPONENT:** Web Client ⚭, ● **ENTITY:** Channel ⚭, ● **ENTITY:** Resource ⚭
○ **DECIDES**	● **TACTIC:** Live Failover Continuity ⚭
○ **ALTERNATIVES**	direct binding of clients to a single provider stream
○ **WHEN**	Productions ship in multiple languages and resolutions and must survive a streaming-provider outage mid-event by switching providers without attendees having to act or even notice.
○ **WHAT**	We model an event as logical Channels each backed by multiple physical Resources, where exactly one Channel and one Resource are active at a time, and switching the active Resource is pushed live to all clients.
○ **WHY**	Separating the audience-facing logical stream from interchangeable provider endpoints makes live failover a state change rather than a reconfiguration, satisfying both the multi-variant and continuity forces; binding clients directly to a single provider was rejected because it would make any provider problem an event-ending failure.

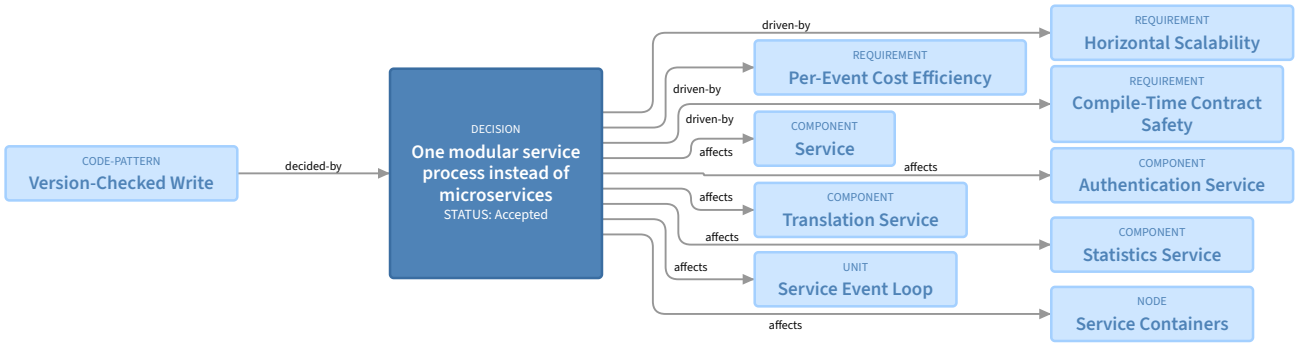
30.06 ● DECISION: Embed provider-delivered video instead of self-hosted

streaming



○ **STATUS**	Accepted
○ **DRIVEN-BY**	● **REQUIREMENT:** Streaming Quality, ● **REQUIREMENT:** Per-Event Cost Efficiency, ● **PREMISE:** Streaming Providers Deliver the Video
○ **AFFECTS**	● **COMPONENT:** Web Client, ● **ENTITY:** Channel, ● **ENTITY:** Resource, ● **ENTITY:** Streaming Provider
○ **DECIDES**	● **TACTIC:** Provider Stream Passthrough
○ **ALTERNATIVES**	self-hosted media server (e.g. OvenMediaEngine, Wowza), relaying the video through the data center
○ **WHEN**	Productions are delivered in 1080p30 in two languages to up to 10000 attendees, the streaming providers already encode and distribute the video worldwide, and the per-event cost has to stay low.
○ **WHAT**	We leave the video entirely with the streaming providers: the client embeds the provider player of the active resource, and the solution carries only the metadata telling every client which resource to play.
○ **WHY**	A provider delivers the produced quality from a global network for a per-event fee, whereas self-hosted streaming would need transcoding and egress capacity for 10000 viewers in the data center; a self-hosted media server was rejected as the single most expensive and most fragile piece the solution could take on.
○ **CONSEQUENCES**	The solution depends on the providers for the entire video path, so the provider failover has to be a first-class mechanism, and the video quality is capped by what the providers deliver rather than tunable by the solution.

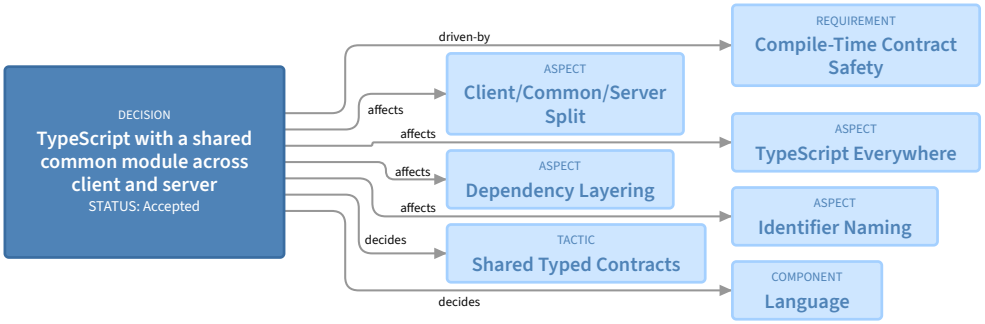
30.07 ● DECISION: One modular service process instead of microservices



○ **STATUS**	Accepted
○ **DRIVEN-BY**	● **REQUIREMENT:** Per-Event Cost Efficiency, ● **REQUIREMENT:** Compile-Time Contract Safety, ● **REQUIREMENT:** Horizontal Scalability

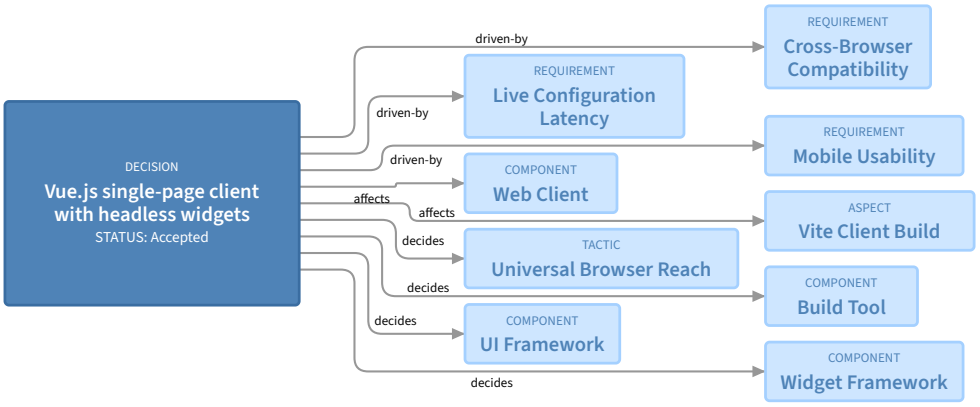
○ AFFECTS	● COMPONENT: Service ⚭, ● COMPONENT: Authentication Service ⚭, ● COMPONENT: Translation Service ⚭, ● COMPONENT: Statistics Service ⚭, ● UNIT: Service Event Loop ⚭, ● NODE: Service Containers ⚭
○ ALTERNATIVES	one service per capability (authentication, messaging, translation, statistics)
○ WHEN	The business logic of events, access, moderation, translation, and statistics is small and shares one data model, the connection load is carried by the relay tier anyway, and a small team operates the solution itself.
○ WHAT	We run the entire business logic as one Node.js service process with authentication, translation, and statistics as modules inside it, scaled by running several identical instances per environment.
○ WHY	The scaling problem of the solution is the connection count, which the relay tier solves, not the business logic, so splitting it would buy nothing but network hops, partial failures, and contracts to keep in sync; per-capability services were rejected because they multiply the operating effort of a self-hosted solution.
○ CONSEQUENCES	Every module shares the release cycle and the failure domain of the service, and a module growing a load profile of its own has to be carved out later at the cost of a refactoring.

30.08 ● DECISION: TypeScript with a shared common module across client and server ⚓



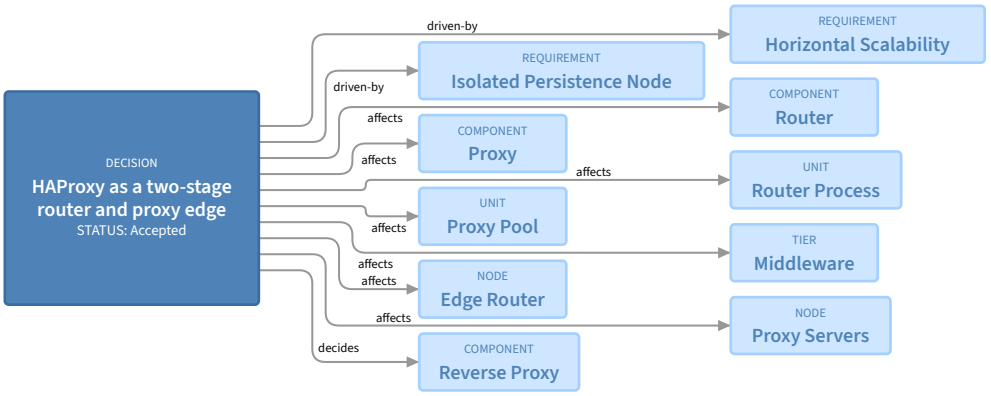
○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Compile-Time Contract Safety ⚭
○ AFFECTS	● ASPECT: Client/Common/Server Split ⚭, ● ASPECT: TypeScript Everywhere ⚭, ● ASPECT: Dependency Layering ⚭, ● ASPECT: Identifier Naming ⚭
○ DECIDES	● TACTIC: Shared Typed Contracts ⚭, ● COMPONENT: Language ⚭
○ ALTERNATIVES	plain JavaScript, per-tier types generated from an interface description (e.g. OpenAPI, AsyncAPI)
○ WHEN	Client and server exchange dozens of message types over MQTT topics, both sides evolve together in one repository, and a contract mismatch would surface only during a live event.
○ WHAT	We write client, server, and a common module in TypeScript, keep every topic and payload type in the common module, and let both sides import it, with the common module depending on neither of them.
○ WHY	Sharing the types directly makes the compiler the contract check with no generator and no drift between a description and the code; plain JavaScript was rejected as it leaves mismatches to runtime, and generated per-tier types as they reintroduce an intermediate description to keep in sync.

30.09 ● DECISION: Vue.js single-page client with headless widgets ⚓



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Cross-Browser Compatibility ⚭, ● REQUIREMENT: Mobile Usability ⚭, ● REQUIREMENT: Live Configuration Latency ⚭
○ AFFECTS	● COMPONENT: Web Client ⚭, ● ASPECT: Vite Client Build ⚭
○ DECIDES	● TACTIC: Universal Browser Reach ⚭, ● COMPONENT: UI Framework ⚭, ● COMPONENT: Widget Framework ⚭, ● COMPONENT: Build Tool ⚭
○ ALTERNATIVES	React, Svelte, server-rendered pages with progressive enhancement
○ WHEN	The attendee and operator screens run in any recent browser on managed and unmanaged desktop and mobile devices, react to server-pushed changes within seconds, and are styled to the corporate design without a native look.
○ WHAT	We build the client as a Vue.js single-page application composed of headless Reka UI widgets styled with Tailwind CSS, bundled by Vite into static files.
○ WHY	A reactive component model maps server-pushed state straight onto the screen, and headless widgets give accessible behavior while leaving the styling free; React was rejected for its heavier toolchain in a small team, Svelte for its thinner ecosystem, and server-rendered pages because they cannot follow live state without reloads.

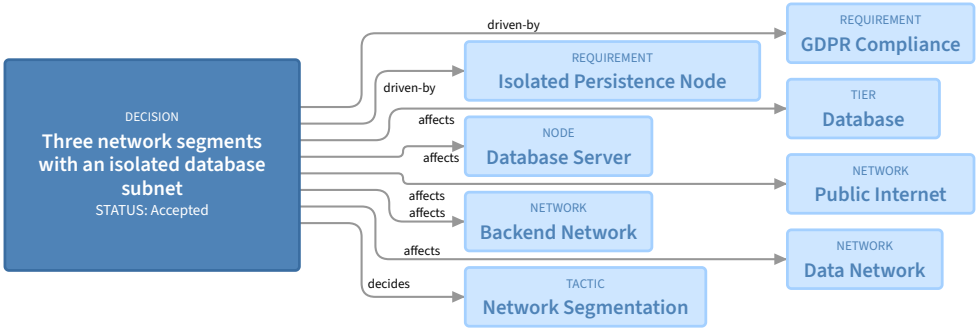
30.10 ● DECISION: HAProxy as a two-stage router and proxy edge ⚓



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Horizontal Scalability ⚭, ● REQUIREMENT: Isolated Persistence Node ⚭
○ AFFECTS	● COMPONENT: Router ⚭, ● COMPONENT: Proxy ⚭, ● UNIT: Router Process ⚭, ● UNIT: Proxy Pool ⚭, ● TIER: Middleware ⚭, ● NODE: Edge Router ⚭, ● NODE: Proxy Servers ⚭

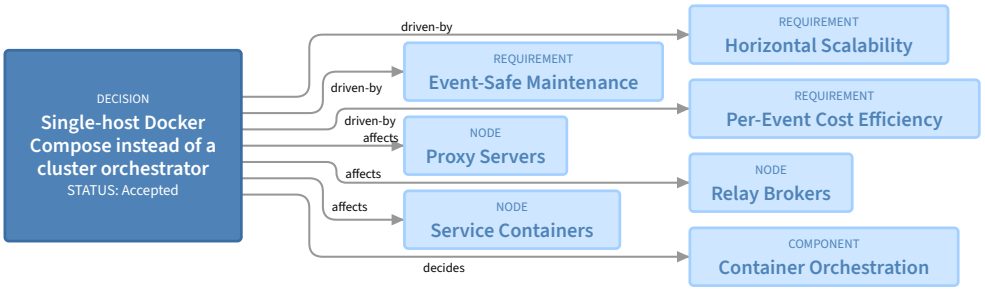
○ DECIDES	● COMPONENT: Reverse Proxy
○ ALTERNATIVES	nginx, Traefik, a managed cloud load balancer
○ WHEN	Ten thousand long-lived WebSocket connections per event have to enter one hardened data center entry point and be spread across the per-environment proxy and relay instances, with dev, QA, and production separated at the edge.
○ WHAT	We terminate all traffic at one HAProxy router process per entry point, firewalled by NFTables, which round-robins it to a pool of HAProxy proxy instances per environment forwarding to the relay brokers.
○ WHY	HAProxy handles WebSocket upgrades and connection counts of this size with a mature configuration model and serves both stages with one product; nginx was rejected for its weaker connection-level balancing, Traefik for its container-centric configuration, and a managed load balancer because the solution is self-hosted.
○ CONSEQUENCES	The router is a single point of entry per data center and its capacity is the ceiling of the environment, and every routing change touches a hand-maintained configuration.

30.11 ● DECISION: Three network segments with an isolated database subnet



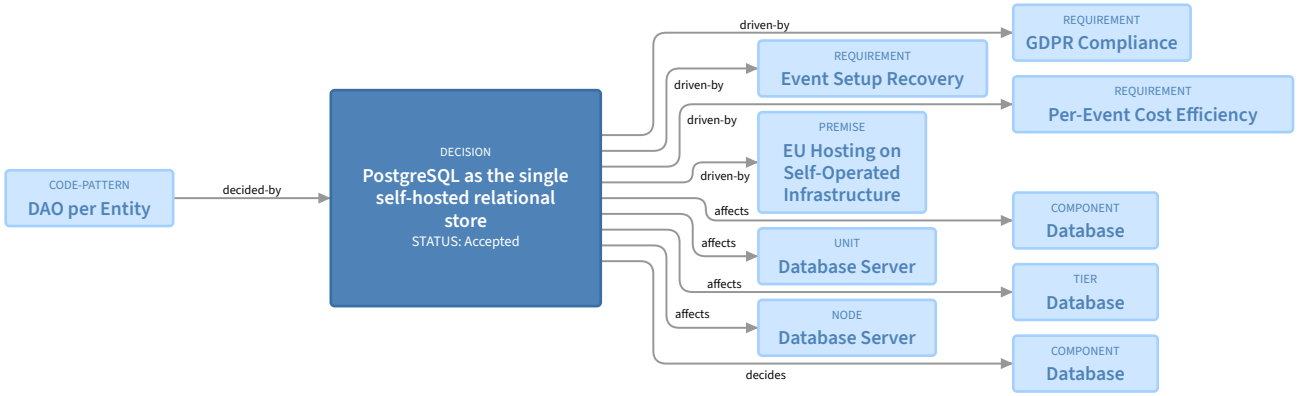
○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Isolated Persistence Node, ● REQUIREMENT: GDPR Compliance
○ AFFECTS	● NETWORK: Public Internet, ● NETWORK: Backend Network, ● NETWORK: Data Network, ● NODE: Database Server, ● TIER: Database
○ DECIDES	● TACTIC: Network Segmentation
○ ALTERNATIVES	one flat private network, database attached to the backend network
○ WHEN	The persisted personal data of the attendees lives on the database server, while the router, proxies, and relays face the Internet and are the parts most likely to be compromised.
○ WHAT	We segment the data center into a private backend VLAN carrying the router, proxies, relays, and services, and an isolated data subnet carrying only the database and the service containers, behind the TLS-only public Internet.
○ WHY	A compromised connection-handling node then still has no route to the database, which is the strongest containment available at no recurring cost on a private vSwitch; a flat network was rejected because it makes every node a stepping stone to the personal data.

30.12 ● DECISION: Single-host Docker Compose instead of a cluster orchestrator



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Per-Event Cost Efficiency, ● REQUIREMENT: Horizontal Scalability, ● REQUIREMENT: Event-Safe Maintenance
○ AFFECTS	● NODE: Service Containers, ● NODE: Proxy Servers, ● NODE: Relay Brokers
○ DECIDES	● COMPONENT: Container Orchestration
○ ALTERNATIVES	Kubernetes, Nomad
○ WHEN	Each environment runs a handful of proxy, relay, and service instances on self-managed servers, scaling is planned ahead of an event rather than reacting to load, and rollouts happen only between events.
○ WHAT	We run and recreate the containers of an environment with Docker Compose from one compose file per environment, adding instances by editing that file and recreating between events.
○ WHY	Planned, per-event scaling and rollouts between events need no scheduler, self-healing, or rolling updates, so a cluster would add an operating burden without a benefit; Kubernetes and Nomad were rejected because operating the cluster would cost more than the solution it hosts.
○ CONSEQUENCES	There is no automatic failover of a crashed instance during an event, and growing beyond one host per environment means revisiting this decision.

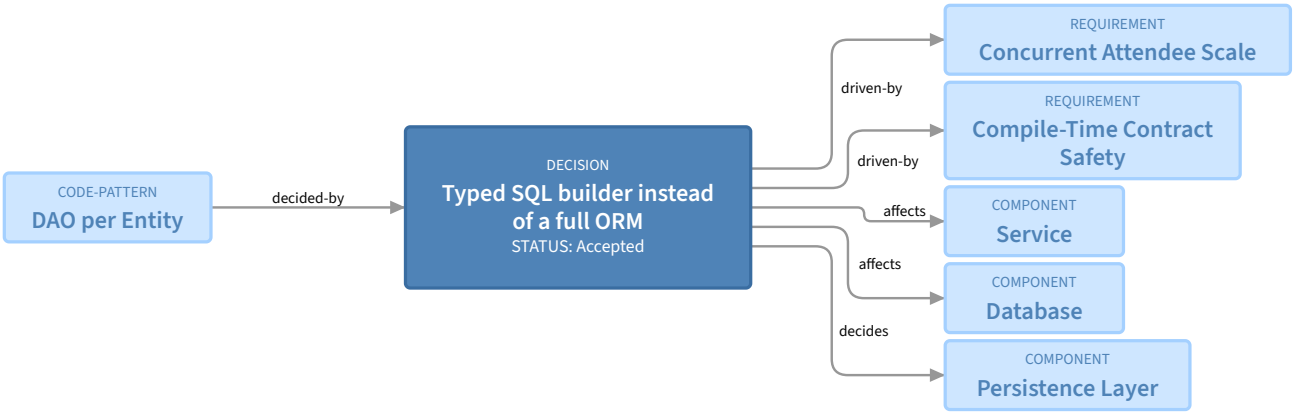
30.13 ● DECISION: PostgreSQL as the single self-hosted relational store



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: GDPR Compliance, ● REQUIREMENT: Event Setup Recovery, ● REQUIREMENT: Per-Event Cost Efficiency, ● PREMISE: EU Hosting on Self-Operated Infrastructure
○ AFFECTS	● COMPONENT: Database, ● UNIT: Database Server, ● NODE: Database Server, ● TIER: Database
○ DECIDES	● COMPONENT: Database

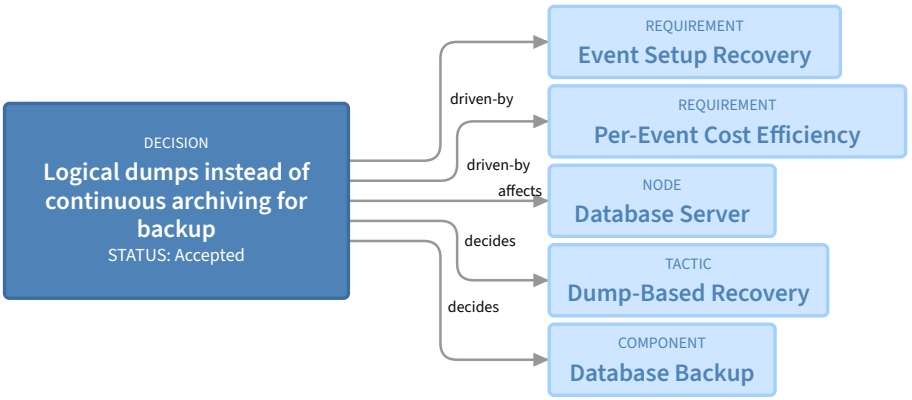
○ ALTERNATIVES	a document store (e.g. MongoDB), an embedded database (e.g. SQLite), a managed cloud database
○ WHEN	Events, channels, messages, tokens, and statistics form a relational model with transactional invariants, the data has to stay in the EU data center, and moderation and reporting need ad-hoc queries over it.
○ WHAT	We persist all state in one self-hosted PostgreSQL primary per environment, accessed by the service alone, with the filesystem beside it for the static assets.
○ WHY	A relational store with ACID transactions matches the model and the consistency the moderation needs, and PostgreSQL self-hosts on the same servers at no license cost; a document store was rejected for its weaker invariants, an embedded database for its single-writer limit across service instances, and a managed database as it leaves the self-hosted boundary.

30.14 ● DECISION: Typed SQL builder instead of a full ORM ⚓



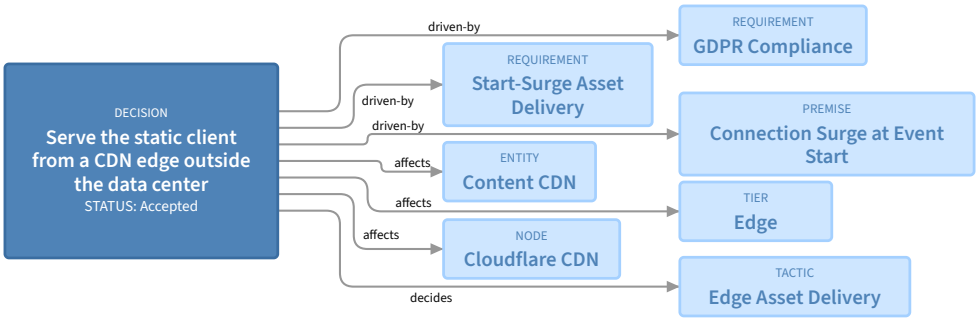
○ STATUS	Accepted
○ DRIVEN-BY	● **REQUIREMENT: Compile-Time Contract Safety** ⚭, ● **REQUIREMENT: Concurrent Attendee Scale** ⚭
○ AFFECTS	● **COMPONENT: Service** ⚭, ● **COMPONENT: Database** ⚭
○ DECIDES	● **COMPONENT: Persistence Layer** ⚭
○ ALTERNATIVES	Prisma, TypeORM
○ WHEN	The service persists the event state through TypeScript against PostgreSQL, the schema is generated spec-first from the data model, and the message handlers have to stay cheap under the load of a large event.
○ WHAT	We access PostgreSQL through Drizzle, a typed SQL query builder and schema definition layer, with the schema derived from the data model.
○ WHY	A thin typed layer keeps the end-to-end type safety of the shared contracts down to the database without the query overhead and abstraction leaks of a heavy ORM; Prisma was rejected for its separate schema language and query engine, TypeORM for its runtime cost and weaker typing.

30.15 ● DECISION: Logical dumps instead of continuous archiving for backup



○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Event Setup Recovery, ● REQUIREMENT: Per-Event Cost Efficiency
○ AFFECTS	● NODE: Database Server
○ DECIDES	● TACTIC: Dump-Based Recovery, ● COMPONENT: Database Backup
○ ALTERNATIVES	pgBackRest, Barman (continuous WAL archiving with point-in-time recovery)
○ WHEN	The recovery objective tolerates the loss of a day of changes (an hour right before an event), the data volume per event is small, and the backup must stay within the EU data center without a backup server of its own.
○ WHAT	We back up with logical pg_dump dumps taken nightly and before each event onto the backup storage of the data center and restore them with pg_restore onto a fresh server.
○ WHY	A logical dump is portable across PostgreSQL versions, needs no additional server, and meets the objectives as the data changes mainly before an event; continuous archiving was rejected because point-in-time recovery buys nothing for data that is anonymized at the end of every event anyway.

30.16 ● DECISION: Serve the static client from a CDN edge outside the data center



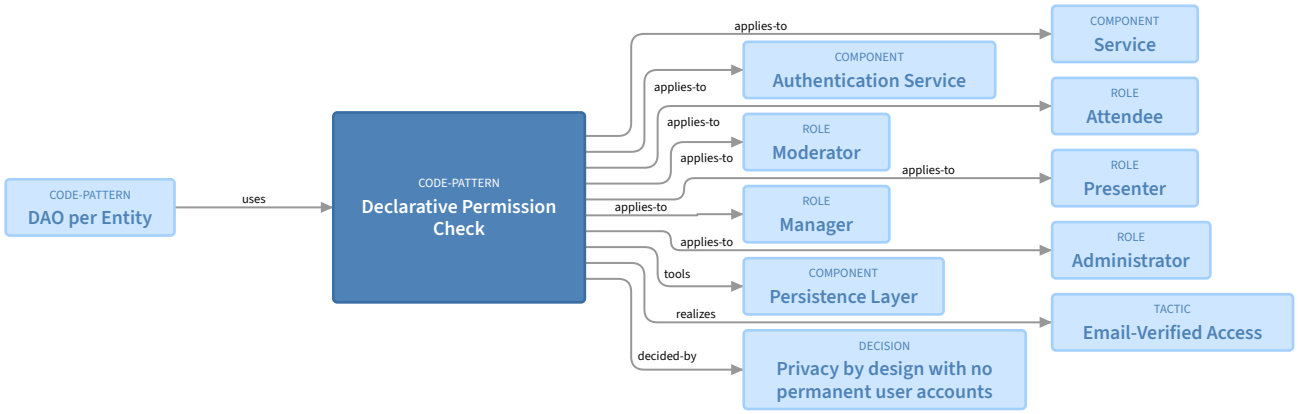
○ STATUS	Accepted
○ DRIVEN-BY	● REQUIREMENT: Start-Surge Asset Delivery, ● REQUIREMENT: GDPR Compliance, ● PREMISE: Connection Surge at Event Start
○ AFFECTS	● TIER: Edge, ● NODE: Cloudflare CDN, ● ENTITY: Content CDN
○ DECIDES	● TACTIC: Edge Asset Delivery
○ ALTERNATIVES	serving the bundle from the data center router, a second Hetzner location

○ **WHEN**	Up to 10000 attendees download the client bundle within the same minute at the start of an event, the data center has one entry point, and the personal data has to stay within the EU.
○ **WHAT**	We serve the static client bundle and static resources from a Cloudflare CDN edge, while every live connection and every request carrying personal data goes to the data center directly.
○ **WHY**	The bundle is public, identical for everyone, and cacheable, exactly what a CDN absorbs at any scale, so the start surge never reaches the data center; serving it from the router was rejected because the surge would compete with the live connections for the same entry point.
○ **CONSEQUENCES**	Only static, non-personal content leaves the EU hosting boundary, but the CDN sees the client addresses, so the data processing agreement with the CDN provider is part of the GDPR compliance evidence.

31 ● ARCH: Code Patterns (CP)

Declarative Permission Check | DAO per Entity | Version-Checked Write

31.01 ● CODE-PATTERN: Declarative Permission Check

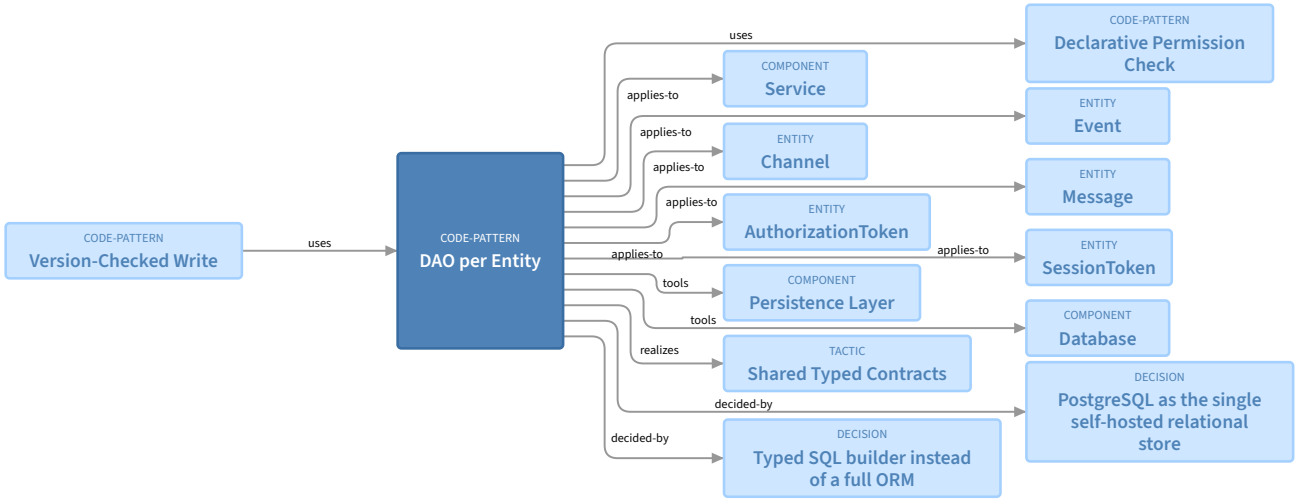


○ **APPLIES-TO**	● **COMPONENT:** Service, ● **COMPONENT:** Authentication Service, ● **ROLE:** Attendee, ● **ROLE:** Moderator, ● **ROLE:** Presenter, ● **ROLE:** Manager, ● **ROLE:** Administrator
○ **REALIZES**	● **TACTIC:** Email-Verified Access
○ **DECIDED-BY**	● **DECISION:** Privacy by design with no permanent user accounts
○ **TOOLS**	● **COMPONENT:** Persistence Layer
○ **AVOID**	role checks in the client alone, or an ad-hoc `if (session.role === ...)` inside a DAO method

The authorization model is coded as one declarative permission table, holding exactly one entry per `PERMISSION` of the Authorization Model (identified by its anchor id) with the role, the entity, the granted operations, the admitted lifecycle states, and an optional condition function, which the single `authorize()` evaluates for every operation after the event scope check, denying whatever no entry grants — **BECAUSE** the table is then reviewed line by line against the specification, while a check spread over the code is neither complete nor auditable.

```typescript
type Permission<E extends Entity> = {
    id:         string
    role:       Role
    entity:     E
    operations: Operation[]
    states?:    string[]
    condition?: Condition<E>
}

const permissions: AnyPermission[] = [
    { id: "moderator-read-messages", role: "moderator", entity: "Message", operations: [ "read" ] },
    { id: "moderator-decide",        role: "moderator", entity: "Message", operations: [ "accept", "reject" ] },
    { id: "moderator-forward",       role: "moderator", entity: "Message", operations: [ "forward", "update" ],
      states: [ "accepted", "forwarded" ],
      condition: ({ operation, obj, changes }) => obj.type === "Question"
          && (operation !== "update" || confinedTo(changes, "presenterAnnotation", "predecessorId", "questionTags")) }
]

export const authorize = async (db: DB, session: Session, operation: Operation,
    entity: Entity, obj: unknown, changes: object = {}): Promise<void> => {
    if (session.role !== "administrator"
        && (session.user === null || await eventOf[entity](db, obj as Ref) !== session.user.eventId))
        throw new AuthorizationError(session, operation, entity)
    const state = (obj as { state?: string }).state ?? ""
    const ctx   = { db, session, operation, obj, changes } as Context<Entity>
    for (const p of permissions) {
        if (p.role !== session.role || p.entity !== entity || !p.operations.includes(operation))
            continue
        if (p.states !== undefined && operation !== "create" && !p.states.includes(state))
            continue
        if (p.condition !== undefined && !(await (p.condition as Condition<Entity>)(ctx)))
            continue
        return
    }
    throw new AuthorizationError(session, operation, entity)
}
```

31.02 ● CODE-PATTERN: DAO per Entity ⚓



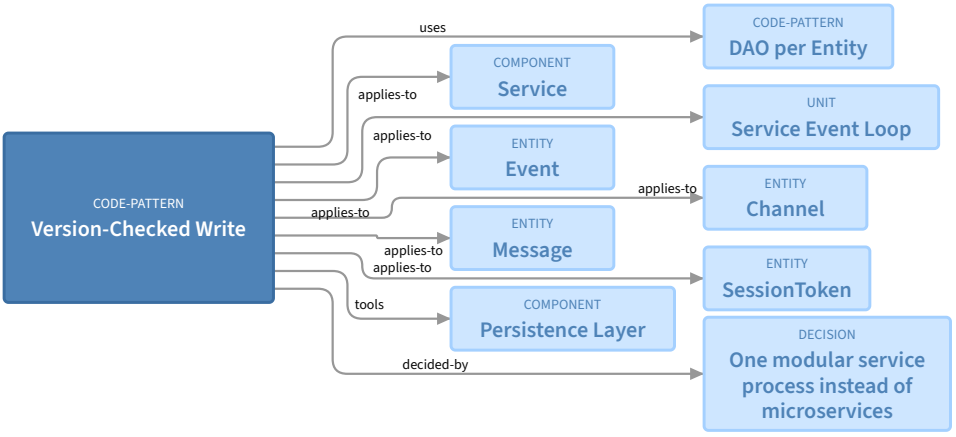
○ APPLIES-TO	● COMPONENT: Service ⚭, ● ENTITY: Event ⚭, ● ENTITY: Channel ⚭, ● ENTITY: Message ⚭, ● ENTITY: AuthorizationToken ⚭, ● ENTITY: SessionToken ⚭
○ REALIZES	● TACTIC: Shared Typed Contracts ⚭
○ DECIDED-BY	● DECISION: Typed SQL builder instead of a full ORM ⚭, ● DECISION: PostgreSQL as the single self-hosted relational store ⚭

○ TOOLS	● COMPONENT: Persistence Layer ⚭, ● COMPONENT: Database ⚭
○ USES	● CODE-PATTERN: Declarative Permission Check ⚭
○ AVOID	a query assembled outside the DAO of its entity, or a DAO writing the table of another entity

Every entity of the Data Model is persisted through exactly one Data Access Object class (`app-db-dao-dm-*.ts`, a lifecycle-carrying entity through `app-db-dao-sm-*.ts`), which takes the session and whole objects (a draft to create, the changed row to update), authorizes every operation through the shared core *before* touching the Drizzle table, and bundles the writes of one operation into one transaction — **BECAUSE** the DAO is then the single place where an entity meets its table, its permissions, and its invariants, so a schema or permission change is applied once.

```
export class SessionTokenDAO {
    constructor (protected core: DB) {}

    async create (session: Session, draft: Draft<SessionToken>): Promise<SessionToken> {
        const db = this.core.require()

        /*  authorize the creation on the draft  */
        await this.core.authorize(session, "create", "SessionToken", draft)

        /*  bundle multiple operations into a single transaction  */
        return db.transaction(async (tx) => {
            /*  close any prior session of the user for the event (SPEC-DR single-session) */
            await tx.delete(schema.sessionTokens).where(and(
                eq(schema.sessionTokens.userId,  draft.userId),
                eq(schema.sessionTokens.eventId, draft.eventId)))

            /*  store the draft  */
            const [ row ] = await tx.insert(schema.sessionTokens).values(draft).returning()
            return row
        })
    }
}
```

31.03 ● CODE-PATTERN: Version-Checked Write ⚓



○ APPLIES-TO	● COMPONENT: Service ⚭, ● UNIT: Service Event Loop ⚭, ● ENTITY: Event ⚭, ● ENTITY: Channel ⚭, ● ENTITY: Message ⚭, ● ENTITY: SessionToken ⚭
○ DECIDED-BY	● DECISION: One modular service process instead of microservices ⚭
○ TOOLS	● COMPONENT: Persistence Layer ⚭
○ USES	● CODE-PATTERN: DAO per Entity ⚭
○ AVOID	a last-writer-wins UPDATE of the whole row, or a read-modify-write across two statements

Every row carries a `version` column, and every update or deletion of a DAO loads the stored row, derives the changed attributes (`changesOf`), and writes them in one statement whose `WHERE` also matches the version the caller based its object on,

incrementing it (bump) on the way and raising a `ConflictError` when no row was hit (`stored`) — **BECAUSE** two moderators editing the same message at once must not silently overwrite each other, and the version check costs one predicate instead of a lock.

```
/*  ensure a version-checked write of an object of an entity returned the
    stored row, i.e. hit the version it was based on  */
stored<T> (obj: T | undefined, entity: Entity, id: string): T {
    if (obj === undefined)
        throw new ConflictError(entity, id)
    return obj
}

/*  the incremented version of a row (optimistic locking)  */
bump (version: PgColumn): SQL {
    return sql`${version} + 1`
}

/*  usage inside a DAO: delete the row, version-checked (optimistic locking)  */
const [ stored ] = await db
    .delete(schema.sessionTokens)
    .where(and(eq(schema.sessionTokens.sessionId, id), eq(schema.sessionTokens.version,
sessionTokens.version)))
    .returning({ sessionId: schema.sessionTokens.sessionId })
this.core.stored(stored, "SessionToken", id)
```

32 ● TEST: Test Cases (TC) ⚓

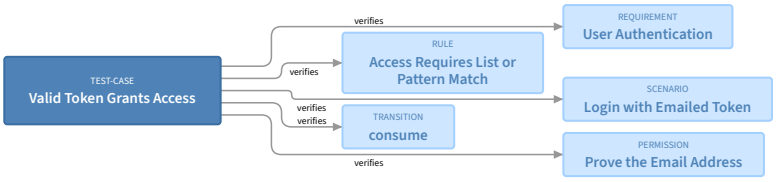
Valid Token Grants Access	Unauthorized Email Rejected	URL Token Skips Login Dialog	Second Login Closes First Session
Moderated Question Starts Pending	Unmoderated Chat Starts Accepted	Chat Cannot Enter Question States	Moderator Message Starts Accepted
Forwarded Message Is Locked	Sentiment Auto-Reject Below Threshold	Sentiment Auto-Accept Above Threshold	Throttled Submission Not Stored
Live Provider Switch Propagates	Channel Activation Deactivates Previous	First Channel Activated on Creation	First Resource Activated on Creation
Active Channel Not Deactivated Directly	Active Resource Not Deactivated Directly	Config Change Reaches Clients	Registration Import Avoids Duplicates
Returned URL Format	Anonymization Removes Personal Data	User Vanishes Without Role or Access	Export Contains Required Fields
Manager Role Survives Finish	Chat Message Shows Configured Name	Like Toggles the Count	Deleted Message Leaves Placeholder
Client-Side Check Holds Back Improper Input	Direct Moderator Reply Stays Private	Resource URL Selects the Resource	Language Switch Applies Immediately
Presenter Sees Forwarded Questions Only	Administrator Sees No Attendee Data	Administrator Enters Without Access List Entry	Administrator Role Granted by Configuration Only
Concurrent Attendee Load			

Coverage	Covered	Total	Ratio
● USE-CASE	9	14	64%
● SCENARIO	21	51	41%
● REQUIREMENT	26	63	41%
● RULE	18	18	100%

● TEST-CASE	Properties & Description

Valid Token Grants Access ⚓

Diagram

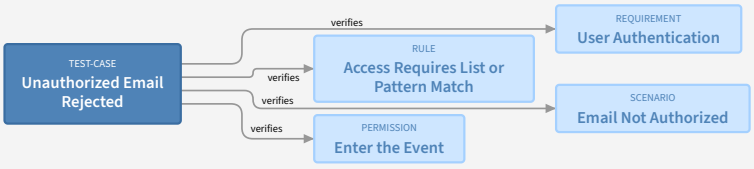


Properties

○ VERIFIES	● REQUIREMENT: User Authentication ⚭, ● SCENARIO: Login with Emailed Token ⚭, ● RULE: Access Requires List or Pattern Match ⚭, ● TRANSITION: consume ⚭, ● PERMISSION: Prove the Email Address ⚭
○ PRE-CONDITION	An invited user with a sent, unexpired authorization token exists for a running event.
○ INPUT	The user submits the correct six-digit token for their email.
○ EXPECTED	The system issues a session token and grants access to the stream.
○ POST-CONDITION	The authorization token is in state used.

Unauthorized Email Rejected ⚓

Diagram

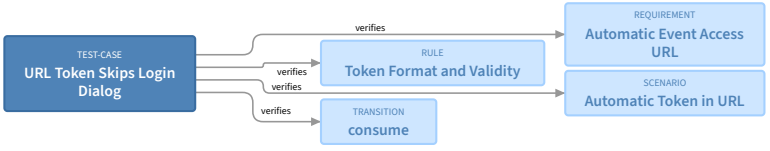


Properties

○ VERIFIES	● REQUIREMENT: User Authentication ⚭, ● SCENARIO: Email Not Authorized ⚭, ● RULE: Access Requires List or Pattern Match ⚭, ● PERMISSION: Enter the Event ⚭
○ PRE-CONDITION	An email not on the access list and not matching the access pattern.
○ INPUT	The user enters that email in the login dialog.
○ EXPECTED	The system denies access and shows a not-authorized notice.
○ POST-CONDITION	No session token exists and no authorization token was sent for the email.

URL Token Skips Login Dialog ⚓

Diagram



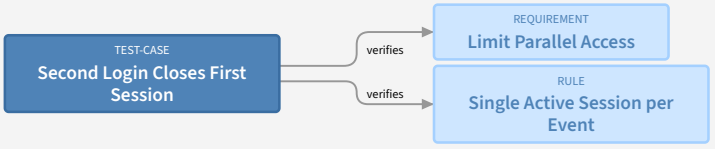
Properties

○ VERIFIES	● REQUIREMENT: Automatic Event Access URL ⚭, ● SCENARIO: Automatic Token in URL ⚭, ● RULE: Token Format and Validity ⚭, ● TRANSITION: consume ⚭

● TEST-CASE	Properties & Description	
	○ PRE-CONDITION	An event allows URL tokens and a pre-generated token exists for an invited email.
	○ INPUT	The user opens the event URL carrying that email and token.
	○ EXPECTED	The system issues a session token and grants access without showing the login dialog.
	○ POST-CONDITION	The user holds an active session token.

Second Login Closes First Session

Diagram



Properties

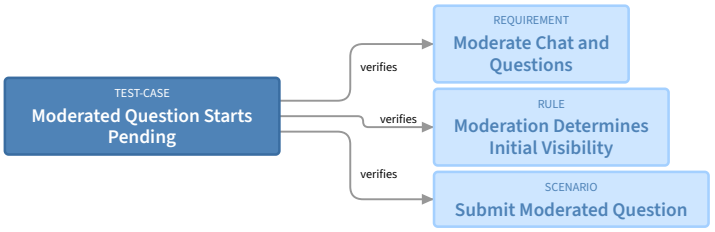
○ VERIFIES	● **REQUIREMENT: Limit Parallel Access**, ● **RULE: Single Active Session per Event**
○ PRE-CONDITION	A user holds an active session for an event from one device.
○ INPUT	The same user completes a login from a second device.
○ EXPECTED	The first connection is closed and only the new session remains active.
○ POST-CONDITION	Exactly one active session token exists for the user and event.

Description

1. The user watches the stream on the first device.
2. The same user completes a login with a fresh authorization token on a second device.
3. The tester observes the first device.

Moderated Question Starts Pending

Diagram

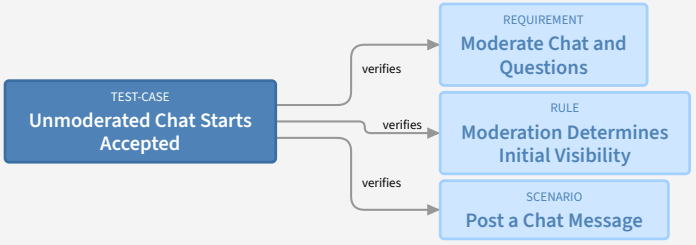


Properties

○ VERIFIES	● **REQUIREMENT: Moderate Chat and Questions**, ● **SCENARIO: Submit Moderated Question**, ● **RULE: Moderation Determines Initial Visibility**
○ PRE-CONDITION	An event with question moderation enabled is running.
○ INPUT	An attendee submits a question.
○ EXPECTED	The question is stored in state pending and is not visible to the audience.

Unmoderated Chat Starts Accepted

Diagram



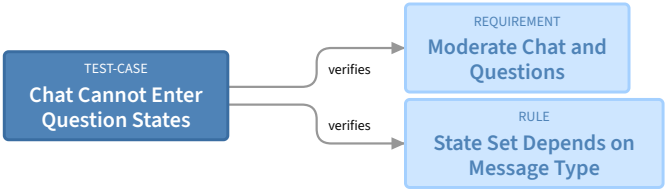
● TEST-CASE	Properties & Description

Properties

○ VERIFIES	● REQUIREMENT: Moderate Chat and Questions, ● SCENARIO: Post a Chat Message, ● RULE: Moderation Determines Initial Visibility
○ PRE-CONDITION	An event with chat moderation disabled is running.
○ INPUT	An attendee submits a chat message.
○ EXPECTED	The message is accepted at once upon creation and is immediately visible.

Chat Cannot Enter Question States

Diagram

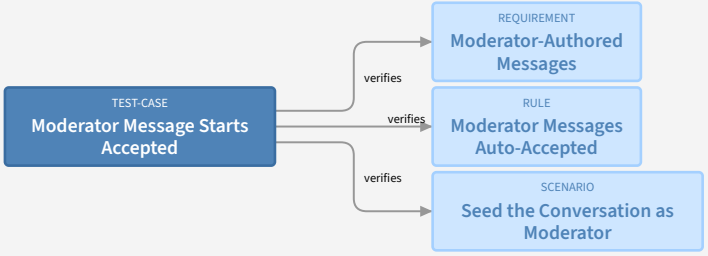


Properties

○ VERIFIES	● REQUIREMENT: Moderate Chat and Questions, ● RULE: State Set Depends on Message Type
○ PRE-CONDITION	An accepted chat message exists in a running event.
○ INPUT	A moderator attempts to forward the chat message to the presenter.
○ EXPECTED	The system refuses the transition, as a chat message permits only the states pending, accepted, and rejected.
○ POST-CONDITION	The chat message remains in state accepted.

Moderator Message Starts Accepted

Diagram

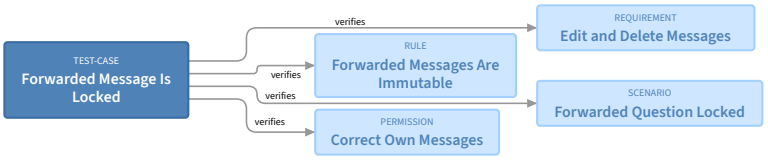


Properties

○ VERIFIES	● REQUIREMENT: Moderator-Authored Messages, ● SCENARIO: Seed the Conversation as Moderator, ● RULE: Moderator Messages Auto-Accepted
○ PRE-CONDITION	An event with chat moderation enabled is running.
○ INPUT	A moderator posts a chat message without overriding the sender name.
○ EXPECTED	The message is accepted at once upon creation with the sender name “Moderator” and is immediately visible.

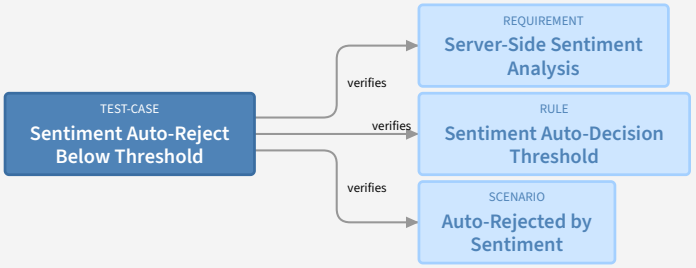
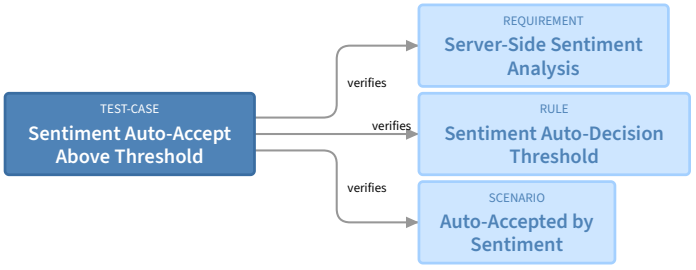
Forwarded Message Is Locked

Diagram



Properties

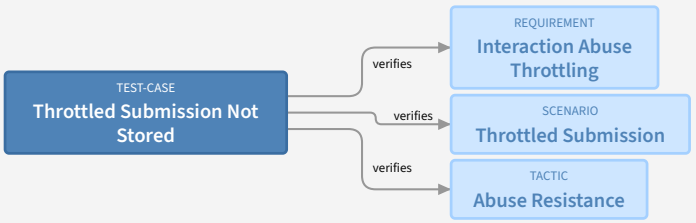
○ VERIFIES	● REQUIREMENT: Edit and Delete Messages, ● SCENARIO: Forwarded Question Locked, ● RULE: Forwarded Messages Are Immutable,

● TEST-CASE	Properties & Description	
		● **PERMISSION:** Correct Own Messages
	○ **PRE-CONDITION**	An accepted question of an attendee exists in a running event.
	○ **INPUT**	The original attendee attempts to edit and then to delete the forwarded message.
	○ **EXPECTED**	The system refuses both the edit and the deletion and the message remains unchanged.
	Description	
	1. The moderator forwards the accepted question to the presenter. 2. The original attendee attempts to edit the text of the question. 3. The original attendee attempts to delete the question.	
Sentiment Auto-Reject Below Threshold	Diagram	
		
	Properties	
	○ **VERIFIES**	● **REQUIREMENT:** Server-Side Sentiment Analysis, ● **SCENARIO:** Auto-Rejected by Sentiment, ● **RULE:** Sentiment Auto-Decision Threshold
	○ **PRE-CONDITION**	Server-side sentiment analysis and auto-reject are enabled.
	○ **INPUT**	An attendee submits a message scoring -0.5.
	○ **EXPECTED**	The system rejects the message at once upon creation, without a moderator decision.
Sentiment Auto-Accept Above Threshold	Diagram	
		
	Properties	
	○ **VERIFIES**	● **REQUIREMENT:** Server-Side Sentiment Analysis, ● **SCENARIO:** Auto-Accepted by Sentiment, ● **RULE:** Sentiment Auto-Decision Threshold
	○ **PRE-CONDITION**	Server-side sentiment analysis and auto-accept are enabled for a moderated event.
	○ **INPUT**	An attendee submits a question scoring +0.5.
	○ **EXPECTED**	The system accepts the question at once upon creation, bypassing the moderator decision.

● TEST-CASE	Properties & Description

Throttled Submission Not Stored

Diagram



Properties

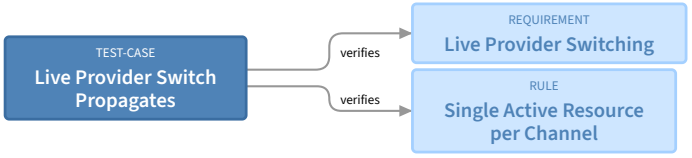
○ VERIFIES	● REQUIREMENT: Interaction Abuse Throttling, ● SCENARIO: Throttled Submission, ● TACTIC: Abuse Resistance
○ PRE-CONDITION	A running event with the default submission limit of 10 per user per minute.
○ INPUT	The attendee submits an eleventh question within the same minute.
○ EXPECTED	The system rejects the eleventh submission and informs the attendee to wait, but accepts the twelfth one after the minute has passed.
○ POST-CONDITION	The eleventh question is not stored, while the ten earlier ones remain unchanged and the twelfth one is stored.

Description

1. The attendee submits ten distinct questions within one minute.
2. The attendee submits an eleventh question before that minute has passed.
3. The attendee submits a twelfth question after the minute has passed.

Live Provider Switch Propagates

Diagram

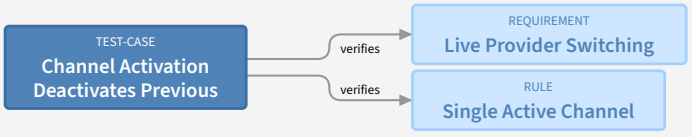


Properties

○ VERIFIES	● REQUIREMENT: Live Provider Switching, ● RULE: Single Active Resource per Channel
○ PRE-CONDITION	A running event with two configured resources and connected clients.
○ INPUT	The manager activates the second resource on the channel.
○ EXPECTED	All connected clients switch to the new stream without user interaction.
○ POST-CONDITION	Exactly one resource of the channel is active.

Channel Activation Deactivates Previous

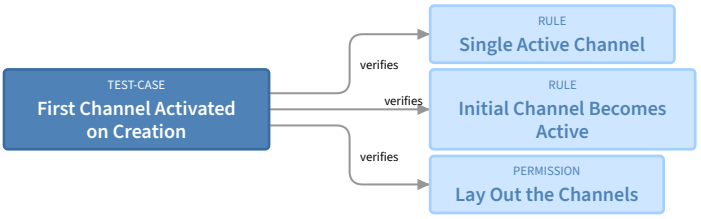
Diagram



Properties

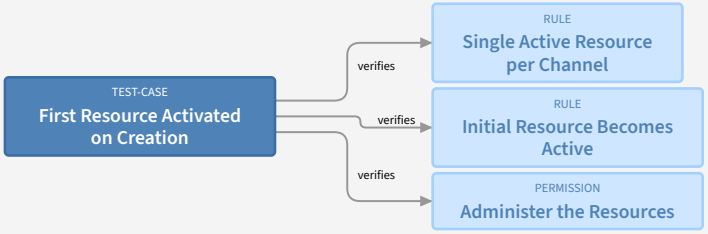
○ VERIFIES	● REQUIREMENT: Live Provider Switching, ● RULE: Single Active Channel
○ PRE-CONDITION	A running event with two configured channels, the first one active.
○ INPUT	The manager activates the second channel.
○ EXPECTED	The second channel becomes active and the first one is deactivated in the same step.

● TEST-CASE	Properties & Description	
	○ POST-CONDITION	Exactly one channel of the event is active.
First Channel Activated on Creation	Diagram	



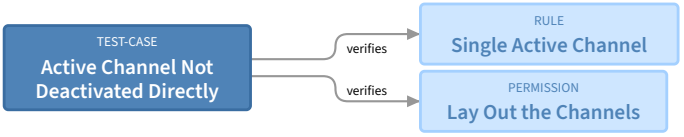
Properties	
○ VERIFIES	● RULE: Initial Channel Becomes Active, ● RULE: Single Active Channel, ● PERMISSION: Lay Out the Channels
○ PRE-CONDITION	A provisioned event without any channel.
○ INPUT	The manager creates a channel without marking it as active.
○ EXPECTED	The system stores the channel as the active one.
○ POST-CONDITION	Exactly one channel of the event is active.

First Resource Activated on Creation	Diagram



Properties	
○ VERIFIES	● RULE: Initial Resource Becomes Active, ● RULE: Single Active Resource per Channel, ● PERMISSION: Administer the Resources
○ PRE-CONDITION	A channel without any resource.
○ INPUT	The administrator creates a resource without marking it as active.
○ EXPECTED	The system stores the resource as the active one.
○ POST-CONDITION	Exactly one resource of the channel is active.

Active Channel Not Deactivated Directly	Diagram

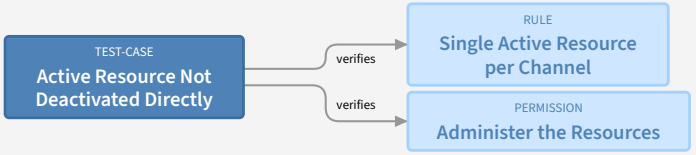


Properties	
○ VERIFIES	● RULE: Single Active Channel, ● PERMISSION: Lay Out the Channels
○ PRE-CONDITION	An event with two channels, the first one active.
○ INPUT	The manager deactivates the first channel directly, then deletes it.
○ EXPECTED	The system rejects both the deactivation and the deletion, as another channel would have to be activated first.
○ POST-CONDITION	The first channel still exists and is the only active channel of the event.

● TEST-CASE	Properties & Description

Active Resource Not Deactivated Directly

Diagram



Properties

○ VERIFIES	● RULE: Single Active Resource per Channel, ● PERMISSION: Administer the Resources
○ PRE-CONDITION	A channel with two resources, the first one active.
○ INPUT	The administrator deactivates the first resource directly, then deletes it.
○ EXPECTED	The system rejects both the deactivation and the deletion, as another resource would have to be activated first.
○ POST-CONDITION	The first resource still exists and is the only active resource of the channel.

Config Change Reaches Clients

Diagram

Properties

○ VERIFIES	● REQUIREMENT: Live Configuration Propagation
○ PRE-CONDITION	A running event with connected clients and chat disabled.
○ INPUT	The manager enables chat for the event.
○ EXPECTED	Connected clients show the chat panel within 2 seconds without a reload.

Registration Import Avoids Duplicates

Diagram

Properties

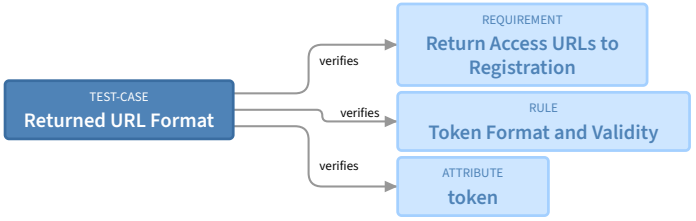
○ VERIFIES	● REQUIREMENT: Import Registered Attendees
○ PRE-CONDITION	A running event with an empty access list.
○ INPUT	The manager imports a second registration sheet overlapping the first one.
○ EXPECTED	The overlapping users are not duplicated, their tokens from the first import are returned unchanged, and only the new users receive fresh tokens.

Description

1. The manager imports a registration sheet with the emails A, B, and C.
2. The manager records the tokens returned for A, B, and C.
3. The manager imports a second registration sheet with the emails B, C, and D.

Returned URL Format

Diagram



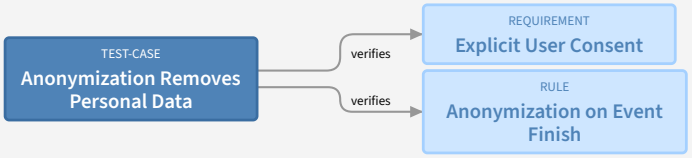
● TEST-CASE	Properties & Description

Properties

○ VERIFIES	● REQUIREMENT: Return Access URLs to Registration, ● RULE: Token Format and Validity, ● ATTRIBUTE: token
○ PRE-CONDITION	A registration import has generated tokens for new users.
○ INPUT	The manager exports the access URLs.
○ EXPECTED	Each URL contains event, user, and a six-digit “NNN-NNN” token in the URL column.

Anonymization Removes Personal Data

Diagram



Properties

○ VERIFIES	● REQUIREMENT: Explicit User Consent, ● RULE: Anonymization on Event Finish
○ PRE-CONDITION	A running event with messages, likes, tokens, and an access list.
○ INPUT	The manager finishes the event.
○ EXPECTED	Sender names become “Anonymous”, likes become bare counts, and tokens, users, and Moderator roles are deleted.
○ POST-CONDITION	No personal attendee data remains and Manager roles are retained.

User Vanishes Without Role or Access

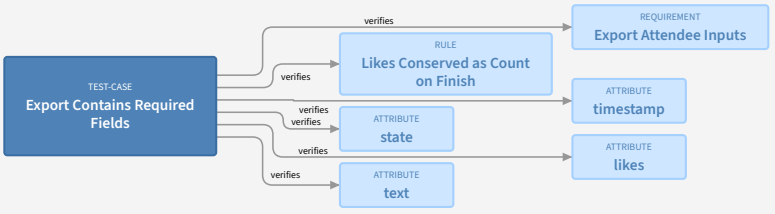
Diagram

Properties

○ VERIFIES	● RULE: No Permanent User Accounts
○ PRE-CONDITION	A user exists solely through the access list of a running event, holding no role.
○ INPUT	The manager removes the user from the access list.
○ EXPECTED	The user record is deleted together with the access list entry, as nothing else keeps it alive.
○ POST-CONDITION	No user record for the email address exists.

Export Contains Required Fields

Diagram



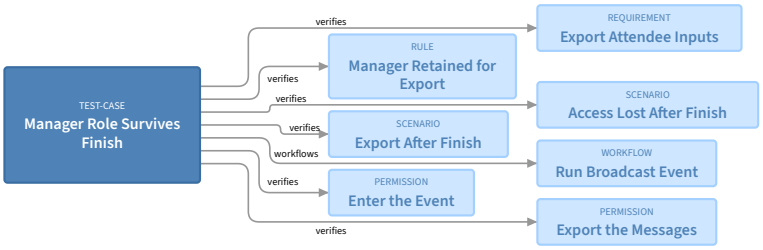
Properties

○ VERIFIES	● REQUIREMENT: Export Attendee Inputs, ● RULE: Likes Conserved as Count on Finish, ● ATTRIBUTE: timestamp, ● ATTRIBUTE: state, ● ATTRIBUTE: likes, ● ATTRIBUTE: text
○ PRE-CONDITION	A finished, anonymized event with messages.
○ INPUT	The manager exports the attendee inputs.

● TEST-CASE	Properties & Description	
	○ EXPECTED	The export includes at least timestamp, state, number of likes, and message text per message.

Manager Role Survives Finish ⚓

Diagram



Properties

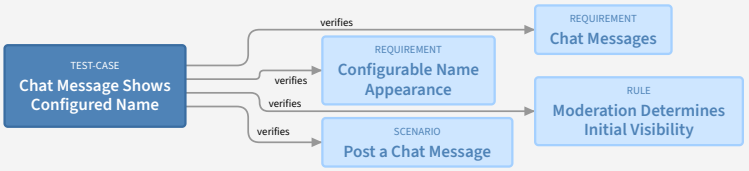
○ VERIFIES	● REQUIREMENT: Export Attendee Inputs ⚭, ● SCENARIO: Export After Finish ⚭, ● SCENARIO: Access Lost After Finish ⚭, ● RULE: Manager Retained for Export ⚭, ● PERMISSION: Export the Messages ⚭, ● PERMISSION: Enter the Event ⚭
○ WORKFLOWS	● WORKFLOW: Run Broadcast Event ⚭
○ PRE-CONDITION	A running event with a manager role and a moderator role assigned.
○ INPUT	The manager exports the event data after having finished the event.
○ EXPECTED	The moderator role is deleted by the anonymization while the manager role remains and can still export the event data.
○ POST-CONDITION	The manager role exists until the event is deleted.

Description

1. The manager finishes the event.
2. The moderator attempts to open the event.
3. The manager exports the attendee inputs of the finished event.

Chat Message Shows Configured Name ⚓

Diagram

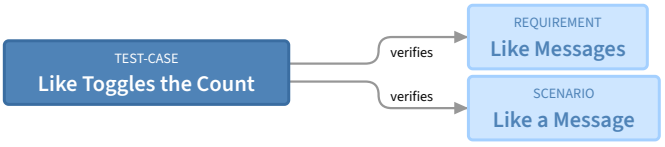


Properties

○ VERIFIES	● REQUIREMENT: Chat Messages ⚭, ● REQUIREMENT: Configurable Name Appearance ⚭, ● SCENARIO: Post a Chat Message ⚭, ● RULE: Moderation Determines Initial Visibility ⚭
○ PRE-CONDITION	A running event with chat enabled, chat moderation disabled, and the name appearance set to first name only.
○ INPUT	An attendee sends a chat message.
○ EXPECTED	The message appears in state accepted under the first name of the attendee, with the email address shown on hover.
○ POST-CONDITION	The message is stored as a chat in state accepted.

Like Toggles the Count ⚓

Diagram



● TEST-CASE	Properties & Description

Properties

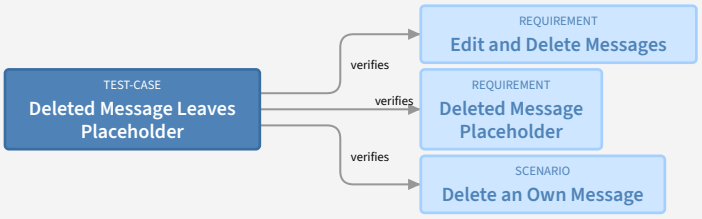
○ VERIFIES	● REQUIREMENT: Like Messages, ● SCENARIO: Like a Message
○ PRE-CONDITION	A visible chat message of another attendee with a like count of 0.
○ INPUT	An attendee likes the message and then undoes the like.
○ EXPECTED	The like count reads 1 after the like and 0 again after the undo.
○ POST-CONDITION	No like of the attendee is recorded for the message.

Description

1. The attendee likes the message.
2. The tester reads the like count shown on the message.
3. The attendee undoes the like on the same message.
4. The tester reads the like count shown on the message again.

Deleted Message Leaves Placeholder

Diagram



Properties

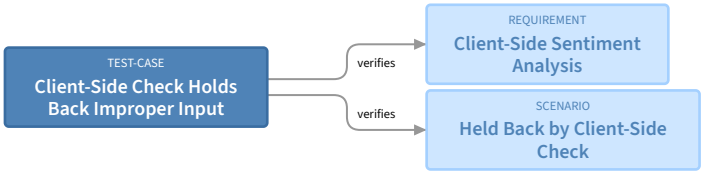
○ VERIFIES	● REQUIREMENT: Edit and Delete Messages, ● REQUIREMENT: Deleted Message Placeholder, ● SCENARIO: Delete an Own Message
○ PRE-CONDITION	A running event with chat enabled and chat moderation disabled, with two attendees watching.
○ INPUT	The first attendee deletes their own message from between the two other messages.
○ EXPECTED	The stream keeps the position of the message and shows a “This message was deleted” placeholder instead of its text on both attendee screens.
○ POST-CONDITION	The message text is no longer retrievable by any attendee.

Description

1. The second attendee posts a chat message.
2. The first attendee posts a chat message.
3. The second attendee posts another chat message.
4. The first attendee deletes their own message.
5. The tester inspects the stream on the screens of both attendees.

Client-Side Check Holds Back Improper Input

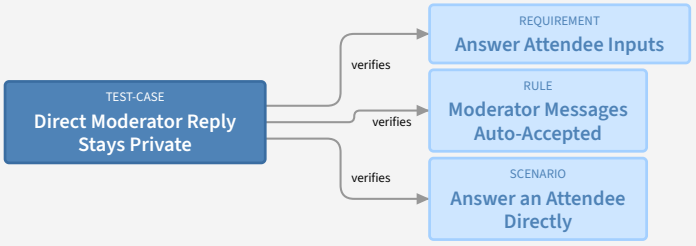
Diagram



Properties

○ VERIFIES	● REQUIREMENT: Client-Side Sentiment Analysis, ● SCENARIO: Held Back by Client-Side Check
○ PRE-CONDITION	A running event with client-side sentiment analysis enabled.
○ INPUT	An attendee submits a question with clearly improper wording.

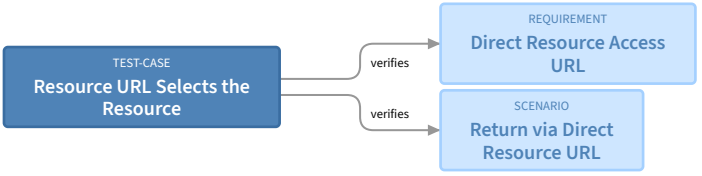
● TEST-CASE	Properties & Description	
	○ EXPECTED	The client prevents the submission and informs the attendee, and no request reaches the server.
	○ POST-CONDITION	No message exists for the submission.
Direct Moderator Reply Stays Private ⚓	Diagram	



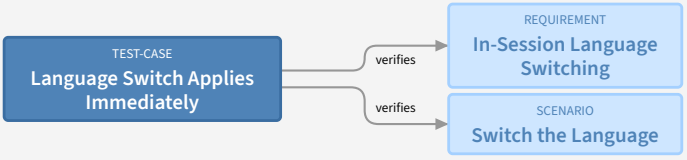
	Properties	
	○ VERIFIES	● **REQUIREMENT:** Answer Attendee Inputs ⚭, ● **SCENARIO:** Answer an Attendee Directly ⚭, ● **RULE:** Moderator Messages Auto-Accepted ⚭
	○ PRE-CONDITION	A running event with chat enabled and two attendees watching the stream.
	○ INPUT	The moderator replies to the message of the first attendee as a direct message.
	○ EXPECTED	The reply reaches the first attendee in state accepted while the second attendee does not see it.
	○ POST-CONDITION	The reply is stored as accepted with the first attendee as its sole recipient.
	Description	

1. The first attendee posts a chat message.
2. The moderator replies to that message as a direct message.
3. The tester inspects the stream on the screens of both attendees.

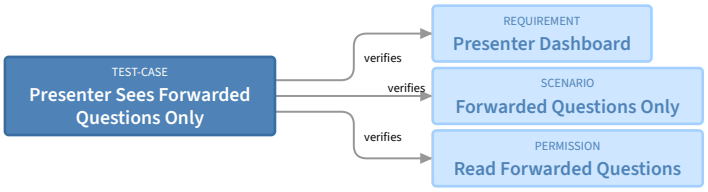
Resource URL Selects the Resource ⚓	Diagram	



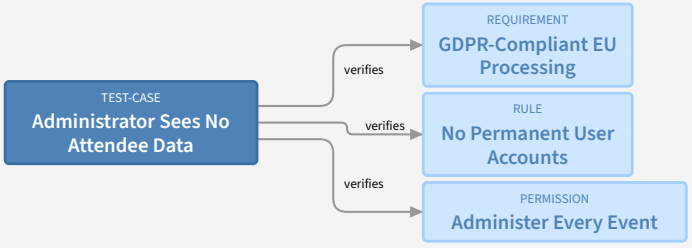
	Properties	
	○ VERIFIES	● **REQUIREMENT:** Direct Resource Access URL ⚭, ● **SCENARIO:** Return via Direct Resource URL ⚭
	○ PRE-CONDITION	A running event with a default stream and a second, static resource, and an attendee holding an active session.
	○ INPUT	The attendee opens the event URL carrying the resource parameter naming the static resource.
	○ EXPECTED	The system shows the static resource instead of the default stream.
	○ POST-CONDITION	The session of the attendee remains active.
Language Switch Applies Immediately ⚓	Diagram	



● TEST-CASE	Properties & Description
	Properties
	○ VERIFIES	● REQUIREMENT: In-Session Language Switching, ● SCENARIO: Switch the Language
	○ PRE-CONDITION	An attendee watches a running event with the interface in German.
	○ INPUT	The attendee switches the language to English in the header.
	○ EXPECTED	The interface and the translated messages appear in English without a page reload.
	○ POST-CONDITION	The stream continues uninterrupted.
Presenter Sees Forwarded Questions Only	Diagram



	Properties
	○ VERIFIES	● REQUIREMENT: Presenter Dashboard, ● SCENARIO: Forwarded Questions Only, ● PERMISSION: Read Forwarded Questions
	○ PRE-CONDITION	A running event holds one question in each of the states pending, accepted, rejected, and forwarded.
	○ INPUT	The presenter opens the presenter view of the event.
	○ EXPECTED	The view lists the forwarded question only, while a direct request for any of the three other questions is refused.
	○ POST-CONDITION	The four questions remain in their states.
Administrator Sees No Attendee Data	Diagram

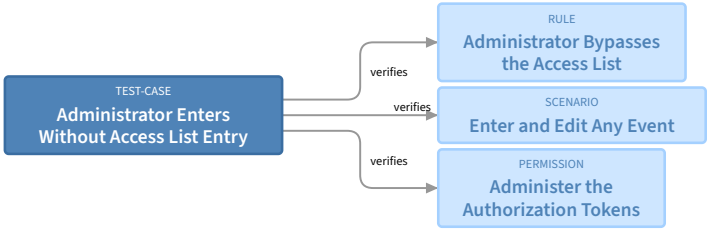


	Properties
	○ VERIFIES	● REQUIREMENT: GDPR-Compliant EU Processing, ● RULE: No Permanent User Accounts, ● PERMISSION: Administer Every Event
	○ PRE-CONDITION	A running event with a populated access list and messages exists, provisioned through the configuration.
	○ INPUT	The administrator reads the event through the configuration and requests its access list and messages.
	○ EXPECTED	The event settings are returned, while the requests for the access list and the messages are refused.
	○ POST-CONDITION	No access list entry or message was disclosed to the administrator.

● TEST-CASE	Properties & Description

Administrator Enters Without Access List Entry

Diagram



Properties

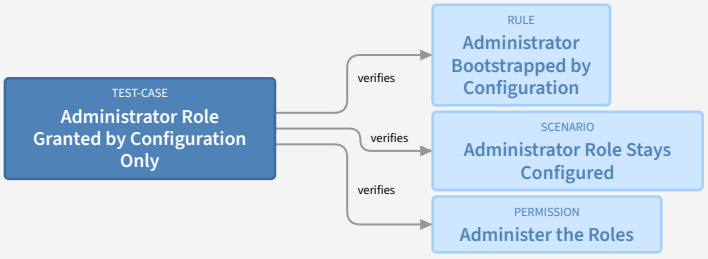
○ **VERIFIES**	● **RULE:** Administrator Bypasses the Access List, ● **SCENARIO:** Enter and Edit Any Event, ● **PERMISSION:** Administer the Authorization Tokens
○ **PRE-CONDITION**	A running event whose access list does not contain the configured administrator email address and whose access email pattern does not match it.
○ **INPUT**	The administrator requests the login challenge for the configured administrator email address at the event URL and returns the received token.
○ **EXPECTED**	The token is issued and sent without consulting the access list, and the administrator is admitted to the event.
○ **POST-CONDITION**	The administrator holds an active session for the event while its access list is unchanged.

Description

1. The administrator opens the event URL and enters the configured administrator email address.
2. The tester verifies that a token was sent to that address.
3. The administrator returns the token.
4. The tester inspects the access list of the event.

Administrator Role Granted by Configuration Only

Diagram



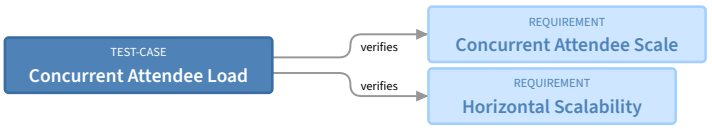
Properties

○ **VERIFIES**	● **RULE:** Administrator Bootstrapped by Configuration, ● **SCENARIO:** Administrator Role Stays Configured, ● **PERMISSION:** Administer the Roles
○ **PRE-CONDITION**	The configuration grants the Administrator role to exactly one email address, and its holder has entered an event.
○ **INPUT**	The administrator attempts to grant the Administrator role to a user of the event and to revoke their own Administrator role through the user interface.
○ **EXPECTED**	Both attempts are refused, while granting the Manager role to the same user succeeds.
○ **POST-CONDITION**	The set of Administrator roles equals the configured one.

Description

1. The administrator attempts to grant the Administrator role to a user of the event.

● TEST-CASE	Properties & Description
	2. The administrator attempts to revoke their own Administrator role. 3. The administrator grants the Manager role to the same user. 4. The tester inspects the roles of the administrator user and of the event user.
Concurrent Attendee Load ⚓	Diagram



Properties	
○ **VERIFIES**	● **REQUIREMENT:** **Concurrent Attendee Scale** ⚭, ● **REQUIREMENT:** **Horizontal Scalability** ⚭
○ **PRE-CONDITION**	A running event deployed with scaled proxy, relay, and server instances.
○ **INPUT**	10000 simulated attendees connect simultaneously and interact.
○ **EXPECTED**	All 10000 attendees hold a served WebSocket connection at the same time, meeting the upper bound of the 2500 to 10000 attendees per event metric.