Debat Direct architecture

Tweede Kamer - Debat Direct app

Application version:
Production: 11.8.15
Latest: 11.10.4

Document version:
2.5.0

Latest update:
13 March 2026

Classificatie: TLP:AMBER

Index

General description

Debat Direct is a (mobile) website and app for iOS, Android, Android TV, Fire TV and Apple TV (tvOS). The most important features are:

- Watch or listen to all the public debates live, with an option to start from the beginning or on demand.
- View this week and today's meetings and subjects in a visual overview and agenda listing.
- Set push notifications on both speakers and debates of interest.
- Get on screen speaker information and on screen notifications.
- Navigate to earlier speakers in a debate.
- See which representatives are attending a debate.
- Click on representatives, Chamber and Committee Rooms for additional information.
- View votings during live and VOD debates
- Download relevant documents.
- Share a link to a debate, speaker or voting on social media or embed the (live) debate on your own website.
- Download a debate or a video fragment.
- Search debates and speech moments.

  Here is an FAQ overview of the all the features and any limitations:
  https://debatdirect.tweedekamer.nl/support/

Frontend

- Debat Direct is a modular React application written in ES6 & TypeScript that runs as a Capacitor App, React Native TV App and as a (mobile) website.
- The Android TV and Apple TV apps are in a separate TV codebase.
- It uses both open source and custom made components.
- For on-demand and live streaming video playback, with support for HLS and Program-Date-Time synchronisation, the proprietary Dolby OptiView Video Player is used as an HTML5 video player (license required).
- The Debat Direct websites, embedded players and the iOS / Android apps rely on availability of multiple backend services (e.g. RESTful API's) that are described below.

Backend

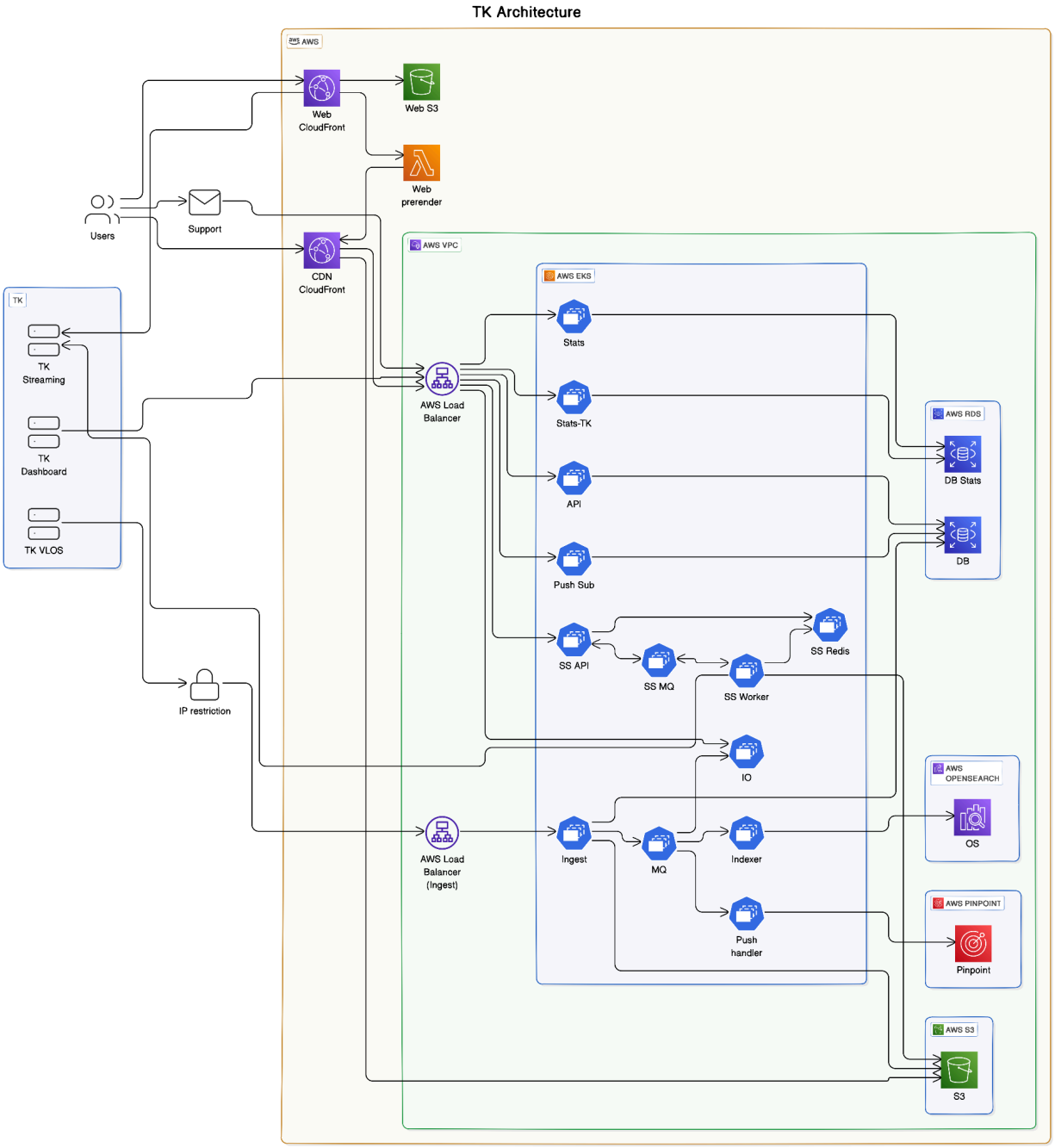
- Debat Direct backend services are built and maintained specifically for Debat Direct and are developed as NodeJS or Go applications.
- All services run in Kubernetes clusters (AWS EKS) or separate AWS services such as AWS OpenSearch, AWS CloudFront, AWS RDS, AWS Lambda, AWS Cognito & AWS Pinpoint.
- The Kubernetes clusters are configured to be in multiple Availability Zones within the AWS Frankfurt region.
- All services run SSL, are IPv6 compatible and need to be set up redundantly (as a best practice).
- DNSSEC is enabled for all domains.
- Most services rely on auto-scaling or auto-schedule capabilities of the (virtual) hosting provider.
- All public-facing endpoints are routed/distributed via CDN's (AWS Cloudfront/Bunny CDN).
- No third party (SAAS) services are used (for tracking); only Sentry (with Sentry Relay) is used for error reporting in agreement with the House of Representatives (Tweede Kamer).
- The backend services are highly dependent on the real time VLOS data (generieke koppelvlak / generic interface) and the live streams (HLS TS, multiple bitrates up to Full HD) provided by the House of Representatives (Tweede Kamer)/ Arbor Media.
- MP4 and MXF (medium) generation of stream with in and out points based on HLS m3u8 manifest.
- For Kamerbeelden MXF high downloads can be created via an integration with Arbor.

System management tasks

- Submit iOS, tvOS. Android, Android TV, and Fire TV (updates) to the relevant stores.
- Keep the Debat Direct app compliant with latest app store guidelines.
- Keep Debat Direct working with the latest OS and browser versions.
- Keep Debat Direct compliant with accessibility guidelines (web guidelines / webrichtlijnen WCAG 2AA).
- Periodic server management including monthly security updates.
- Periodic security audits reporting.

- Respond to CVE vulnerability alerts and work together with the TK security department on structural security improvements.
- Minor app and backend updates to stay compatible with the latest VLOS application.
- Make sure that ‘Sprekersmarkeringen’ stays <0.2 seconds accurate.
- Provide automated device lab and manual test reports with each release.
- 24/7 monitoring and troubleshooting (according to SLA) via Betterstack and AWS CloudWatch.
- Continuous monitoring and standards-compliant validation of live and on-demand HLS/DASH streams.
- Javascript services dependencies checks with package managers (outdated/audit) and GitHub dependabot
- Code quality checks with SonarLint and SonarCloud
- Run License checks with https://www.npmjs.com/package/js-green-licenses (Javascript)
- Penetration and vulnerability tests are executed yearly.
- Maintain code in GitHub and submit exports to escrow (at least yearly)

System architecture



Debat Direct services

Available environments:

- Test (test.debatdirect.nl)
- Acceptance (acc.debatdirect.nl)
- TK Acceptance (tk-acc.debatdirect.nl) only used for testing TK VLOS Acceptance
- Production (debatdirect.tweedekamer.nl / debatdirect.nl)

DNS routing for debatdirect.nl is managed via AWS Route53. The production domain tweedekamer.nl DNS is managed by the Tweede Kamer. All domains have DNSSEC enabled. All environments except production are IP restricted using AWS WAF.

Description of backend services

Ingest

The Ingest service's main responsibility is to receive Tweedekamer debate data in a secure way. Debate data consists of agenda-data such as starting times and venues, actor-data such as tweedekamer-leden and photo's, and debate-updates like the starting of a debate or when a politician gets the floor or gets interrupted. The Ingest service allows Tweedekamer to insert this data into Debat Direct as soon as this data becomes available. For this insertion, Tweedekamer uses a modified version of its own VLOS data format (SecondScreen XSD XML).
Another responsibility of the Ingest service is to let other services know that new data was received.
(NodeJS)

API

The API allows the Debat Direct user applications (Website and mobile apps) access to the debate data that was received through the Ingest service, but also some static data like information on locations and frequently asked questions. Support requests are created in a support ticket system. With straightforward HTTP requests, the user applications can demand the actual state of agenda- and debate-data at any time. Endpoints are cached by Varnish.
(Nodejs)

Stats

Specific user actions are recorded and sent to the Stats service. This makes it possible to analyze which parts of the application are used most or least extensively, and to draw conclusions from overall application use over time. The Stats service allows administrators and developers to get insight into the use of the Debat Direct user applications.
(Matomo)

I/O

Where the API allows the Debat Direct user applications to do requests on the actual state of debates, the I/O service allows the backend system to actively inform the user applications of updates. As soon as a user opens the application, a socket connection is set up between application and I/O server. This connection allows the backend system to notify the application as soon as a new update was received by Ingest. For example, this service is responsible for keeping the list of ‘sprekers’ up to date during a debate.
(NodeJs)

Message Queue

The message queue enables multiple backend services to communicate with each other.
For example, this service allows the Ingest service to let the Push handler service know that a debate has started, so that users can be notified. Or to let the I/O service know that it should inform all Debat Direct users following a certain debate, that a politician is interrupted.
(RabbitMq)

DB

This is the database service storing application data, both static and dynamically ingested.
(AWS Aurora MySQL Compatible)

Web

The web (browser) version of the Debat Direct application is hosted with AWS Cloudfront and responds to social media bots to serve relevant metadata for the requested page in static form if users share a debate or speech moment. Prerendering for SEO/crawler optimization is done with AWS Lambda NodeJS functions.
(React / AWS Lambda NodeJS /AWS CloudFront)

Push subscriber

Maintains user subscriptions to be notified when certain events occur, like a debate starting or a politician taking the floor. These subscriptions are stored in AWS Pinpoint Endpoints.
(Go / AWS Pinpoint)

Note: AWS will end support for AWS Pinpoint on October 30, 2026 and needs to be replaced in order to continue supporting Push notifications in the Debat Direct applications.

Push handler

The push handler starts AWS pinpoint campaigns based on triggers from the Ingest service. The trigger contains information to create a pinpoint segment which is a filter on pinpoint Endpoints that were created by the Push subscriber. The campaign will then send users notifications based on their subscriptions. AWS Pinpoint handles the connection to FCM (Firebase Cloud Messaging) and APNS (Apple Push) which deliver the notifications to user devices.
(Go / AWS Pinpoint)

Note: AWS will end support for AWS Pinpoint on October 30, 2026 and needs to be replaced in order to continue supporting Push notifications in the Debat Direct applications.

Stream Share

MP4 / MXF generation of streams with in and out points based on HLS m3u8 manifest. MP4 files are created internally in this service using FFmpeg. MXF medium (MPEG2, 1920 × 1080, 50 Mbps) files are created with AWS MediaConvert. MXF high is managed via an Arbor API service.
(Nodejs/ffmpeg/AWS MediaConvert)

Search

The search service enables the web and iOS/Android applications to search for debates and speech moments. The search requests are handled by a search API function to abstract search calls and make it easier for the website and applications to perform search operations. Responses are cached by AWS CloudFront. The connection runs through an authorization endpoint so outside calls can only read data stored in elasticsearch/opensearch.
(Nodejs / AWS Elasticsearch/Opensearch / AWS CloudFront)

Auth

For authentication of Kamerbeelden users, we integrate with Amazon Cognito. Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.
The Debat Direct website has implemented the AWS SDKs and login UI so that users can log in directly from the Debat Direct website. (AWS Cognito)