Smart pi ECOsystem



Smart Public Information for the smart city.

Smart pi takes information from across the city and delivers it to the people who need it: as an announcement, on a display, through a help point or straight to a hearing aid, the moment it matters.

One ecosystem, designed to look, feel and work as one

Every product works on its own, and better together.

IN

Where information comes from

- RTA
- MIC
- PBX
- SPL
- Fire panel

SMART PI CORE

Decides what to say, where and when

- DVA
- DVA-GTFS
- EYE
- DSP
- SIP
- FSM
- LOG
- NMS

OUT

How it reaches people: hear, see, ask

- NAM
- NAM-HP
- EDC
- EYE-HD
- HLP

Smart pi DVA-GTFS

Automatic announcements

Turn timetable and realtime changes into the announcements passengers need.

GTFS

From service data to spoken information

Let the service information drive the announcement, so passengers hear the change when it matters.

Smart pi DVA GTFS reads GTFS timetables and realtime updates, evaluates the business rules for each stop and sends announcement orders to Smart pi DVA.

Routine arrivals, delays, cancellations and next-service updates follow the same configured workflow. Operators can inspect the service state and trace the rule behind an order.

Before arrival

Prepare passengers for the next service.

When plans change

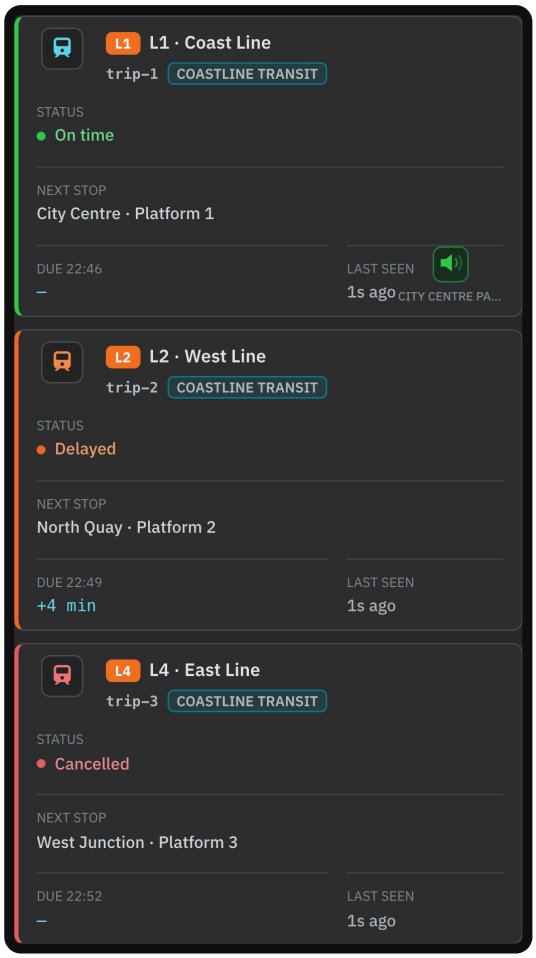
Explain delays and cancellations.

Between services

Repeat useful next-departure information.

During review

Find what triggered the announcement.



Follow a service
A service's status, next stop and update age in the mobile view.

29
Trigger choices in the current rule editor.

2 s
Rule evaluation interval in the current processor.

16
CPU cores in the published starting server specification.

100 GB
Storage in the published starting server specification.

Current application views with fictional sample data. Coastline Transit is an illustrative network. Server requirements depend on the deployment.

A clear chain from feed to channel

Separate the data, decision and playback roles so each handover can be checked.

READ
GTFS
Timetable, stops and routes.

→

UPDATE
Realtime
Expected times and service changes.

→

DECIDE
Business rules
Evaluate stop, trigger and timing.

→

BUILD
Audio actions
Assemble the announcement order.

→

DELIVER
Smart pi DVA
Play on the assigned channels.

OBSERVE
Status and order history
Check service state, inspect a trip and review the rule and destination behind an order.

COORDINATE
Smart pi EYE
Send configured text, messages or mode changes.

- DVA GTFS decides what to order. DVA manages audio playback on the destination channels.

Keep the handovers explicit

- **Feed selection:** bind a rule to the intended timetable and stop.
- **Channel assignment:** connect the rule to the correct DVA destination.
- **Priority:** agree how automatic orders share playback with operator messages.

Other feed formats

Smart pi RTA can convert supported SIRI or custom data sources into GTFS for the announcement workflow. Confirm the source mapping and update behaviour for the actual feed.

Better together

Smart pi RTA
Prepare the timetable and realtime data.

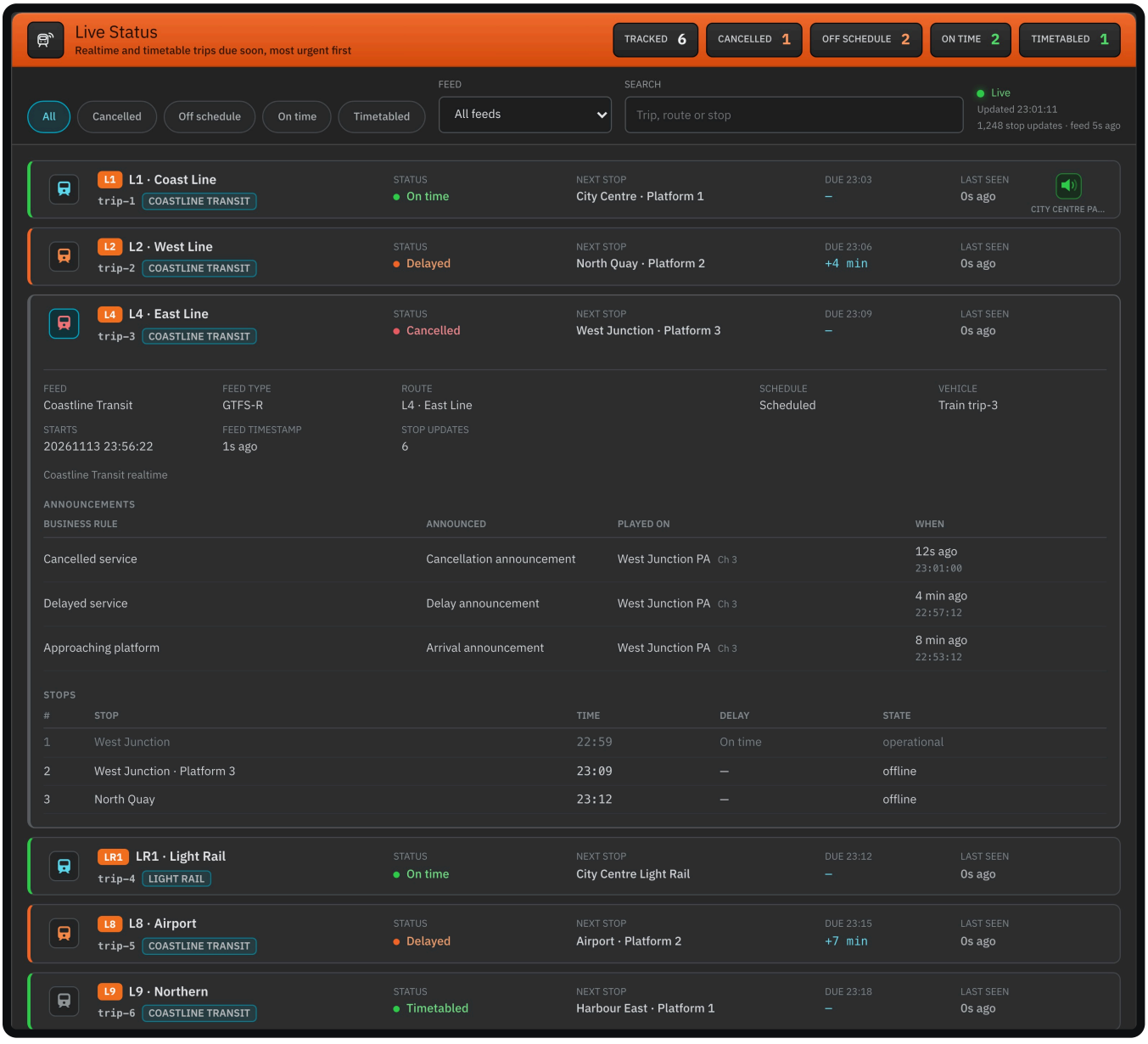
Smart pi DVA
Queue and play the announcement audio.

Smart pi EYE
Coordinate spoken and visual information.

Smart pi SIP
Request live service information for help-point callers.

The service. The rule. The announcement.

Expand a service to see which rules triggered, what they announced, where and when.



Service state and announcement history in one window
The selected service shows arrival, delay and cancellation rule outcomes together.

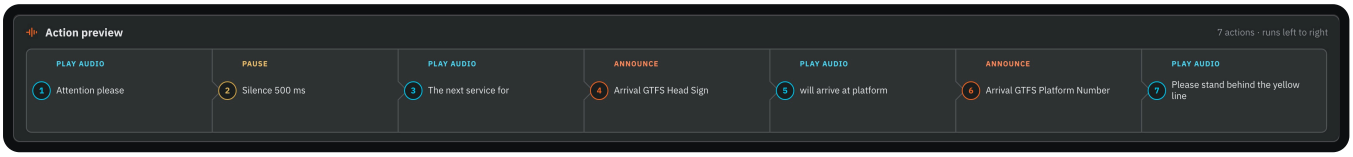
Read the decision alongside the service

Business rule → announcement type → DVA node and channel → time. This view shows the available history from the application's current run; use Order Logs for the retained order history.

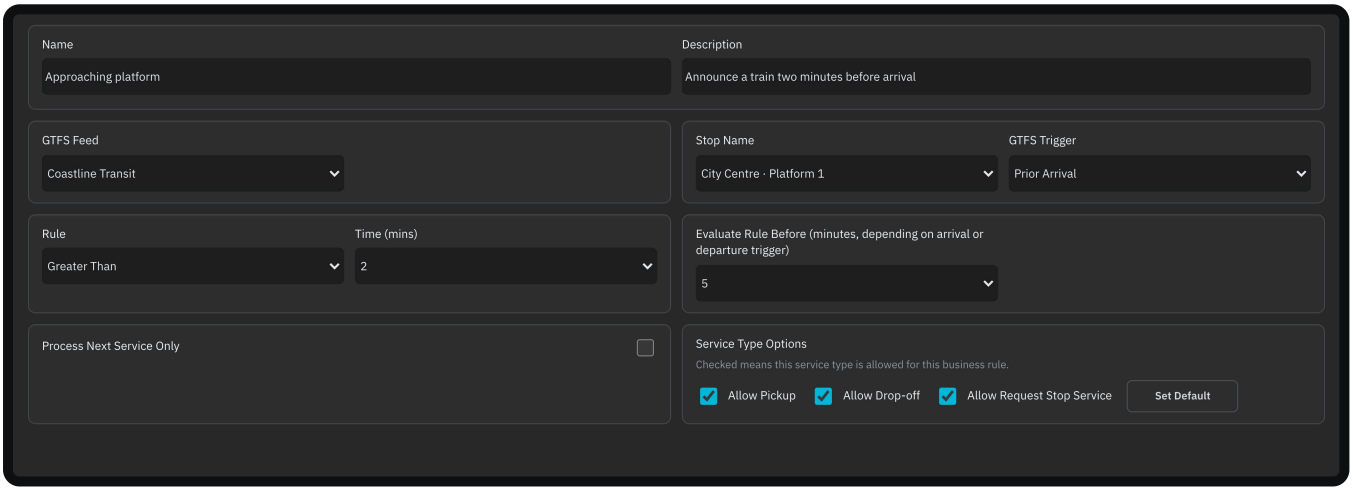
Current application views with fictional Coastline Transit data. History shown is illustrative.

One condition. A useful passenger message.

Build rules around the event at the stop, then connect them to the channels that should announce it.



See the announcement being assembled
Recorded audio, a deliberate gap and GTFS-derived speech appear in their configured order.



Define the condition
Feed, stop, trigger, comparison and time window identify when the rule applies.

Start with a specific stop

Choose the feed and stop before selecting the trigger. An arrival rule and a departure rule refer to different events, so their timing needs to match the operating intent.

Make the boundary deliberate

Use the comparison, time and evaluation window appropriate to the trigger. Test services just inside and outside the boundary.

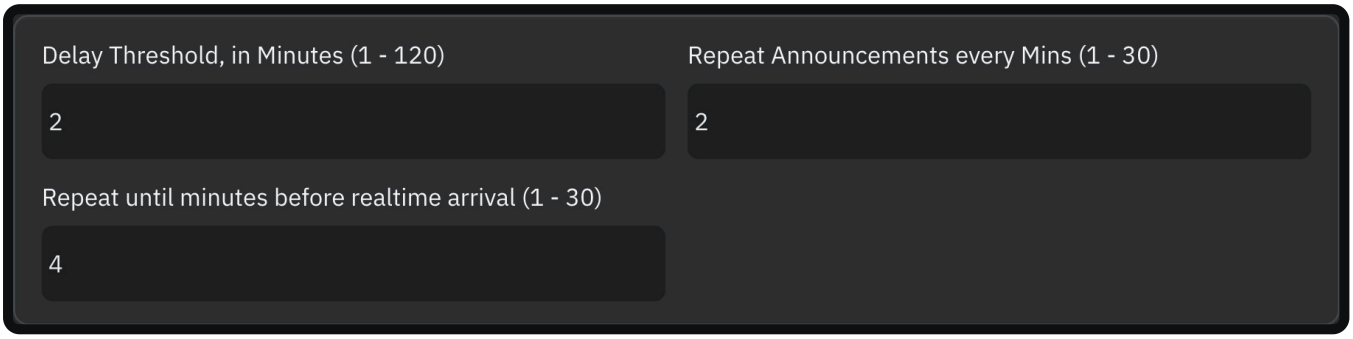
From a rule to a destination

Assign the rule to a DVA node and channel, with the required priority. Enabling a rule and assigning its output are separate parts of the setup.

Current application views with fictional sample data. Coastline Transit is an illustrative network.

Keep the update current. Know when to stop.

A repeated delay message needs to follow the service, not a stale prediction.



Repeat controls for a delayed arrival
Set the qualifying delay, repeat period and stop-before boundary.

1 · Qualify

Check whether the service meets the delay condition and is still in the evaluation window.

2 · Refresh

Before a repeat, retrieve the current stop-time information for the service.

3 · Recheck

Only create another order while the repeat remains valid and before the configured stop boundary.

Use the latest expected time

Repeats refresh the service state rather than replaying the original timing indefinitely. If the refreshed service cannot be found, the repeat is skipped and logged.

Stop before it becomes unhelpful

The stop-before window is calculated from the realtime event time. A repeat is not scheduled when its next due time would reach that boundary.

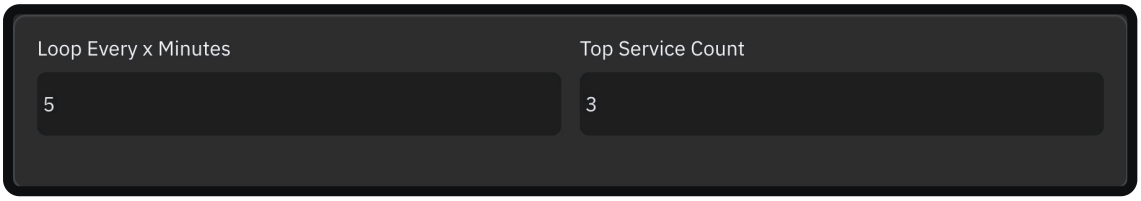
Example · a changing delay

A service is initially four minutes late. Set repeat announcements at a useful interval, then verify what happens when the prediction moves, the delay clears or the service reaches its stop-before window.

Current application views with fictional sample data. Coastline Transit is an illustrative network. Example policy only; the actual timings are configured per rule.

A useful summary. Even when the trip changes.

Cover the regular wait as well as the less usual service pattern.



Regular next-service announcements
Choose the repeat interval and the number of services to include.

Give the waiting passenger a summary

Loop announcements can provide regular next-service information with “due in” times. Choose how often to repeat and how many of the next services to include.

Focus on the next service

The next-service-only option can restrict applicable rules at a busy stop, helping avoid repetitive calls for later services.

When the service changes shape

Headsign changes, splits and joins have dedicated trigger choices. Trip transformations can suppress normal triggers and use alternate announcement audio for the affected services.

Test the passenger’s journey

Changes destination

Check that the revised headsign is reflected in the message.

Splits into services

Check which platforms and service legs need the alternate wording.

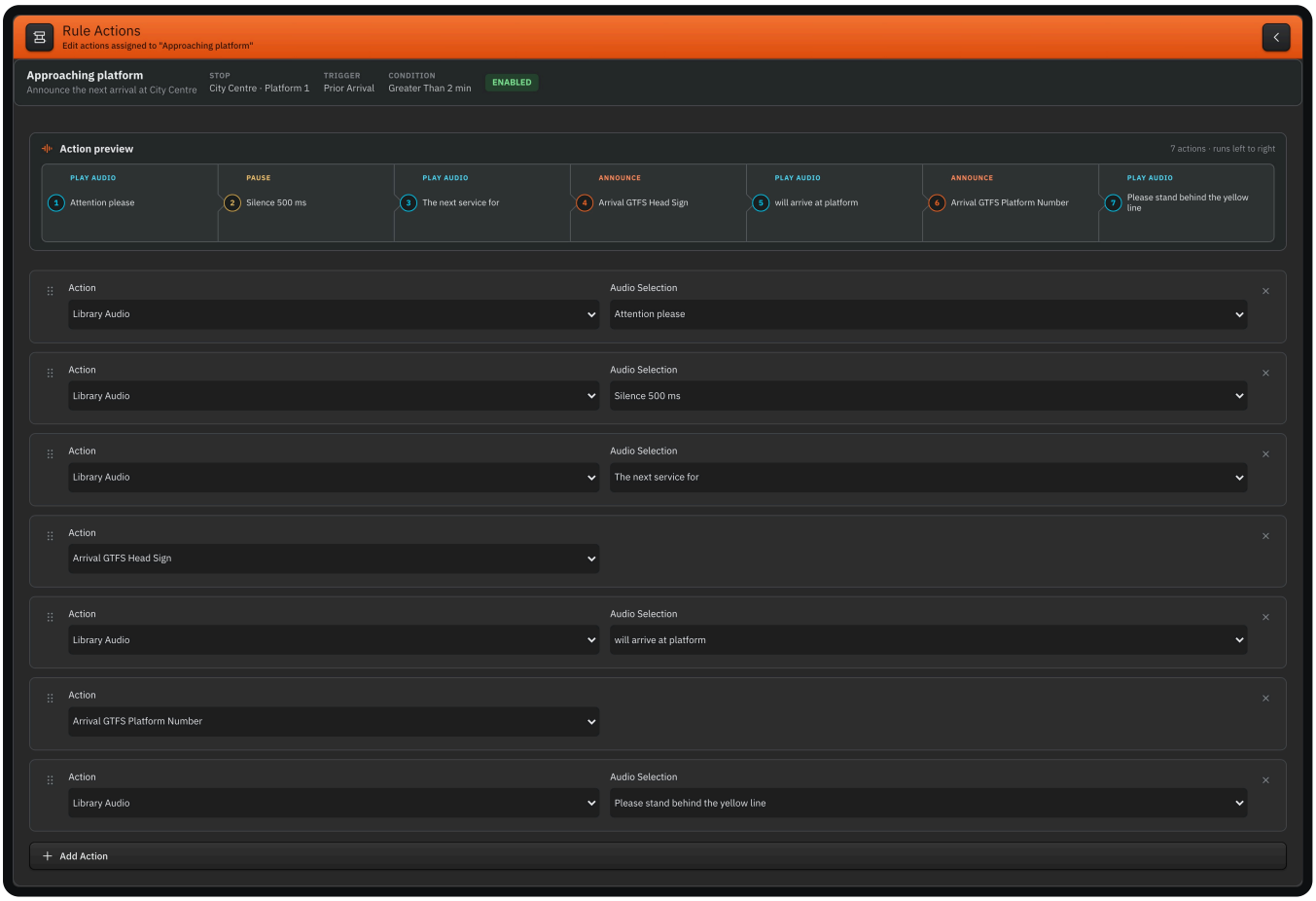
Joins another service

Check suppression of normal calls so the combined service is described consistently.

Current application views with fictional sample data. Coastline Transit is an illustrative network. Transformation behaviour is described from the product source; it is not executed by this demo.

Build the message. See every step.

Combine recorded speech, timed gaps and live service details in one visual sequence.



The Rule Actions editor
The sequence preview stays above the actions that build it. Drag handles change their order.

Play. Pause. Announce.

Choose a library clip, insert a silence clip for a gap, then add service-data actions such as the arrival headsign or platform number.

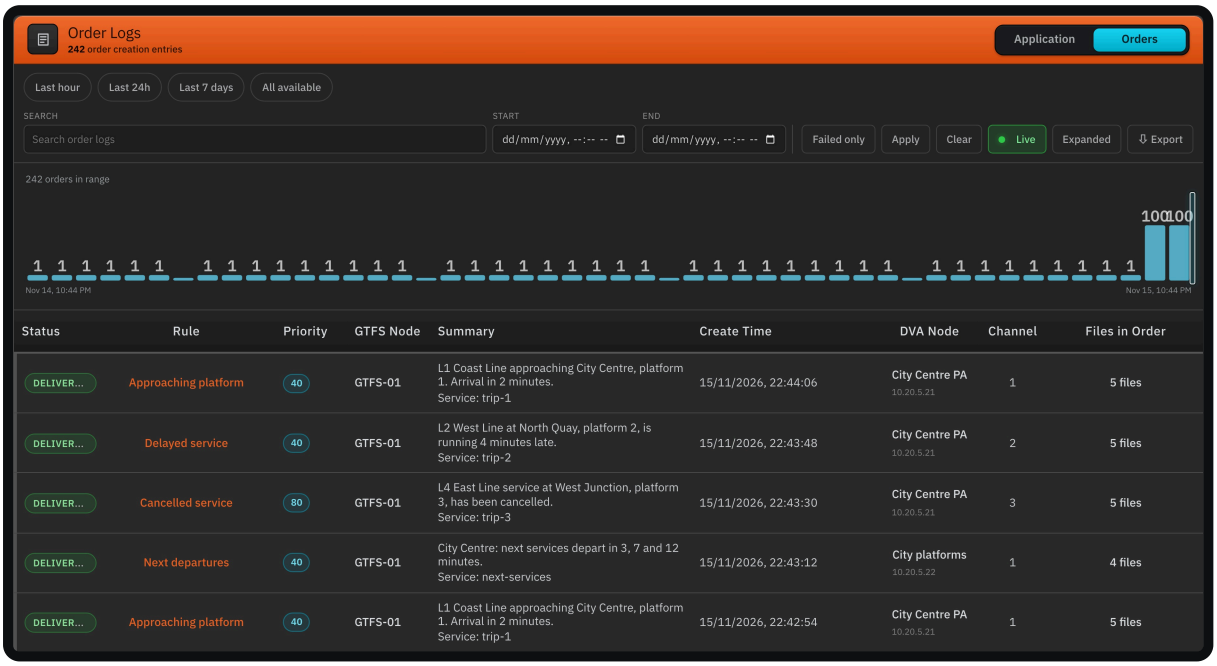
See the sentence before it plays

The preview links each numbered step to its editable action. Change the order or selection, then review the complete message with representative feed data.

Current application views with fictional Coastline Transit data. History shown is illustrative. Native Rule Actions view rendered with sample library options.

What was announced? Which rule triggered it?

Make the decision traceable, from service conditions to the destination of the order.



Order creation history
Timeline and entries link the triggering rule to priority, trip, DVA node, channel and audio files.

Start with the question

Find a service or message in the order history, then inspect the rule, channel and dispatch time. Use date ranges and failed-only filtering to narrow the investigation.

Follow the sequence

Compare the order's audio files with the intended rule actions. Export the selected history for a commissioning or support review.

An order is not an acoustic measurement

A delivered order records the application's dispatch result. Confirm DVA playback and the audio path separately when proving what reached a passenger area.

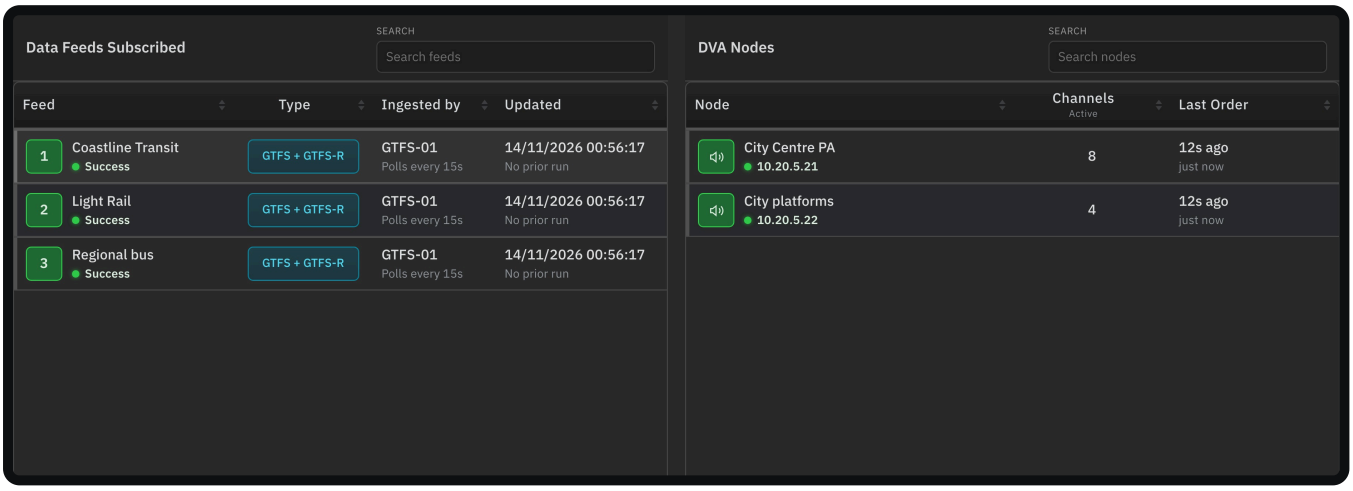
Keep application logs beside order history

Order logs explain the announcement request. Application logs help investigate feed, processing and delivery issues around it. Agree retention and cleanup settings during setup.

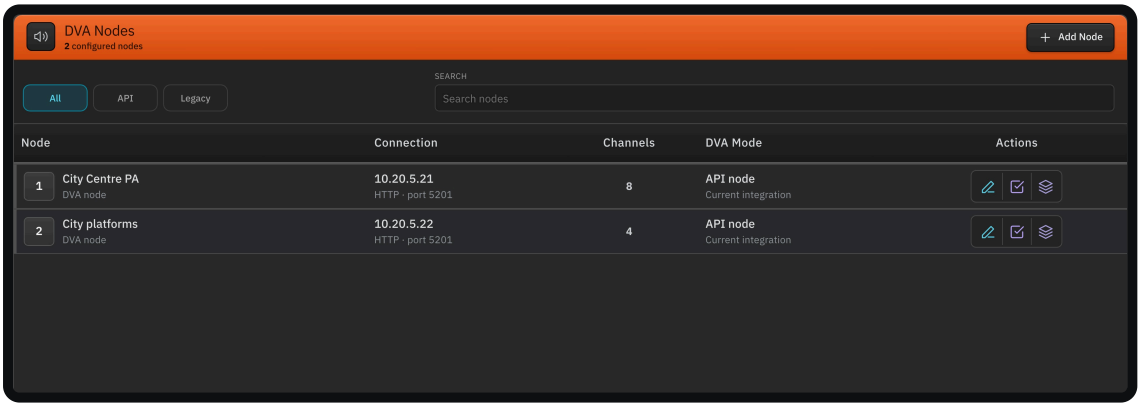
Current application views with fictional sample data. Coastline Transit is an illustrative network. Dispatch status is sample data, not proof of audible delivery.

The right information. The right destination.

Check both sides of the chain: where the service data comes from and where the announcement goes.



Subscribed feeds and output destinations
Review the data subscriptions and DVA endpoints together.



DVA node configuration
Connect the rules to the playback endpoints and channels.

Connect visual information

Configured actions can send EYE realtime text, messages or mode changes alongside an announcement. Confirm the matching display destinations.

Answer a help-point request

Smart pi SIP can pass the stop and DVA channel when a call connects, so the caller receives the current service information for that location.

Current application views with fictional sample data. Coastline Transit is an illustrative network.

From a sample feed to a proven announcement

1. **Confirm the feeds**
   Map timetable and realtime sources, stop identities, update intervals and ownership.

2. **Prepare the audio**
   Load the library and check stop names, route names, timing phrases and closing inflection.

3. **Build the rules**
   Set triggers, timing windows and service-type filters for the intended passenger information.

4. **Assign the destinations**
   Map rules to DVA nodes and channels, agree priorities and configure any EYE or SIP integration.

5. **Prove the edge cases**
   Test delay changes, cancellations, repeat boundaries, timetable-only services and failed orders. Review the logs.

Published starting server specification

16 CPU cores	16 GB memory	100 GB storage
Starting point for the server.	Starting point for the server.	Starting point for the server.

Windows or Linux, on premises or in the cloud. Confirm sizing for the feed volume, active services, rules and retention needed by the installation.

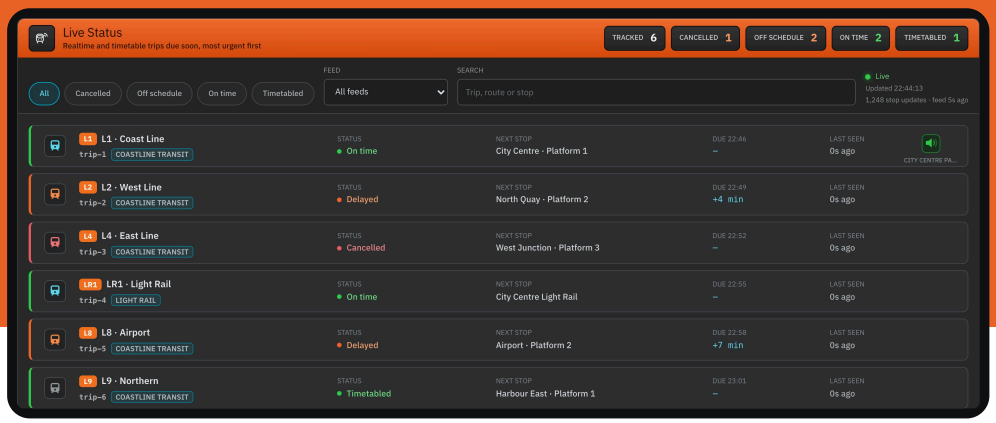
Hand over a repeatable test

Keep the sample service, expected trigger, output channel and resulting order together in the acceptance record.

SMART PI DVA GTFS

Make the next update clear to everyone.

Plan your feeds, business rules and announcement workflow with our team.



Web
www.smart-pi.info

Email
info@smart-pi.info

Sales in Australia and New Zealand: Qvest
+61 2 8011 0500

Scan to visit smart-pi.info

Smart pi ECOsystem

Smart pi is a T&M Media Pty Ltd brand · ABN 67 612 111 369
Interface views use fictional sample data. Specifications may change without notice.