VONTAS ONCALL

Teaching Machines to Speak Transit

Turning noisy radio calls into clean records, then into safe and useful action

Presenters

Henry Liao
Principal Engineer of IP Communications
Vontas

Sabah Lone
Technical Solutions Consultant
Vontas

What Will be Covered

1

AI's impact on voice communication.
Push-to-talk is VoIP now, and that changes what software can do with a radio call.

2

Agentic enablement and practical use cases.

Cleaning input data and building knowledge graphs for agentic operations and automation powered by speech.

3

Where this goes next.
The paths that open up once the machine speaks your agency's language.

One thread runs through all three: get the words right, and everything downstream gets better.

OBJECTIVE 1 OF 3

AI comes to transit operations

The wave has reached transit, moving from dashboards that report to agents that act. One operational layer got left behind: the radio.

From Dashboards to Doing

Agentic AI has taken off since fall 2025.

AI is moving from **answering** to **acting**. Most AI you've met replies when asked. The **AI agent** takes a goal and gets something done:

- **Perceive** what's happening, **reason** about what it means, **act** through real tools, and **learn** from the outcome under human oversight
- The same loop shows up everywhere, from coding assistants to customer support to fleet operations.

Analysts expect 40% of enterprise software to embed task-specific agents by the end of 2026, up from under 5%. Transit is on that curve.

The Wave has reached Transit

AI is showing up in transit:

Predictive Maintenance

Catches failures before they strand a bus, with agencies reporting roughly **30% lower maintenance cost** and far fewer breakdowns.

Scheduling and Routing

Match vehicles to demand in real time.

Computer Vision

Watches for safety incidents, and purpose-built **operations agents** now help plan and dispatch.

About **70% of agencies** plan to grow AI investment over the next three years.

Radio is next

Radio communications is ripe for AI enablement:

- Push-to-talk is **voice over IP (VoIP)** now, so every conversation is already stored as data packets that software can read and interpret.
- **Still on analog?** Capture-and-convert gateways pull analog radio into the same digital stream today, and **hybrid** analog/VoIP deployments are already in the field.
- Troves of VoIP data are sitting in archives, waiting to be used to enable agentic AI workflows.
- It's also the **most time-critical** channel: operators call in problems in real time, often before anything else in the stack knows.

OBJECTIVE 2 OF 3

Agentic Enablement

Accurate transcription, knowledge graphs, and use cases

Accurate transcription comes first

- Transcription is the foundation of the workflow. Intent recognition, the knowledge graph, and every automated action are built on the words we capture
- Accuracy is what makes automation safe to trust. You can only auto-fire a workflow when you're confident you heard the request correctly.

The Radio Check

A driver keys the mic and says "**2661, radio check**" to confirm it's working. Dispatch replies "loud and clear." It's the simplest thing on the radio, and the thread we'll pull all the way through this talk.

- It's the **#1 most common message** we found. One large transit agency's radio holds 13,573 of them.
- That volume is why automating it is worth **about $700,000 a year**.

One phrase, followed end to end: from how it's misheard, to how it gets handled without a human.

Where The Words Get Lost

Automating a radio check means first **recognizing** one reliably, over engine noise, in about two seconds. Here's what off-the-shelf speech AI writes down when a driver says "**radio check**":

"video check"	← actually said: radio check
"radio tech"	← actually said: radio check
"ready to check"	← actually said: radio check

- **Two-second bursts** over a noisy radio. 33.1% of all messages are that short.
- **Local words:** vehicle "run numbers," stop names, agency slang. Every property talks a little differently.
- Everyday AI learned from **podcasts and YouTube**. Transit radio sounds nothing like that.

Off-the-shelf AI solutions don't speak transit

Two Ways to Fix It

Off-the-shelf speech AI still guesses at your radio. There are two paths to close the gap:

	A) throw a bigger model at it	B) teach the AI your agency's language by building a knowledge layer
Cost	≈ $0.02–0.10 / min, forever	one-time mining (≈ $0 ongoing)
Runtime	seconds per clip, at scale	a lookup, sub-millisecond
Your words	still guesses "video check"	fixes "video check" → "radio check"

The bigger model still guesses at your radio. The knowledge layer is cheap, aimed right at the gap, and works with most general models.

A Dictionary + Rulebook for Your Radio

It's an organized map of your agency's world: the words people use, what they mean, and how they fit together.

- The technical name for this is a **transit ontology**.
- The word sounds fancy. It's really just three parts.

Words · Shape · Rules.

Part 1: Words

Every real phrase, plus the ways it's actually spoken **and mis-heard:**

means	heard as
RADIO CHECK	radio check · video check · radio tech · radio jack · ready to check

We **mined this from 92,233 real calls:** 119 core phrases and 1,964 spelling and mishearing variants.

Now when the AI hears "video check," the map says: that's a radio check.

Part 2: Shape

Messages follow a pattern, with **blanks to fill in** (we call them slots):

a driver says	the shape
"2661, radio check"	[run #] radio check
"2643, no-show at 14 Elm"	[run #] no-show at [stop]

The **run number** and the **stop** are slots. They change every time, so they sit apart from the phrase itself.

The shape is what lets the system pull out "who" and "where," along with "what."

Part 3: Rules

A few common-sense guardrails the computer can actually check:

- A no-show needs a stop to be logged. If the stop is missing, ask a human.
- An emergency always goes to a person.

Written down once, and we can prove the rules agree with each other before anything ships.

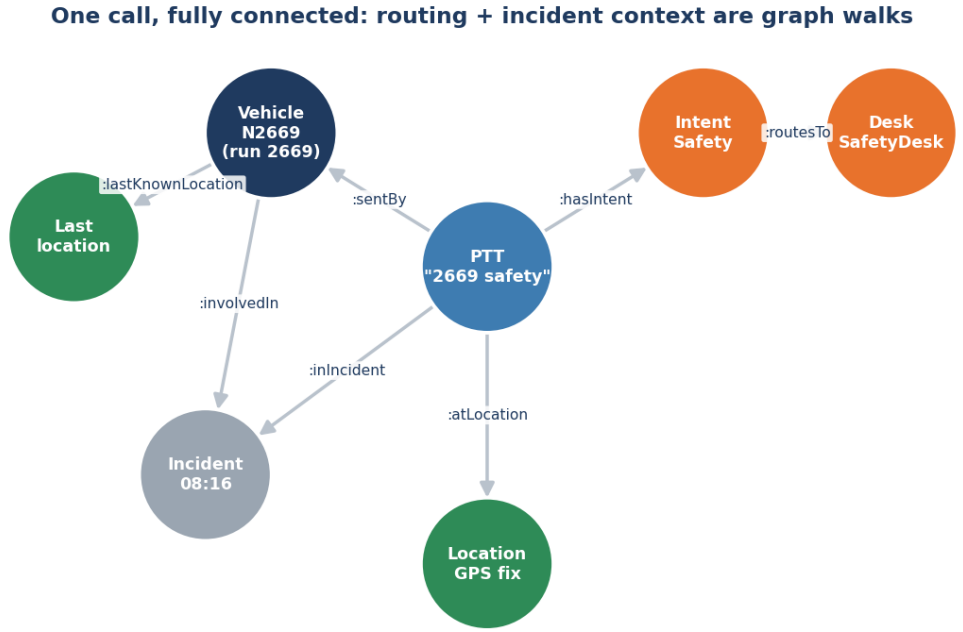
Structured input => helpful output

- The system reads your own radio traffic and discovers your words, your shapes, your mishearings.
- The word list maintains itself as the traffic keeps coming in.
- Every agency grows its own map, discovered straight from real audio.

Mine data from real comms traffic and map the entries to known entities and relationships.

From a Word List to a Knowledge Graph

The map so far is the **schema**: the words, shapes, and rules (an **ontology**). Point it at real traffic and it fills in with **connected records**, a **knowledge graph**:



Ontology is the rulebook; the graph is it **populated with what actually happened**, all linked. So the hard questions (where should this call go? what's the full picture of this incident?) become short walks across it.

How an Agent actually uses the Graph

A language model is fluent but **ungrounded**: it guesses at facts and struggles to chain several together. The graph gives the agent four things it can't get from words alone:

- **Ground the facts.** Instead of guessing, the agent **looks up** who "2669" is, where it is, and what group it's on. No invented vehicles or stops.
- **Reason in hops.** Chained questions ("this call → its speaker → their last location → their recent calls") become a **single walk**, not brittle multi-table joins.
- **Assemble context.** Hand the agent a **connected slice** of the graph as its working memory (retrieval over a graph, sometimes called GraphRAG), then let it decide.
- **Stay auditable.** Every fact it used is a **real node and edge** you can point to afterward, which is exactly what the oversight rules want.

The model brings language; the graph brings truth and memory. The routing and incident use cases ahead are exactly these walks.

Are we building this from scratch? No, use AI to accelerate

The **nouns** of transit are already standardized, so we reuse them and only build the missing piece:

- **Reuse** open standards for stops, routes, runs, vehicles (GTFS, Transmodel) and for GPS (the sensor standard SOSA).
- **Build new** the one layer nobody has: the **spoken-operations** vocabulary, what a radio call means and should do.
- The closest prior art is **air traffic control**, which formalized controller speech. Transit radio has had no such thing.

Align to what exists; build only the genuine gap. That gap is the voice of the radio.

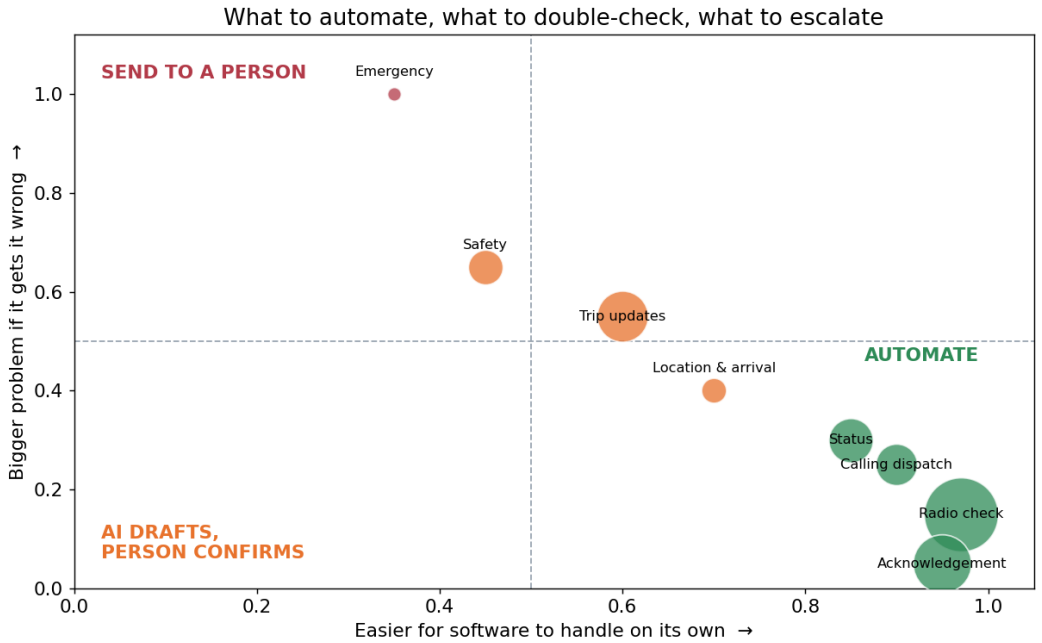
Two clocks: build it carefully and deliberately, use it fast

The heavy thinking happens **offline first**. The live radio performs a faster lookup.

	Historical setup (offline)	Real-time (per call)
What runs	mine the vocabulary, build the graph, prove the rules	recognize intent, walk the graph, route
How often	nightly / as traffic accumulates	the moment the mic is keyed
Speed	minutes to hours, fine	sub-millisecond lookup

We reason at build time and compile to fast artifacts.

Automate the Routine. Assist the Judgement. Escalate the Critical.



- **Automate** the low-stakes and unmistakable (radio check, status).
- **Assist** when judgement helps: a no-show, where AI drafts and a person confirms.
- **Escalate** anything safety-critical straight to a human.

Two use cases next show these bands working together. (run numbers like 2661 identify the vehicle.)

Use Case 1: Intelligent Call Routing

A call comes in. The system recognizes it, resolves the speaker's run, and hands the **right desk a call that already has its context attached**. Two real examples from the graph:

heard on the radio	the graph walk	result
"video check, 3105" (misheard radio check)	fix words → run 3105 → RadioCheck	auto-answered, no human
"2669, safety"	run 2669 → last location + recent calls → Safety desk	to safety desk, dispatcher confirms

Routing isn't keyword matching. It's a walk across the graph: **call → speaker → run → location → recent calls → desk**, with the bundle built along the way.

The desk doesn't get a bare call. It gets the call, the vehicle, where it is, and what it just said.

Use Case 2: Automated Incident Response

When a safety call anchors an incident, the graph already holds the whole picture. From **one node**, the system assembles a draft investigation folder, run live on a **real 2026-03-30 safety incident**:

from the incident node, gather	on real data
every vehicle involved + its last known GPS	12 vehicles located
the substantive timeline (routine chatter dropped)	89 of 92 calls kept
the intent mix across the incident	Safety 13 · Emergency 2 · Status 5

No hand-collation: the folder is a single traversal from the incident outward. It's a **draft for a dispatcher**, never an auto-resolution.

Minutes of gathering, done in a query. The person reviews and decides.

Use Case 3: Simulate Routes to Train your People

The same map that **reads** real radio can also **generate** it. An agent role-plays a route so dispatchers and operators drill comms and SOPs on demand, no live operations tied up, done today:

- **An agent plays the driver.** It keys realistic calls (radio checks, no-shows, a duress call) in your agency's own phrasing, mined from real traffic.
- **People practice the SOP.** Route a call, run the emergency procedure, handle a rollover, scored against the playbook.
- **Rare, high-stakes scenarios on tap.** Rehearse the once-a-year emergency a hundred times before it happens for real.

The same knowledge that reads the radio can rehearse it, training the humans and the AI on the same synthetic traffic.

So what does the AI keep improving?

Coding assistants iterate on code: edit, test, repeat. Here the same loop runs in two places.

- **In the moment** (seconds): on a live call the agent gathers context, proposes an action, and a person accepts or corrects. Edit-test-refine, applied to handling the call instead of writing code.
- **Over time** (the real one): the **rulebook is the codebase**. Every misheard word and every human correction is a change request against it. Mine the new phrase, add the rule, prove it stays consistent, ship it.

The model is a frozen commodity. Your agency's rulebook is the living thing that keeps getting better, learned from your own radio traffic.

Safe by Design, and it Earns More over Time

- The AI **proposes**; a person **decides** on anything high-stakes.
- The rulebook makes "auto-fire an emergency" **impossible to even ship**.
- Every action is **logged and reviewable**: who, what, when, and why.
- A new intent starts in **Escalate** or **Assist** with a human on every one, then **graduates inward** (Escalate → Assist → Automate) as its track record proves out. Emergencies stay with a person by rule, always.

Trust is the product. Ship the safe part first, earn the rest. The safe band only grows.

OBJECTIVE 3 OF 3

What’s Next?

The paths that open up once the machine speaks your agency's language

The Paths that Open Up

Once the words are right and the intents are known, the same foundation reaches further:

- **Live, in the moment.** On-demand transcription today; the same model moving toward real-time as the call happens.
- **From words to workflows.** Recognized intents feed dispatch: auto-drafted incident reports, evidence gathered on a duress call, ETAs pushed to riders.
- **Agencies that learn from each other.** Share the **method**, keep the words private: think **"GTFS for the radio,"** the free standard that put schedules into every maps app, applied to radio talk. Every property's system gets smarter without giving up its private vocabulary.

The transcript was step one. The action layer on top is where the next few years live.

Voice as One Signal in Connected Transportation

Voice doesn't stay on an island. The knowledge graph is the **join point** where a spoken call meets everything else the vehicle is already telling you:

- **GPS and telemetry** already flow through the same platform. The graph links "2669, safety" to where 2669 is and what it's doing.
- **Schedules and trips** (the GTFS/Transmodel nouns) connect a call to the run it belongs to.
- As vehicles get more connected, voice becomes one more real-time signal fused with the rest, not a separate radio channel.

Connected transportation is many signals in one picture. Voice, finally, is one of them.

The Dispatch Center in a Few Years

The pattern the whole industry is converging on: the dispatcher stops **doing** the routine work and starts **supervising a team of agents** that do it.

- Agents handle the routine, answer checks, log no-shows, draft notifications, and **escalate the exceptions** to a person.
- They act **proactively**: watch live conditions, propose a spare, pre-draft the rider alert, before the call even comes in.
- The dispatcher sets the guardrails, watches the exception queue, and takes control when judgement is needed.

Not fewer people, more reach. One dispatcher, supervising the routine instead of drowning in it.

How Agents get New Abilities: Skills

Across the AI world, agents are extended with **skills**: a self-contained bundle of instructions (and the tools to run them) that teaches an agent **one capability**, loaded on demand.

- **Portable.** Write a skill once, and any compatible agent can pick it up, the way an app installs a plug-in.
- **Composable.** Snap several together for a bigger job, each one small and testable.
- **Shareable.** It's becoming the common format for distributing agent know-how, the way libraries package code.

A skill is a reusable unit of "how to do this," that travels between agents.

Take it and teach your own radio.

Open-sourcing the **transit-ops ontology** and **sample workflow skills** as a shared core, built to be **forked per region and locale**. Grow your own vocabulary, run your own SOPs, keep your words private.

Questions?

Henry Liao
Principal Engineer of IP Communications
henry.liao@vontas.com

Sabah Lone
Technical Solutions Consultant
sabah.lone@vontas.com

Thank You!