Tech4Biz

Excellence Guaranted!

Tech4Biz
Excellence Guaranteed!

CASE STUDY 02

WAREHOUSE & AMR TASK INSTRUCTION

Natural Language Fleet Orchestration on the AMD Kria KR260

A quantized language model on the warehouse edge that turns spoken operator intent into a validated, schema checked mission plan for an autonomous mobile robot fleet. The model understands and plans. Perception, navigation and safety remain fully deterministic.

The Business Problem

How instructions reach robots today

Autonomous mobile robot fleets are commanded through predefined workflows, a warehouse management system integration, or a custom dashboard built during commissioning. Each of those works well for the tasks that were anticipated, and poorly for everything else.

The practical consequence is that the operation adapts to the tool rather than the tool adapting to the operation. A supervisor who wants to sequence two tasks conditionally, or route around a blocked aisle for the afternoon, or prioritise one dock over another because a truck is late, is either configuring a workflow or calling an integrator.

Where the cost accumulates

- **Configuration effort.** Every new task shape becomes an engineering ticket rather than an instruction.
- **Training overhead.** Each supervisor has to learn a dashboard, and staff turnover in warehousing is high enough that this is a recurring cost, not a one off.
- **Under-utilised fleet.** Robots idle because expressing the task takes longer than the task.
- **Integrator dependency.** Operational flexibility is gated by a third party's availability.
- **Connectivity exposure.** Cloud hosted planning stops the floor when the link stops.

What the operator actually wants to say

SUPERVISOR

"Move pallet 23 from Dock 5 to Storage Zone B after charging Robot 4."

That single sentence contains an entity reference, two locations, a precedence constraint, an implied robot allocation and an implied battery condition. Expressing it in a dashboard takes several screens. Expressing it in speech takes four seconds.

> THE REQUIREMENT IN ONE SENTENCE
>
> Let the supervisor state the operation in plain speech, convert it into a mission plan that is validated against the real fleet and the real floor, and reject anything that cannot be grounded, all without a network dependency.

Why the edge, specifically

Warehouse floors are radio hostile environments full of metal racking, and site policy frequently forbids operational data leaving the building at all. Running the planning layer on the floor removes both problems, and it also removes the per instruction inference cost that would otherwise scale with throughput.

Solution Overview

A single Kria KR260 node on the warehouse network hosts the reasoning and planning layer. It listens to the supervisor, resolves what they said against a live registry of real robots and real locations, produces a structured mission plan, and hands that plan to the existing fleet manager and ROS2 stack.

Layered architecture

LAYER 6 INTENT
Speech capture and transcription
Industrial microphone, noise suppression, Whisper Tiny on device.

LAYER 5 UNDERSTANDING
4-bit quantized language model
Resolves entities, sequence, conditions and implied constraints from the utterance.

LAYER 4 GROUNDING
Live world model
Registry of robots, docks, racks, zones and their current state, refreshed from the fleet manager.

LAYER 3 PLANNING
Deterministic task planner
Dependency ordering, constraint checking, allocation and feasibility.

LAYER 2 HANDOFF
Schema validated mission plan
JSON contract, versioned, auditable, rejected if it does not validate.

LAYER 1 EXECUTION
Fleet manager, ROS2, robot controllers
Unchanged. Navigation, perception and safety behave exactly as they do today.

The boundary that makes this deployable

The most important design decision in this architecture is what the model is not allowed to do.

The model does	The model never does
Understand operator intent	Drive the robot
Resolve names to real entities	Compute a path
Order tasks and dependencies	Interpret sensor data
Propose an allocation	Override a safety stop
Emit a structured plan	Write directly to a controller
Ask when uncertain	Guess a destination

WHY THIS MATTERS COMMERCIALLY

Keeping perception, navigation and safety deterministic means the existing safety case, the existing certification and the existing integrator relationships all stay intact. The language layer is additive. That is what makes it possible to deploy on a live floor rather than in a lab.

Hardware and Sensing

Edge node

Element	Configuration
Platform	AMD Kria KR260 robotics starter kit
System on module	Kria K26, Zynq UltraScale+ MPSoC
Processors	Quad core Cortex-A53 with dual core Cortex-R5F
Memory	4 GB on-module DDR4
Acceleration	Programmable fabric with DPU for vision and audio offload
Networking	Multiple Gigabit Ethernet on the carrier
Storage	NVMe or eMMC for model, index and audit log

Robot side sensing

Where the KR260 is deployed on the robot rather than at a supervisor station, it also carries the perception payload. In the reference design the two roles are separated, with a planning node on the network and perception handled on each robot.

2D and 3D LiDAR · RGB camera · Depth camera · Wheel encoders · IMU · Safety rated laser scanners

Operator station

Industrial noise cancelling microphone · Push to talk headset · Confirmation display or tablet · Optional fixed station microphones per zone

Acoustic conditions drive this choice more than anything else. A warehouse floor typically sits well above conversational noise levels, with reverberation from hard surfaces and racking. A headset with a close talking microphone will outperform a far field array in most facilities, and it also makes it unambiguous who is issuing an instruction.

> **DESIGN DECISION TO TAKE EARLY**
>
> Whether instructions are issued from fixed supervisor stations, from a roaming headset, or from both changes the audio architecture, the authorisation model and the confirmation flow. Decide this before detailed design.

Placement on the network

The planning node sits alongside the fleet manager on the operational network, not on the corporate network and not on the internet. It needs to reach the fleet manager and the message broker. It does not need, and in the reference design does not have, an outbound path.

The World Model

A language model that does not know what robots exist will invent one. The world model is the component that prevents that, and it is the difference between a demonstration and a deployable system.

What it holds

Entity class	Attributes tracked
Robots	Identifier, type, payload capability, current task, battery state, location, availability
Docks and bays	Identifier, aliases, current occupancy, access restrictions
Racks and storage	Identifier, aisle, level, capacity, current contents where known
Zones	Boundaries, access rules, temporary exclusions, speed restrictions
Charging points	Identifier, occupancy, compatibility by robot type
Loads	Pallet or tote identifiers currently known to the system

How it stays current

- Subscribed to fleet manager state over MQTT or the site's existing broker.
- Warehouse management system provides load and location identifiers where an integration exists.
- Staleness is tracked explicitly. An entity whose state is older than a configured threshold is treated as unknown rather than as last known good.
- Aliases are first class. Operators say "the north dock" and "Dock 5" interchangeably, and the registry has to resolve both.

Grounding contract

Every entity the model names in a plan must resolve to a real record in the world model. There is no path by which an unresolved name reaches the fleet manager.

Condition	Behaviour
Name resolves uniquely	Proceed.
Name resolves to several entities	Ask the operator to disambiguate, listing the candidates.
Name does not resolve	Reject and say so. Never substitute the nearest match.
Entity exists but is unavailable	Report why, and offer the feasible alternative if one exists.
Entity state is stale	Treat as unknown and request confirmation.

> **FAIL CLOSED, ALWAYS**
>
> An ambiguous instruction that becomes a wrong movement is far more expensive than an instruction that gets questioned. The system is designed so that uncertainty produces a question, never a guess. Operators tolerate being asked. They do not tolerate a pallet arriving in the wrong aisle.

From Language to Mission Plan

Constrained generation

The model does not write free text that is then parsed. It generates directly into a schema, using grammar constrained decoding so that structurally invalid output is not merely rejected but is impossible to produce.

01
Transcript plus world model snapshot
The model is given only the entities that currently exist, not an open vocabulary.

02
Grammar constrained decoding
Output is forced to conform to the mission plan grammar at generation time.

03
Schema validation
Types, enumerations, required fields and value ranges checked against the contract.

04
Semantic validation
Entities must exist, be available, and be compatible with the requested operation.

05
Operator confirmation
Plan read back in plain language before dispatch, for anything above a configured significance threshold.

What the model is given

- The transcript.
- A filtered snapshot of currently valid entities.
- The active zone and speed restrictions.
- Nothing else. No site history, no free text, no unbounded context.

Mission plan contract

```
{
  "plan_id"       : "MP-2026-07-29-0143",
  "issued_by"     : "supervisor-station-2",
  "tasks": [
    { "seq"           : 1,
      "type"          : "charge",
      "robot"         : "AMR-04",
      "station"       : "CHG-02",
      "until_soc_pct" : 60 },
    { "seq"           : 2,
      "type"          : "transport",
      "robot"         : "AMR-04",
      "load"          : "PLT-23",
      "pickup"        : "Dock-5",
      "destination"   : "Zone-B",
      "depends_on"    : [1],
      "priority"      : "normal",
      "avoid"         : ["Zone-C"],
      "battery_check" : true }
  ],
  "world_snapshot" : "ws-88214",
  "schema_version" : "2.1"
}
```

WHY THE SNAPSHOT REFERENCE IS IN THE PLAN

Recording which world model snapshot the plan was built against makes the whole decision reproducible after the fact. When something goes wrong on a floor, the first question is always what the system believed at the time, and this answers it exactly.

The Task Planner

Why planning is separate from the model

Language models are good at understanding what someone meant. They are unreliable at constraint satisfaction, and unreliable in a way that is hard to test. The planner is therefore a conventional deterministic component that takes the model's interpretation and turns it into something feasible.

Planner responsibility	What it checks
Dependency ordering	Precedence expressed in the instruction is turned into an acyclic task graph and validated.
Feasibility	Is there a robot of the right type, free within the window, able to reach both locations.
Battery and duty	Does the assigned robot have enough charge for the full task chain plus a return margin.
Zone and access rules	Are all traversals and all destinations permitted for this robot type at this time.
Capacity	Does the destination have space, and is the load within payload limits.
Conflict	Does this plan contend with an already dispatched plan for the same robot, load or location.

Allocation

Where the instruction does not name a robot, the planner proposes one. The allocation policy is configuration, not model behaviour, so a site can tune it without retraining anything.

- Nearest suitable robot by travel time rather than by straight line distance.
- Preference for robots already near the pickup, to reduce empty travel.
- Charge state weighted against the length of the task chain.
- Load balancing across the fleet over a shift, so duty cycle stays even.

> **ARBITRATION STAYS WITH THE FLEET MANAGER**
>
> The planner proposes. The existing fleet manager remains the single authority that dispatches, sequences against other traffic and resolves contention. Introducing a second scheduler would be the fastest way to create deadlock on a live floor, so the architecture deliberately avoids it.

Rejection paths

Every rejection is returned to the operator with the specific reason and, where one exists, a feasible alternative. A plan that fails silently is worse than no plan, because the operator believes work is underway.

Inference Stack and Memory Budget

Why four bit

A three billion parameter model at sixteen bit precision needs roughly six gigabytes for weights alone, against four gigabytes available on the module. At four bit the same model comes down to approximately 1.7 gigabytes, leaving room for the speech model, the world model, the index and the operating system.

Throughput is bound by memory bandwidth rather than by arithmetic, because every generated token requires the full weight set to stream out of DDR. Halving the bytes per weight roughly doubles the achievable token rate.

Component	Allocation
Quantized model weights	~1.7 to 2.2 GB
Key value cache at working context	~150 to 250 MB
Speech recognition model	~75 MB
World model and entity index	~100 to 200 MB
Runtime, ROS2 bridge and OS	~600 to 800 MB
Headroom retained	**Reserved, not consumed**

The DDR pipeline

Theoretical bandwidth on the module is a ceiling nobody reaches. Closing the gap between theoretical and achieved bandwidth is where the engineering effort actually goes.

TILING
Weight matrices tiled to on-chip memory
Geometry chosen against available BRAM and URAM.

DOUBLE BUFFERING
Transfer overlapped with compute
One tile processed while the next streams in over AXI.

PREFETCH
Next layer fetched during the current one
Layer boundaries are the most common stall and are removed deliberately.

ON-FABRIC DEQUANT
Four bit bytes cross the bus
Dequantization after transfer, never before.

STATUS OF FIGURES
Memory allocations and performance characteristics here are design targets for the reference configuration. They are confirmed on customer hardware and workload during the engineering phase.

Execution and Robotics Integration

Integration surface

Component	Role
ROS2	Middleware and node graph across the robot fleet.
Nav2	Path planning, obstacle avoidance and recovery behaviours.
MoveIt	Manipulation where the robot carries an arm.
VSLAM	Localisation and map maintenance.
Object detection	Load identification and pick verification on the DPU.
Fleet manager	Dispatch, traffic control and contention resolution.
MQTT	State distribution and plan handoff.
OPC-UA	Integration with conveyors, doors and fixed automation.

What stays deterministic

Everything below the mission plan. Localisation, path planning, obstacle avoidance, docking, pick verification and every safety function operate exactly as they did before the language layer was introduced, and they are validated the same way.

Safety architecture

Safety is not implemented in software that a language model can influence. It is implemented where it always was, and the language layer sits entirely outside that boundary.

Safety function	Where it lives
Emergency stop	Hardwired, independent of all software.
Protective laser scanners	Safety rated devices on the robot, directly interlocked.
Speed and separation	Safety controller on the robot.
Zone interlocks	Safety PLC on the fixed installation.
Traffic management	Fleet manager, unchanged.
Mission planning	The only place the language layer participates.

STANDARDS CONTEXT

Industrial mobile robot installations are governed by the applicable regional standards for driverless industrial trucks and their systems. Because the language layer cannot reach any safety function, the existing conformity assessment for the installation is unaffected by its introduction. This position is confirmed with the site's safety authority as part of Phase 0.

Worked Example, End to End

SUPERVISOR

"Move pallet 23 from Dock 5 to Storage Zone B after charging Robot 4."

STEP 1 TRANSCRIBE
On device speech recognition
Close talking microphone, transcript produced locally.

STEP 2 RESOLVE
Entities matched to the world model
PLT-23 exists at Dock-5. Zone-B exists and has capacity. AMR-04 exists at 38 percent charge.

STEP 3 INTERPRET
Precedence and conditions extracted
Charging must complete before transport begins. Battery check implied by the instruction.

STEP 4 PLAN
Feasibility and constraints checked
Charge duration estimated, task chain fits the shift, Zone-C exclusion active this afternoon and applied automatically.

STEP 5 CONFIRM
Read back to the operator
Plain language summary, one confirmation, then dispatch.

SYSTEM

"AMR-04 will charge to 60 percent at station 2, then move pallet 23 from Dock 5 to Zone B, avoiding Zone C. Estimated start in 18 minutes. Confirm?"

And when it cannot be done

SUPERVISOR

"Move pallet 31 to Rack B5."

SYSTEM

"Pallet 31 is not currently known at any dock. Rack B5 is full. Did you mean pallet 13, which is at Dock 2?"

THIS IS THE BEHAVIOUR THAT MATTERS MOST

Two separate failures caught in one turn, both reported specifically, with a grounded suggestion rather than a guess. No plan was dispatched. Systems that get this wrong are not merely unhelpful, they move physical goods to the wrong place.

Audit, Deployment and Lifecycle

Audit trail

Every instruction produces a record, held locally. This is what makes the system defensible when a movement is questioned weeks later.

Recorded	Purpose
Transcript	What was actually said, as understood.
World snapshot reference	What the system believed at that moment.
Generated plan	The exact structured output.
Validation outcome	Accepted, rejected, or amended, and why.
Operator confirmation	Who authorised dispatch.
Execution result	Returned from the fleet manager.

Release management

Model, grammar, schema and application are released as one signed bundle with a single version number. The mission plan schema version travels inside every plan, so a fleet manager receiving a plan always knows which contract it was built against.

Delivery phases

Phase	Scope	Exit criteria
Phase 0 **Assessment**	Fleet manager interface review, entity taxonomy, acoustic survey, safety authority engagement.	Agreed schema and integration approach.
Phase 1 **Shadow mode**	Plans generated and validated but never dispatched, run alongside existing operation.	Grounding accuracy and rejection behaviour measured against real instructions.
Phase 2 **Supervised dispatch**	Plans dispatched with mandatory operator confirmation on every instruction.	Zero ungrounded dispatches over an agreed window.
Phase 3 **Routine operation**	Confirmation required only above a significance threshold, tuned per site.	Throughput and instruction latency targets met.
Phase 4 **Expansion**	Additional zones, task types, languages and sites.	Repeatable deployment package.

Benefits and Measurement

Benefits realised

No operator training
The interface is the operation described in plain speech, so onboarding is close to zero.

Faster instruction
Multi step, conditional tasks issued in one utterance instead of several dashboard screens.

Less programming effort
New task shapes stop requiring workflow configuration or an integrator ticket.

Offline task planning
The floor keeps running when the link to the enterprise or the internet is down.

Secure deployment
No operational data, floor layout or inventory position leaves the site.

Auditable handoff
Every dispatched plan is a structured, versioned record rather than a free text command.

Higher fleet utilisation
Robots idle less because expressing the task is no longer the bottleneck.

Predictable cost
Inference is capital rather than a per instruction charge that grows with throughput.

KPI framework

Measure	What it tells you
Grounding accuracy	Share of instructions correctly resolved to real entities.
Ambiguity rate	How often the system has to ask, which should fall as aliases are learned.
Ungrounded dispatch count	Must be zero. This is the safety relevant number.
Instruction to dispatch time	End to end latency the supervisor experiences.
Instructions per shift	Whether the interface is actually being used.
Fleet utilisation	Idle time recovered.
Configuration tickets raised	Integrator dependency reduced.

BASELINE FIRST

Capture instruction volume, configuration ticket rate and fleet idle time before deployment. Without a baseline the pilot produces numbers that cannot be defended, which is the usual reason a working pilot fails to secure production funding.

Engineering Considerations and Risks

Known hard problems

Risk	Mitigation
Acoustic environment	Acoustic survey in Phase 0, close talking microphones preferred, per zone tuning rather than a single reference setup.
Entity drift	Live subscription with explicit staleness handling, and unknown rather than last known good when the threshold is exceeded.
Ungrounded dispatch	Grammar constrained generation, schema validation, semantic validation and confirmation, in that order. Four independent gates.
Operator trust	Shadow mode before dispatch, so the operation sees the system be right before it is allowed to act.
Scheduler conflict	Never introduce a second dispatcher. The fleet manager remains the sole authority.
Accent and language variation	Site specific speech tuning, and alias learning from the audit trail rather than from a generic vocabulary.
Schema evolution	Version carried inside every plan, with an explicit compatibility policy on the fleet manager side.

Open questions for the customer

- Which fleet manager is in use, and what interface does it expose for plan submission and state subscription?
- Does a warehouse management system integration exist that can supply load identifiers, or is the world model limited to fleet state only?
- Are instructions issued from fixed stations, roaming headsets, or both?
- What is the authorisation model? Can any operator issue any instruction, or is it role based?
- Which languages and how many sites are in scope for the first release?
- Who is the safety authority for the installation, and what evidence will they want that the language layer cannot reach a safety function?

> **RECOMMENDED FIRST STEP**
>
> A shadow mode pilot on one zone with one fleet. It generates and validates plans against real instructions without dispatching anything, which produces measured grounding accuracy at effectively zero operational risk, and gives the operation the evidence it needs before anything is allowed to move.

Contact

Website

https://tech4bizsolutions.com

Contact details

If you would like to know more about Tech4Biz Solutions
and our products please contact us via email contact@tech4biz.io

Address

Bangalore
NGR Arcade,
HSR Layout Sector 1,
Bangalore, Karnataka

Surat
306-307,
Millionaire Business Park,
LP Savani Road, Adajan,
Surat, Gujarat - 395009



Tech4Biz

Excellence Guaranted!

Tech4Biz is a provider of comprehensive engineering and IT solutions for businesses of all sizes. From cloud computing and data management to security, networking and embedded intelligence, we have the expertise and experience to help your business stay ahead of the curve.

Our mission is to provide businesses with the best possible technology solutions. We understand that a reliable and efficient technology foundation is crucial in an increasingly competitive marketplace. That is why we offer a wide range of services, from managed IT and cloud computing to edge and embedded AI engineering.

This communication contains general information only, and Tech4Biz Solutions Private Limited is not, by means of this communication, rendering professional advice or services. Architecture descriptions, performance characteristics, memory budgets and integration details in this document represent a reference design and design targets for the stated configuration. They are confirmed against the specific customer hardware, workload and duty cycle during the engineering phase. Third party product names are the property of their respective owners. Before making any decision or taking any action that may affect your finances or your business, you should consult a qualified professional adviser. Tech4Biz Solutions Private Limited shall not be responsible for any loss whatsoever sustained by any person who relies on this communication.

© 2026 Tech4Biz Solutions Private Limited