appliedAI Developers | e·on

MODULE 2

Behavior – Explicit, Inspectable Planning

Production architecture deep-dives and live reviews

Brinda Rao
Data Scientist @ E.ON

Andrei Beliankou
Tech Lead Data & AI @ E.ON

Tanja Fenn
Data Scientist @E.ON

Munich AI Nexus

Why Munich AI Nexus Exists

- AI demos are easy
- Production failures are hard
- Most failures aren’t model problems
- They’re system and architecture problems

We focus on what breaks *after* the demo works.

The Production Readiness Blueprint

- Context
- Behavior ← today
- Execution
- State
- Collaboration
- Observability
- Security

Recap: What We’ve Learned So Far

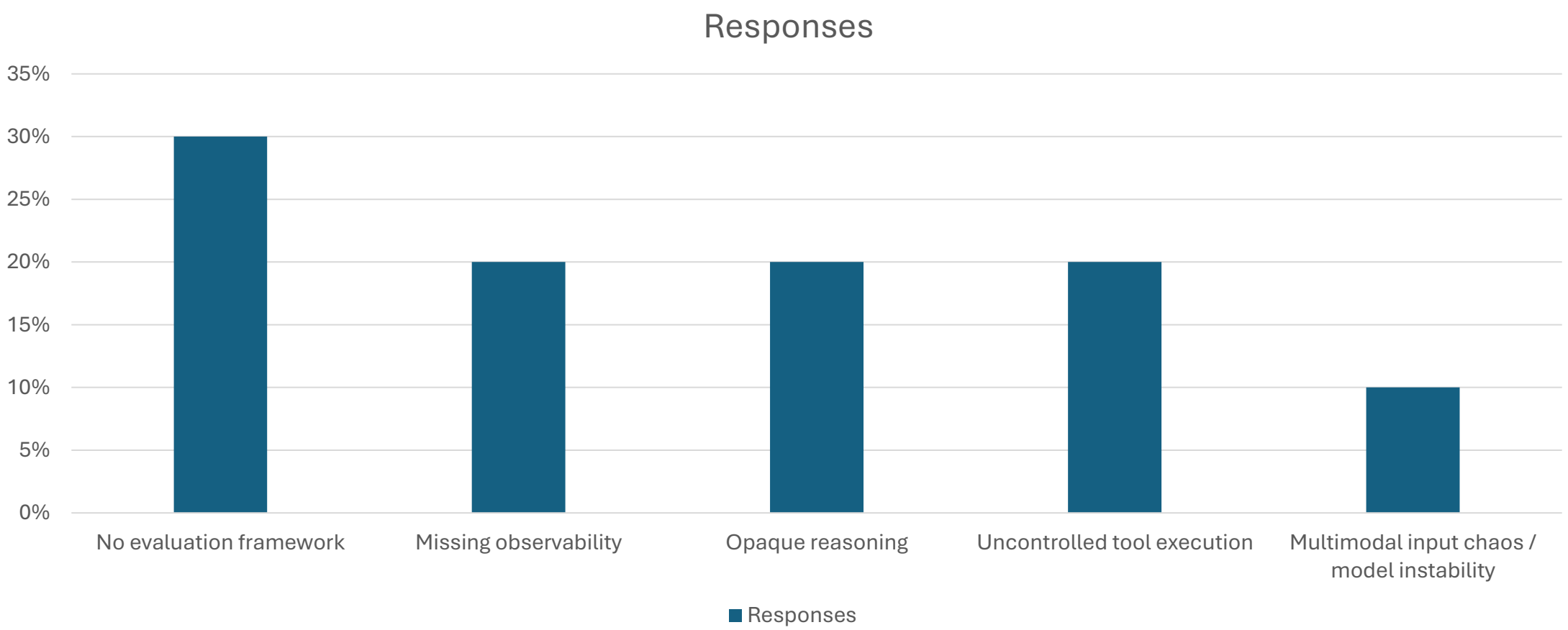
- Kickoff: context & collaboration are the most overlooked layers
- Module 1 (Context): evaluation and validation are missing almost everywhere
- Insight: fixing issues upstream is cheaper than debugging downstream

Tonight: Behavior Under Real Constraints

- Planning vs execution
- Opaque reasoning vs explicit decisions
- Autonomy vs control
- Behavior under rate limits, failures, and messy input

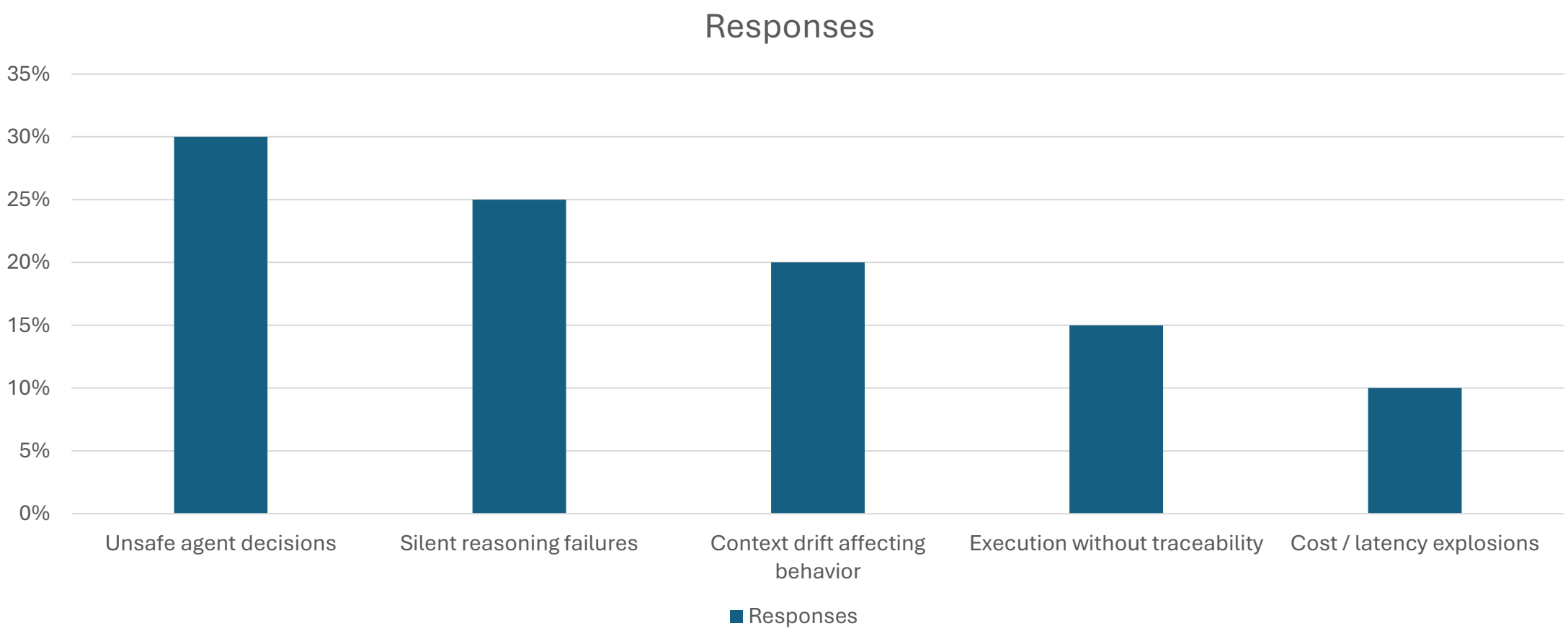
This is about systems that behave predictably when the world doesn’t.

Biggest blockers to reliable execution:



Most teams are flying blind. They don’t lack intelligence — they lack visibility and control.”

Long-term failure modes:



The biggest fear is not wrong answers — it's *undetectable* wrong behavior.

What this room wants to talk about:

- Debugging agent decisions → ~35%
- Bounded autonomy / policy enforcement → ~25%
- Fallback strategies under failure → ~20%
- Shipping fast without losing control → ~15%

What’s missing is not intelligence.
It’s control.

Evaluation, observability, traceability, and bounded behavior are now bigger bottlenecks than models.

Talk: Routing, Throttling, and Reliability

- What breaks first at scale
- How routing and fallback evolve under pressure
- What still feels brittle

What We Just Saw

- Execution reality shapes behavior
- Routing is policy, not magic
- Fallbacks reveal architecture quality

Break (45 min)

Second half is interactive

Challenge: Multimodal Intake as a Planning Problem

- Emails, PDFs, images, attachments
- Partial intent
- Real-world mess

Instructions

- Form groups of **3–4 people**
- You have **15 minutes**
- Focus on the planning and *decisions*, not the full pipeline
- You will report back **one insight**, not a solution

Challenge: Planning Multimodal Intake for Long, Messy Documents

- You receive a **multi-page document**:
  - scanned PDFs
  - blurry photos
  - mixed quality pages
  - information spread across pages
- Example:
  A 30-page notary contract where critical fields appear only once — possibly at the end.
- **Your task is planning, not execution!**

Context: You receive a **multi-page document**

Your task:

1. Define the **first 3 planning steps** (before any model runs)
2. Decide **what you extract vs summarize** (page-by-page vs whole-document)
3. Name **one tradeoff you accept** (cost or latency to guarantee completeness?)
4. Name **one thing you refuse to automate**

Optional:

- Where would you introduce **specialized models** (e.g. YOLO, OCR) instead of a frontier multimodal model?

What Emerged Tonight

- Behavior breaks before models do
- Planning needs to be explicit
- Tradeoffs are unavoidable
- Observability determines trust

Get Involved

- Looking for speakers, case studies, challenges
- Especially teams operating in production
- Rough problems > polished demos
- If you’re hiring or looking — talk to each other.

Thank You

- Thanks to speakers
- Thanks to e.on for hosting
- Next module teaser: Execution, March 3th

If your system behaves well only in demos, it doesn’t behave well.

We’re building the shared architecture for real-world AI systems.

https://github.com/Munich-AI-Nexus/production-readiness-blueprint/

Talk: real tradeoffs, brittleness, decision logic.

- Can you walk us through one specific moment where the system had to choose between *latency*, *cost*, and *model quality* — and which one lost?
- Which part of this routing or fallback logic still makes you uncomfortable today?
  - “Who has something similar that still feels brittle?”
- What signal do you actually trust when deciding to downgrade or reroute — and which signals turned out to be misleading?
- If you had to rebuild this from scratch tomorrow, what would you *not* rebuild?

Challenge: planning ambiguity & competing strategies

- When a vague email with attachments comes in, what is the *first irreversible decision* your system has to make?
- Who here summarizes first? Who extracts first? Who sends everything to the model?
- If you had to optimize for only one — cost, latency, or correctness — which would you pick, and what would you sacrifice?
- At what point do you *stop* the system and require human review?