Foundations of Dependable Agentic AI

Ten requirements that determine whether an agentic system holds up under real production conditions, particularly in regulated or safety-relevant environments.

TERENCE KOK · 10 AUGUST 2026

The ten foundations

01 Bounded task specification

02 Context as an engineered resource

03 Tool surfaces for model consumption

04 Closed-loop verification

05 Least-privilege, staged autonomy

06 Adversarial assumptions on tool output

07 Trajectory-granularity observability

08 Evaluation on trajectories, not outputs

09 Failure and recovery design

10 Deterministic code where available

Bounded task specification

An agent needs a machine-checkable definition of success and an explicit termination condition. Where success can't be verified programmatically, the task is decision support, not automation.

MISSING TERMINATION CONDITION → RUNAWAY LOOP, UNBOUNDED COST

Context as an engineered resource

- → Retrieval returns the minimum sufficient evidence, not the maximum available
- → State lives in a durable store, not accumulated conversation history
- → Prior steps compact into structured summaries on a schedule
- → Retrieved content carries provenance tags, separating instruction from data

MODEL PERFORMANCE DEGRADES NON-LINEARLY AS CONTEXT FILLS

15 beats 80

Task-shaped tools outperform a raw API surface

Exposing eighty raw endpoints reliably produces worse tool-selection accuracy than fifteen consolidated, task-shaped tools with narrow, validated schemas.

Closed-loop verification

Compilers, unit tests, schema validators, simulators, reconciliation checks, constraint solvers. All give the model a ground-truth signal it can act on.

No verifier, no way to tell a plausible output from a correct one.

Least-privilege authority, staged

ACTION CLASS	CONTROL
Read, non-sensitive	Autonomous
Write, reversible	Autonomous, compensating transaction defined
Write, irreversible	Human confirmation before commit
OT / safety-instrumented systems	No direct write path; proposals only

SCOPE CREDENTIALS PER TASK INSTANCE, NOT PER SERVICE ACCOUNT

"Every tool result is untrusted input capable of carrying instructions."

Indirect prompt injection is the expected condition wherever an agent processes third-party content, not a residual risk.

07 **Trajectory-granularity observability**

Log every step: prompt, model, tool, args, result, latency, tokens, policy decisions.

08 **Evaluate trajectories, not outputs**

Step-level scoring, cost and latency distributions, pass rates across repeated trials.

09 **Failure and recovery design**

Checkpoint after each visible action, detect loops, enforce spend ceilings outside the model.

10 **Deterministic code where available**

Reserve the agentic loop for judgement under ambiguity; code the rest.

Specify the verifier, the authority boundary, and the audit record first.

Then determine what degree of autonomy those constraints will support. That is also the only approach that produces a defensible safety case.

Terence Kok
AI & Innovation Strategist

SAVE · COMMENT · FOLLOW