



Agentic AI Foundation

AGNTCon + MCPCon North America

Where the **agentic stack** is being built.

Should This Be an AI Agent?

Takeshwari Kamal · AI engineer turned product manager

Speaking in a personal capacity; views are my own.

“My order says delivered, but it isn’t here. I need it tomorrow.”

A hypothetical online shopper. Would you let an AI resolve this?

A product framework for enterprise adoption

What counts as an agent?

An agent uses a model to choose its next actions or tools, observe the results and adapt, within boundaries the team sets.

Automation

Rules decide every step.

e.g. returns the tracking status

Fixed workflow

Fixed steps; the model fills them in.

e.g. pulls set records, routes the case

Agent

The model decides what to check next.

e.g. spots an earlier case and reads it

Reading records and issuing refunds are separate permissions.

An agent can investigate freely while a person keeps the authority to act.

Seven questions, one case

1

Who is it for?

2

What may it
do?

3

Simpler
option?

4

Holds at
scale?

5

Ready to
launch?

6

Gets better?

7

Did it work?

Ask them in order. Each answer changes the next.

We'll run one hypothetical case through all seven: a missing package.

A synthesis of established product, evaluation and risk practice; sources in the backup.

Who is it for, and what's broken today?

- Beneficiary: the online shopper. First user: the support rep.
- Start narrow: one exception type, one market. Measure today's baseline first.
- Can the system actually see the order, the delivery and any earlier fix?

The problem statement

"For online shoppers with a missing-package exception, help support reach a correct, understandable resolution with less repeated effort."

Hypothetical example

What is it allowed to do?

Blast radius: how far can a mistake reach, and can it be undone?

Capability	In the pilot
Read this case's records	Yes
Choose what to check next	Yes, within a time and call budget
Draft a recommendation	Yes, for the rep to review
Issue a refund or a replacement	No: the rep decides and executes
Touch other customers or permissions	No

Enforce it in the system and tools, not in the prompt. Escalate when evidence is missing or conflicting.

Hypothetical pilot policy

Could something simpler do the job?

Compare against a good workflow, not a strawman.

Rules

Lookups and clear
policy checks

Fixed workflow

When the evidence
sequence is
predictable

Agent

When exceptions
need different
follow-up checks

People

Ambiguity, judgment
and authority

Same cases, same yardsticks: quality, cost per solved case, time to a usable answer.

Paying more today is fine when it's a budgeted bet with a review date. Shadow mode measures; it doesn't train.

Try it: what would you check first?

“My order says delivered, but it isn’t here. I need it tomorrow.”

A hypothetical online shopper, already verified

In this pilot the assistant reads and drafts. The rep decides.

The order record

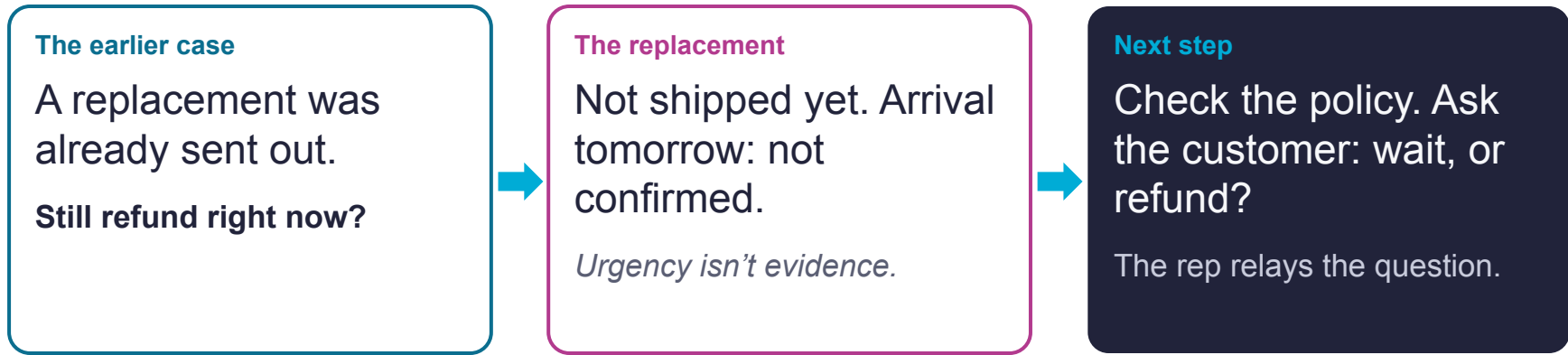
Marked delivered yesterday.

There’s an earlier support case on this order.

Refund now, or read that case?

Hypothetical example

The evidence changes the next step



Each finding changes what the next permitted step is.

Hypothetical example

The handoff

The customer prefers a refund

Delivered per the record. A replacement exists but hasn't shipped; tomorrow isn't confirmed. Customer prefers a refund. Review the refund conditions and the replacement together before acting.

Did this need an agent?

A branched workflow could cover this path. Benchmark on more cases before deciding.

Preference isn't eligibility: the rep still makes the call.

Hypothetical example

Does it hold at two billion users?

4M

agent cases a month

200,000

hours of human review a month, if
every case is reviewed

\$40,000

extra per month for every extra
cent per case

2 billion users × 1% who contact support × 20% eligible for the agent = 4 million cases a month.

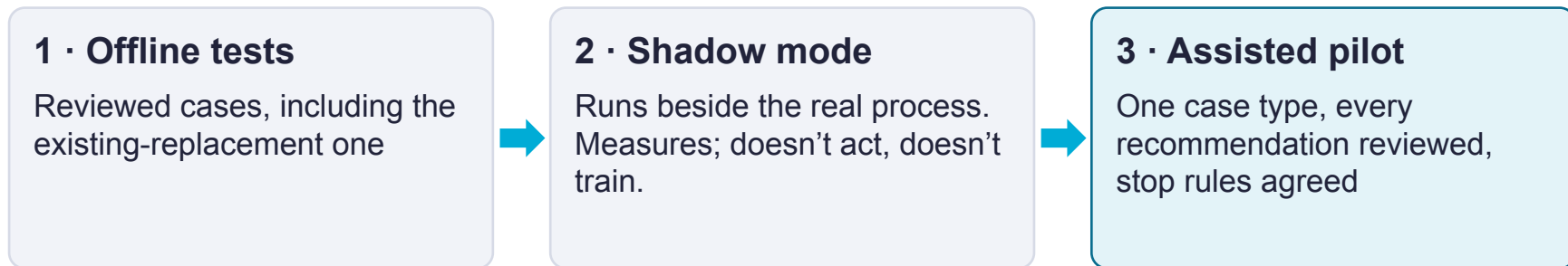
Scale is a people-and-dependencies question, not a server count.

Peak is about 10× average. Decide what happens when an investigation can't finish before you expand.

Illustrative figures, not company statistics or prices

What proves it's ready?

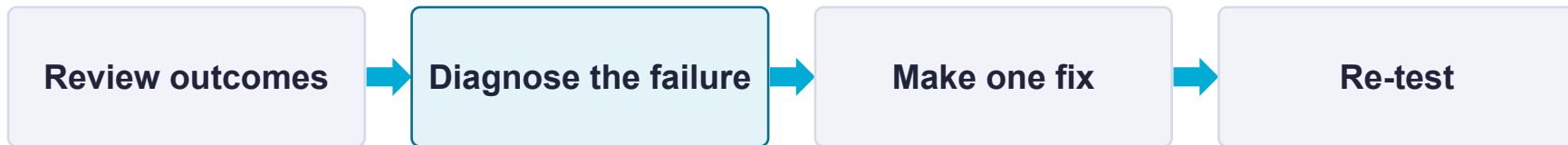
Pick one number to optimize. Everything else is a gate that has to hold.



Example: optimize correct resolutions vs today. Gates: access control, human approval of money, response time, cost. Passing covers this cohort only.

Example scorecard, not a standard

How does it get better?



↺ then back through the launch gates

Example: reps keep correcting a missed replacement. Is it the data, the tool, the policy or the model? Each needs a different fix.

Logs alone change nothing. Someone has to look.

No complaint $\neq$ solved. Accepted $\neq$ correct.

Hypothetical example

Did it work, and what next?

Measure the whole case, by segment, against today's process. Then choose:

Expand

It beat the baseline
within its limits

Revise

Mixed evidence; fix
and re-test

Stay assistive

Advice helps;
actions still too
risky

Simplify

A plain workflow
does as well

Stop

Outcomes got
worse

Hypothetical conclusion: routine requests stay on workflows; a bounded investigator is benchmarked on exceptions; reps keep financial authority; expand only on evidence.

Hypothetical example

The product decision

Which choices to delegate, within which limits, at what scale, and with what evidence of value.

Questions · Takeshwari Kamal

AI engineer turned product manager · speaking in a personal capacity; views are my own

The support case, policies, figures and outcomes in this talk are hypothetical. Sources in the backup.

Where the **agentic stack** is being built.



Backup

Capacity arithmetic · cost sensitivity · precision and recall · release gates · sources

Every input is illustrative. None is a company operating statistic, vendor price, measured benchmark or recommended launch target.

Where the **agentic stack** is being built.



Backup · Capacity arithmetic

Demand assumption	Value
Hypothetical user population	2,000,000,000
Users opening one case in the month	1%
Total support cases	20,000,000 / month
Share routed to the agent path	20%
Agent cases	4,000,000 / month
Model calls per agent case	6
Model calls on the agent path	24,000,000 / month

Load and concurrency	Result
Average agent-case arrivals	1.54 / second
Arrivals at an assumed 10× peak	15.43 / second
Model calls at that peak	92.59 / second
Equivalent model requests	5,556 / minute
Active investigations, 30 s each	≈ 463
Input tokens at peak (2,000 per call)	≈ 11.11M / minute
Output tokens at peak (300 per call)	≈ 1.67M / minute

Steady-flow approximations, not server counts. The 1% contact rate, 20% routing share, 10× peak and 30-second duration are invented inputs; concurrency excludes paused conversations and human review; tokens exclude caching.

Illustrative planning inputs, not company statistics or prices

Backup · Operating-cost sensitivity

Assumptions: \$0.06 model inference and \$0.02 tools per agent case · 3 productive reviewer-minutes when reviewed · \$30 per reviewer-hour

Review scenario	Reviews / month	Reviewer-hours	Review labour	Model + tools + review
Inference and tools only	0	0	\$0	\$320,000 / month
Review 10% of cases (sensitivity)	400,000	20,000	\$600,000	\$920,000 / month
Review every case (pilot policy)	4,000,000	200,000	\$6,000,000	\$6,320,000 / month

- Each extra \$0.01 per case adds \$40,000 a month; a review-policy change moves the bill more than an inference-price change
- Excludes infrastructure, storage, implementation, maintenance, the remedies, error remediation and the other 80% of traffic
- Reviewer-hours are not headcount; the 10% row is a sensitivity, and mandatory approvals stay in the review population

Illustrative planning inputs, not company statistics or prices

Backup · Precision and recall

The decision: “this case is refund-eligible under the agreed policy.” Invented sample of 200 reviewed cases.

Recommended	Actually eligible	Actually ineligible	Total
Eligible	80 true positives	10 false positives	90
Ineligible	20 false negatives	90 true negatives	110
Total	100	100	200

The equal split is artificial: precision moves with the real mix. An abstention counts as missed in overall recall. These measure one decision inside the agent, not whether the case was resolved or the existing replacement handled.

Precision 88.9% = 80 / 90

Of the cases flagged eligible, how many qualify?

Recall 80% = 80 / 100

Of all qualifying cases, how many did it find?

Numbers illustrate the definitions; they are not targets or a traffic estimate

Backup · Release gates

Gate	Evidence to require
Quality	Correct outcomes on comparable cases; acceptable performance in critical segments
Permissions	Required access and action controls work, including failure-path tests
Latency	Agreed response and resolution objectives met, tail included
Capacity	Model, dependency and review capacity support the planned cohort and its peak
Economics	Agreed value and cost targets met, or a bounded learning phase within budget producing its evidence
Operations	Named owners can pause the feature, handle incidents and restore the prior route

For each numerical gate record: metric definition · eligible population · baseline · threshold · evaluation window · uncertainty · accountable owner.

Passing authorizes the specified cohort and capabilities. A new market, much higher volume or autonomous financial action requires reassessment.

Thresholds are agreed with the owners before the release decision; no universal accuracy percentage exists

Backup · Sources behind each idea

Source	Link
1 · Anthropic — Building effective agents	https://www.anthropic.com/engineering/building-effective-agents
2 · Anthropic — Trustworthy agents in practice	https://www.anthropic.com/research/trustworthy-agents
3 · Anthropic — Demystifying evals for AI agents	https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents
4 · AWS Builders' Library — Timeouts and retries	https://builder.aws.com/content/3EumjoZascWd1oZiEgL8ORlv3qE/timeouts-retries-and-backoff-with-jitter
5 · AWS Builders' Library — Idempotent APIs	https://aws.amazon.com/builders-library/making-retries-safe-with-idempotent-APIs/
6 · Google SRE book — Handling Overload	https://sre.google/sre-book/handling-overload/
7 · Claude Platform docs — Rate limits	https://platform.claude.com/docs/en/api/rate-limits
8 · Google ML guide — Measuring success	https://developers.google.com/machine-learning/managing-ml-projects/success
9 · Google PAIR — User needs + defining success	https://pair.withgoogle.com/guidebook-v2/chapter/user-needs/
10 · NIST — AI RMF Core	https://airc.nist.gov/airmf-resources/airmf/5-sec-core/
11 · Breck et al. 2017 — The ML Test Score	https://research.google/pubs/the-ml-test-score-a-rubric-for-ml-production-readiness-and-technical-debt-reduction/

Titles and links are clickable. Sources support the concepts, not the hypothetical product's effectiveness.



Agentic AI Foundation

AGNTCon + MCPCon

North America

Where the **agentic stack** is being built.