MODULE 3 · SESSION 2

Control Inference Spend at Scale

How to reduce what each request costs without giving up the quality and coverage the product depends on.

Theory session

Beginner

Running case: VerdantCart Markets

ORIENTATION

Where this session sits

3.1 made it reliable; 3.2 makes it affordable at volume

MODULE 1
AI System Foundations

MODULE 2
Controlled AI Operations

MODULE 3
Strategic AI Operating Choices

SESSIONS IN THIS MODULE

3.1 Engineer Reliable AI Operations
Failure tolerance, service objectives and customer impact

3.2 Control Inference Spend at Scale — YOU ARE HERE
Reducing cost without giving up quality or coverage

3.3 Select an AI Vendor Portfolio
Capability, switching exposure and strategic control

SESSION OUTCOMES

What this session explains

Treating cost per request as a design variable

01

Unit cost decides what you can build

At 20% coverage the retailer had a sampling tool; at 100% it had a network control.

02

Cost is reduced at five places

Model choice, prompt and context, caching, batching, and the workload itself.

03

Cheaper is not the goal

The goal is more useful work per unit of spend, which is a different number.

04

Measure cost per outcome

Cost per request tells you very little on its own.

CONCEPT 1 · THE REFRAME

Cost per useful outcome, not cost per call

The metric that prevents the most common false economy

WORKING DEFINITION

Divide total inference spend by the number of outcomes the business actually wanted — resolved cases, correct decisions, avoided losses — rather than by the number of model calls made.

1

Why per-call misleads

A cheap model that answers wrongly twice, then escalates, costs more than one good answer.

2

What the case measured

$28m of shrinkage reduction against $420k of cost — an outcome ratio, not a per-call price.

3

The practical version

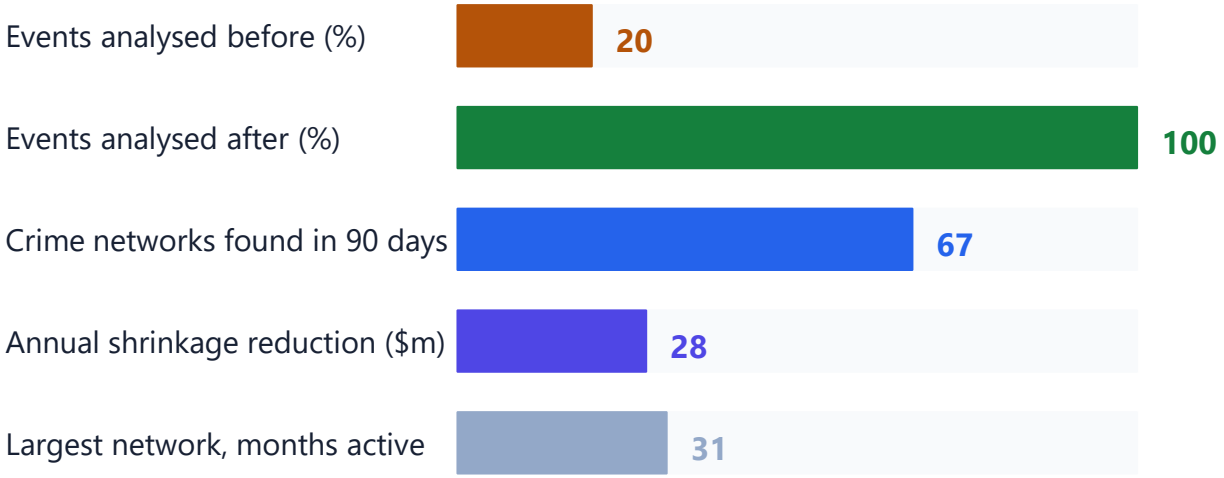
Report spend per thousand resolved cases, tracked monthly alongside quality.

CONCEPT 2 · EVIDENCE

What falling unit cost made possible

The change was in coverage, not in the model

REPORTED FIGURES FROM THE DEPLOYMENT



A 1,000× cost improvement is the reason the second bar was reachable at all.

WHY PARTIAL COVERAGE FAILED

The pattern was the evidence

Organised activity spread across stores and accounts is invisible when you look at one event in five.

THE STRATEGIC POINT

Cost is a capability constraint

A cost reduction large enough stops being an efficiency and becomes a new product.

THE CAUTION

Check the outcome, not the coverage

100% coverage is only worth paying for because of the $28m — not because it is complete.

CONCEPT 3 · LEVERS

Five places inference cost is actually reduced

Roughly in order of effort, easiest first

Lever	What you change	Typical trade-off
Right-size the model	Use the smallest model that clears the quality floor for each route.	Needs the per-route measurement from Session 1.3.
Trim the context	Send only what the model needs, not everything available.	Cutting too far degrades quality in ways that are hard to attribute.
Cache	Reuse answers for repeated or near-identical requests.	Stale answers — the freshness problem from Session 3.1.
Batch and schedule	Group work that does not need an immediate answer.	Latency. Only works for genuinely asynchronous work.
Redesign the workload	Change what gets asked, not just how it is asked.	The largest gains, and the most engineering — the 1,000× case.

THE DIFFERENCE THAT MATTERS

The first four make the same product cheaper. The fifth changes what the product can be, which is where the interesting gains are.

CONCEPT 4 · THRESHOLDS

Where cost reduction changes what is possible

Two questions that sort efficiency from capability

SIZE OF THE COST REDUCTION ▲

MARGIN IMPROVEMENT

Large saving, same product. Real money, no strategic change.

NEW CAPABILITY

Large saving that crosses a threshold. 20% coverage becomes 100%.

HOUSEKEEPING

Small saving, same scope. Worth doing, not worth a roadmap slot.

INCREMENTAL REACH

Small saving that extends scope modestly. Useful and rarely transformative.

COVERAGE OR SCOPE IT UNLOCKS →

THE QUESTION TO ASK

What would we do if this were free?

If the answer is “the same thing, more profitably”, it is efficiency. If it is something else, it is strategy.

THE RETAILER'S ANSWER

Reason over everything

Complete coverage was already the desired control. Only the price stood in the way.

CONCEPT 5 · TWO MINDSETS

Cutting cost and buying coverage

The same saving spent two different ways

CUT

Bank the saving

WHAT YOU DO

Reduce spend, keep the product as it is.

WHEN IT IS RIGHT

Margins are the binding constraint, or the product is already doing its whole job.

THE RISK

Optimising a product that a competitor is about to redefine with the same saving.

REINVEST

Buy scope with it

WHAT YOU DO

Spend the saving on coverage, depth or frequency.

WHEN IT IS RIGHT

A known blind spot exists and the only reason it persists is price.

THE EVIDENCE

20% to 100% of events, and 67 networks found in the first 90 days.

THE CHOICE TO MAKE EXPLICITLY

Decide this deliberately. A cost reduction that quietly becomes margin is a decision made by default rather than on purpose.

CONCEPT 6 · AGGRESSIVENESS

How hard to push on cost

The damage from over-optimising arrives later than the savings

UNDER-MANAGED

Nobody knows the cost per outcome. Spend grows with volume and surprises everyone at scale.

MANAGED

A quality floor, a cost target, and per-route reporting reviewed monthly.

OVER-OPTIMISED

Quality traded away invisibly. Savings appear this month; churn appears next quarter.

THREE GUARDRAILS

Guardrail	What it protects
A quality floor per route.	Stops cost work from silently degrading a segment.
Cost reported next to quality, always.	Prevents a saving being celebrated without its price.
Shadow-test before every change.	Measures what the cheaper path would have got wrong.

CONCEPT 7 · FAILURE MODES

Three ways cost programmes go wrong

All three look like good engineering at the time

Optimising the wrong number

Looks like Cost per call falling steadily.

Why it fails Cheap wrong answers generate retries, escalations and churn that cost more.

Fix Track cost per resolved outcome instead.

Caching without a freshness rule

Looks like A large, immediate saving.

Why it fails Stale answers are wrong answers, and the cache hides the decay.

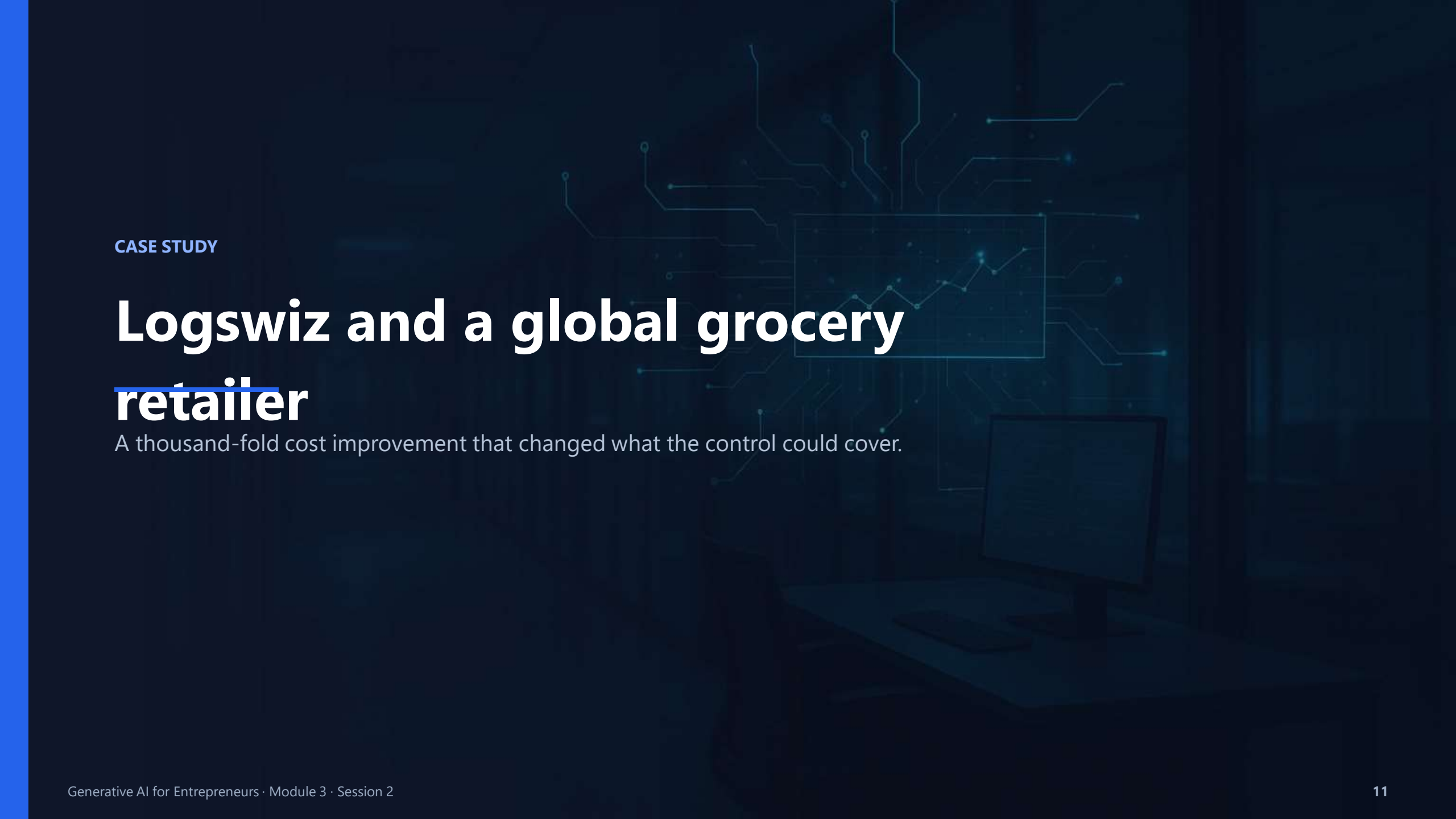
Fix Every cached value carries an age limit derived from how fast the truth changes.

Saving without deciding what to do with it

Looks like A successful efficiency project.

Why it fails The strategic option the saving created is never considered.

Fix Ask what you would build if inference were free, before banking the money.



CASE STUDY

Logswiz and a global grocery retailer

A thousand-fold cost improvement that changed what the control could cover.

CASE STUDY · LOGSWIZ AND A GLOBAL GROCERY RETAILER

From 20% of events to all of them

Cost reduction as the enabler of a network-wide control

BEFORE

Twenty per cent of the picture

THE ESTATE

A global grocery retailer operating across 2,800 locations.

THE COVERAGE

About 20% of point-of-sale events were analysed.

WHAT WAS MISSED

Organised retail crime distributed across stores, accounts, timing and repeated behaviour — patterns invisible in a sample.

AFTER

Complete coverage, and what it found

THE ENABLER

A 1,000× inference cost improvement made reasoning over 100% of transaction events viable.

THE FINDINGS

67 organised retail crime networks in the first 90 days; the largest had run 31 months and caused $8.4m in shrinkage.

THE RETURN

$28m annual shrinkage reduction, against $420k of cost in the first year.

READING THE CASE

Inference cost reduction was the key enabler for full coverage. The goal was not to make each inference cheaper — it was to make a stronger operating model possible.

Drawn from the session case pack.

APPLICATION

Bringing the theory to VerdantCart Markets

Managing inference spend as a design variable

Step	What you do	VerdantCart example
Define the outcome unit	What the business is actually buying.	A correctly fulfilled order line, and an avoided substitution complaint.
Report cost per outcome	Monthly, next to quality, per route.	Spend per thousand order lines, split by request type.
Right-size each route	Smallest model above the quality floor.	Availability scoring on the small model; exception review on the strong one.
Cache with an age limit	Derived from how fast the truth changes.	Availability scores cached for fifteen minutes, never longer.
Ask the free-inference question	Before banking any large saving.	At one-tenth the cost, score every item in every store continuously.
Guard the floor	No cost change ships without a shadow test.	One per cent double-run for two weeks before any route change.

WHAT COMES NEXT

Session 3.3 turns to who supplies all of this, and what depending on them costs you.

SUMMARY

What to carry out of this session

1. **Measure cost per outcome, not per call.**
   Cheap wrong answers are expensive once retries and escalations are counted.

2. **Five levers, in order of effort.**
   Right-sizing, context, caching, batching, and redesigning the workload.

3. **A large enough saving is a strategy question.**
   20% coverage became 100%, and found 67 networks in 90 days.

4. **Ask what you would build if it were free.**
   The answer tells you whether to bank the saving or spend it.

5. **Report cost next to quality, always.**
   A saving without its quality cost is not yet a result.

FURTHER READING

Sources for this session

Primary references behind each concept

COST-AWARE SERVING Chen, L., Zaharia, M. & Zou, J. (2023). FrugalGPT: how to use large language models while reducing cost and improving performance. *arXiv:2305.05176*.

DISTILLATION Hinton, G., Vinyals, O. & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv:1503.02531*.

SERVING EFFICIENCY Kwon, W. et al. (2023). Efficient memory management for large language model serving with PagedAttention. *SOSP 2023*.

UNIT ECONOMICS Anderson, E. & Kumar, N. (2007). Price competition with repeat, loyal buyers. *Quantitative Marketing and Economics*, 5(4), 333–359.

COST AND DEMAND Jevons, W. S. (1865). *The Coal Question*. London: Macmillan.

CLOUD ECONOMICS Armbrust, M. et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58.

ML SYSTEMS Huyen, C. (2022). *Designing Machine Learning Systems*. Sebastopol: O'Reilly.