

Select an AI Vendor Portfolio

How to balance capability, switching exposure, reliability and strategic control when choosing who supplies your AI stack.

Theory session

Beginner

Running case: VerdantCart Markets



Where this session sits

The final session: who supplies the stack, and what that costs you

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
Reducing cost without giving up quality or coverage

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

YOU ARE HERE

What this session explains

Choosing suppliers without giving away your future options

01

Buy, build or partner is a strategic choice

It depends on how central the capability is, not on which is cheaper today.

02

Switching cost is the real exposure

Not the price you pay now, but what it would take to leave.

03

Abstraction buys you options

Wegmans' API-first façade is what makes each specialist vendor replaceable.

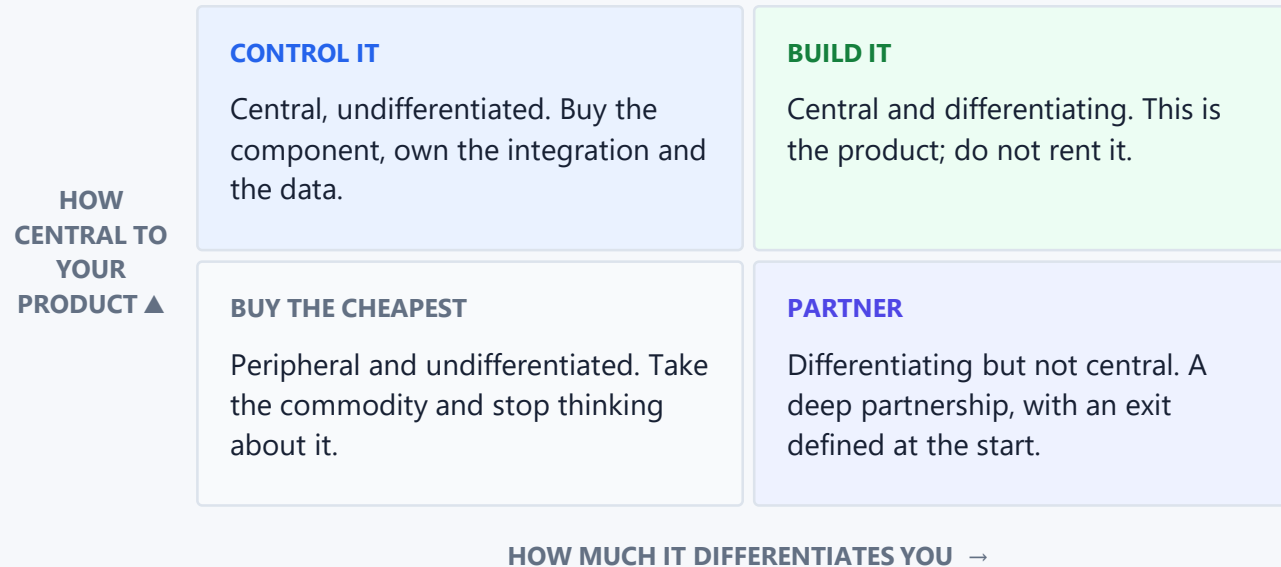
04

A portfolio beats a platform

Provided you own the layer that holds it together.

Two questions decide build, buy or partner

How central the capability is, and how much it differentiates you



THE COMMON ERROR

Buying the top-right

Renting the thing that makes you different means your advantage is available to every competitor at list price.

THE WEGMANS READING

Specialists around an owned core

Search, content, commerce and delivery were bought. The façade layer that joins them was not.

Williamson, O. (1985). The Economic Institutions of Capitalism. New York: Free Press.

Switching cost is what you are really buying

Measured in time and risk, not in licence fees

WORKING DEFINITION

Your exposure to a supplier is not the price you pay — it is what it would cost in money, time and risk to replace them next year. That number rises quietly every month you do not measure it.

1

Data gravity

The more of your data lives in their format, the more expensive leaving becomes.

2

Interface specificity

Code written against their API is code you rewrite when you leave.

3

Operational habit

Trained people, tuned workflows and runbooks are real switching costs nobody puts in the business case.

Shapiro, C. & Varian, H. (1999). Information Rules. Boston: Harvard Business School Press.

The layer that keeps suppliers replaceable

Wegmans' API-first façade, generalised

YOUR PRODUCT LOGIC

What makes the experience yours. Never delegated to a supplier.

YOUR FAÇADE LAYER

A stable interface your code calls. You own it, and it hides which vendor is behind it.

VENDOR ADAPTERS

One thin translation per supplier. The only place vendor-specific code is allowed to live.

VENDORS

Search, content, commerce, models.
Replaceable individually, without touching the layers above.

THE RULE

Vendor names appear in one layer only

If a supplier's SDK is imported across your codebase, you have no portfolio — you have a dependency.

THE COST

An abstraction is real work

It is worth building where switching is plausible, and over-engineering where it is not.

THE PAYOFF

Options you can actually exercise

A second supplier you can switch to in a week also changes what the first one charges.

Six criteria for an AI supplier

Capability is only the first, and rarely the deciding one

Criterion	The question to ask	How it bites later
Capability	Does it do the job well enough today?	Easy to assess, and the fastest to change.
Reliability	What is their stated objective, and their record?	Their outage becomes your outage, in your customers' eyes.
Economics	How does the price behave at ten times the volume?	A unit price that looked fine becomes the business model.
Portability	Can you export your data and prompts in a usable form?	Discovered at exactly the moment you want to leave.
Data terms	What can they do with what you send?	A clause agreed quickly becomes a compliance problem later.
Direction	Is their roadmap heading where you are?	A supplier moving away from your use case degrades slowly.

THE STEP EVERYONE SKIPS

Run this on the incumbent too, once a year. Most vendor problems are not bad choices — they are good choices that were never re-examined.

One platform or a portfolio of specialists

Wegmans moved from the first to the second, deliberately

SINGLE PLATFORM

One supplier, everything

WHAT YOU GET

One contract, one integration, one support line. Genuinely simpler.

WHAT YOU GIVE UP

Your pace of change becomes their roadmap — the legacy constraint Wegmans described.

GOOD WHEN

The team is small, the domain is standard, and speed of launch dominates.

PORTFOLIO

Specialists behind a façade

WHAT YOU GET

Best-of-breed in each area, and the ability to replace one part at a time.

WHAT YOU GIVE UP

Integration work, more contracts, and an abstraction layer you must maintain.

GOOD WHEN

Performance or feature pace is a competitive issue — 2× page performance, AI features at launch.

THE CONDITION THAT MAKES IT WORK

The portfolio only works if you own the joining layer. Without it, a portfolio is simply several lock-ins instead of one.

How many suppliers for the same capability

Redundancy has a price, and so does its absence

SINGLE SOURCE Cheapest and simplest. Their outage, price rise or policy change is yours with no alternative.	PRIMARY PLUS TESTED ALTERNATE One main supplier, a second integrated and periodically exercised. The usual right answer.	FULLY MULTI-SOURCED Traffic split across several. Maximum resilience, and real complexity and cost.
--	--	---

DECIDING PER CAPABILITY

If this is true	Then
An outage here stops the business.	A tested alternate is not optional.
Switching would take over three months.	Build the abstraction now, before you need it.
The capability is commodity.	Single-source it and spend the attention elsewhere.

Three vendor decisions that age badly

Each is entirely reasonable on the day it is made

Choosing on today's capability alone

Looks like A thorough feature comparison.

Why it fails Capability is the fastest-changing criterion and the easiest to switch for.

Fix Weight portability, economics at scale and direction at least as heavily.

An untested alternate

Looks like A second supplier named in the continuity plan.

Why it fails An integration never exercised does not work when you need it.

Fix Route real traffic to it on a schedule, however small.

Never re-examining the incumbent

Looks like A renewal signed without review.

Why it fails Switching cost grows quietly every month until the choice is no longer yours.

Fix Score the incumbent annually on the same six criteria as any challenger.

CASE STUDY

Wegmans Food Markets

A closed legacy platform replaced by a composable stack of specialist vendors.

A composable store built for resilience and speed

Specialist vendors behind an API-first façade the retailer owns

BEFORE

One closed platform

THE CONSTRAINT

A legacy closed third-party platform limited both performance and the delivery of new features.

THE EXPOSURE

The pace of new customer experiences was set by a single supplier's roadmap.

THE STRESS TEST

The legacy system had failed under conditions like the ones that would later arrive.

AFTER

Composable, best-of-breed

THE ARCHITECTURE

Azure serverless, Databricks event hubs and API-first façade services; live 2 April 2025 after beta testing with all customers.

THE SUPPLIERS

commercetools, Algolia, Contentstack, Adobe, Instacart Connect, Cooklist and Vercel, integrated rather than replaced by one platform.

THE EVIDENCE

Page performance doubled; digital orders grew 20%+ over the 2025 holidays and surged 50%+ during Winter Storm Fern in January 2026.

READING THE CASE

Vendor strategy became a route to resilience, speed and AI-enabled value — not a procurement exercise. The façade layer is what made the suppliers replaceable.

Bringing the theory to VerdantCart Markets

Assembling the supplier portfolio

Step	What you decide	VerdantCart example
Map each capability	Central or peripheral, differentiating or not.	Availability scoring is central and differentiating; search is central and not.
Set build, buy or partner	From the quadrant, not from price.	Build the availability model; buy search; partner for delivery.
Own the joining layer	A façade your code calls, vendor names in one place.	One internal API for search, so the provider can change in a sprint.
Score suppliers on six criteria	Including the incumbent.	Annual review scheduled with the contract renewal date.
Pick a concentration level	Per capability, not once for everything.	Single-source content; primary-plus-alternate for model inference.
Exercise the alternate	Real traffic, on a schedule.	Two per cent of inference to the secondary provider every week.

WHAT COMES NEXT

That closes the course: an architecture, trustworthy data, a routing policy, a measurement loop, bounded autonomy, a response plan, and the reliability, cost and supplier choices that make it a business.

What to carry out of this session

1 **Build what differentiates; buy what does not.**

Renting your advantage makes it available to everyone at list price.

2 **Your exposure is the switching cost, not the price.**

Data gravity, interface specificity and operational habit all raise it quietly.

3 **Own the layer that joins the suppliers.**

Without it, a portfolio is several lock-ins instead of one.

4 **Score suppliers on six criteria, annually.**

Capability, reliability, economics, portability, data terms and direction — the incumbent included.

5 **An untested alternate is not an alternate.**

Route real traffic to it on a schedule, however small.

Sources for this session

Primary references behind each concept

TRANSACTION COSTS	Williamson, O. (1985). <i>The Economic Institutions of Capitalism</i> . New York: Free Press.
LOCK-IN AND SWITCHING	Shapiro, C. & Varian, H. (1999). <i>Information Rules</i> . Boston: Harvard Business School Press.
MODULARITY AS OPTIONS	Baldwin, C. & Clark, K. (2000). <i>Design Rules: The Power of Modularity</i> . Cambridge, MA: MIT Press.
RESOURCE DEPENDENCE	Pfeffer, J. & Salancik, G. (1978). <i>The External Control of Organizations</i> . New York: Harper & Row.
CORE COMPETENCE	Prahalad, C. K. & Hamel, G. (1990). The core competence of the corporation. <i>Harvard Business Review</i> , 68(3), 79–91.
PLATFORM STRATEGY	Parker, G., Van Alstyne, M. & Choudary, S. (2016). <i>Platform Revolution</i> . New York: Norton.
ARCHITECTURE PRACTICE	Newman, S. (2021). <i>Building Microservices</i> (2nd ed.). Sebastopol: O'Reilly.