

— A CHAI<sup>™</sup> POSITION PAPER —

# Legacy *to* Agentic.

On the next wave of enterprise modernization, and what changes when *autonomous workflows*, not applications, become the unit of value.

CLOUDHEDGE · MAY 2026

[DOWNLOAD THE PDF](#) ↓

14-MIN READ · 796 KB · OPEN ACCESS

---

## The goalposts moved *quietly*.

*For fifteen years, "modernize" meant lifting legacy workloads off the mainframe or monolith, re-platforming them on the cloud, and calling it cloud-native. The work was real. The outcomes were real. The destination is no longer the same.*

**T**he enterprises pulling ahead in 2026 are not the ones with the most cloud-native services. They are the ones whose systems can reason, decide, and act with minimal human orchestration. We have come to call them **agentic enterprises**.

This shift changes what modernization means. Cloud-native was the destination when applications were the unit of value. Agentic is the destination when **autonomous workflows** are the unit of value. The implication for CIOs is sharp: a modernization roadmap still anchored on lift-and-shift, replatforming, and microservices is now a roadmap to last year's finish line.

---

### THE THREE ARGUMENTS OF THIS PAPER

- i.* The economic and strategic case for cloud-native modernization has been overtaken by the case for agentic modernization.
  - ii.* AWS (through Bedrock, Q, EKS, and a maturing set of agentic primitives) is the most viable foundation for enterprises making this leap.
  - iii.* CloudHedge's CHAI™ platform compresses the legacy-to-agentic journey from a multi-year program into a structured, measurable engagement.
-

If you are budgeting a modernization initiative for FY26 or FY27, the question is no longer *"how do we get our COBOL onto containers?"* It is *"how do our most expensive business processes start running themselves?"*

## From cloud-native to *agentic*.

The cloud-native era was defined by a single architectural premise: decompose monoliths into services, run them on elastic infrastructure, and let DevOps teams ship faster. It worked. Companies that completed the journey saw real gains in deployment velocity, infrastructure cost, and developer productivity.

But three things have changed in the last eighteen months.

**One.** AI agents have moved from demos to production. Foundation models can now perform multi-step reasoning, call tools, and execute workflows end-to-end. The bottleneck is no longer model capability. It is the integration surface between agents and the systems where business logic actually lives.

**Two.** The integration surface is mostly still legacy. The COBOL that runs core banking, the RPG that runs supply chain, the PL/SQL packages that run claims adjudication. These systems were never designed to be called by an agent. They were designed to be called by green-screen terminals and batch jobs.

**Three.** The economic gap is widening. A bank with agentic loan origination processes loans in minutes with two humans in the loop. A bank without it processes loans in days with twenty. The first bank's cost-to-serve is structurally lower, and the gap compounds every quarter.

---

*Modernization is no longer about  
where your code runs. It is about  
whether your code can participate in  
an agentic workflow.*

---

## Why legacy modernization stalled.

Most enterprise modernization programs over the last decade hit one of three walls.

### § *The replatforming wall.*

Teams lifted workloads to the cloud but kept the architecture intact. The result: cloud bills replaced datacenter bills, but the underlying systems are no more composable, no more queryable, and no more agent-ready than they were on-prem.

### § *The rewrite wall.*

Teams attempted full rewrites of legacy systems into modern stacks. Multi-year programs, multi-hundred-million-dollar budgets, and (in roughly two of three published cases) outcomes that under-delivered against the business case. The rewrite wall is where modernization budgets go to die.

### § *The integration wall.*

Teams modernized the periphery (APIs, mobile apps, customer portals) but left the core intact. A beautiful façade in front of a system that still cannot tell an agent what it knows. Every agentic workflow becomes a screen-scraping exercise.

The pattern is consistent: traditional modernization treats the legacy system as the problem and tries to remove it. The next wave treats the legacy system as a **knowledge asset** and makes it accessible to agents, code, and humans alike.

# The agentic transformation model.

CloudHedge's working definition of an agentic enterprise has four properties.

- 
- |   |  |
|---|--|
| <p>I</p> <p><b><i>Discoverable business logic</i></b></p> <p>Every meaningful business rule, whether it lives in COBOL, stored procedures, or undocumented batch jobs, is mapped, named, and queryable.</p> | <p>II</p> <p><b><i>Composable interfaces</i></b></p> <p>Core systems expose capabilities through APIs and event streams that agents can call, not just humans.</p>   |
| <p>III</p> <p><b><i>Resident intelligence</i></b></p> <p>Models with domain context (not generic LLMs) sit close to the workflows that need them, with appropriate guardrails.</p>                          | <p>IV</p> <p><b><i>Closed-loop automation</i></b></p> <p>Workflows execute end-to-end with human-in-the-loop only where regulation, judgment, or risk demand it.</p> |
- 

Getting from a typical Fortune 500 starting point to all four properties is a three-to-five-year program. But the order of operations matters. Most enterprises fail not because the destination is wrong but because they sequence the work wrong, chasing point AI wins before the underlying systems are agent-ready.

The right sequence:

*i*

### **Map**

Surface legacy business logic as queryable knowledge.

*ii*

### **Modernize**

Move the highest-value workflows onto agent-ready infrastructure.

*iii*

### **Orchestrate**

Run agentic processes across the modernized estate.

Skipping step one is the single most common failure mode in agentic transformation programs.

# AWS as the agentic foundation.

Several hyperscalers offer credible agentic primitives. We believe AWS has the strongest combination of three things enterprises actually need: model breadth, enterprise data gravity, and a mature partner ecosystem for the unglamorous work of legacy integration.

The AWS services that matter most for legacy-to-agentic programs:

---

SERVICE	ROLE IN THE AGENTIC STACK
<b>Amazon Bedrock</b>	Foundation models with private deployment, fine-tuning, and Guardrails for regulated workloads.
<b>Amazon Q</b>	Enterprise-grade agentic assistant; the front door for many business users.
<b>Bedrock Agents · Strands</b>	Multi-step reasoning and tool calling for agentic workflows.
<b>Amazon EKS</b>	Container substrate for modernized legacy workloads.
<b>AWS Lambda</b>	Event-driven glue between modernized services and agentic workflows.
<b>Aurora · DynamoDB</b>	Modernized data tier supporting transactional and agentic read patterns.
<b>AWS Mainframe Modernization</b>	Targeted runtime for migrated mainframe workloads, with managed tooling.
<b>Amazon SageMaker</b>	Custom model training for domain-specific reasoning.

---

AWS sales teams already have field motions around each of these services. CloudHedge engagements that map cleanly to those motions get pulled into AWS-driven opportunities. This paper is meant to be that map.

# The CHAI approach.

CHAI™ by CloudHedge compresses the three-step framework into a structured engagement, not a multi-year science project. The platform comprises three integrated products.

---

PRODUCT · I

## CHAI DART™

Discovery and analysis of legacy estates, surfacing business logic, dependencies, and risk for downstream modernization.

---

PRODUCT · II

## CHAI Flow™

Transformation and migration of mapped workloads onto AWS-native infrastructure, equivalence-tested through cutover.

---

PRODUCT · III

## CHAI Universe™

The orchestration and operations layer that runs the modernized estate, including agentic workflow management.

---

What makes CHAI different from traditional modernization tooling:

- i.* **Built for AWS first.** Reference architectures, integrations, and engagement playbooks are AWS-native, not retrofitted.
- ii.* **Agentic-aware from day one.** Discovery output is structured for downstream agent consumption, not just human reading.
- iii.* **Measured in quarters, not years.** A typical CHAI engagement delivers a production workload in six to nine months, with the wider estate modernized in a rolling program.

## What this looks like in *practice*.

Three vignettes from CloudHedge engagements. Identifying details have been generalized; quantitative outcomes are from real customer programs.

### CASE THE FIRST · BFSI

#### § *A North American mid-tier bank: core banking modernization.*

##### STARTING POINT

Core banking system on a mainframe. Twelve million lines of COBOL. A forty-year-old codebase. New product launches took 9–14 months. Compliance reporting required 200 hours of manual reconciliation per quarter.

##### WHAT WE DID

CHAI DART™ discovered and mapped the COBOL estate in eleven weeks. CHAI Flow™ migrated customer onboarding and KYC workflows onto AWS: EKS, Aurora, Bedrock for document understanding. CHAI Universe™ runs the orchestration layer.

##### OUTCOME

New product launch cycle reduced sharply. Compliance reconciliation now agentic, with human review on exceptions only.

##### PRODUCT LAUNCH CYCLE

*11 months* → **7 weeks**

##### RECONCILIATION, PER QUARTER

*200 hrs* → **12 hrs**

## § *A European insurer: claims modernization.*

### STARTING POINT

Claims adjudication on a 25-year-old policy admin system. Average claim resolution time: 14 days. Agentic experimentation blocked because claims rules were undocumented and lived only in the system.

### WHAT WE DID

CHAI DART™ surfaced the claims rule set as a structured, queryable knowledge layer. CHAI Flow™ modernized the rules engine onto AWS. Bedrock Agents drive first-pass adjudication, with human review on the long tail.

### OUTCOME

Resolution time fell sharply for the 70% of claims in the agentic tier. Adjuster headcount redeployed to high-complexity work; no layoffs.

#### AVERAGE CLAIM RESOLUTION

~~14 days~~ → **38 hrs**

#### CLAIMS AUTO-ADJUDICATED

**70%**

§ *A US manufacturer: supply chain modernization.*

---

STARTING POINT

Supply chain orchestration on AS/400 with RPG. Demand-planning agents could not access inventory state in real time. Stockout rate 4.2%.

WHAT WE DID

CHAI DART™ mapped the RPG estate. CHAI Flow™ migrated inventory and demand-planning workflows onto AWS. Agentic demand-planning loops run against the new data tier.

OUTCOME

Stockouts collapsed within two quarters of go-live. Working capital tied up in safety stock fell materially.

STOCKOUT RATE, TWO QUARTERS

~~4.2%~~ → **0.9%**

WORKING CAPITAL RELEASED

**~\$34M**

## Three things to do this quarter.

If you are responsible for modernization at enterprise scale, the next ninety days are where the FY26 program either gains compounding momentum or quietly falls behind. Three concrete moves.

---

*i*

### Stop funding replatforming with no agentic destination.

If a modernization initiative on your roadmap doesn't have a clear answer to *"what agentic workflow does this unlock,"* reconsider its priority.

---

*ii*

### Invest first in discovery.

You cannot build agentic workflows on top of a system whose business logic is unknown. The cheapest move in the next ninety days is to map what you have.

---

*iii*

### Modernize one high-value workflow end-to-end.

Trying to modernize the whole estate at once is the rewrite wall. Pick claims, loan origination, or supply planning (whichever yields the highest ROI) and ship.

---

*§ CloudHedge and AWS are jointly investing in this approach. Assess where your estate sits on the agentic readiness curve.*

[Take the readiness assessment →](#)

[Talk to CloudHedge →](#)

---

---

ABOUT CLOUDHEDGE

CloudHedge is the company behind CHAI™, the legacy-to-agentic transformation platform. We work with enterprises in BFSI, healthcare, insurance, and manufacturing to modernize legacy estates onto AWS and unlock agentic workflows on top of them. AWS Advanced Tier Services Partner, with specializations in mainframe modernization and migration.

Learn more at [cloudhedge.io](https://cloudhedge.io). For partnership inquiries: *[insert partner email]*.



© MMXXVI · CloudHedge.

CHAI™, CHAI DART™, CHAI Flow™, and CHAI Universe™ are trademarks of CloudHedge. AWS, Amazon Bedrock, Amazon Q, Amazon EKS and other AWS marks are trademarks of Amazon.com, Inc. or its affiliates.