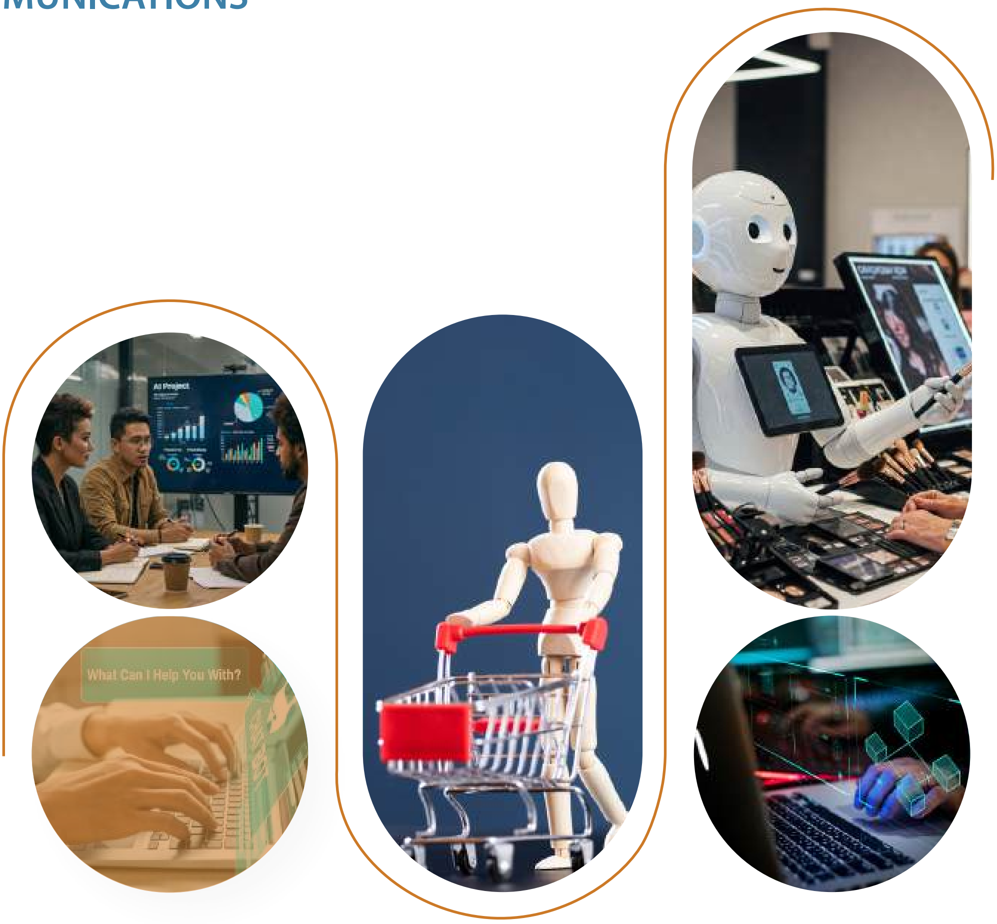


BUILD VS. PARTNER:

A THREE-YEAR TCO ANALYSIS OF MULTI-AGENT AI DEPLOYMENT FOR COMMERCE AT SCALE

ABOUT THE AUTHOR

Ankur Jain
Associate Vice President, Tata Communications
Ankur Jain is a seasoned AI and digital transformation expert with 15+ years of experience driving product innovation, business strategy, and operational excellence across global markets and high-growth tech environments.

EXECUTIVE SUMMARY

The AI revolution is redefining the future of commerce. With millions of monthly active users, global platforms are expected to deliver real-time, hyper-personalised experiences across languages, categories, and channels. The complexity of deploying generative AI at this scale is immense. Enterprises must decide: should they build their own AI stack from scratch, or partner with a specialist? This whitepaper explores the total cost of ownership (TCO) over three years for deploying a multi-agent AI system, comparing the build-it-yourself path with partnering through Tata Communications' CXaaS offering powered by Vayu Cloud. The analysis reveals a potential **40% TCO** reduction through partnership—maintaining the same level of performance and control.

LET'S TAKE A SCENARIO

Imagine Acme Commerce, a global B2C giant with 80 million monthly active users spanning fashion, electronics, daily groceries and multiple other categories. The business thrives on rapid customer service, rich personalisation, and real-time recommendations.

However delivering this requires AI agents that can handle vast volumes, respond in multiple languages, and learn from highly diverse products and customer datasets.

At this scale, traditional deterministic AI isn't enough. Acme needs a multi-agent system leveraging multiple foundation models, each fine-tuned for specialised tasks and working in orchestration. This brings scale, speed, and flexibility, but it also introduces significant complexity, infrastructure requirements, and cost inefficiency.

But before we dive into the specifics, let's first clearly understand what multi-agent AI entails.

WHY LEVERAGE MULTIPLE AGENTS WITH DIFFERENT FOUNDATIONAL MODELS?

A multi-agent AI system doesn't rely on a single AI model to do everything. Instead, it distributes tasks across a set of agents. These agents operate in a shared context, call each other when needed, and are coordinated by a central orchestrator. The benefits?

Specialisation for efficiency:

Different foundational models, such as Qwin, Llama, and DeepSeek, are built with unique strengths. For instance, Qwin excels in customer Q&A, Llama shines in generating creative text, and DeepSeek is designed for complex reasoning and multi-step problem-solving. We can ensure that the most capable model handles each task by deploying multiple specialised agents.

Optimised load-balancing

As user bases scale—potentially reaching 50-100 million monthly active users—it's crucial to distribute workloads across multiple models to maintain optimal performance. This approach enhances system efficiency and mitigates the risk of overloading any single model, ensuring consistent service delivery even during peak times.

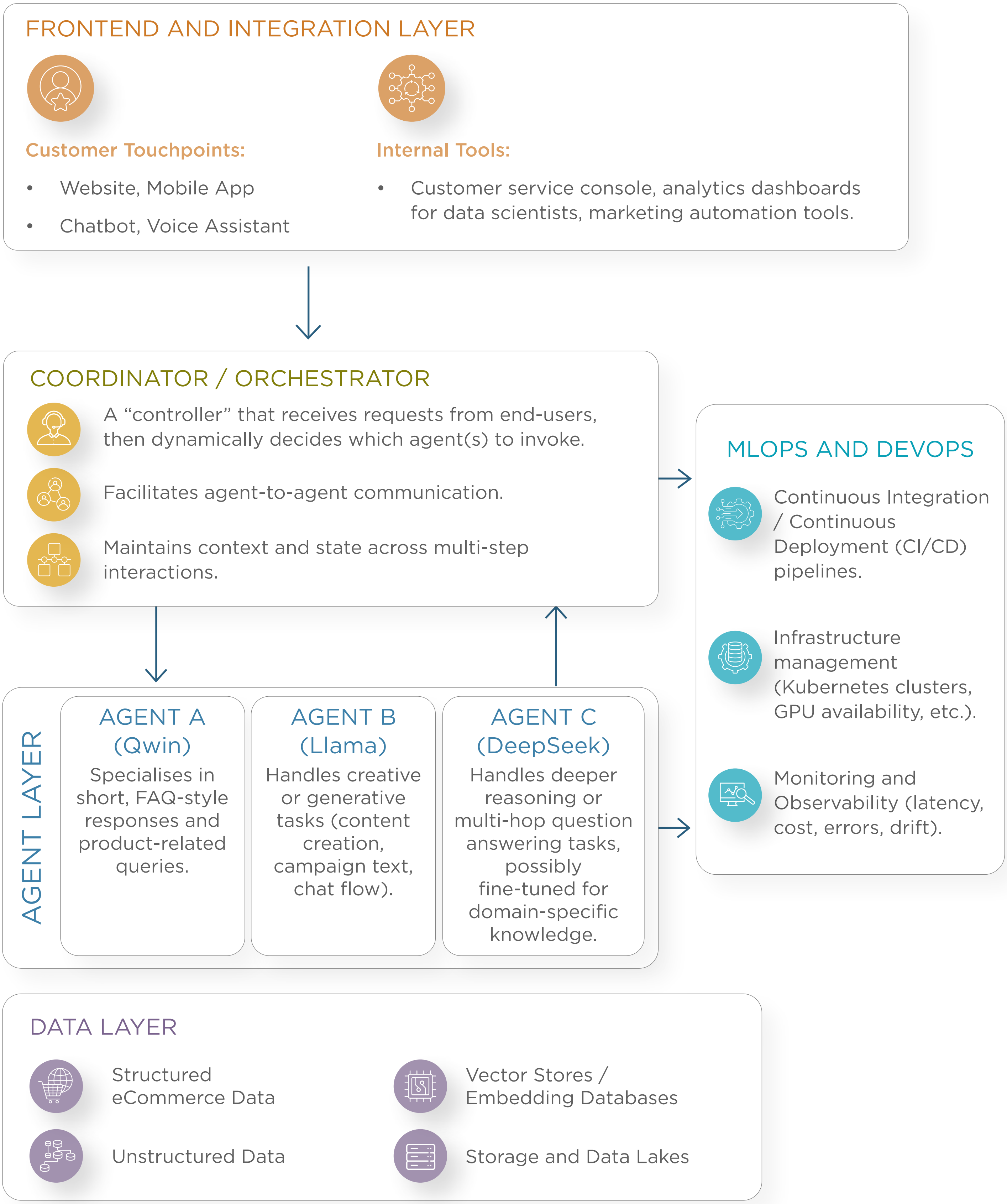
Non-deterministic, autonomous interactions:

These agents are not isolated entities; they operate autonomously, calling on each other—or even external services—when necessary to complete tasks. For example, suppose Llama requires a fact to substantiate its output. In that case, it can automatically call on DeepSeek's retrieval agent to validate the data, fostering seamless collaboration and enhancing overall task execution.

However, designing such a system demands a complete architecture—from data ingestion to inferencing to orchestration logic—and a strategy for managing latency, concurrency, and compliance.

OVERVIEW OF AN AGENTIC AI WORKFLOW

HIGH-LEVEL ARCHITECTURE



FRONTEND AND INTEGRATION LAYER

Customer Touchpoints:

- Website, Mobile App
- Chatbot, Voice Assistant

Internal Tools:

- Customer service console, analytics dashboards for data scientists, marketing automation tools.

COORDINATOR / ORCHESTRATOR

A "controller" that receives requests from end-users, then dynamically decides which agent(s) to invoke.

Facilitates agent-to-agent communication.

Maintains context and state across multi-step interactions.

MLOPS AND DEVOPS

Continuous Integration / Continuous Deployment (CI/CD) pipelines.

Infrastructure management (Kubernetes clusters, GPU availability, etc.).

Monitoring and Observability (latency, cost, errors, drift).

AGENT LAYER

AGENT A (Qwin)

Specialises in short, FAQ-style responses and product-related queries.

AGENT B (Llama)

Handles creative or generative tasks (content creation, campaign text, chat flow).

AGENT C (DeepSeek)

Handles deeper reasoning or multi-hop question answering tasks, possibly fine-tuned for domain-specific knowledge.

DATA LAYER

Structured eCommerce Data

Vector Stores / Embedding Databases

Unstructured Data

Storage and Data Lakes

PHASES OF AN AGENTIC AI PROJECT

Building an agentic AI solution requires a strategic, phased approach. While certain stages may run in parallel, a linear view helps align stakeholders, budget cycles, and execution timelines. Below is a high-level overview of the five critical phases, each with clear outcomes and interdependencies.

01 DATA PREPARATION: THE FOUNDATION LAYER

This phase sets the stage for everything downstream. It involves:

- Consolidation and Cleaning: Ingesting structured and unstructured data across sources (eCommerce transactions, user logs, chat transcripts).
- Transformation and Governance: Normalising formats, applying anonymisation protocols, and ensuring compliance (e.g., GDPR).
- Optional Embeddings: Generating vector representations if planning Retrieval-Augmented Generation (RAG).

Outcome: Clean, standardised, and secure datasets ready for model training, with clear governance protocols.

02 LLM TRAINING (RAG / FINE-TUNING): BUILDING INTELLIGENCE

This phase aligns the model(s) with domain-specific context and user expectations:

- Model Strategy: Choosing between fine-tuning, few-shot learning, or RAG-based enhancement.
- Infrastructure Setup: Leveraging GPU/TPU or cloud environments for scalable training.
- Evaluation and Alignment: Ensuring brand tone, factual accuracy, and performance under load.

Outcome: Domain-tuned models or RAG pipelines benchmarked for quality and ready for agent integration.

03 AGENT BUILDING: CREATING SPECIALISED CAPABILITIES

Here, intelligent agents are constructed and tuned for specific tasks:

- Role Assignment: Different agents (e.g., Q&A, reasoning, creative writing) mapped to models like Qwin, Llama, DeepSeek.
- Personality and Protocols: Defining how agents behave, respond, and collaborate.
- Compliance Controls: Role-based access and security protocols embedded

Outcome: Modular, tested agents with defined roles and behaviour, ready for orchestration.

04 INFERENCING: REAL-TIME INTELLIGENCE DELIVERY

This operational phase ensures low-latency, scalable, and observable inferencing:

- Deployment: Real-time model execution environments with auto-scaling.
- Monitoring: Token usage, latency, and user interactions logged and visualised.
- Optimisation: Performance tuning through prompt engineering, model quantisation, and A/B testing.

Outcome: Production-ready inferencing infrastructure continuously monitored and optimised.

05 ORCHESTRATION: SEAMLESS AGENT COLLABORATION

The final phase focuses on ensuring the agents work in harmony:

- Central Orchestrator: Directs traffic between agents and manages handoffs and fallbacks.
- Context Management: Maintains memory and ensures relevant information sharing.
- Governance: Enforces SLAs, budgets, and non-deterministic agent behaviour safely.

Outcome: Fully operational multi-agent ecosystem governed by SLAs, with cost and latency safeguards.

For a deeper dive into tools, techniques, and detailed workflows in each phase, refer to the **Appendix: Section 1 - Phases of an Agentic AI project**

COST CONSIDERATIONS ACROSS THESE PHASES

Implementing AI at scale involves many cost factors—some visible upfront, others emerging across the lifecycle. Enterprises must look beyond infrastructure and licensing to include the full spectrum of development, orchestration, and ongoing maintenance.

DATA PREPARATION

This includes setting up data pipelines, ETL tooling, storage, data labelling, and ensuring governance and compliance. The scope and scale of your dataset— especially across multiple languages, categories, and systems—can significantly influence this cost layer.

LLM TRAINING (FINE-TUNING AND RAG)

Beyond compute-intensive training runs, costs can balloon due to hyperparameter tuning, validation cycles, licensing models, and managing model versions. Compute is typically the largest line item here.

AGENT DEVELOPMENT

Building intelligent agents requires skilled resources, integration with internal and third-party APIs, orchestration frameworks, and secure, compliant workflows. Documentation and testing add to the cost—especially for multi-agent environments.

INFERENCE AT SCALE

Once deployed, inference costs can exceed training, especially with high query volumes. Autoscaling, load balancing, observability, and continuous optimisation must be factored into ongoing operational expenses.

ORCHESTRATION ACROSS AGENTS

For systems using multiple AI agents, there are additional costs associated with communication, shared memory, fallback logic, and traceability. Interactions between agents can carry infrastructure and compute overhead.

CROSS-CUTTING OVERHEADS

Salaries, compliance, project coordination, and contingency planning often account for the largest hidden costs in enterprise AI. These are essential to budget for, especially in long-term programs involving continuous updates.

For a comprehensive view of specific cost drivers, variables, and decision-making considerations across each stage, please refer to the **Appendix: Section 2 - Enterprise AI Cost Guide.**

PUTTING IT ALL TOGETHER: A PRACTICAL FRAMEWORK FOR TCO ESTIMATION

Below is a structured example of how to approach this calculation. The actual figures will depend on your cloud provider's quotes, internal resource costs, project timelines, and more.

Phase	Component	Description	Est. Cost
Data Preparation (Year 1)	Data Pipeline Engineering	2 FTEs × 6 months = 12 FTE-months @ $X/month	$X
	Data Storage	100 TB @ $Y/TB/month × 12 months	$Z
	Labeling / Annotation	1M items × $0.02/item	$20,000
	Governance Tool	Annual licensing	$A
Model Training (Year 1, repeatable)	Model Licensing	Annual or usage-based fee	$X
	GPU Compute (Training)	500 GPU hours @ $Y/hour	$Z
	Hyperparameter Tuning	200 GPU hours @ $Y/hour	$Z
	Evaluation & QA	1 FTE × 2 months = 2 FTE-months	$W
AI Development (Year 1–2)	Agent Development	3 AI Engineers × 6 months = 18 FTE-months	$X
	Integration and Security	2 Security Engineers × 2 months	$Y
	Testing and Documentation	1 QA Engineer × 3 months	$Z
Inference and Operations (Year 2+)	Production Compute	10,000 QPS N GPU nodes @ $X/node/month × 12	$...
	Autoscaling / Orchestration	Kubernetes overhead, etc.	$...
	Logging and Monitoring	$A per million logs, scaled to MAUs	$...
Multi-Agent Orchestration (Ongoing)	Controller Dev and Maintenance	2 FTEs × 12 months = 24 FTE-months Maintenance	$X
	Context Storage	Redis or similar @ $Y/month	$Z/year
	Intra-Agent Overhead	Avg. 1.5 model calls/request + 50% inference cost	Adjusted %

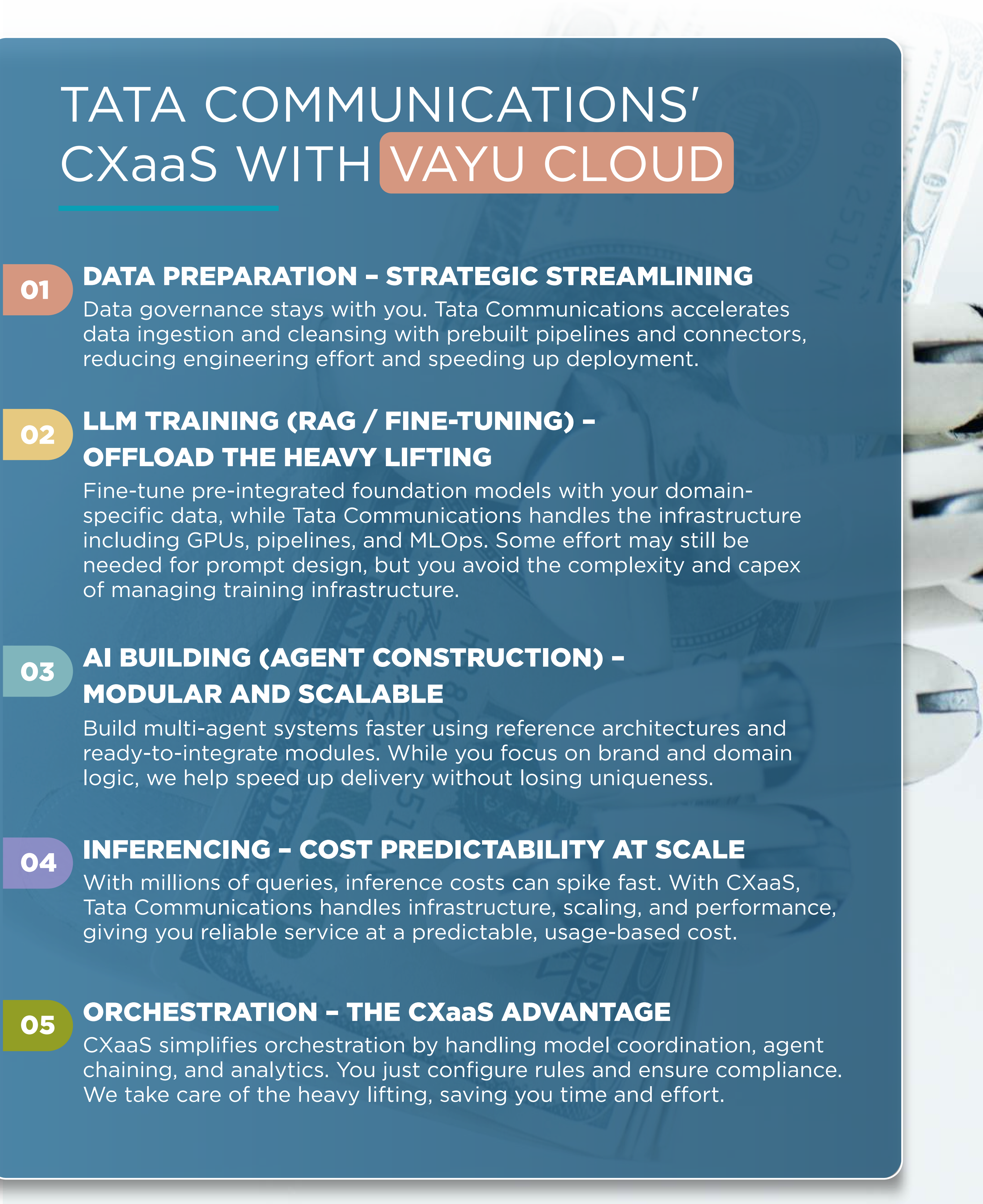
FINAL CALCULATION: CAPEX + OPEX

Once itemised, add these costs across all phases for a comprehensive TCO. You may choose to categorise them as:

- CapEx: Upfront investments in data preparation, engineering, and training
- OpEx: Ongoing costs tied to inference, monitoring, and maintenance

This structure gives finance and technology leaders the granularity to model different what-if scenarios while aligning investment decisions with long-term business goals.

Scalable AI isn't just about cost—it's shaped by user demand, model complexity, orchestration depth, and more. See **Appendix: Section 3 - Key Variables and Considerations for the list of impact factors.**



TATA COMMUNICATIONS' CXaaS WITH VAYU CLOUD

01

DATA PREPARATION – STRATEGIC STREAMLINING

Data governance stays with you. Tata Communications accelerates data ingestion and cleansing with prebuilt pipelines and connectors, reducing engineering effort and speeding up deployment.

02

LLM TRAINING (RAG / FINE-TUNING) – OFFLOAD THE HEAVY LIFTING

Fine-tune pre-integrated foundation models with your domain-specific data, while Tata Communications handles the infrastructure including GPUs, pipelines, and MLOps. Some effort may still be needed for prompt design, but you avoid the complexity and capex of managing training infrastructure.

03

AI BUILDING (AGENT CONSTRUCTION) – MODULAR AND SCALABLE

Build multi-agent systems faster using reference architectures and ready-to-integrate modules. While you focus on brand and domain logic, we help speed up delivery without losing uniqueness.

04

INFERENCING – COST PREDICTABILITY AT SCALE

With millions of queries, inference costs can spike fast. With CXaaS, Tata Communications handles infrastructure, scaling, and performance, giving you reliable service at a predictable, usage-based cost.

05

ORCHESTRATION – THE CXaaS ADVANTAGE

CXaaS simplifies orchestration by handling model coordination, agent chaining, and analytics. You just configure rules and ensure compliance. We take care of the heavy lifting, saving you time and effort.

IN SUMMARY

Tata Communications' CXaaS + Vayu Cloud

is purpose-built to simplify enterprise AI deployment—reducing time, cost, and complexity across the entire lifecycle. From training to orchestration, it shifts the burden of infrastructure and integration to a proven, end-to-end platform—without compromising control, scalability, or performance. Your teams stay focused on strategy and outcomes while the heavy lifting is handled by an enterprise-grade AI fabric designed for tomorrow's intelligent businesses.

THE DECISION POINT: BUILD VS. PARTNERING WITH TATA COMMUNICATIONS

THREE-YEAR TCO: A COMPARATIVE SNAPSHOT

The table compares the TCO of building the AI system in-house versus partnering with Tata Communications for a multi-agent deployment (50–100 million monthly users). Costs are broken down by phase and summarised annually. All figures are illustrative and subject to vary based on factors such as usage patterns, model size, and negotiated terms.

YEAR-BY-YEAR TCO

[refer Appendix Section 4: TCO ILLUSTRATION (BUILD/DIY) for detailed calculations]

Phase	Build (Year 1)	Partner (Year 1)	Build (Year 2)	Partner (Year 2)	Build (Year 3)	Partner (Year 3)
Data Prep	$220k	$150k	$80k	$60k	$60k	$40k
LLM Training	$130k	$70k	$200k	$120k	$120k	$80k
AI Building	$280k	$150k	$100k	$70k	$50k	$30k
Inferencing	$400k	$280k	$500k	$360k	$600k	$420k
Orchestration	$520k	$200k	$300k	$150k	$200k	$100k
Annual Sub-Total	**$1.55M**	**$850k**	**$1.18M**	**$760k**	**$1.03M**	**$670k**

This data shows significant cost savings when partnering, thanks to prebuilt pipelines, scalable GPU infrastructure, and reduced engineering overhead. Inference costs drop by 25–30%, and orchestration is significantly more cost-efficient when outsourced.

Across three years, partnering with Tata Communications can yield nearly $1.5 million in savings, representing a 40% TCO reduction. These savings stem from avoiding capital expenditures on GPUs, lower orchestration development costs, and streamlined integration provided by CXaaS. However, the actual savings will vary based on contract specifics, scale, and customisation needs.

QUALITATIVE BENEFITS OF PARTNERING WITH TATA COMMUNICATIONS

Partnering with Tata Communications for CXaaS offers a wealth of strategic advantages that extend beyond direct cost savings:

01

ACCELERATED TIME TO VALUE

Leverage our prebuilt modules rather than investing time and resources into building your own orchestration, vector stores, and retrieval flows from scratch. This streamlined approach can shorten your time to market by several months, if not a year.

02

SEAMLESS SCALABILITY AND ELASTICITY

As you experience fluctuations in user demand—whether from large campaigns or seasonal surges—Tata Communications enables effortless infrastructure scaling. This alleviates the burden on your internal teams for capacity planning, GPU provisioning, and queuing strategies, ensuring optimal performance at all times.

03

ACCESS TO WORLD-CLASS EXPERTISE

Multi-agent AI orchestrations require specialised skills in engineering, MLOps, and advanced machine learning. Partnering with Tata Communications grants you access to our seasoned experts, significantly reducing your risk and minimising the need for expensive training.

04

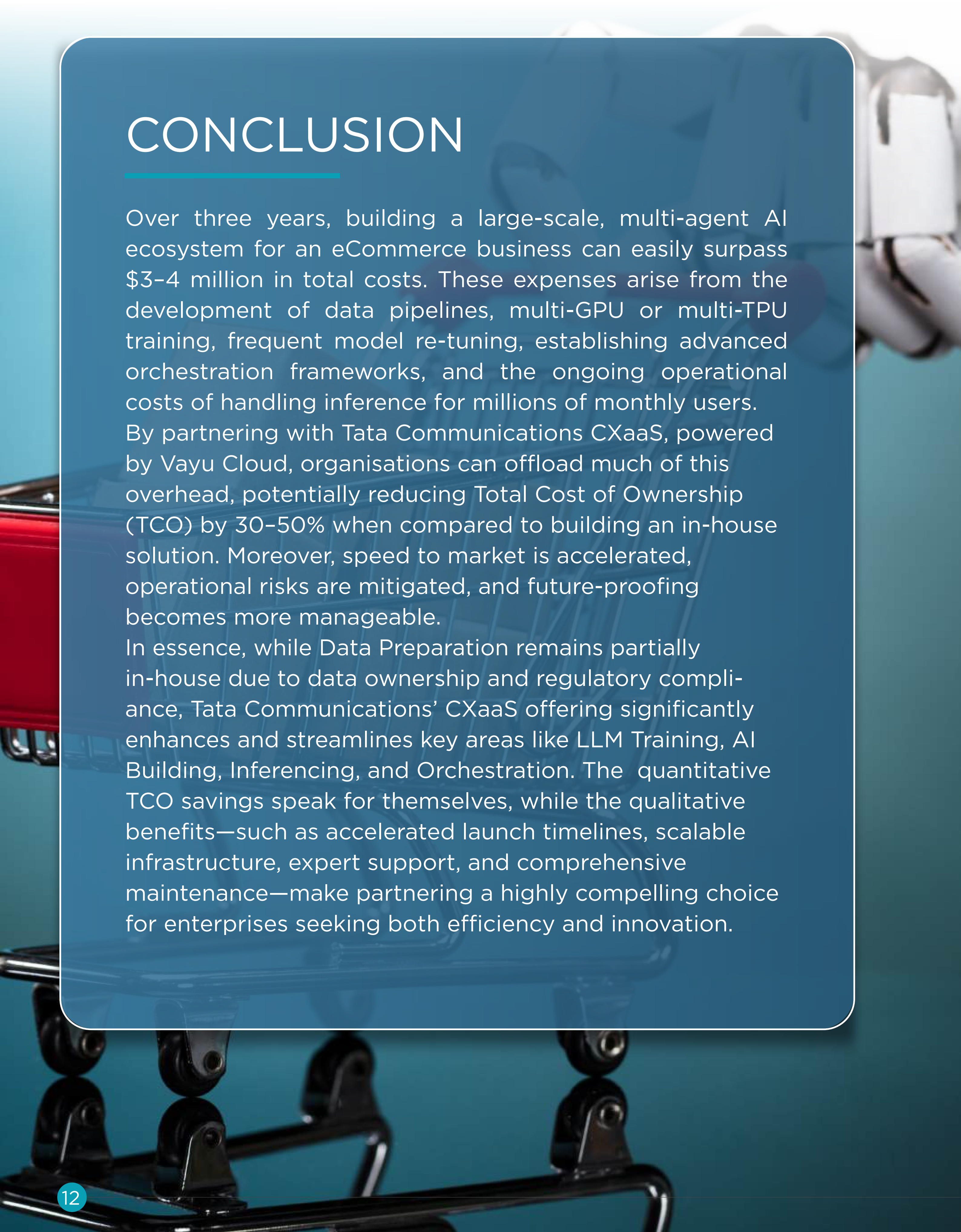
COMPREHENSIVE MONITORING AND MAINTENANCE

Tata Communications manages the ongoing operational tasks of logging, monitoring, compliance, and updates, freeing your MLOps teams from time-consuming maintenance duties. This allows you to focus on strategic initiatives rather than routine upkeep.

05

FUTURE-PROOFING YOUR AI ECOSYSTEM

As AI technology evolves, Tata Communications remains at the cutting edge, proactively upgrading your systems to ensure you stay competitive. With our expertise, you won't need to worry about re-architecting your entire solution whenever new advancements emerge.



CONCLUSION

Over three years, building a large-scale, multi-agent AI ecosystem for an eCommerce business can easily surpass $3–4 million in total costs. These expenses arise from the development of data pipelines, multi-GPU or multi-TPU training, frequent model re-tuning, establishing advanced orchestration frameworks, and the ongoing operational costs of handling inference for millions of monthly users.

By partnering with Tata Communications CXaaS, powered by Vayu Cloud, organisations can offload much of this overhead, potentially reducing Total Cost of Ownership (TCO) by 30–50% when compared to building an in-house solution. Moreover, speed to market is accelerated, operational risks are mitigated, and future-proofing becomes more manageable.

In essence, while Data Preparation remains partially in-house due to data ownership and regulatory compliance, Tata Communications’ CXaaS offering significantly enhances and streamlines key areas like LLM Training, AI Building, Inferencing, and Orchestration. The quantitative TCO savings speak for themselves, while the qualitative benefits—such as accelerated launch timelines, scalable infrastructure, expert support, and comprehensive maintenance—make partnering a highly compelling choice for enterprises seeking both efficiency and innovation.

APPENDIX:

SECTION 1: PHASES OF AN AGENTIC AI PROJECT

Below are the key phases, each culminating in deliverables and requiring iteration. These phases often happen in parallel or with overlaps, but it's helpful to structure them linearly for budgeting and planning.

DATA PREPARATION

DATA INGESTION AND CONSOLIDATION

- ETL pipelines for eCommerce data (transactions, inventory, user behaviours).
- Aggregation of user interaction logs, reviews, and chat transcripts.
- Tools involved: Spark, Kafka, Flume, or cloud-based data pipeline services.

DATA CLEANING AND TRANSFORMATION

- Handling duplicates, missing values, outliers.
- Normalising data formats (JSON, CSV, Parquet, etc.).
- Ensuring consistent data across all markets and platforms.

DATA LABELING AND ANNOTATION

- For supervised fine-tuning or for RAG (Retrieval-Augmented Generation) test sets.
- Tools: Labelling platforms (Labelbox, internal custom labeling system, etc.).

DATA GOVERNANCE AND SECURITY

- Access control, GDPR/CCPA compliance, anonymisation.
- Setting up data retention policies.

EMBEDDING GENERATION (optional within Data Preparation or early LLM Training)

- Generating embeddings for each chunk of text if you plan to do RAG.
- Tools: Sentence-transformers, OpenAI embeddings, or a self-hosted embeddings model.

DELIVERABLES

Clean, standardised datasets ready for training/fine-tuning.

Proper data lineage documentation and governance structure.

Vector store populated (if using RAG).

LLM TRAINING (RAG / FINE-TUNING)

FOUNDATIONAL MODEL SELECTION AND ARCHITECTURE

- You have multiple models (Qwin, Llama, DeepSeek). Decide which tasks each model is responsible for.
- Some might remain few-shot; others might be thoroughly fine-tuned.

DATA PREPARATION FOR TRAINING

- Splitting data into training/validation/test sets.
- Creating domain-specific corpora (product descriptions, chat logs, etc.).
- Potentially transforming data into a knowledge base for RAG.

FINE-TUNING / TRAINING STRATEGY

- **RAG Approach:** Use a pre-trained model + a retrieval mechanism (vector store).
  - Add domain knowledge by chunking relevant documents and injecting them into prompts.
- **Full Fine-tuning:** Train the model on eCommerce domain data.
  - Integrate RLHF (Reinforcement Learning from Human Feedback) if you want higher alignment.

TRAINING INFRASTRUCTURE SETUP

- Access control, GDPR/CCPA compliance, anonymisation.
- Setting up data retention policies.

MODEL VALIDATION AND EVALUATION

- GPU/TPU clusters, HPC environment or a cloud-based training pipeline.
- Possibly parallel training or multi-node training for large data sets.

DELIVERABLES

Domain-tuned LLM(s) or RAG pipeline with tested retrieval.

Clear evaluation metrics and performance thresholds.

AGENT BUILDING

AGENT CONSTRUCTION

- **Agent A (Qwin):** Specialised for Q&A. Possibly lighter model with narrower scope.
- **Agent B (Llama):** Larger generative tasks; may integrate marketing or campaign text generation.
- **Agent C (DeepSeek):** More advanced reasoning tasks or multi-hop queries.

AGENT "PERSONALITY" AND PROTOCOLS

- Defining how each agent "speaks" or responds (tone, style).
- Standardising prompt format and context injection.
- Setting up query limits, timeouts, and fallback behaviours.

AGENT INTEGRATION

- Restful APIs or gRPC endpoints for each model.
- Mechanisms to hand off partial context or partial solutions from one agent to another.

SECURITY AND COMPLIANCE

- Ensuring PII does not leak during agent interactions.
- Role-based access control (some agents might handle sensitive data, others not).

TESTING AND VALIDATION

- End-to-end tests covering multi-agent orchestration.
- Non-deterministic scenario tests (agent autonomy).

DELIVERABLES

Agent-based microservices or orchestrated systems.

Thoroughly tested multi-agent flow.

INFERENCING

DEPLOYMENT ENVIRONMENT

- Real-time inference on GPUs or CPU-based servers (depending on scale and model size).
- Low-latency architecture for user-facing queries.
- Potential offline batch processing for marketing or recommendation tasks.

AUTOSCALING AND LOAD MANAGEMENT

- Horizontal scaling for peak traffic.
- Caching partial answers or intermediate results.
- Consider multi-region deployment to reduce latency globally.

MONITORING AND LOGGING

- Observability into latencies, token usage, and concurrency.
- Logging for error analysis and user behaviour analysis.

PERFORMANCE OPTIMISATION

- Prompt engineering to reduce token usage or improve speed.
- Quantisation or distillation of models if cost or latency is too high.

A/B TESTING AND CONTINUOUS IMPROVEMENT

- Test different agent strategies and model versions.
- Gradual rollout of new model versions to subsets of traffic.

DELIVERABLES

Production-grade inferencing stack.

Monitoring dashboards, scaling policies, and performance metrics.

ORCHESTRATION BETWEEN MULTIPLE AGENTS

CENTRAL ORCHESTRATOR / CONTROLLER

- Receives requests and decides which agent(s) to call.
- Potentially uses a chain-of-thought or blackboard approach to pass partial results.
- Manages concurrency, agent selection logic, and fallback.

CONTEXT MANAGEMENT

- Shared memory or ephemeral context store (like a conversation memory).
- Ensuring each agent sees the relevant portion of the conversation or data only.

AGENT AUTONOMY / NON-DETERMINISM

- Agents can call each other: for example Llama calls DeepSeek if it needs extra knowledge.
- Must track calls to prevent infinite loops or repeated queries.

EXCEPTION HANDLING

- If an agent is at capacity or fails, fallback to a second-best agent.
- If an agent's output is ambiguous, controller might re-ask or clarify.

COST AND LATENCY GOVERNANCE

- Each agent call has a cost in tokens and compute.
- The orchestrator can set a maximum budget (time or cost) per request.

DELIVERABLES

Fully functional multi-agent orchestration platform.

SLA definitions for each agent (e.g., must respond within X ms).

SECTION 2: ENTERPRISE AI COST CONSIDERATIONS

A. DATA PREPARATION COSTS

Cost Driver	Key Variables	Notes
Pipeline Setup	Engineer hours, cloud infra (VMs, storage)	One-time setup + ongoing maintenance
ETL Tools and Licensing	License fees, number of seats	Consider open-source or in-house alternatives
Storage	Data volume (TB), storage type (hot/cold)	For raw, processed, and backup datasets
Labeling and Annotation	Data size, cost per label/hour	Cost scales with category, language, and brand coverage
Cleaning & QA	Engineering time, QA tooling	Often iterative—especially for large/unstructured datasets
Governance and Compliance	Tool licenses, compliance staff time	Include anonymisation tools and regulatory overhead (e.g., GDPR, CCPA)

B. LLM TRAINING (RAG / FINE-TUNING) COSTS

Cost Driver	Key Variables	Notes
Model Licensing	Commercial/open-source license terms	Check usage restrictions (e.g., LLaMA family)
Compute (GPU/TPU)	Instance type, hours, provider (cloud/on-prem)	Major cost driver for large-scale training
Training Data Prep	Engineering hours, dataset curation	Often iterative and reused across cycles
Fine-Tuning and RAG	Dev hours, vector DB licensing	Includes integration, test cycles, and data pipeline tuning
Hyperparameter Tuning	Additional GPU hours	20–40% of overall compute cost
Validation and Evaluation	Dataset cost, data scientist time	May include external evaluation for brand compliance
Model Storage	Checkpoint size, artifact management system	Consider MLOps platform overhead

C. AGENT BUILDING (AGENT CONSTRUCTION)

Cost Driver	Key Variables	Notes
Development Resources	FTE hours (AI, MLOps, architects)	Multi-agent designs add complexity
Agent Framework	License or usage fees	Open-source options may need custom extensions
API Integration	Number of APIs, dev time per integration	Includes internal (e.g., product DB) and external platforms
Security and Compliance	Dev hours, tooling	Prompt protection, audit logging, secure API layers
Testing & QA	Dev/QA hours, test infra	Complex flows require advanced testing scenarios
Documentation and Training	Internal guides, external contractor support	Important for scale and multilingual ops

D. INFERENCE COSTS

Cost Driver	Key Variables	Notes
Compute (GPU/CPU)	QPS, concurrency, cost/hour	Largest ongoing cost at scale
Autoscaling and Load Balancing	Cluster infra, geo-distribution	Kubernetes, edge balancing costs
Model Serving	Instance types, infra setup	May require specialised LLM hardware
Maintenance and Optimisation	Dev hours for updates, model tweaks	24/7 operational coverage
Monitoring and Observability	Logging volume, metrics stack	Tools like ELK, Prometheus, Datadog
A/B Testing and Iteration	Canary deployments, duplicate model runs	Essential but compute-intensive

E. MULTI-AGENT ORCHESTRATION

Cost Driver	Key Variables	Notes
Orchestration Logic	Engineering hours	Logic complexity grows with agent count and fallbacks
Context and Memory Sharing	DB/in-memory store (e.g., Redis), scaling requirements	Needed for shared state and coordination
Agent Communication	Network calls, microservice overhead	Each inter-agent call may count as an inference
Monitoring and Debugging	Granular logs, trace tools	Required to analyse multi-agent behavior
Access Control	Role-based restrictions, data protection	Sensitive info may be handled by specific agents

F. CROSS-CUTTING AND OVERHEAD COSTS

Cost Driver	Key Variables	Notes
Personnel	Salaries (Data Science, ML, DevOps, PMs)	Often the largest cost in long-term programs
Project Management	PMO/Scrum coordination overhead	For cross-functional execution
Legal and Compliance	GDPR, CCPA, PCI requirements	May require consultants or legal reviews
Risk and Contingency	10–20% budget buffer	For emerging tech and unknowns
Continuous Improvement	Annual/quarterly update cycles	New features, retraining, performance boosts

SECTION 3: KEY VARIABLES AND CONSIDERATIONS

01

USER CONCURRENCY AND QPS

With 50–100 million MAUs, you may see tens of thousands of queries per second during peak times, directly influencing your inference hardware and scaling strategies.

02

DEPTH OF ORCHESTRATION

Multiple agent calls per user request can exponentially increase inference costs, potentially doubling or tripling expenses.

03

MODEL SIZE AND COMPLEXITY

Larger models demand more GPU memory, leading to higher inference latency and costs. Smaller, distilled, or quantised models may offer cost-effective alternatives.

04

TRAINING FREQUENCY

More frequent retraining or fine-tuning (monthly or quarterly) increases GPU costs, requiring strategic planning to balance performance and budget.

05

DATA GROWTH

As your data grows—logs, new products, languages—embedding updates for RAG models become essential for maintaining performance.

06

LICENSING

Some foundational models come with commercial usage restrictions or fees, and certain vector databases or specialised frameworks may not be fully open-source.

07

RISK FACTOR

Multi-agent systems can introduce unforeseen overheads, such as debugging complexity, concurrency challenges, and potential cost overruns.

08

MLOPS MATURITY

A robust MLOps pipeline can significantly lower operational overhead, automating orchestration (CI/CD for models, automated retraining) to drive efficiencies.

SECTION 4: TCO ILLUSTRATION (BUILD/DIY)

Phase	Line Item	Quantity / Rate	Estimated Cost (USD)
A. Data Preparation	Data Pipeline Engineering (2 FTEs, 6 months)	2 FTE × 6 mo × $12k/mo = $144k	$144,000
	Data Storage (100 TB, $20/TB/mo, 12 mo)	100 TB × $20 × 12 = $24k	$24,000
	Data Labeling (1M items at $0.02/item)	1,000,000 × $0.02 = $20k	$20,000
	Data Governance Tool License (annual)	Flat $30k	$30,000
	Subtotal (Data Prep)		**$218,000**
B. LLM Training	Model Licensing (annual fee, if applicable)	$100,000	$100,000
	GPU Compute for Training (500 hrs @ $5/hr)	500 × $5 = $2,500	$2,500
	Hyperparameter Tuning and QA (200 hrs @ $5/hr)	200 × $5 = $1,000	$1,000
	HPC Environment Overheads (misc)	$26,000	$26,000
	Subtotal (LLM Training)		**$129,500**
C. AI Building	Agent Development (3 AI Eng, 6 months @ $12k/mo)	3 × 6 × $12k = $216,000	$216,000
	Integration & Security (2 Sec. Eng, 2 months @ $12k/mo)	2 × 2 × $12k = $48,000	$48,000
	Testing and Documentation	Flat $20,000	$20,000
	Subtotal (AI Building)		**$284,000**
D. Inferencing	Production GPU Nodes (10 nodes @ $3k/mo, 12 mo)	10 × $3k × 12 = $360,000	$360,000
	Logging & Monitoring (100M logs/mo @ $0.01 per 1k logs)	$1,000/mo × 12 = $12,000	$12,000
	HPC Cluster for Auto-Scaling (misc)	$50,000	$50,000
	Subtotal (Inferencing)		**$422,000**
E. Orchestration	Controller Dev and Maintenance (2 FTE, 12 months)	2 × 12 × $12k = $288,000	$288,000
	Context Storage (Redis, etc., $5k/mo)	$5k × 12 = $60,000	$60,000
	Intra-Agent Overhead (50% extra inference cost)	~$180,000 (synthetic example)	$180,000
	Subtotal (Orchestration)		**$528,000**
Total (All Phases)			**$1,581,500**

Phase	Year 1 (USD)	Year 2 (USD)	Year 3 (USD)	3 Years total (USD)	Notes / Assumptions
Data Preparation	218,000	80,000	60,000	358,000	Year 1: Major data engineering, labelling, governance setup. Years 2 and 3: Lower overhead, mostly incremental updates.
LLM Training	129,500	200,000	120,000	449,500	Year 1: Initial fine-tuning/RAG training. Year 2: Additional or larger re-training cycles as usage expands. Year 3: Some ongoing retraining.
AI Building	284,000	100,000	50,000	434,000	Year 1: Heavy initial dev for multi-agent system and integration. Year 2: New features, security patches. Year 3: Minimal new dev.
Inferencing	422,000	500,000	600,000	1,522,000	• Inference grows as monthly active users grow. • Additional GPU/CPU resources in Years 2 and 3.
Orchestration	528,000	300,000	200,000	1,028,000	Year 1: Building robust orchestration and multi-agent calling. Years 2 and 3: Maintenance, scaling, partial refactoring.
Total (All Phases)	**$1,581,500**	**$1,180,000**	**$1,030,000**	**$3,791,500**	

RATIONALE FOR EACH YEAR

YEAR 1:

- High costs in Data Preparation, AI Building, and Orchestration. You're doing the initial heavy lifting: setting up data pipelines, fine-tuning or training LLMs, building multi-agent orchestrators, etc.
- Inferencing cost starts moderately high because you're launching to potentially tens of millions of MAUs.

YEAR 2:

- Data Preparation cost decreases as most of the pipelines are built.
- LLM Training cost might increase if you do more frequent re-training as you discover new use cases or expand to additional languages or product lines.
- AI Building costs decline but still requires budget for new features, security, and improvements.
- Inferencing grows along with user traffic or additional agent calls.
- Orchestration cost declines from Year 1 because the core infrastructure is stable, but ongoing maintenance and improvements remain.

YEAR 3:

- Data Preparation is mostly incremental.
- LLM Training is moderate: you might retrain for new products or additional fine-tuning.
- AI Building is minimal if the core system is stable.
- Inferencing cost continue to climb due to more usage or more complex agent calls.
- Orchestration cost is lower but still present.

About us

Tata Communications leads the digital revolution by empowering enterprises globally to seize growth opportunities. With borderless connectivity and innovative solutions, we drive digital transformation for Fortune 500 companies. Our robust infrastructure powers unparalleled intelligence in cloud, IoT, and network services, connecting businesses to 80% of top cloud providers and four out of five mobile subscribers. Through strategic partnerships like Formula 1®, we deliver world-class experiences, showcasing our commitment to innovation and empowering global connectivity.

For more information, visit us at www.tatacommunications.com
or email us LeadershipConnect@tatacommunications.com