

Chattermill

# Building an AI Agent for Customer Insights

When to build an AI agent for customer feedback analytics – and when not to.

# Table of Contents

|                                                                                        |           |
|----------------------------------------------------------------------------------------|-----------|
| <b>Introduction</b>                                                                    | <b>01</b> |
| Where agents add real value                                                            | 02        |
| Why agents fail on raw customer feedback data                                          | 03        |
| <b>Here's what you need to build an effective agent for customer feedback analysis</b> | <b>05</b> |
| Data ingestion & unification                                                           | 06        |
| AI analysis pipeline                                                                   | 08        |
| Guardrails (governance & compliance)                                                   | 11        |
| Analytical features                                                                    | 12        |
| Agentic capabilities                                                                   | 14        |
| <b>Why is it harder than it looks?</b>                                                 | <b>16</b> |
| Stage 1: Agent on raw data                                                             | 16        |
| Stage 2: Structuring the data                                                          | 16        |
| Stage 3: Adding guardrails                                                             | 17        |
| Stage 4: Think you're done                                                             | 17        |
| Stage 4: Full stack                                                                    | 17        |
| <b>Why the Chattermill agentic CXI architecture is the smarter path forward</b>        | <b>20</b> |
| Why Chattermill is the platform designed for the agentic era                           | 20        |
| The five-layer stack behind Chattermill's agentic CXI architecture                     | 21        |
| The foundation: data ingestion & unification                                           | 22        |
| The analytical layer: Lyra AI insights engine                                          | 23        |
| CX context engine                                                                      | 25        |
| AI agents                                                                              | 27        |
| Why this architecture scales                                                           | 29        |

# Introduction

At Chattermill, we have engaged with many companies facing challenges in understanding customers at scale across the entire journey. What they are really trying to achieve is Experience-Led Growth – transforming customer experience from a business challenge into a key driver of sustained, profitable growth.

To get there, these companies need the right technology and expertise to analyze and act on customer feedback at scale – connecting signals across every channel and touchpoint, measuring the impact on business outcomes, and turning that intelligence into decisions that drive growth.

With the rise of AI agents, that ambition feels more achievable than ever. Companies are already connecting AI agents to their internal tools and data systems – and customer feedback is no exception. The promise is compelling: an agent that can query, synthesize, and surface insights from vast amounts of unstructured feedback at scale, on demand.

But building an agent for customer feedback analysis that would drive Experience-Led Growth is fundamentally more complex than connecting an AI agent to raw data. It requires rigour – in how data is structured, how models are trained and governed, how insights are validated, and how outputs are held to a standard that the entire organization can make decisions on.

**The agent is only as good as the foundation it sits on — the data, the structure, the models, and the domain expertise behind it.**

This paper offers decision-makers a clear framework for navigating that complexity – covering what it actually takes to build and maintain an agent-powered feedback analytics solution, and how to make an informed decision about whether building in-house is the right path for your organization.

# Where agents add real value

Before exploring what it takes to build a reliable solution, it is worth being precise about what AI agents actually bring to customer feedback analytics – and why the opportunity is real.

AI adoption is accelerating fast. Companies that were experimenting a year ago are now allocating serious budget and engineering resources to building with agents. The models are evolving week after week, continuously expanding what is possible – and data analysis is one of the use cases where teams are starting to see real value.

When it comes to customer feedback specifically, here is what agents do well.

- **Synthesizing across everything at once.** A question like "what changed this quarter and why does it matter for retention?" requires pulling together signals from multiple sources, time periods, and customer segments simultaneously. An agent can hold all of it at once and surface what actually matters, without requiring an analyst to manually piece it together.
- **Explaining in context.** Traditional analytics tells you that theme X went up 12%. An agent tells you the full story – when it started, what drove it, which customer segments are most affected, and what it likely means for the business.
- **Answering questions nobody prescribed.** Every dashboard was built around a fixed set of questions. The moment the business asks something different – "we're launching in a new market, what does our feedback say?" – an agent can answer it on the spot, without anyone having to build a new report first.
- **Connecting dots across use cases.** Patterns across teams and data sources that no single report surfaces. A product issue showing up in support tickets before it appears in reviews. A regional trend invisible at the aggregate level. An agent can draw connections across the entire landscape.
- **Making the architecture accessible.** Instead of navigating dashboards or waiting for reports, you ask a question and get an answer. The whole experience feels like a natural conversation - making it accessible to everyone in the organization.

These are the capabilities that make agents compelling for customer feedback analysis. But they come with a set of limitations that are just as important to understand – and that have significant implications for how a reliable solution needs to be built.

# Why agents fail on raw customer feedback data

The capabilities described above are real and valuable – but they depend entirely on what the agent is working with. When agents are applied directly to raw, unstructured customer feedback at enterprise scale, their probabilistic nature becomes a fundamental problem.

- **Inconsistent and unreliable outputs.** When an agent processes fifty thousand customer comments, it performs semantic classification, theme extraction, sentiment attribution, and cross-segment aggregation – and none of that is deterministic. Ask the same question tomorrow and it answers differently – not because the data changed, but because the agent improvised differently. For an enterprise making decisions on that data, that is not acceptable.
- **Context windows have limits.** Agents can only process a certain amount of text at once. With tens of thousands of customer comments, the agent has to sample, truncate, or process in batches – and each approach introduces inconsistency. When those outputs are the basis for business decisions, the consequences are real.
- **No stable taxonomy.** A reliable taxonomy is the foundation of any trend analysis – you can only track what something becomes over time if it was consistently defined from the start. When an agent constructs categories on the fly, that foundation doesn't exist. The same issue gets labelled differently across time periods, teams, or data sources.

On top of that, LLMs are constrained by token windows, meaning they process feedback in portions rather than across the full dataset – which makes consistent categorisation even harder.

Building a taxonomy that holds up at enterprise scale requires domain expertise and precision that an agent alone cannot provide.

- **Agents lack accuracy governance.** An agent produces confident answers regardless of whether those answers are correct. It has no mechanism to detect when its own outputs have degraded – because the underlying model changed, data quality dropped, or the categories it is using have drifted from what the business actually means.

The only way to catch it is if a human notices something looks off – which at enterprise scale is not a reliable quality control mechanism.

- **Deterministic analysis is impossible.** Some questions require the same answer every time they are asked – measuring whether a product change improved customer sentiment, or flagging an unusual spike in complaints.
  - If the way data is categorised or interpreted shifts slightly each time – which is what agents do by nature – it becomes impossible to tell whether a change in the data is real or simply the agent improvising differently. The result is chasing noise instead of signal.
- **Scaling across teams creates fragmentation.** A CX team, a product team, and a marketing team all querying the same feedback data will arrive at different answers – not because the data is different, but because the agent interpreted each question differently.
- **Security and compliance risks.** Role-based access, PII filtering, audit logs, and incident response are non-negotiable for enterprise buyers. None of these come with a general-purpose model. They have to be deliberately engineered into the product serving the use case, and that engineering is significant.

Taken together, these limitations define the problem. Building a reliable agent for customer feedback analysis is not about finding a better model – it is about building the right foundation underneath it. The next section covers exactly what that foundation should look like.

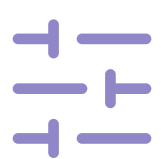
## CHAPTER 1.

# Here's what you need to build an effective agent for customer feedback analysis

Building an in-house agent for customer feedback analysis means more than connecting a model to your data. It means engineering a system that can unify feedback across every channel, process it consistently at scale, and produce outputs that the entire organization can trust and make business decisions on.

That's why for an agent to be genuinely useful in an enterprise context, its outputs need to be accurate, reliable, and trustworthy. Same input, same output. Testable, auditable, and consistent across teams.

Most organizations underestimate what that actually requires. Before committing to building in-house, there are five essential layers to consider.



## Data ingestion & unification

- Connections
- Normalization
- PII Removal
- Denoising
- Transcription
- Translations



## AI analysis pipeline

- Supervised taxonomy
- ABSA
- Precision insights
- Taxonomy management
- Robustness over time



## Guardrails

- Audit trail
- AI security
- Evaluations
- LLM governance
- Cost monitoring
- Incident response
- Roles & permissions



## Analytical features

- Impact analysis
- Observations
- Anomaly detection
- Dashboards
- Deterministic outputs



## Skills & MCP

- MCP
- Skills
- Proactive agent
- Evaluations
- Knowledge base
- Memory

# Data ingestion & unification

The capabilities described above are real and valuable – but they depend entirely on what the agent is working with. When agents are applied directly to raw, unstructured customer feedback at enterprise scale, their probabilistic nature becomes a fundamental problem.

To ensure high-quality analysis, ingested data must first be cleaned, structured, and transformed into a format that machine learning models can process. This involves eliminating noise, standardizing inconsistent data, and aligning different data sources to ensure consistency.

## Things to consider:

- **Connections:** Connect with all voice of customer tools - survey platforms, support ticketing systems, CRM - as well as data warehouses (Databricks, Snowflake, BigQuery, Redshift), unifying customer feedback from dozens of sources.
- **Normalization:** Standardizing data from various sources – survey responses, support chat transcripts, call transcriptions, app reviews, NPS comments – into a single, unified structure is essential for a reliable analysis. Without this, you risk inconsistencies that can skew your results and make it harder to draw conclusions.
- **Public data:** Consider how to connect with or scrape data from public channels – app store reviews (iOS, Android), review platforms (Trustpilot, G2, Capterra), and social media (X, Reddit, Facebook) – to capture the voice of the customer beyond your own feedback channels.
- **PII Removal:** Always consider how to automatically detect and redact personally identifiable information – names, emails, phone numbers, credit card details, addresses – before data enters the analysis pipeline, protecting customer privacy and meeting compliance requirements.
- **Denoising:** Make sure to filter out irrelevant content that would confuse the analysis model – automated messages ("Your ticket has been updated"), bot responses, greetings, signatures, disclaimers, duplicate entries, and low-signal filler text – so analysis focuses on genuine customer sentiment.

- **Transcription:** If voice data is part of your feedback mix – calls, voicemails, video feedback – it needs to be transcribed accurately with speaker diarisation to distinguish customer from agent, because without reliable diarisation, you cannot attribute sentiment or themes to the right party.
- **Translations & Language Support:** Remember to support multi-language feedback by translating customer input while preserving meaning and handling company-specific terminology.



"The quality of the insights you get from customer feedback analysis depends entirely on the quality of the data you put in. If the data is messy, inconsistent, full of duplicates and errors, the results will be simply inconsistent and unreliable."

**Dr Aji Ghose**

VP of Data & Research

**Chattermill**

# AI analysis pipeline

At the core of any feedback analytics system is the ability to turn unstructured text into structured, tagged data. That means building a taxonomy – a consistent set of categories and themes – and applying it reliably across every piece of feedback, at scale, over time.

## Things to consider:

- **Supervised Taxonomy:** Building a taxonomy means creating a structured system to classify feedback into relevant groups that reflect common topics or concerns that are robust and cover different areas of your business.  
The goal is to organize feedback into categories and themes that allow you to quickly identify trends and key insights. Most likely, you'll need a robust, curated taxonomy covering the key aspects of your customer experience – adapted to specific use cases (NPS, support, product feedback) and optimized for accuracy and actionability. Typically ~100 themes, each precisely defined and validated on your data.
- **Aspect-Based Sentiment Analysis:** Make sure your solution can identify both positive and negative sentiment within a single piece of feedback, supporting multiple themes per item for deeper context – so "Love the design but the battery is terrible" gets tagged as positive for Design and negative for Battery.
- **Precision Insights:** No static taxonomy can capture the full richness of your customer experience – you likely have thousands of distinct issues, requests, and patterns that your teams need at full granularity to action.

Make sure your solution can surface insights like "Delivery delayed by 3+ days with no tracking update" vs "Wrong item delivered" vs "Package left in rain" – each quantified with volume, sentiment, and trend data.

- **Taxonomy Management:** Enable customization of the taxonomy and themes, with a collaborative approach where your team can propose changes and your partner works closely with you to ensure analytical integrity. This starts to become important when your business needs evolve, new themes and concepts start to emerge, or there are new projects, products or business lines on the horizon.
- **Robustness Over Time:** Ensure your analysis results are not subject to swings from unexpected model changes – so you can rely on over-time patterns and trend comparisons without wondering whether a shift is real or an artifact of a model update.

- **Analysis Across Data Types:** Different types of feedback data – voice transcripts, live chat conversations, social media posts, survey responses, app reviews – require fundamentally different analysis approaches. A single prompt or model cannot handle all of them well. Make sure to build separate analysis pipelines (or at minimum separate prompts/models) per data type – one-size-fits-all will produce poor results



"While you can start by creating a basic taxonomy for your feedback analysis, the real challenge arises when you need more detailed, precise insights. This complexity grows as your system scales, turning what was once a manageable task into months of work and significant costs, all while maintaining and refining the taxonomy to uncover the most valuable insights."

**Dr Aji Ghose**

VP of Data & Research

**Chattermill**

# Guardrails (governance & compliance)

A tagging engine that works well on day one is not guaranteed to work well on day ninety. Models change, data quality shifts, edge cases accumulate, and without something actively catching it, the system degrades silently.

Guardrails are everything that prevents that from happening – the validation checks, confidence thresholds, drift detection, and human review mechanisms that keep the tagging system accurate over time. Without them, there is no reliable way to know when outputs have become inaccurate, which means decisions get made on bad data without anyone realising it.

## Things to consider:

- **Audit Trail:** Complete audit trail for regulatory and operational reasons – Who changed the taxonomy? When was this dashboard edited? Who accessed sensitive customer feedback? – with immutable logs for compliance.
- **AI Security:** Build security and safety guardrails for AI-powered analysis – preventing PII leakage in summaries, detecting adversarial inputs, understanding model confidence and limitations, and versioning models for reproducibility.
- **Evaluations:** Build a continuous evaluation and monitoring system for your models – classification accuracy per theme, sentiment detection precision/recall, summarization quality – with automated testing and alerting when accuracy drops below acceptable thresholds.
- **Cost Monitoring & FinOps:** Build cost tracking and optimization for your platform. Transparent cost breakdown – API calls, LLM inference, storage – will help you understand the cost drivers so you can optimize accordingly (trade off real-time vs batch, cheaper models vs better quality, storage retention policies).
- **LLM Governance:** Consider adding explicit control over which LLMs are used for which tasks – different models for different tradeoffs (cost, latency, quality), version management, ability to swap models, and transparent cost/performance tracking for each decision point.
- **Incident Response:** When something breaks – model accuracy drops, data pipeline stalls, data exposure – you'll need to have a playbook to detect it quickly, communicate impact, and remediate. This includes status pages, incident notifications, and RCA tooling.

- **Roles & Permissions:** Ensure you've provided fine-grained access control – executives see high-level dashboards, support leads manage their queue, analysts access raw data – with audit trails showing who accessed what.



"LLMs are completion engines. Given partial data, they fill the gap – not out of deception, but out of design. Whether the underlying data is clean, corrupted, or the taxonomy has drifted, the output looks the same: confident. That's why guardrails can't be optional – they're the only mechanism that encodes what an agent cannot self-generate: the instinct that honest incompleteness is always safer than plausible fabrication. Without them, degradation is invisible – and that is exactly when it does the most damage."

**Ugo Anomelechi**

VP Engineering, Chattermill

**Chattermill**

# Analytical features

An agent is only as useful as the analytical infrastructure it sits on. Behind every reliable answer sits a layer of structured analytical capabilities – the ability to track metrics over time, connect feedback to business outcomes, surface trends, detect anomalies, and deliver insights to the right people in the right format. These are the features that allow your organization to truly make decisions from customer data.

## Things to consider:

- **Metrics:** Make sure your solution is capable of tracking KPIs that matter – NPS, CSAT, CES, sentiment trends, theme distribution, and more – with clear definitions, automated calculation from your data, and historical trending.
- **Impact Analysis:** Think about how you will connect feedback to business outcomes – which themes correlate with retention, which drive revenue, which affect NPS, which predict churn. These correlations, combined with volume data, are what allow you to prioritize which customer signals are worth acting on.
- **Dashboards:** Your solution should allow you to build customizable dashboards with filters, time ranges, comparisons, and AI summaries so stakeholders can self-serve insights without waiting for reports.
- **Drilldown:** Your solution needs to support seamless navigation from aggregate views down to the raw voice of the customer – from a theme on a dashboard, to AI summaries, to representative snippets, to the full original feedback. A user should be able to click a metric and trace it all the way back to the individual comments supporting it, without losing context at any step.
- **Summarization:** Users should be able to get an instant answer – what are the top themes this week, what changed since last week, what are customers saying about this feature – in plain language that an executive can read in thirty seconds. These summaries need to be accurate and reliable enough that business decisions can be made on them.
- **Anomaly Detection & Alerting:** Leaders should not be finding out about critical signals in the weekly report. Build a system that detects and alerts on NPS drops, sentiment spikes, theme volume surges, and churn risk signals in real time – so the right people are notified immediately when something worth acting on emerges.

- **Report & Dashboard Sharing:** Stakeholders need access to insights without necessarily having access to the full platform. Executives need a monthly CX summary, support leadership a daily overview, product teams a weekly feature request digest. Make sure your solution supports scheduled sharing of dashboards and reports so the right information reaches the right people, on the right cadence.
- **Issue Tracking Integrations:** Make sure your issue tracking platforms are connected to your agent – Jira, Linear, GitHub – so feedback-driven issues move through your development pipeline with full context attached. Engineers should be able to trace any issue back to the customer feedback that drove it.



"Connecting an agent to a database and letting it run queries is the easy part. The hard part is getting it to answer complex questions reliably – to perform genuine analysis, surface the right patterns, and produce results you can trust consistently. An agent without the right infrastructure underneath will give you a different answer every time you ask the same question."

**Mikhail Dubov**

CEO & Co-Founder

**Chattermill**

# Agentic capabilities

Once the foundational layers are in place – your data infrastructure, tagging engine, guardrails, and analytical capabilities – you can build the agentic layer on top.

These additional capabilities will determine how your agent operates, how it connects to the rest of your organization, and how it accumulates the context needed to get smarter over time.

## Things to consider:

- **MCP:** Build an MCP-compatible layer on top of your insights data. A standardised protocol that lets every AI agent deployed across your organization – support bots, product copilots, internal assistants – connect directly to your customer insights platform as a data source, rather than each team building bespoke integrations.
- **Skills:** Build a library of reusable analytical skills – pre-built instructions designed to answer the questions your business asks repeatedly. What is driving NPS decline? What are the top feature requests this quarter? Where are we losing customers?

Define the methodology for each analysis type – NPS drivers, trend detection, feature request ranking, churn signals – so results are consistent, comprehensive, and statistically sound regardless of who runs them or when.

- **Guardrails:** Build a guardrail framework around your agentic pipelines. Safety and quality controls that constrain what agents can do, what data they can access, and how they present results – preventing hallucinated metrics, unauthorised data access, or misleading summaries from reaching stakeholders.
- **Proactive Agent:** Most teams think about agents reactively – you ask a question, you get an answer. But some of the most valuable insights are the ones nobody thought to ask about. Build a proactive agent that runs on a schedule, continuously mining your feedback data for emerging issues, trending topics, and anomalies, and surfacing them to the right team at the right time. The goal is a system that tells you what you need to know before you know to ask.
- **Evaluation Framework:** Build a comprehensive evaluation framework that covers the entire agentic setup – from individual skill accuracy to end-to-end pipeline correctness. As your data grows and your models evolve, this is what ensures your agent continues to deliver reliable, reproducible results. Without it, degradation is invisible.

- **Knowledge Base:** Make sure your agent has all the business context it needs to reason about your data correctly. Build a structured knowledge base it can query at inference time – your company's terminology, customer segments, or metric definitions. Without it, the agent does not know that "Premium" refers to your enterprise tier, that NPS is calculated excluding internal testers, or that Q4 data includes Black Friday spikes that should not be treated as anomalies.
- **Memory:** Make sure your agent retains context across interactions rather than starting from scratch every time. Build memory at three levels: at the individual level, the agent should remember what each analyst has been working on and what they care about. At the team level, it should retain shared context – current focus areas, known issues, at-risk accounts. At the company level, it should stay grounded in organization-wide priorities – OKRs, recent product launches, known data issues. Without this, every conversation starts cold and the agent cannot build on what it already knows



"What sits underneath the agent is what determines whether it is useful or unreliable. When all five layers are in place, the agent stops being a sophisticated interface and becomes something genuinely different: a system where every answer is grounded, consistent, and traceable back to the data that produced it – so that when someone asks a question, the answer is the same today as it will be next month. That is what it means to build a Customer Experience Intelligence platform."

**Mikhail Dubov**

CEO & Co-Founder

**Chattermill**

## CHAPTER 2.

# Why is it harder than it looks?

The architecture described in the previous section represents the destination. In practice, most teams discover it incrementally – adding layers as they encounter problems they didn't anticipate.

Here is what we typically see happen when organizations begin building an in-house agent for customer feedback analysis.

## Stage 1: Agent on raw data

Most teams start with the same instinct. Take the raw, unstructured feedback data, point an agent at it, and let it handle everything. The initial results can look promising – especially on a small sample. The agent surfaces themes, generates summaries, answers questions in natural language. It feels like the fastest route to value.

At enterprise scale, however, the approach breaks down. Agents are probabilistic by nature, and on large volumes of data that translates into inconsistency, hallucinations, and results that cannot be trusted or acted on.

The same question asked on different days returns different answers – not because the data changed, but because the agent interpreted it differently. Categories shift without warning, and the agent expresses confidence regardless of whether its outputs are accurate.

Teams quickly discover that a capable model pointed at raw data is not a reliable foundation for business decisions.

## Stage 2: Structuring the data

The natural response is to go back and structure the data. Teams build a tagging engine, define a taxonomy, and create a real system of record – bringing consistency to how feedback is categorised and labelled.

That is genuine progress, and it solves a real problem. The data becomes more reliable and the agent has something more stable to work with.

But even with a well-structured taxonomy and clean, tagged data underneath it, an agent may still produce different answers to the same question on different days. The structure helps – but it does not make the agent deterministic.

## **Stage 3: Adding guardrails**

Teams that reach this point add guardrails on top of their tagging engine – validation checks, confidence thresholds, drift detection mechanisms – that minimize hallucinations and preserve accuracy.

The system starts to feel robust. Outputs are more consistent, edge cases are handled, and the agent is no longer producing results that are obviously wrong. This is meaningful progress, and it represents a level of maturity that many teams never reach. It is also where almost every team stops, believing the hard work is done.

## **Stage 4: Think you're done**

The tagging engine is in place, the guardrails are holding, and the agent is performing well enough in demos and spot checks.

What is not yet visible are the two layers that sit between the tagging engine and the agent – analytical features and skills. These are the layers most teams don't know they need until they realize their agent still isn't giving them answers they can trust consistently enough to base business decisions on.

Analytical capabilities - impact analysis, anomaly detection, interactive reports and dashboards - are the features that allow your organization to truly make decisions from customer data.

## **Stage 5: Full stack**

The teams that get all the way there arrive at a fundamentally different kind of system. The agent no longer sits on top of raw data – it sits on top of a complete architecture: data ingestion, tagging engine, guardrails, analytical features, and skills.

The five layers process all the feedback, structure it, validate it, and make it analytically meaningful. The agent then reasons about the structured intelligence those layers produce, dipping into raw feedback only when it needs specific examples to support an answer.

The result is a system that returns the same answer, every time, at scale. With the full architecture beneath it, the agent becomes a decision system that the entire organization can rely on.

## CHAPTER 3.

# Why the Chattermill Agentic CXI Architecture is the smarter path forward

Each of the five layers described in this paper represents months of development, testing, and refinement – and the domain expertise required to get them right is not something that can be engineered from first principles.

The decision to build in-house is a decision to start that journey from scratch – to spend the next months solving problems that have already been solved, before you can even begin to address the questions that are specific to your business.

The alternative is to build on a foundation that has already been developed, tested, and refined over a decade, with the benefit of lessons learned across hundreds of enterprise deployments.

That means your engineering team spends its time on the problems only you can solve rather than reconstructing infrastructure that already exists.

## Why Chattermill is the platform designed for the agentic era

Chattermill didn't just add AI agents on top of its existing platform. We built the agentic Customer Experience Intelligence architecture to support them – from the ground up.

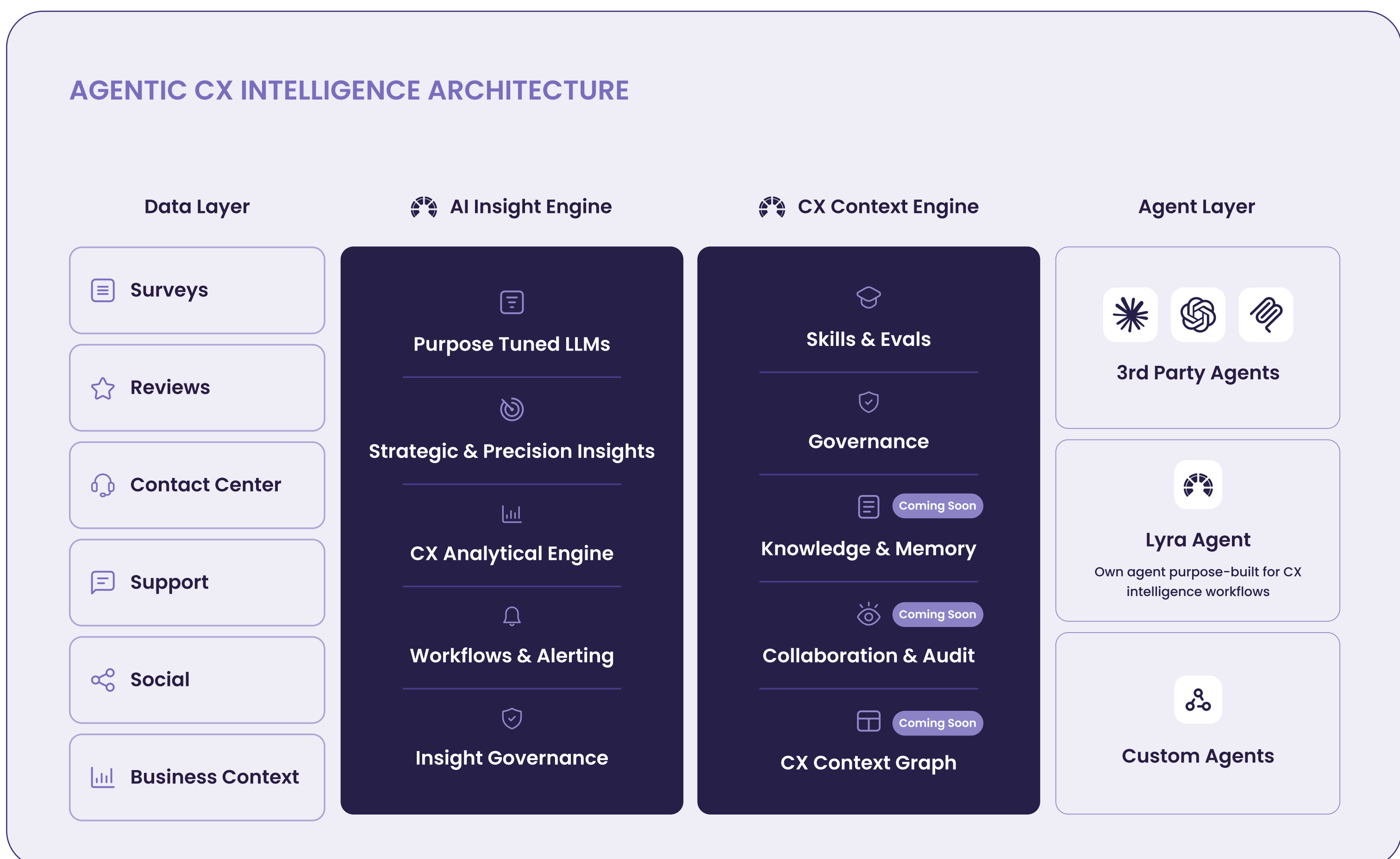
The result is a multi-layer system where data ingestion, the insights engine, analytical features, AI agents, agentic intelligence, and domain expertise are connected by design, each layer feeding into the next to produce outputs that are deterministic, auditable, and reliable at enterprise scale.

Understanding the Chattermill agentic CXI architecture matters if you are evaluating enterprise AI feedback analytics vendors, if you are building agents on top of the Chattermill platform, or trying to understand what agentic feedback analysis actually looks like in practice.

The following section walks through each layer – what it does, how it connects to the others, and why the architecture as a whole is what makes agents trustworthy enough to base business decisions on.

## The four-layer stack behind Chattermill's agentic CXI architecture

The Chattermill CXI architecture is built in layers, where each layer processes, structures, and validates what the next one depends on. An agent sitting on top of this stack is not improvising – it is reasoning on top of intelligence that has already been structured, validated, and made analytically meaningful at every level beneath it.



Here is how the stack is structured:

- 1. Data ingestion and unification** – Chattermill unifies customer feedback from over sixty data sources into a consistent schema, with transcription, translation, denoising, and PII redaction handled in pipeline – so every piece of feedback is clean, structured, and ready for analysis before it enters the insights engine.
- 2. Lyra AI insights engine** – Chattermill's proprietary NLP platform leverages a range of technologies, including aspect-based sentiment analysis, bottom-up clustering, and purpose-tuned LLMs – all operating on structured data, never on raw feedback. Combined with analytical features including metrics tracking, impact analysis, or anomaly detection, Lyra AI is what turns clean data into intelligence that agents can act on.
- 3. CX context engine** – Skills, knowledge base, memory, evaluation framework, guardrails – the capabilities that determine how intelligently and safely agents operate on top of everything beneath them.
- 4. AI agents** – Lyra Agent, Chattermill's own agent purpose-built for CX intelligence, continuously monitoring feedback data and surfacing insights proactively. Or any MCP-compatible agent – such as Claude, ChatGPT – connected directly to the Chattermill CXI architecture via the native MCP server, querying the same reliable, structured intelligence underneath.

Each layer deserves a closer look.

## The foundation: data ingestion & unification

An agent is only as good as the data it can access. Chattermill's answer to this is the data layer ingestion & unification – a unified, clean, and consistently structured foundation that consolidates customer feedback from every channel and touchpoint before it enters the analysis pipeline.

**Chattermill connects to over sixty data sources out of the box** – survey platforms, support tools, CRMs, and direct warehouse integrations into Databricks, Snowflake, BigQuery, and Redshift. All data is normalised into a consistent structure with source-specific metadata preserved.

**Transcription, translation, and denoising** are handled in pipeline – automatically filtering out automated system messages, bot responses, greetings, signatures, disclaimers, duplicates, and low-signal content.

Before data enters the analysis pipeline, **Chattermill automatically detects and redacts personally identifiable information** – names, emails, phone numbers, credit card numbers, and addresses – ensuring customer privacy is protected and compliance requirements are met from the point of ingestion.

### Why this matters for agents

When an agent queries customer feedback through the Chattermill's CXI architecture, it is not working with raw, unprocessed data.

Every piece of feedback has already been normalised into a consistent structure, translated, cleaned of noise, and stripped of PII. The result is a consolidated, clean, and consistently structured dataset – prepared, validated, and unified across every channel and touchpoint.

That is the foundation the agent builds on, and it is what ensures the highest quality of analysis at every subsequent layer.

## The analytical layer: Lyra AI Insights Engine

The intelligence layer in the Chattermill agentic CXI architecture sits between the data layer and the agentic capabilities. Chattermill built this around **Lyra AI** – proprietary technology developed over a decade of experience, combining best-of-breed models purpose-built for customer feedback analysis.

Together with the analytical features built on top, this layer gives agents and human analysts the infrastructure to track metrics, surface trends, and connect feedback to business outcomes.

### What is Lyra AI

Lyra AI is Chattermill's proprietary technology – the insights engine that drives how customer feedback is analyzed, structured, and transformed into intelligence that agents and analysts can act on.

Unlike general-purpose models that apply a single approach uniformly across all data, Lyra AI is purpose-built for customer feedback analysis, combining the best available technologies for each specific analytical task.

Lyra AI is built on three foundational pillars:

- 1. Aspect-Based Sentiment Analysis** – unlike traditional sentiment analysis which assigns a single score to an entire piece of feedback, ABSA analyzes theme and sentiment together. Each verbatim is decomposed into the specific attributes a customer is reacting to – with sentiment determined independently for each theme.
- 2. Observations Pipeline** – a granular analytical layer that goes beyond themes to surface the most precise and specific issues within customer feedback. The result is the ability to discover, extract, and quantify specific customer issues with a level of precision that general-purpose LLMs alone cannot achieve at scale.
- 3. Best-in-class LLMs and SLMs** – delivering the last mile of insight by turning structured data into clear, digestible summaries. The LLM and SLM layer in Lyra AI operates only on the structured data produced by ABSA and Observations, never on raw feedback – ensuring outputs are grounded in accurate, pre-processed data rather than improvised from unstructured text.

The result is two levels of intelligence: strategic insights that track broader trends over time, and precision insights that surface exactly what needs immediate attention – giving agents and analysts the foundation to answer any CX question with confidence.

Those questions can be answered through a range of analytical capabilities available within the platform – including reports and dashboards, metrics tracking, impact analysis, or anomaly detection. Together, these capabilities transform structured customer intelligence into decisions that the entire organization can act on.

## Why this matters for agents

An agent is only as good as the intelligence it reasons on top of. Because Lyra AI has already classified, analyzed, and validated every piece of feedback before an agent ever queries it, the agent is not guessing – it is retrieving structured, accurate intelligence.

The result is deterministic: the same question asked on Monday returns the same answer on Tuesday. The answers are consistent, traceable, and grounded in evidence – which is what makes them reliable enough to base business decisions on.

# The Science Behind Customer Experience Intelligence

Read the full technical breakdown of Lyra AI – how Chattermill approaches feedback analysis, what makes the technology different, and why it matters.

[Read more](#)

## CX context engine

The foundational layers – data ingestion and the insights engine – produce reliable, structured intelligence. The agentic layer is what determines how that intelligence is accessed, distributed, and acted upon across the organization.

This is what completes the Chattermill CXI architecture – turning a reliable analytical foundation into a system that any team, and any agent, can interact with directly.

### Skills

Chattermill skills library teaches agents how to use the insights engine properly – how to triangulate evidence, how to dig deeper rather than stop at the first plausible answer, and how to perform the data quality checks agents naturally skip because of their built-in bias toward getting to an answer quickly.

Our expertise is drawn from hundreds of real insights we have delivered to customers over years of analyst work. When we teach an agent to do this job, we are teaching it to approximate the best analysts in the world.

## Evaluations

Chattermill maintains continuous evaluation across every agentic component – skill-level evals that test each analytical recipe against known datasets, pipeline evals that verify end-to-end correctness from data ingestion through to final output, and regression evals that catch when model updates or data changes break existing functionality. All results are tracked over time with automated alerting on degradation.

## Knowledge & memory

COMING SOON

Chattermill will maintain a managed knowledge layer that agents consult automatically – your product taxonomy, metric definitions, customer segment rules, business-specific terminology, and known data quirks, all codified and versioned.

Built on top of that will be a three-layer memory system, so every interaction builds on what the system already knows.

**Employee memory** will track each user's analysis history and past findings.

**Team memory** will maintain shared context around current focus areas and known issues.

**Company memory** will capture organization-wide context – OKRs, strategic priorities, recent product launches – ensuring every interaction is grounded in what matters to the business right now.

## Governance

Chattermill enforces multi-layer controls across the agentic stack – so every output can be trusted.

Output validation checks that cited numbers match the underlying data. Access controls ensure agents only query data within their permission scope. Content filters prevent PII leakage in summaries. And confidence scoring flags low-certainty outputs for human review, rather than auto-publishing them.

The result: agents that reason over customer feedback reliably, safely, and at the standard enterprise companies require.

## Collaboration & audit COMING SOON

Chattermill will maintain a complete immutable audit trail of all system activity – data access, configuration changes, deletions, user actions. Logs will be retained for compliance and exportable for regulatory audits.

## CX context graph COMING SOON

Our vision is a structured, queryable record of every insight path an organization produces – modelled as a connected chain: Question → Insight → Action → Impact.

When a regional manager asks "why are renewals dropping in Benelux," the agent wouldn't start from zero. It would already know whether a similar pattern emerged elsewhere, what the root cause was, and whether the intervention worked – producing a sharper answer every time, and improving with every decision made on top of it.

## AI agents

The Chattermill agentic CXI architecture is designed for both teams and agents.

Teams can use Lyra Agent – Chattermill's own agent purpose-built for customer experience intelligence – directly within the platform, or connect any MCP-compatible agent, such as Claude or ChatGPT, via the native MCP server. Either way, the agent is querying the same reliable, structured intelligence underneath.

### 3rd party agents

The Chattermill MCP server makes that intelligence available wherever your teams already work. Product, CX, marketing, and support teams can query customer feedback directly within the AI tools they use every day – Claude, ChatGPT, Cursor, Gemini, Notion Agent, and more – without leaving their current environment.

Once connected, any query to the MCP server returns relevant insights drawn from the full Chattermill CXI architecture underneath, with all the reliability and structure that entails.

## Lyra Agent

In 2026, Chattermill will introduce **Lyra Agent** that monitors customer feedback data around the clock and continuously surfaces what matters without anyone having to ask.

Lyra Agent will run on your customer feedback data, scanning for emerging themes, sentiment shifts, volume anomalies, and new patterns, and routing findings to the relevant team.

Users will also be able to interact with Lyra directly through a conversational interface that lets anyone ask any business question and get back relevant insights, backed by customer quotes and visualisations. The experience is designed to feel like working with a human analyst: ask a question, get an answer, follow up, go deeper.

## Custom agents

Teams building their own internal agents can connect them to the Chattermill CXI architecture through the same MCP server. Whether it is a support automation tool, a product planning agent, or an internal assistant, any MCP-compatible agent can query Chattermill's structured customer intelligence directly – with the full reliability of the stack underneath.

### Why this matters

Rather than facing a binary choice between building everything from scratch or adopting a fixed solution, Chattermill offers a third path.

Teams that have already standardised on a particular agent – Claude, ChatGPT, or an internal tool – can connect it directly to the Chattermill CXI architecture via MCP, with the full stack available underneath: clean, unified data, deterministic insights from Lyra AI, guardrails, analytical features, and domain expertise.

Teams that prefer a ready-to-use solution can deploy Lyra Agent directly within the platform. Either way, the foundational infrastructure is already in place – and the only decision is how to interact with it.

# Why this architecture scales

The Chattermill agentic CXI architecture is the foundation on which enterprises can build and deploy agents for customer feedback analysis – reliably, at scale, and with results that hold up under scrutiny.

The data is clean, consolidated, and consistently structured before it reaches the insights engine.

The insight engine produces deterministic, validated, and accurate insights that agents can depend on.

The agentic layer determines how that intelligence is accessed, distributed, and acted upon across the organization.

And the governance layer ensures the entire system operates within enterprise security, compliance, and data protection requirements at every step.

The result is a multi-layer system where every component is connected by design, producing outputs that are deterministic, auditable, and reliable at enterprise scale – giving agents and the humans working alongside them a foundation they can actually build on.

Underpinning the entire architecture is a layer of domain expertise accumulated over a decade of working with the world's most customer-centric enterprises, and operationalised through a strategic partnership that helps organizations connect customer intelligence to business outcomes.

# Exploring how AI agents can work for your customer feedback analytics?

If you are evaluating how to bring AI agents to your customer feedback analytics, speak with one of our experts. We will walk you through the architecture, show you what it looks like in practice, and help you understand what the right approach looks like for your organization.

[Book a personalized demo](#)

## About Chattermill

Chattermill, the Customer Experience Intelligence company, enables leading brands to unlock their customers' reality. Global enterprises like Amazon, H&M, and Uber rely on Chattermill to operationalize CX data. Chattermill is the best-in-class AI-powered platform for CX, Product, Marketing, and Support teams to action insights from their customer data at scale, ranked no.1 in multiple G2 categories.

### Follow us

 [LinkedIn](#)

 [YouTube](#)

 [How to Love a Customer](#)

[Blog](#)

[Chattermill.com](#)