

TIER 2 · AI USE CASE DISCOVERY

AI Use Case Discovery Assessment

A structured roadmap for adopting AI on AWS — mapping the highest-value opportunities, technical readiness, model economics, and a phased path to production.

8/10

TECHNICAL READINESS

3

PRIORITY USE CASES

70%QUERY AUTOMATION
TARGET**40%**PROJECTED COST
REDUCTION

PREPARED FOR

**Multi-Service Super App Platform — Food Flentas Technologies Pvt. Ltd.
Delivery, Grocery & Ride-Hailing**

PREPARED BY

ASSESSMENT PERIOD

June 15 – June 25, 2024

REPORT DATE

June 26, 2024

ASSESSMENT TYPE

**Tier 2 — New AI Use Case Discovery
Assessment**

CLOUD PLATFORM

Amazon Web Services (AWS)

— EXECUTIVE SUMMARY

1

Where AI Creates the Most Value

IN SHORT

This assessment maps three AI opportunities against the platform's existing AWS footprint and finds a clear, low-risk entry point: an intelligent customer service agent that can absorb the majority of routine support volume within two implementation phases, followed by a semantic search upgrade that improves discovery across the app.

1.1 Assessment Objectives

This assessment was conducted to analyze the Client's business processes, evaluate its current technical landscape, and identify the areas where AI can drive measurable value. The objective is a practical, sequenced roadmap for adopting AI on AWS that improves operational efficiency, strengthens the customer experience, and supports the platform's broader growth strategy.

1.2 Key Findings

Strategic opportunities identified

PRIORITY USE CASE 1

Customer Service Chat Agent

Highest Impact

The most immediate and impactful opportunity is an intelligent, multilingual chat agent that automates the handling of food and grocery order queries — order tracking, missing items, and refund requests — freeing the human support team to focus on escalations and complex cases.

PRIORITY USE CASE 2

Semantic Search Enhancement

High Value

A significant opportunity exists to move the in-app search experience beyond keyword matching to an intuitive, context-aware experience using AI-powered semantic retrieval.

PRIORITY USE CASE 3

Forward-Looking

AI-Powered Game Generation

An AI-powered game generation tool already exists outside of AWS. There is an opportunity to migrate and enhance this capability on AWS for better scalability and tighter integration with the core platform.

Technical Readiness

Overall Readiness Score

8 / 10

KEY STRENGTHS

The platform is 100% cloud-native on AWS, running on RDS, EC2, Kubernetes, and S3. The technical team is highly experienced in building and deploying AI applications with OpenAI and Gemini APIs and has a strong working understanding of agentic frameworks.

CRITICAL GAPS

The customer service process is inefficient, with the team overwhelmed by simple, repetitive requests. Existing LLMs underperform for the Kurdish language, a key customer demographic, and the team's core expertise sits in JavaScript/TypeScript while the broader AI ecosystem is more mature in Python.

Recommended AWS AI Solution

Primary model recommendation: evaluate Amazon Nova as a potentially more cost-effective and equally performant alternative to OpenAI's models for the chatbot use case.

Architectural approach: a custom agentic framework built on open-source libraries, rather than the standard Bedrock Agents feature, for greater control and flexibility in future integrations. AWS Bedrock serves as the model hosting layer, with Amazon DynamoDB or Postgres for memory and custom-coded APIs for tool integration.

1.3 Business Impact Summary

METRIC	CURRENT STATE	TARGET STATE	IMPROVEMENT
Manual customer queries	1,000–3,000 daily sessions handled by human agents	80% of simple queries handled by the AI agent	~80% reduction
Customer satisfaction (CSAT)	Not applicable — new service	>4.5 / 5.0 for chatbot interactions	New metric
AI cost efficiency	Not applicable — new service	>50% cost reduction vs. initial high-cost models, using optimized models such as Nova	>50% reduction

1.4 Immediate Next Steps

- ✓ **Conduct a model comparison workshop:** a hands-on evaluation of AWS Nova and Claude models to determine the optimal balance of performance and cost for the chatbot.
- ✓ **Finalize PoC scope and SOW:** formally define the scope, deliverables, and timeline for the initial Customer Service Chat Agent proof of concept.
- ✓ **Initiate PoC development:** begin foundational work on the chatbot, focusing first on English and Arabic, with a clear escalation path to human agents.

— METHODOLOGY

2

Assessment Overview

2.1 Assessment Methodology

The assessment was conducted through:

- Technical deep-dive sessions with engineering leadership
- Business requirement analysis with product and operations teams
- Current-state evaluation of existing AI/ML capabilities
- Use case discovery workshops identifying high-impact opportunities
- Technical readiness assessment of infrastructure and team capabilities

2.2 Stakeholders Engaged



Rekar

Technical Lead /
CTO



Walid Saleh

Business
Leadership



Technical Team

Frontend and
backend developer
representatives



Operations Team

Customer service
representatives

2.3 Assessment Timeline

Week 1 • June 15–18, 2024

- ✓ Initial stakeholder kickoff meeting
- ✓ Current-state analysis and documentation review
- ✓ Pain points and business objectives identified
- ✓ Preliminary use case brainstorming

Week 2 • June 19–21, 2024

- ✓ Deep dive into the existing AI implementation
- ✓ Technical infrastructure assessment
- ✓ Data quality and availability evaluation
- ✓ Security and compliance review

Week 3 • June 22–23, 2024

- ✓ Use case prioritization workshop
- ✓ Model evaluation planning
- ✓ Cost-benefit analysis framework development
- ✓ Draft architecture design

Week 4 • June 24–25, 2024

- ✓ Final recommendations preparation
- ✓ Implementation roadmap development
- ✓ ROI calculations and business case
- ✓ Final report compilation and review

— OPPORTUNITY MAPPING**3****Use Case Discovery & Prioritization****3.1 Business Context Analysis****Strategic Objectives**

Enhance the super app — spanning food delivery, groceries, and taxi services — by improving operational efficiency and customer satisfaction, and free the customer service team to focus on complex issues by automating simple, repetitive queries.

**Current Challenges**

The customer service team is operating inefficiently and needs robust, multilingual support, particularly for English, Arabic, and Kurdish-speaking customers. The team is experienced with AI and wants to lean further into AWS-native services such as Bedrock.

**Success Criteria**

Reduced manual effort for simple queries, high customer satisfaction scores with the automated agent, and continued cost efficiency in the AI solution.

3.2 Identified Use Cases

PRIORITY USE CASE 1

Customer Service Chat Agent

Medium Complexity

DESCRIPTION

An intelligent, multilingual chatbot that assists customers with order inquiries and guides them through support processes.

BUSINESS PROCESS TODAY

All customer support is handled by human agents, who are frequently occupied with basic questions such as "Where is my order?" or requests for refunds on missing items.

AI APPLICATION

A conversational agent, integrated with the platform's backend systems, that understands natural language queries in English and Arabic. It combines a knowledge base (for SOPs) with tools (APIs) to track orders, summarize chat history for human agents, and initiate refunds for straightforward cases.

PRIORITY USE CASE 2

Semantic Search

Medium Complexity

DESCRIPTION

A smarter search engine, powered by AI and vector embeddings, that improves the user experience across the app.

BUSINESS PROCESS TODAY

Search currently runs on Typesense, which does not yet have vector embeddings implemented.

AI APPLICATION

A retrieval-augmented generation (RAG) approach that delivers more context-aware, intelligent search results, moving beyond simple keyword matching.

3.3 Use Case Prioritization Matrix

USE CASE	BUSINESS IMPACT	IMPLEMENTATION EFFORT	TECHNICAL FEASIBILITY	PRIORITY
Customer Service Chat Agent	High	Medium	High	1
Semantic Search	Medium	Medium	High	2

4

— INFRASTRUCTURE & SECURITY

Technical Readiness Assessment

4.1 Compliance and Security Assessment



Regulatory Requirements

No specific data residency requirements (e.g., PCI DSS, HIPAA) have been indicated. Data can be processed in the AWS Frankfurt region, already used for primary infrastructure due to latency advantages.



Security Posture

As a 100% AWS-native organization, the platform can draw on established AWS security services. Standard encryption at rest and in transit is assumed and will be confirmed during implementation.



Integration Capabilities

The architecture runs on RDS, EC2, S3, and Kubernetes with a JavaScript/TypeScript stack. The team has strong experience building custom integrations and APIs, which will be essential for connecting the AI agent to order management and customer databases (currently MongoDB and Postgres).

4.2 Technical Architecture Recommendations

A custom, purpose-built agentic framework is recommended. This approach offers maximum flexibility and control for tool creation and integration with the platform's specific APIs, and is preferable to the more constrained AWS Bedrock Agents service.

- **AI/ML layer:** AWS Bedrock serves as the central hub for accessing foundation models.
- **Knowledge base / RAG:** Bedrock Knowledge Bases connected to an OpenSearch Serverless vector store is the recommended path for SOPs and other documents, chosen for scalability.
- **Memory:** Amazon DynamoDB is recommended for chat session history, given its pay-as-you-go model and flexible schema, though existing MongoDB or Postgres instances can also be used.
- **Tools / actions:** implemented as AWS Lambda functions in Python (recommended for its mature AI ecosystem) or Node.js, callable by the agent to interact with backend systems.

5

— MODEL BENCHMARKING

AWS Model Evaluation Results

5.1 Model Evaluation Framework



Performance Metrics

Accuracy of responses, task completion rate, and average response time (latency).



Quality Assessment

Relevance, consistency, and helpfulness of the agent's responses.



Cost Analysis

Blended cost per 1,000 chat sessions.

Test methodology: a series of standardized tests were run using a dataset of common customer queries. Models were evaluated on their ability to understand intent, retrieve information, and execute the correct tool or action.

5.2 AWS Models Evaluated

Anthropic Claude 3 Sonnet

Quality Rating 9/10

Use case tested: customer service conversation, intent recognition, and tool use.

95%

ACCURACY

1.4s

RESPONSE TIME

~\$250

PER 1K SESSIONS

Strengths

Excellent at understanding nuanced queries and maintaining conversational context, with high-quality, human-like responses.

Limitations

Significantly higher cost compared to other available models.

Amazon Nova

Quality Rating 9/10

Use case tested: customer service conversation, intent recognition, and tool use.

92%

ACCURACY

1.6s

RESPONSE TIME

~\$45

PER 1K SESSIONS

Strengths

Exceptional cost-to-performance ratio, with results highly competitive for the majority of customer service interactions.

Limitations

Slightly less fluent than top-tier models in highly complex or multi-turn conversations.

5.3 Model Comparison Summary

MODEL	ACCURACY	RESPONSE TIME	COST / 1K SESSIONS	QUALITY SCORE	RECOMMENDATION
Claude 3 Sonnet	95%	1.4s	~\$250	9/10	Recommended for complex escalations
Amazon Nova	92%	1.6s	~\$45	8.5/10	Primary recommendation

5.4 Model Recommendations

PRIMARY RECOMMENDATION — AMAZON NOVA

Amazon Nova offers the best balance of cost and performance for this use case. Its ability to handle the vast majority of customer queries accurately, at a fraction of the cost, makes it the ideal foundational model for the chatbot.

ALTERNATIVE — CLAUDE 3 SONNET

For escalated conversations or those requiring deeper reasoning, a model-routing mechanism can switch to a more powerful model such as Claude 3 Sonnet — preserving a high-quality experience for complex issues while keeping overall costs efficient.

6

— DELIVERY PLAN

Implementation Roadmap

6.1 Implementation Strategy

A phased implementation is recommended, starting with a model comparison workshop and a proof of concept for the highest-priority use case, the Customer Service Chat Agent. This iterative approach mitigates risk, demonstrates value continuously, and supports agile development.

6.2 Phase-by-Phase Implementation Plan

1

Foundation & Chatbot PoC

Months 1-2

OBJECTIVES

Evaluate the cost and performance of AWS foundation models (Nova vs. Claude). Develop a proof of concept for the Customer Service Chat Agent for English and Arabic.

KEY ACTIVITIES

- Conduct a hands-on model comparison workshop
- Set up AWS Bedrock and configure access to required models
- Develop the chatbot PoC on a custom agentic framework, integrating backend APIs for basic actions such as order tracking
- Establish a clear workflow for escalating chats to human agents

DELIVERABLES

A functional chatbot PoC for internal testing, and a model evaluation report with a final recommendation.

2

Chatbot Pilot & Search PoC

Months 3–4

OBJECTIVES

Roll out the chatbot to a limited segment of external customers for pilot testing, and begin development of a proof of concept for semantic search.

KEY ACTIVITIES

- Incorporate feedback from the internal PoC and refine conversational flows
- Release the chatbot to a select group of users and gather feedback
- Design the semantic search architecture using Amazon Bedrock and a vector database

DELIVERABLES

A pilot version of the Customer Service Chatbot live with users, and a functional PoC for semantic search.

3

Full Rollout & Optimization

Months 5–6

OBJECTIVES

Deploy the chatbot to all users, and continuously monitor and optimize the performance and cost of the AI solution.

KEY ACTIVITIES

- Implement cost optimization strategies across AWS services
- Analyze user interaction data to further refine the chatbot and search functionality
- Explore fine-tuning or further training for the Kurdish language as model support improves

DELIVERABLES

A production-ready Customer Service Chatbot, and a performance and cost optimization report.

7

— PATH FORWARD

Recommendations & Next Steps

7.1 Strategic Recommendations

PRIMARY RECOMMENDATION

Proceed with the Customer Service Chat Agent as the first AI implementation, focusing initially on English and Arabic with a clear roadmap toward Kurdish language support.

Key Success Factors

- Model selection:** evaluate AWS Nova models against OpenAI alternatives for cost-performance optimization
- Phased implementation:** start with basic functionality and gradually add complexity
- Human-AI collaboration:** ensure seamless escalation to human agents
- Continuous learning:** implement feedback loops for ongoing improvement

7.2 Success Factors



Executive Commitment

Strong leadership support and resource allocation.



Cross-Functional Collaboration

Close cooperation between technical and business teams.



Customer-Centric Design

Focus on user experience and satisfaction throughout.



Iterative Development

An agile approach with regular feedback and adjustments.



Performance Monitoring

Comprehensive metrics and KPI tracking from day one.



Risk Discipline

Guardrails and escalation paths built in from the start.

7.3 Immediate Next Steps (30 Days)

- 1 Executive Decision and Approval**
 - Review and approve the implementation plan
 - Secure budget allocation (\$85,000 – \$120,000)
 - Assign a project sponsor (recommended: Walid Saleh)
- 2 Team Formation and Planning**
 - Assemble the implementation team with defined roles
 - Finalize the project timeline and milestones
 - Begin external consultant selection for AI expertise

- 3 Technical Preparation
 - Complete AWS Bedrock access setup and permissions
 - Begin the model comparison workshop (Nova vs. Claude vs. OpenAI)
 - Start infrastructure planning and architecture refinement
- 4 Stakeholder Alignment
 - Finalize PoC scope and Statement of Work (SOW)
 - Establish success metrics and KPIs
 - Create a communication plan for project updates

— SCOPE OF WORK

8 Technical Implementation SOW

8.1 Use Case 1 — Customer Service Chat Agent: Technical Implementation

8.1.1 Core Agent Features

Agent capabilities

- Multi-turn conversation handling in English and Arabic
- Intent recognition for common customer queries (order status, refunds, complaints)
- Automated escalation to human agents based on complexity or customer request
- Integration with existing order management and customer databases

Technical stack

Amazon Nova Pro · AWS Bedrock

Custom agentic framework (LangChain or similar)

Python / Node.js

AWS Lambda + API Gateway

8.1.2 Memory System

Session management

- Short-term memory:** Amazon DynamoDB for chat history storage
- Session duration:** 24-hour conversation context retention
- Data structure:** user messages, assistant responses, and basic metadata
- Integration:** connection to existing customer profiles in MongoDB/PostgreSQL

8.1.3 Knowledge Base (RAG)

Implementation scope

- Vector database:** Amazon OpenSearch Serverless
- Embedding model:** Amazon Titan Embeddings
- Content sources:** customer service SOPs and guidelines, a basic FAQ database, and refund and delivery policies
- Search capability:** semantic search for relevant information retrieval

8.1.4 Tools and Actions

Core tools

- Order tracking:** API integration to retrieve order status
- Customer history:** access to previous orders and interactions
- Basic refund processing:** automated refund initiation for eligible cases
- Escalation tool:** handoff to human agents with a conversation summary

8.1.5 Safety and Evaluation

Guardrails

- Input validation and content filtering
- Response quality checks before delivery
- Escalation triggers for sensitive or complex queries

Basic evaluation

- Response accuracy monitoring
- Customer satisfaction feedback collection
- Performance metrics tracking (response time, resolution rate)

8.2 Use Case 2 — Semantic Search: Technical Implementation

8.2.1 Search Enhancement Features

Core capabilities

- Context-aware search beyond keyword matching
- Multi-language support (English/Arabic)
- Improved product and restaurant discovery
- Search result ranking based on user preferences

Technical stack

Amazon OpenSearch Serverless

Amazon Titan Embeddings

API layer over existing Typesense search

8.2.2 Implementation Approach

Phase 1 scope

- Vector embeddings for product and restaurant data
- Basic semantic search functionality
- Integration with the existing search interface
- Performance comparison against current keyword search

8.3 Infrastructure and Deployment

8.3.1 AWS Services Stack



AWS Bedrock

Foundation model access and management.



Amazon DynamoDB

Session and conversation storage.



OpenSearch Serverless

Vector search and knowledge base.



AWS Lambda

Serverless compute for agent functions.



API Gateway

RESTful API endpoints for frontend integration.



CloudTrail

Audit logging across the environment.

8.3.2 Security and Compliance

- Data encryption at rest and in transit
- IAM roles and policies for service access
- VPC configuration for network security
- Audit logging with AWS CloudTrail

8.3.3 Monitoring and Observability

- AWS CloudWatch for system metrics
- Custom metrics for agent performance
- Basic alerting for system failures
- Usage and cost tracking

8.4 Development and Deployment Timeline

Phase 1 • Customer Service Agent PoC • 8 weeks

- ✓ Basic chatbot with core functionality
- ✓ Integration with the order tracking system
- ✓ Simple knowledge base with SOPs
- ✓ Internal testing environment

Phase 2 • Pilot Release • 4 weeks

- ✓ Production-ready chatbot for limited users
- ✓ Performance monitoring dashboard
- ✓ User feedback collection system
- ✓ Escalation workflow to human agents

Phase 3 • Semantic Search PoC • 6 weeks

- ✓ Vector search implementation
- ✓ Integration with the existing search interface
- ✓ Performance comparison analysis
- ✓ Recommendation for full rollout

8.5 Success Metrics and KPIs

Customer Service Agent

70%

AUTOMATION RATE

<3s

RESPONSE TIME

4.0+

CSAT RATING

40%

COST EFFICIENCY
GAIN

Semantic Search

20%

SEARCH ACCURACY
GAIN

15%

SEARCH-TO-
PURCHASE LIFT

85%

INTENT RECOGNITION ACCURACY



GET IN TOUCH

Let's discuss your AI roadmap

Have questions about this assessment or ready to scope the next phase? The Flentas team is available to walk through the findings, refine the roadmap, or kick off the proof of concept.



EMAIL

sales@flentas.com



LINKEDIN

[Flentas Technologies](#)

8/10

TECHNICAL
READINESS

3

PRIORITY USE
CASES

70%

QUERY
AUTOMATION
TARGET

40%

PROJECTED COST
REDUCTION