# h1 Architecture Vision Document

## h2 Problem Description

### h3 Problem Statement

Large enterprises handle thousands of IT incident tickets every month. These tickets often contain redundant issues, verbose logs, and inconsistent resolution documentation. As a result, service agents—especially newer or L1 support personnel—struggle to quickly identify resolution steps, leading to longer response times and reduced service quality

This problem is compounded by a lack of effective knowledge reuse and manual triaging.

This challenge invites participants to build a Generative AI-powered system that ingests historical ITSM data, automatically summarizes incoming tickets, and suggests resolutions by referencing prior solved incidents—empowering faster, consistent, and more intelligent IT support.

**Table 1 - Showing Stakeholders vs. Business Capability Function vs Business Concerns**

| Stakeholder | Business Capability / Function | Business Problem/Concerns |
|---|---|---|
| L1 IT Support Agents | IT Support | Limited technical knowledge; unable to resolve issues like 404 errors independently; rely heavily on escalation to L2 for complex problems |
| L2 IT Support | IT Support | Overburdened with tickets that could be resolved at L1 if proper guidance/tools were available |
| L1/L2 IT Support Agents | Account | Secure login - own secure username and password or SSO |
| Alpaka AI System | IT Support (Tools) | Ingests historical ITSM data, summarises new tickets, and recommends resolutions by referencing prior incidents and knowledge base entries<br><br>Connects UI, AI engine, ITSM system, and knowledge base; must be scalable, secure, |

| Stakeholder | Business Capability / Function | Business Problem/Concerns |
|---|---|---|
| | | and performant<br><br>Need reliable data pipelines and feedback loops; responsible for ensuring that AI suggestions improve over time<br><br>Web-based Chatbot UI serves as the primary tool for L1 agents to interact with the AI assistant, search historical data, and receive contextual recommendations<br><br>Integrates with MS Teams App<br><br>Knowlede base matches incoming tickets with possible fixes; must be accurate, searchable, and well-maintained |
| ITSM Platform | IT Support (Tools) | Stores prior tickets and resolutions; inconsistent documentation or verbosity can make knowledge retrieval challenging |
| End Users / Employees (i.e. customers requiring IT support) | Customers | Expect faster, more accurate resolutions; often submit vague or repetitive tickets, making summarisation and suggestion features critical for support agents |
| Risk and Compliance Management (optional) | Business Management | Depending on the needs of the organisation, stricter security protocols may need to be implemented (MFA? Documentation of testing? GDPR?) |

✓

h3 **Business Vision Statement**

The TeBS vision is to be able to empower L1 IT support agents to independently resolve a broader range of incident tickets (and thus reducing reliance on L2 support) by leveraging a cost-effective, AI-powered assistant embedded directly within Microsoft Teams, and/or with a custom web application. The assistant will be accessible via a user-friendly chatbot embedded in Microsoft Teams, and/or a React-based web interface.

The solution will be cost-effective, modular, and built using Microsoft Azure products and services that enable powerful language understanding, semantic search, and pattern recognition.

This AI-driven platform will feature:

- A **conversational assistant** that allows agents to ask natural language questions like *"Has this issue occurred before?"* and receive context-aware proven resolutions.
- Optional advanced functionality such as **root cause pattern detection** and **proactive alerting** based on cluster analysis of historical ticket trends.

This solution leverages the capabilities of existing ITSM tools.

### h3 Change Drivers & Opportunities

Here is a list of key business drivers and opportunities to implement the AI-powered assistant:

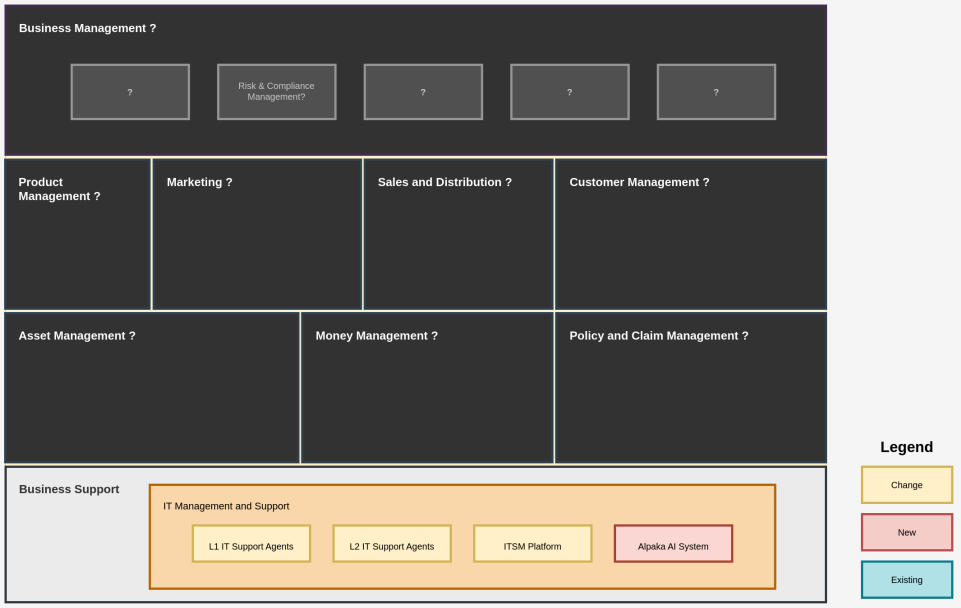
- The business outcome is to
  i. Reduce ticket resolution times
  ii. Empower L1 support to handle more incidents independently
  iii. Improve overall service quality
- The solution should integrate with MS Teams
- The web-based assistant interface must be accessible only to authenticated users to protect internal systems and data
- The web site's availability should support a 24/7 uptime
- Consider responsiveness
- Privacy of customer data should be handled as confidential and security should be implemented

### h3 Business Capability Impact

Based on the limited information, it seems the solution should be flexible enough that it can be easily adapted to a variety of Business Capability Models, and drive positive change for IT Management and Support.s

Below is a high-level Business Capacity map inspired by the Panorama 360 Reference Model, that showcases this flexibility.

**Figure 1 - High-Level Business Capability Map Showing the Alpaka System**




Business Functions outside of IT Management and Support can be considered a “black box”, but the system should be flexible enough to take them into account, if necessary.

**Table 2 - Showing How the Business Capabilities Are Impacted by the Alpaka AI System**

| Capability | Description | New/Migrate/Change |
|---|---|---|
| AI System (Alpaka) | AI system that allows agents to ask natural language questions like *“Has this issue occurred before?”* and receive context-aware proven resolutions with optional advanced functionality such as **root cause pattern detection** and **proactive alerting** based on cluster analysis of historical ticket trends. | New |
| ITSM System | Provide historical support data. May need to find a way to feed the data automatically to the AI system | Change? |
| Risk & Compliance (optional) | Risk & Compliance will need to check compliance needs | Change? |
| L1 IT Support Agents | Processing of IT Service incidents will change due to L1 being empowered to take on more incidents, and AI System will be integrated into their workflow | Change |
| L2 IT Support Agents | Reduced incidents they have to deal with?<br><br>(L2 taking on L3-level support incidents has not been discussed as a requirement by the stakeholders) | Change |

## h2 Architecture Vision

To provide an AI-powered assistant to enhance the IT Support activities of the organisation. The solution architecture should implement the solution using the following architecture principles:

- **Business continuity** - ensure the system has a disaster recovery plan included
- **Ease of use** - Keep the technology selection simply and easy to adapt and use
- **Data security** - ensure the data is secure at rest and in transit
- **Azure** - The design should include architecture components that run on Azure services

### h3 Architecture Assumptions

The following architecture assumptions are made based on the drivers and objectives of the business problem:

- Sample historical ITSM data is provided, and future ITSM data will follow the same structure/schema for consistency in processing and search (ITSM may have API?).
- The assistant will integrate with **Microsoft Teams** as a registered app, accessible only to authenticated users via MS Teams SSO.
- The system will use **Azure OpenAI** for summarisation and natural language generation tasks such as ticket interpretation and suggestion generation.
- **Azure Cosmos DB** will be used to store processed ticket metadata, embeddings, and vector indexes to support semantic search at scale.
- **Azure Cognitive Search / AI Search** will be used to index and query both structured ticket data and unstructured knowledge base content.
- Embedding generation and similarity matching will be powered by **vector indexing**, assumed to be supported either via Azure AI Search or a custom service.
- Backend logic, including API routes such as `/recommend-resolution` and `/ask-assistant`, will be hosted on **Azure Functions** for scalability and cost efficiency.

### h3 Constraints

- Solution must be completed in 8 days (I found out about the Hackathon late)

### h3 Risks

| Risk | Mitigation | Owner |
| --- | --- | --- |
| Project not delivered on time | Focus on MVP and solution architecture | Product Owner (Alex) |
| IT Operational cost might be high | Solution Architecture should include auto scaling capacity and services should be turned off after hackathon. Additionally, make sure to use the Azure Free Trial, which includes $200 worth of Azure credits. | Solution Architect (still Alex) |

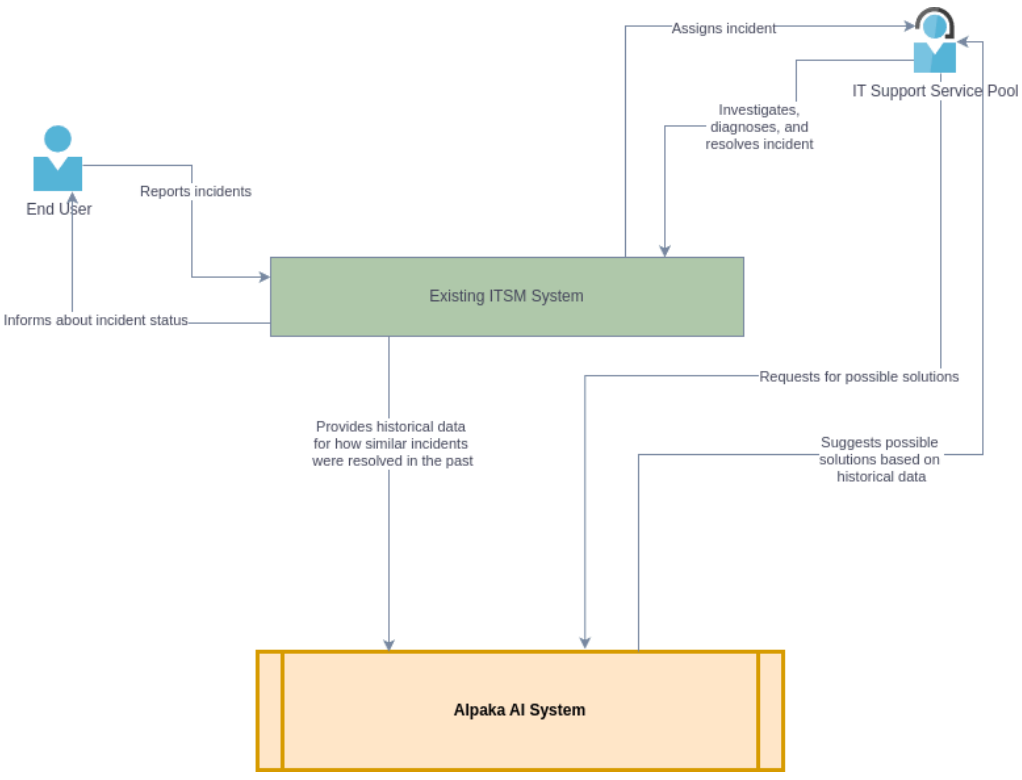
### As-is Conceptual Architecture

There is no as-is architecture as this is a new solution integrating with existing systems.

### To-Be Conceptual Architecture

The below context diagram depicts the interactions between the different systems and actors.

**Figure 2 - Showing Context Diagram**




The following interactions are expected:

- The end users will interact with the existing ITSM system by reporting incidents
- The existing ITSM system will provide historical data for how incidents have been resolved in the past
- The Alpaka AI System will interact with the historical ITSM data by:
  - Taking a new incident and returning probable resolutions based on semantic similarity with historical tickets
  - Return past tickets with high similarity to the given incident using vector-based search
  - Provide natural language response based on user query
- The IT Support agents will interact with the Alpaka AI System by sending natural language queries and opening

## High-Level Non-Functional Requirements

### Availability

- The solution should be highly available 24/7.
- Planned downtime (if any) must be scheduled and communicated in advance.

### Performance

- The web client should remain responsive across device types, loading initial components in under 2 seconds.
- Solution should allow for auto scalability on evenings and weekends

### Volumes

- The system must support a minimum of 1027 historical incident records initially, with the ability to scale to 5000 records.
- Vector database and semantic index must support real-time similarity matching against large-scale embeddings
  - The system converts tickets or user queries into embeddings, and stores those embeddings into a vector database. The system then needs to quickly search for similar items (e.g. past incidents) based on semantic similarity.

The sample historical data set contains 1027 records

### h3 Business Continuity

- Include automated backups (e.g. with **Cosmos DB**), with geo-replication enabled across **Azure** regions.

### h3 Security

- All data should be encrypted at rest and in transit on all services.
- Authentication via **Microsoft Teams** SSO
- Role-based control for different user types
- Implement audit controls on data changes
- Audit logs must be maintained for all interactions with the assistant and backend APIs
- Rate limiting and input validation must be enforced
- X509 Certificate (e.g. SSL/TLS)

### h3 Operations and Monitoring

- Implement monitoring for both the health of the web site and the database via **Azure Monitor** or **Application Insights**
- Set up alerts for failures, high latency, or downtime across critical services (e.g. API, OpenAI, DB).
- The solution should include exception handling and alert on any errors

### h3 Networking

- Create a public subnet for the web frontend and private subnets for the backend database and other services
- Create firewall rules for traffic for the API (check for in-house Azure solution)

### h3 User Interface Requirements

- The chatbot assistant must be accessible via Microsoft Teams as a registered app
- A separate web UI must be available

- It can allow for support agents to explore historical data, similar cases, and common solutions using natural language queries
- It may contain a web-based chatbot UI in addition to or in lieu of the MS Teams integration
- It must be Accessible

### h3 Architectural Requirements

- Use Azure Cloud services
- Implement a Dev, QA, UAT, and Production
- Detail of the development stack will be in the solution options
- Solution should follow microservices architecture by implementingg the solution components in containers
- Use the DevOps pipeline to be able to cater for automated deployment

## h2 Proposed Solution Option

- **Azure OpenAI** – for ticket summarisation and resolution suggestion via natural language processing.
- **Azure Cosmos DB** – to store and query historical tickets, incident metadata, and resolution mappings.
- **Azure Cognitive Search / AI Search** – to index and query both historical tickets and knowledge base content semantically.
- **Vector Indexing** – to power similarity-based search (e.g. `/search-similar-incidents`) using embeddings.
- **Azure Functions** – to implement lightweight, scalable backend APIs such as `/recommend-resolution`, `/ask-assistant`, and `/search-similar-incidents`.
- **LangChain / Semantic Kernel** – to orchestrate prompt chaining and multi-step reasoning in the assistant.
- **AI Foundry** – to manage and deploy AI capabilities with monitoring and feedback loops.
- **React-based UI** – for the web assistant chatbot, designed in Figma and optimised for usability. (optional)
- **Microsoft Teams App** – for seamless integration of the assistant into existing agent workflows using secure login via MS Teams.

- [ ] mention Retrieval Augmented Architecture

    - information retrieval system provides ggrounding data the LLM can use when formulating a response
    - contrain generative AI to the business' content sourced from vectorised documents and other data formats that you have embeddings for
    - should provide
        - indexin strateies that load and refresh at scale, for all the content, at the frequency you require

        - query capabilities and relevance tuning. the system sohuld return relevant results, in short-form formats necessary for meeting the token lenth requirmetns of LLM inputs
        - Integration with embeddin models for indexing, and chat models or language understanding models for retrieval
- approaches for RAG in Azure AI Search

    - Azure AI Foundry, use a vector index and retrieval augmentation.
    - Azure OpenAI, use a search index with or without vectors.
    - Azure Machine Learning, use a search index as a vector store in a prompt flow.
- RAG Pattern for Azure AI Search

    i. Start with a user question or request (prompt).
    ii. Send it to Azure AI Search to find relevant information.
    iii. Return the top ranked search results to an LLM.
    iv. Use the natural language understanding and reasoning capabilities of the LLM to generate a response to the initial prompt.
    v. Optionally, use agentic retrieval where an agent evaluates an answer and finds a better one if the original answer is incomplete or low quality.
- Azure AI Search provides inputs to the LLM prompt, but does not train the model. In RAG architecture, there's no extra traninig. the LLM is pretrained using public datam but it generates responses that are augmented by information from the retriever - in this carse Azure AI Search

the app server or orchestrator is the interation code that coordinates the handoffs between information retrieval and the LLM
common solutions include LanChain, which interates with Azure aAi Search, which makes it easier to include Azure AI Search as a retrieview int he eworkflow

The information retrieval system provides the searchable index, query logic, and the payload (query response). The search index can contain vectors or nonvector content. Although most samples and demos include vector fields, it's not a requirement. The query is executed using the existing search engine in Azure AI Search, which can handle keyword (or term) and vector queries. The index is created in advance, based on a schema you define, and loaded with your content that's sourced from files, databases, or storage.

ure AI Search doesn't provide native LLM integration for prompt flows or chat preservation, so you need to write code that handles orchestration and state. You can review demo source (Azure-Samples/azure-search-openai-demo) for a blueprint of what a full solution entails. We also recommend Azure AI Foundry to create RAG-based Azure AI Search solutions that integrate with LLMs..

## h2 References


- https://learn.microsoft.com/en-us/azure/search/retrieval-augmented-generation-overview?tabs=docs