Scaling AI with Large Language Models (LLMs)

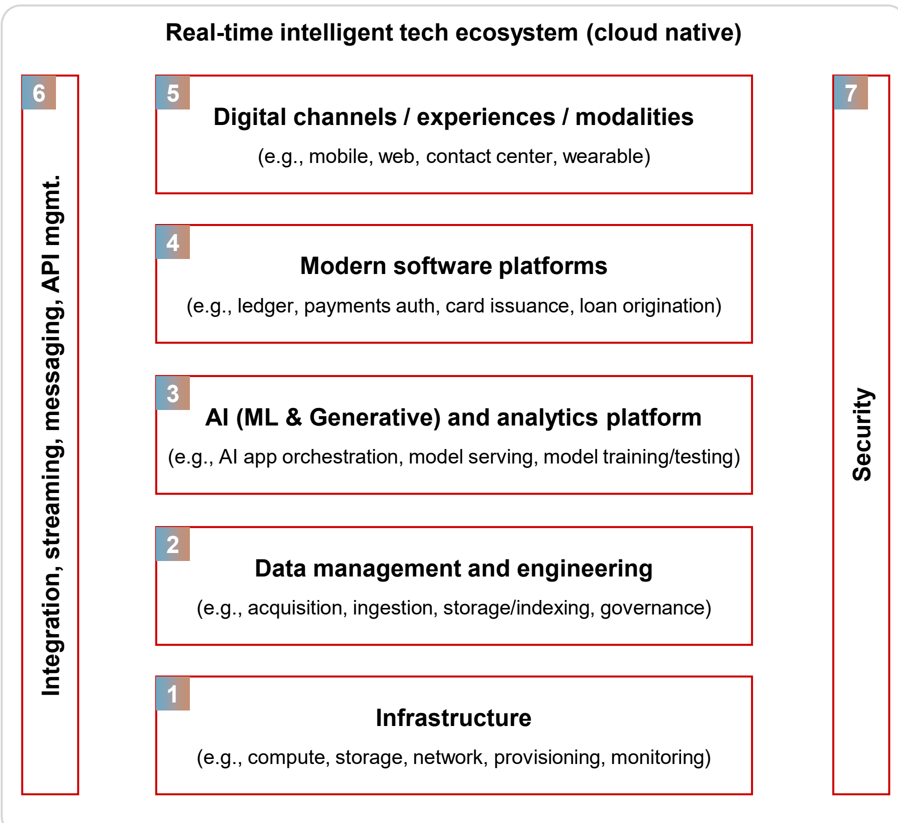
Meethun Kumar Panda

Conf42 Large Language Models (LLM) Conference
06-Mar-2025

Key design principles to enable Gen AI use case at scale utilizing cloud

Generative AI Characteristics	Key Design Principles	
it is important to be able to experiment with different models and element technologies.	**Flexibility**	• **Replaceable modules:** The components of the application are modular and technology-agnostic • **Flexible** storage: Scalable, platform-agnostic storage solutions • **Self-healing architecture:** the ability to identify and recover from errors and misconfigurations without human intervention
Focusing on larger-scale unstructured data than conventional AI/ML models	**Scalability**	• **Reusable module:** Deploy a repeatable infrastructure stack using "infrastructure as code (IAC)" • Coordination: Control application traffic and other architectural components, such as DNS and load balancing • **Elastic scaling / hosting:** Ability to automatically add/remove infrastructure based on changes in traffic patterns (such as tuning compute configurations based on CPU, memory, and storage)
Foundation models are 'black boxes' in structure, with low explanatory power and error traceability	**Monitoring**	• **Process transparency:** Enables all processes in the architecture to be observable and proactively diagnose issues • **Drift/Deflection Control:** Implement restrictive measures and mechanisms to monitor/manage drift and bias in the process • **Continuous improvement:** monitoring for performance benchmarking; Implement a systematic process for incorporating feedback
It is common for foundation models to be connected externally through APIs.	**Security**	• **Data management:** Secure management of sensitive data (such as PII) • **Access management:** Enforce access management to restrict access to infrastructure and authenticate users of applications • **API key and secret** management: (1) centralized key vault, (2) dynamic secret, (3) automatic secret injection, and (4) monitoring/auditing for secret activities

Architecting AI-Driven Ecosystems: A Bottom-Up and Destination-Back Approach



7 **Security:** security embedded in engineering and ops for all dimensions of the stack, network, infrastructure, data, models, software, APIs, etc.

6 **Integration:** pervasive use of enterprise data, event, message streaming pipelines (“data is air”) and enterprise API management platform

5 **Digital:** flexible and adaptable digital channels, experiences across modalities able to easily embed AI output into real-time experiences

4 **Software Platforms:** software architected for real-time, event-driven and micro-services and to embed AI into workflow, experiences, and decisions

3 **AI Platform:** flexible AI platform for AI app orchestration, model serving, integration, model training, data engineering, and utilities

2 **Data Management:** world-class unstructured (and structured) data acquisition, ingestion, storage, indexing, and governance capabilities

1 **Infrastructure:** all-in on cloud and advanced “GPU stacks” computing to support compute storage intensity, elasticity and scalability required for AI

AI platform reference architecture (traditional & Gen AI)

Legend: Traditional AI | Gen AI | Both Traditional & Gen AI | ★ Hotspot of change due to Gen AI

6. API Management

5. Digital Channels

4. AI Application / Software

Application Frontend
- UI/UX
- Web App | React / Vue
- FE Database
- Error Handling
- …

Application Backend / AI Logic
- Decision / Rules Engine
- Optimization Process
- Simulation Process
- Prompt Engine / Template
- On Demand Code Gen
- Accuracy /Bias Eval.

Gen AI Application Logic Patterns 6
- Research: *Summarization, Doc Classification, Sentiment Analysis, …*
- Search: *RAG / Knowledge, Info Extraction, Structured Query, …*
- Comm.: *Dialog Response, Intent Tracking, Query Classifying, …*
- Content Gen: *Text Content Gen, Speech Gen, Image Gen, …*
- Workflow: *Decision Making, Flow Control, State Management, …*
- Software Dev: *Code Comment, Code Gen / Analysis, Documentation, …*

Application Service
- Development IDE
- Load Balancing
- App Logging / Monitoring
- CI/CD
- Batching
- …
- Container Mgmt.
- Autoscaling
- …

3. AI Model 5

AI Model Training / Fine-Tuning
- Model Training IDE (*e.g., Jupyter, Databricks, SageMaker Studio)*
- SQL Tool
- Data Mining: Correlation, Clustering, Association Rule, Pattern, Feature Importance
- Machine Learning & Deep Learning: Regression, Bayesian, Time Series, ABM, Segmentation, System Dynamic, Anomaly Detection, KNN, Tree based: RF,TN, NLP, NN/ANN/RNN, Transformer
- Rules Engine & Dev
- Hyper-Parameter Tuning
- MaaS Finetuning
- …

AI Model Deployment & Servicing
- Data Connectors
- Model Validation
- LLM Broker
- LLM Inference
- Data Pipeline
- ML Flow
- 4 LLM Evaluation / Selection ★
- BYO LLM Inference
- Champion / Challenger
- AutoML
- Feedback Loop
- Model Cost / Performance Monitoring
- Scheduling / Triggering
- Error Handling
- Model Sharding / Parallel Inference for Custom/New LLM (e.g., TensorRT) ★
- …

2. Data Management & Engineering
- Manual Data Capture ★
- External Data Intake ★
- Batch / Change Data Capture ★
- Streaming Data Capture ★
- Data Transformation
- Data Querying
- Data Indexing
- Data Inventory
- 3 DW, Lake, Delta, Lake House
- Datamart | Business Ready Data
- Document / Graph Database ★
- 2 Vector Database ★
- Master Data Management
- Meta Data Management
- Data Quality & Remediation
- Data Lineage
- Data Virtualization
- Data Visualization
- Feature Store
- Sensitive Data Tokenization
- Document Loading & Parsing ★
- Document Chunking ★
- Data Vectorization / Embedding ★
- Metadata Enhancement
- Lexical & Semantic Search
- Semantic Caching DB ★
- Voice, Image, Video Data Mgmt ★
- …

1. Infrastructure 1
- Cloud & Distributed Computing ★
- GPU, Advanced, Accelerated, High-Performance Compute ★
- Serverless / MaaS ★
- Network and VPCs / interlinks
- Storage ★
- Infrastructure Provisioning
- Ops / Monitoring
- BCP & DR
- FinOps ★
- …

7. Security
- Prompt Guard
- Content Moderation
- Security Logging
- Kill Switch
- Alert
- User Auth
- AD
- IAM
- Encryption
- Security Ops
- Security Scanning
- Endpoint, Intrusion & Network Security
- VPC
- Firewall

Strategic foundational choices on Generative AI building blocks

1 **Infrastructure**	Single cloud; Multi-cloud; SaaS foundation model API; Self-hosted hybrid/on-premise GPU full-stack AI platform (NVIDIA); Cloud vendor specialized AI chips; Niche provider AI chips
2 **Vector database**	Enterprise-level vector storage; Application-level vector storage Vector indexing added to existing data store; Specialized vector data store
3 **Data architecture**	Single integrated data lake; Multiple federated data lakes / data mesh Integrated structured and unstructured storage (e.g. Delta Lake); Separate structured and unstructured/vector data stores; Separate structured, unstructured, and vector stores
4 **LLM(s) selection**	Large foundation model; Customized large foundation model; Multiple smaller specialized FMs Open source/proprietary; Commercial
5 **AI / ML Ops platforms**	Open source; Niche best of breed; Hyperscale cloud vendor (e.g., AWS SageMaker); GPU full-stack AI platform (e.g., NVIDIA)
6 **Gen AI app multi-cloud architecture**	All on same cloud as LLM (ingest, vector db, AI app FE/BE, LLM); AI app front-end on current cloud, all else on same cloud as LLM; All on current cloud; LLM hosted on relevant LLM cloud

Unstructured data readiness for “AI everywhere” is a critical pre-requisite

Three key data management challenges

	Data requirements for Gen AI	Why it’s challenging	Ecosystem perspective
Organizing, cleaning, & labeling and catalog unstructured data	Data must be **structured, accurate, and contextually “sound”** before it’s fed into foundation models for Gen AI solutions	**Most data is unstructured** (e.g., live in PDFs or images) and must be **organized, cleaned, and labeled** prior to being used in a data pipeline	“*Finding and labeling data for a given use case has been the singular bottleneck - You’re set up for success when you can properly label the data for the specific use case at scale.*”
Storing data to be effectively used by Gen AI solutions	Data needs to be stored in a way that’s **compatible with being “fed” into foundation models** (e.g., in vector databases)	Company **with limited or poorly maintained** data storage practices will have a **vast amount of data that is not ready for ingestion** into foundation models	“*Model output is directly related to how good of an input you can put into the model – the benefit of working with a database of real data is that it's generally really good input.*”
Extracting, transforming, & loading data into foundation models	Foundation models **continuously learn and adapt**, which requires **constant managing of data pipelines** to ensure compliance, accuracy, and sustaining of data privacy	Companies must establish **appropriate data pipelines** and integration processes in line with overall **data governance strategies**	“*It's not only about cleaning data at rest. It's about being able to exchange or move that data around... If we can have data in a state that allows us to move it elsewhere and plug it into [FMs] it makes a huge difference.*”

What is hard about building and operationalizing Generative AI beyond typical AI/ML project challenges

TECHNICAL CHALLENGES

- Decreasing **hallucinatory response** and **improving quality of output** requires **new prompt engineering skills** and **new technical skills** (e.g., grounding, fine-tuning, chaining)
- **Vastly different MLOps** with many tools nascent or non-existing (e.g., vector DBs, p-stores, chaining, failovers, guardrails)
- Creation of **grounding databases** (e.g., sources, chunking)
- **New types of cyber-security risks** (e.g., chat prompt injection)
- Increasing **scalability and deployment** efficiency (incl. caching)
- More and **diverse data improves performance** but requires **extensive integration**
- **Balancing data privacy and security** with cloud LLMaaS

OPERATIONAL CHALLENGES

- Picking smart (build vs buy) & **building a roadmap** that balances today forward with future-back
- Research **user experience and unmet needs** to develop hypotheses on agent automation
- Need for a different type of **UI innovation**
- **System to identify issues** with response pathways to feed back into development
- Estimating value and implementing **systems to track value and prove real value**
- Workforce **engagement and training** at scale
- Planning for **radically redesigned processes** and **potential organizational changes**

GenAI tech architecture is structured around six key elements

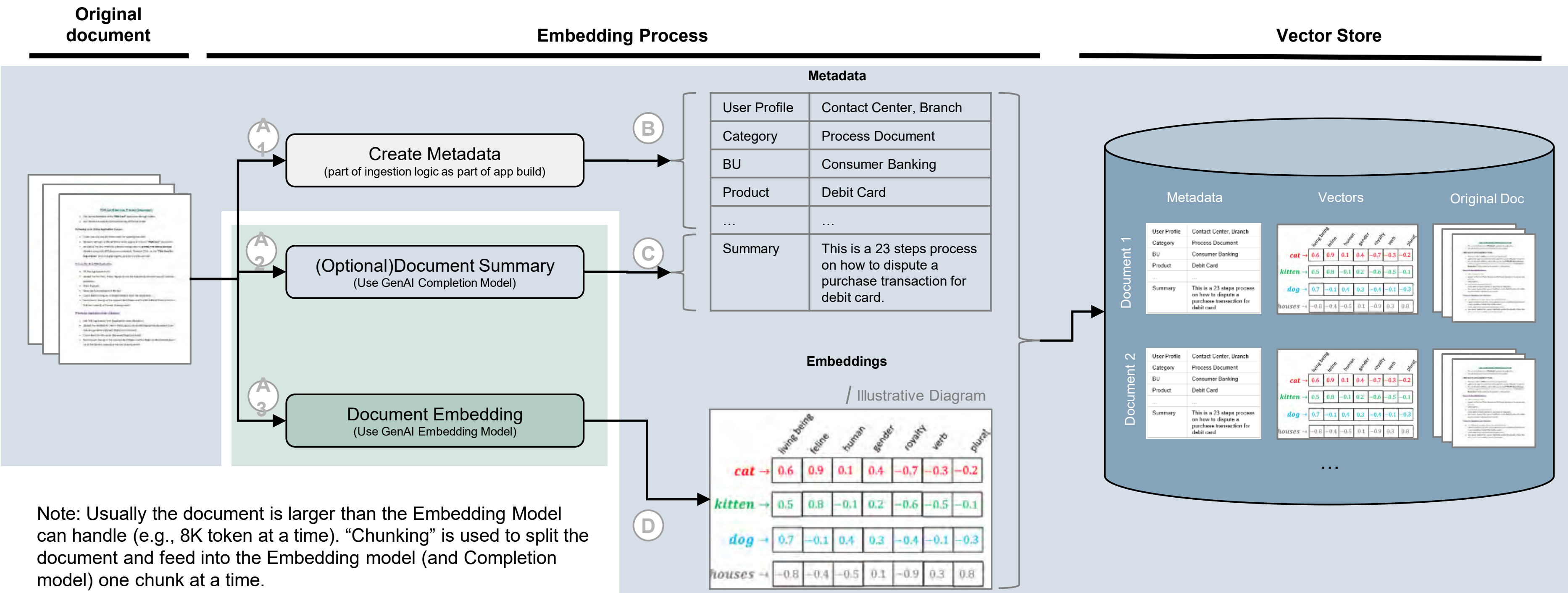
	GenAI architecture elements	Description	Considerations for decisions
GENAI TECH STACK	LARGE LANGUAGE MODELS	• Model trained on **vast amounts of data** to understand the structure of natural language • Enables generation of **novel content** and response to questions	• Capabilities • Data security • Ease of use and deployment • Multilingual capabilities
	EMBEDDINGS	• Representation of content and meaning that captures **semantic and contextual relationships** • Enables comparison of user questions with data in knowledge base **to determine relevance**	
	VECTOR DATABASE	• **Storage of data** with their embeddings • Enables **retrieval of relevant context** for specific queries based on semantic and contextual relationships	• Performance and scalability • Flexibility • Current availability
DATA & INFRASTRUCTURE	PLATFORM	• **Consolidates data from various sources** and provides tools for analysis, application development, and visualizations • **Accelerates AI application** and insights delivery	• Readiness of the options • Compliance with security norms • Data access availability • Scalability to accommodate new use cases
	HOSTING	• **Computational resources** and storage to run applications and store application data • **Enables end users to utilize application** in line with data security requirements	
	DATA STORE	• **Data utilized for each use case**, stored in business applications and IT systems • Data needs to be provided and pre-processed **in compliance with data security and privacy requirements**	

Semantic knowledge search use case example

AWS
Azure

Step 1: Preparation Knowledge Vector Store

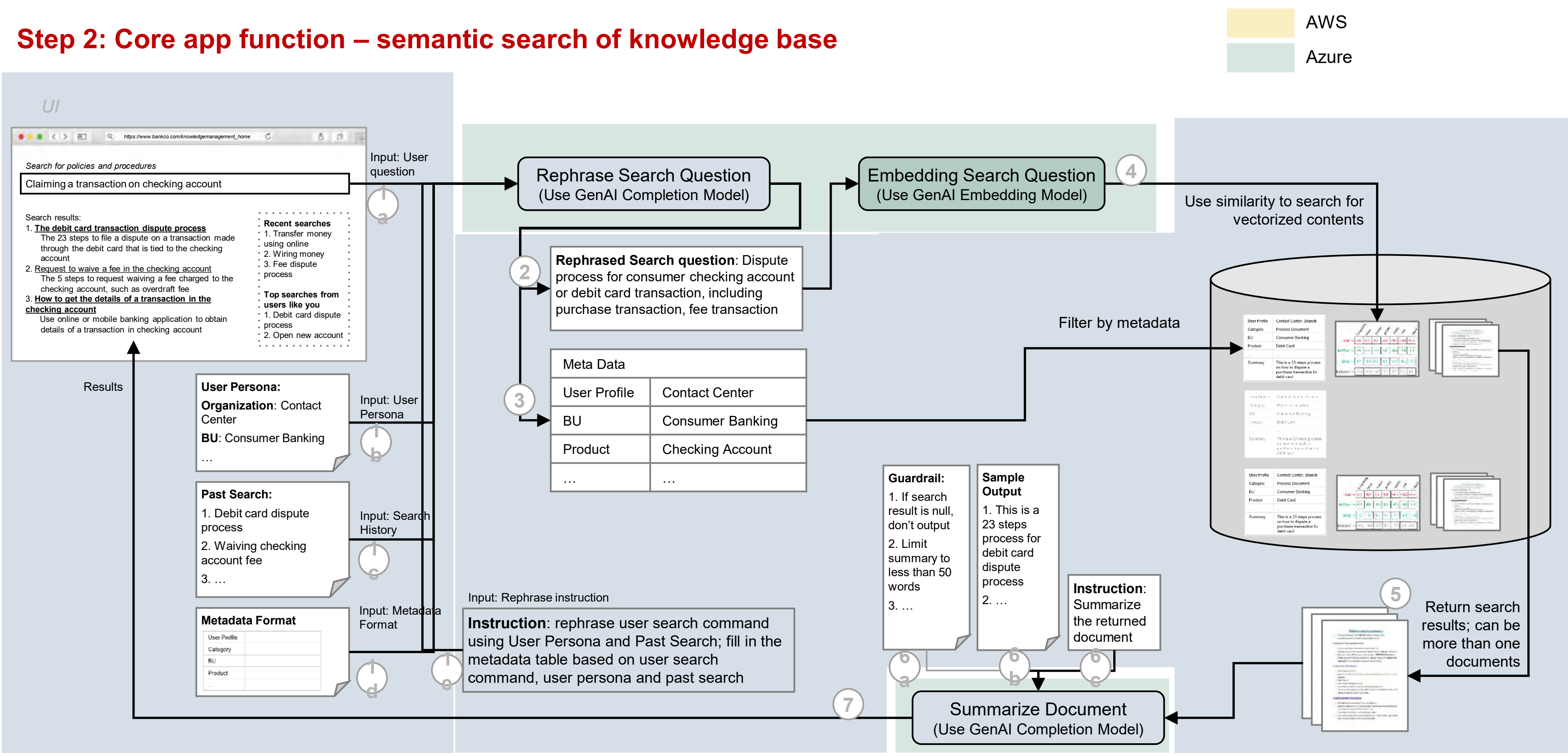
A key step is to ingest the document and prepare the vector store. Both metadata and document embeddings are prepared. The more precise and complete the metadata, the better the search result.



Note: Usually the document is larger than the Embedding Model can handle (e.g., 8K token at a time). “Chunking” is used to split the document and feed into the Embedding model (and Completion model) one chunk at a time.

Semantic knowledge search use case example

Step 2: Core app function – semantic search of knowledge base



Use case delivery requires multiple technical roles and resources

Use case delivery requires multiple technical roles and requirements

Process	1 Use Case Definition	2 Discovery	3 Set up	4 Application Development	5 Deployment and monitoring	6 Application Maintenance & Support
Core Activities	Define use cases to prioritize (activities targeted, hypothesis on GAI applications) Develop basic business case (benefits, resources, investments, ROI)	Data sourcing and ingestion, integration Define data sources Define hosting environment Determine info security requirements and access Develop application layer priorities to enable selected use cases Define model features and prototypes MVP development: prompts/embeddings/fine turning (if required) Implementation project management and oversight	Implement automated data and model training pipeline/MLOps Determine high level reference architecture for LLM applications Connect GPT models to other applications (plug-ins) and chain modules to create end-to-end application Establish test and security protocols	Design and code user interface Build API for integration with internal models Model testing accuracy (accuracy, bias, truthfulness, performance, etc) Risk assessment and monitoring setup (resilience, security, etc) Operations and user deployment Integration with existing infrastructure; any needed build or improvements	Pilot design and deployment Technical operations and monitoring Change management with impacted business functions Train users (e.g., on query formation) Design feedback loops and process for model improvement Monitor implementation and results/impact	Build out talent as needed, including data and analytics, enterprise technology, and business implementation expertise Application and knowledge base updates and improvements Maintenance, including: Tracking model performance and accuracy Risk and security monitoring Software patches and updates Application performance monitoring Troubleshooting/issue resolution
Key Resources	Business analysts Process SMEs Data Scientists	Data scientists Data engineers DevOps/Infra engineers	Data Scientists ML Engineers Data engineers DevOps engineers	Product managers Data scientists ML engineers Data/back-end/systems engineers Front-end engineers	Site reliability engineers ML engineers Process owners and change management Business analysts	Site reliability engineers User support Training and onboarding