Evaluating, Governing, and Supporting LLMs in Production

Penguicon 2024
Ryan Kather

Agenda

- About Me
- Defining Terms
- AI Hype
- Foundation Model Risks
- Trustworthy AI
- xOps
- Platform Considerations
- Evaluations and Metrics
- Resources / Q & A

About Me

Principal AI Engineer – IBM Data & AI

Old Hat Linux Guy, OSS Contributor, Father, Bad Musician and Wearer of Many Hats

Areas of AI

Artificial Intelligence

AI involves techniques that equip computers to emulate human behavior, enabling them to learn, make decisions, recognize patterns, and solve complex problems in a manner akin to human intelligence.

Machine Learning

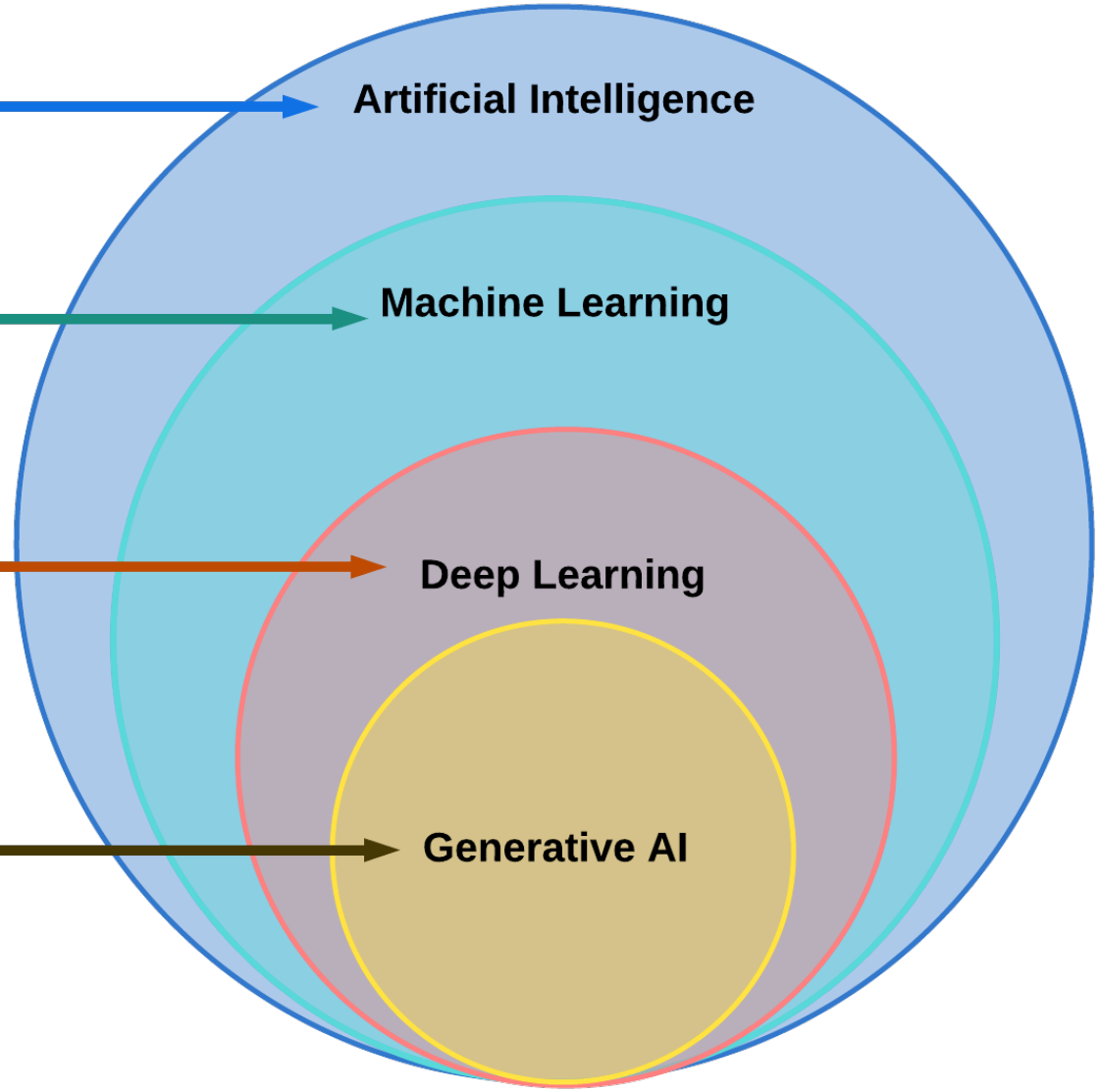
ML is a subset of AI, uses advanced algorithms to detect patterns in large data sets, allowing machines to learn and adapt. ML algorithms use supervised or unsupervised learning methods.

Deep Learning

DL is a subset of ML which uses neural networks for in-depth data processing and analytical tasks. DL leverages multiple layers of artificial neural networks to extract high-level features from raw input data, simulating the way human brains perceive and understand the world.

Generative AI

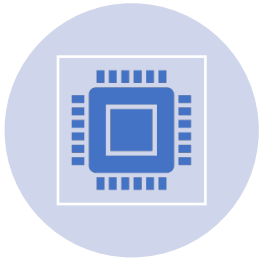
Generative AI is a subset of DL models that generates content like text, images, or code based on provided input. Trained on vast data sets, these models detect patterns and create outputs without explicit instruction, using a mix of supervised and unsupervised learning.



Unraveling AI Complexity - A Comparative View of AI, Machine Learning, Deep Learning, and Generative AI.

(Created by Dr. Lily Popova Zhuhadar, 07, 29, 2023)

Let’s Define Some Terms



LLM: Large Language Model, an AI model that has been trained on a vast amount of text data ranging from books and articles to social media posts and online conversations. The architecture of an LLM consists of layers of neural networks that learn to generate language in a way that is similar to how humans use language.

Generative AI: Generative AI refers to artificial intelligence systems that can generate new data or content, often based on a given set of input parameters or data. This can include text, images, audio, or other forms of data.

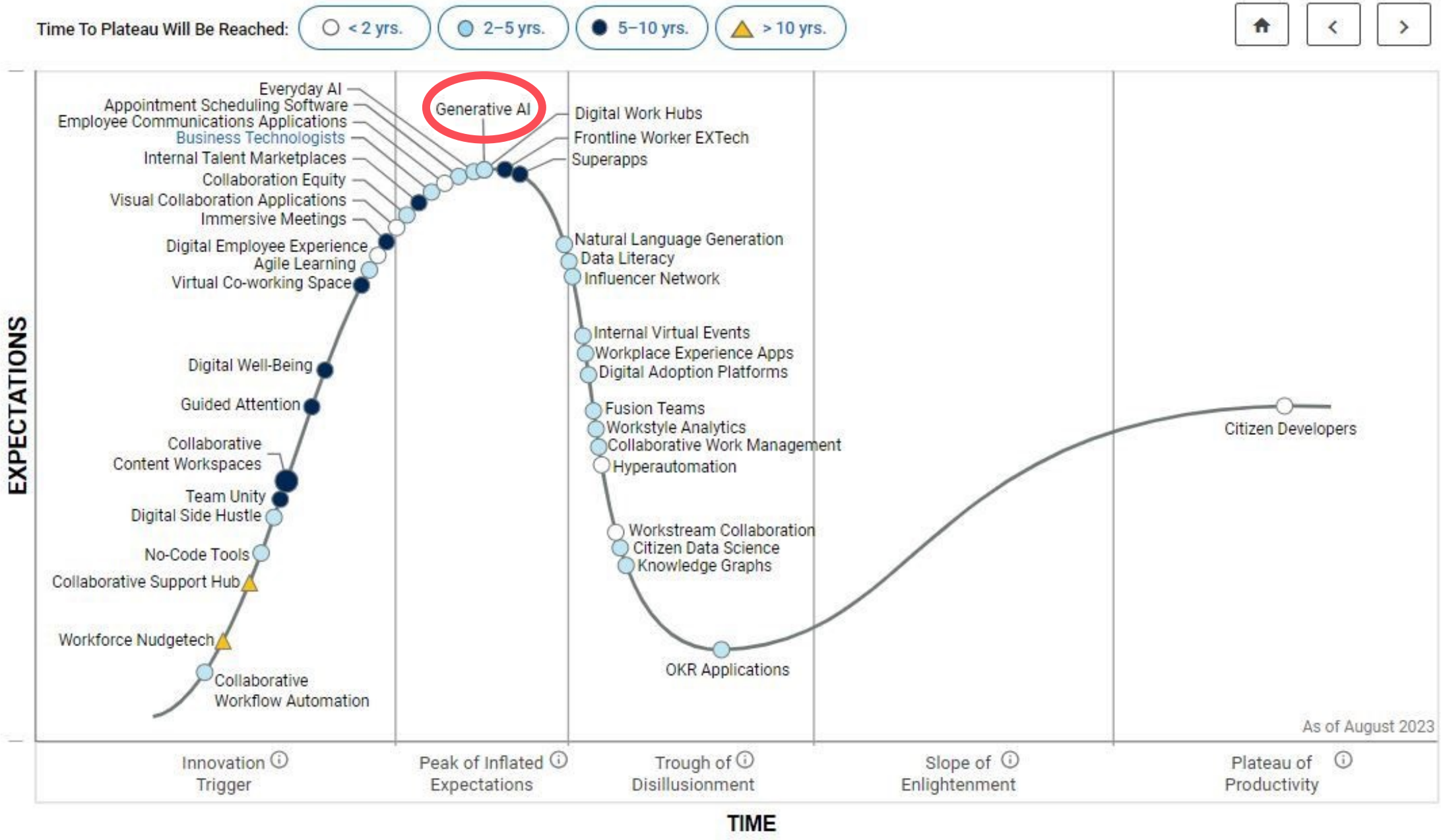
ChatGPT: ChatGPT is a generative AI chatbot developed by AI research lab OpenAI and launched in November 30, 2022. GPT stands for Generative Pre-trained Transformer and refers to a family of neural network-based Large Language Models developed by OpenAI.

Weights: Parameters in a neural network that determine the strength of connections between neurons. They are adjusted during the training process to minimize the error in the model's predictions.

Embeddings: A technique used to represent words, phrases, or even whole sentences as fixed-size vectors in a continuous space. These vector representations capture semantic and syntactic information, allowing models to process and understand text more effectively.

Foundation Model: A term first defined by Stanford’s Institute for Human-Centered Artificial Intelligence, a model that has been trained on a broad set of unlabeled data that can be used for different tasks, with minimal fine-tuning.

Generative AI is at the peak of hype cycle!



NYC’s government chatbot is lying about city laws and regulations You can be evicted for not paying rent, despite what the "MyCity" chatbot says.
-- Arstechnica

Foundation model risks

Risk Associated with Input

Training and Fine-tuning Phase

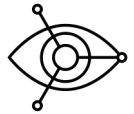
- Bias
- Data poisoning attacks
- Legal restrictions on data
  - Copyright and other IP issues
  - Inclusion of PI and SPI
- Data transparency challenges

Inference Phase

- Disclosure of PI/SPI/Copyright/other IP information as a part of prompt
- Adversarial attacks like evasion, prompt injection, prompt leaking, and jail breaking

Risk Associated with Output

- Bias in generated content
- Performance disparity
- Copyright infringement
- Value alignment issues (e.g., Hallucination)
- Misuse
- Exposing PI and SPI in the output
- Explainability challenges
- Traceability challenges



Challenges

- Transparency challenges
- Challenge around assigning responsibility
- IP issues
- Human exploitation
- Impact on jobs
- Environmental Impact
- Diversity and Inclusion
- Human agency
- Impact on education

What makes a generative AI system trustworthy?

How was it trained?

- Garbage in, garbage out
- An enterprise should not use a foundation model trained with a Wikipedia crawl
- The training material must be huge and comprehensive, but must also be curated

Can it detect & minimize bias & hallucination?

- How does the platform detect and correct bias?
- How can it prevent hallucination (providing random and untrue answers with absolute aplomb and conviction)?

Is it transparent?

- Open vs. black-box
- How to audit and explain a model and the answers it generates?
- Does the model track drift and bias? And how does it address them?

Does it support regulatory compliance?

- How do foundation models and their usage comply with privacy and government regulations?
- What are the guardrails?
- Who is responsible for inadvertently exposed personal identifiable information or a “wrong answer”?

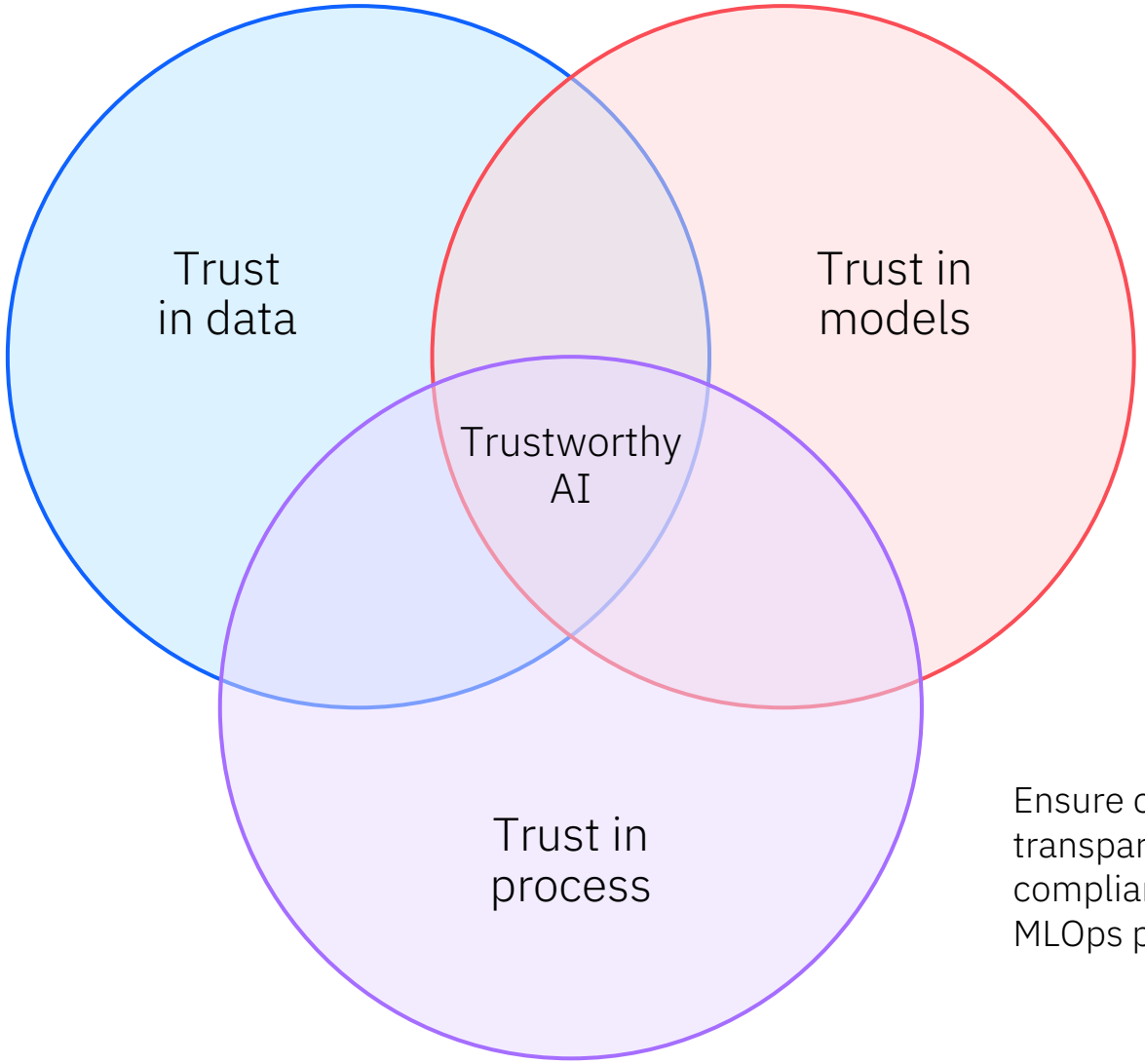
Is it safe?

- Who has control over the model, input data, and output data?
- Can you ensure that confidential information is not given out?
- How is it monitored?
- What safety features and guardrails are in place?

Can it be customized?

- Hybrid and multicloud?
- Can the model be fine-tuned with your data?
- Can it be enhanced and extended to make it more suitable for specific use cases?
- How will it integrate with other applications?

Trustworthy AI



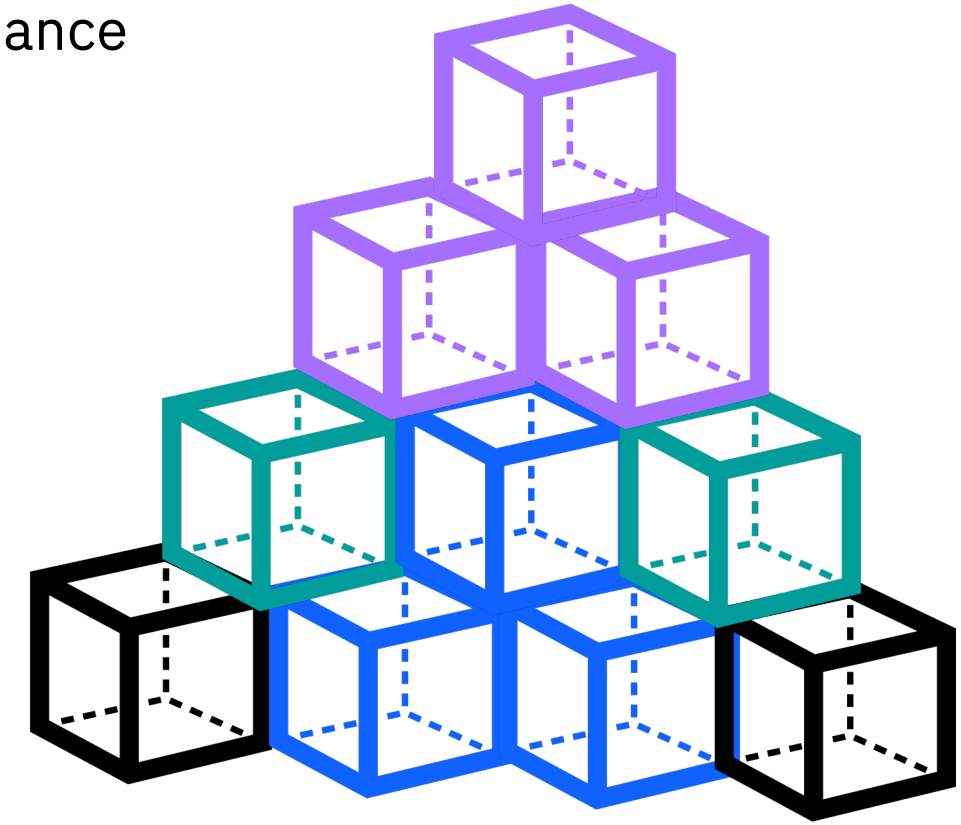
Measure model bias, explainability, and robustness

Ensure consistency, transparency, and compliance throughout MLOps process

IBM's approach to selecting models in watsonx.ai

Technical considerations

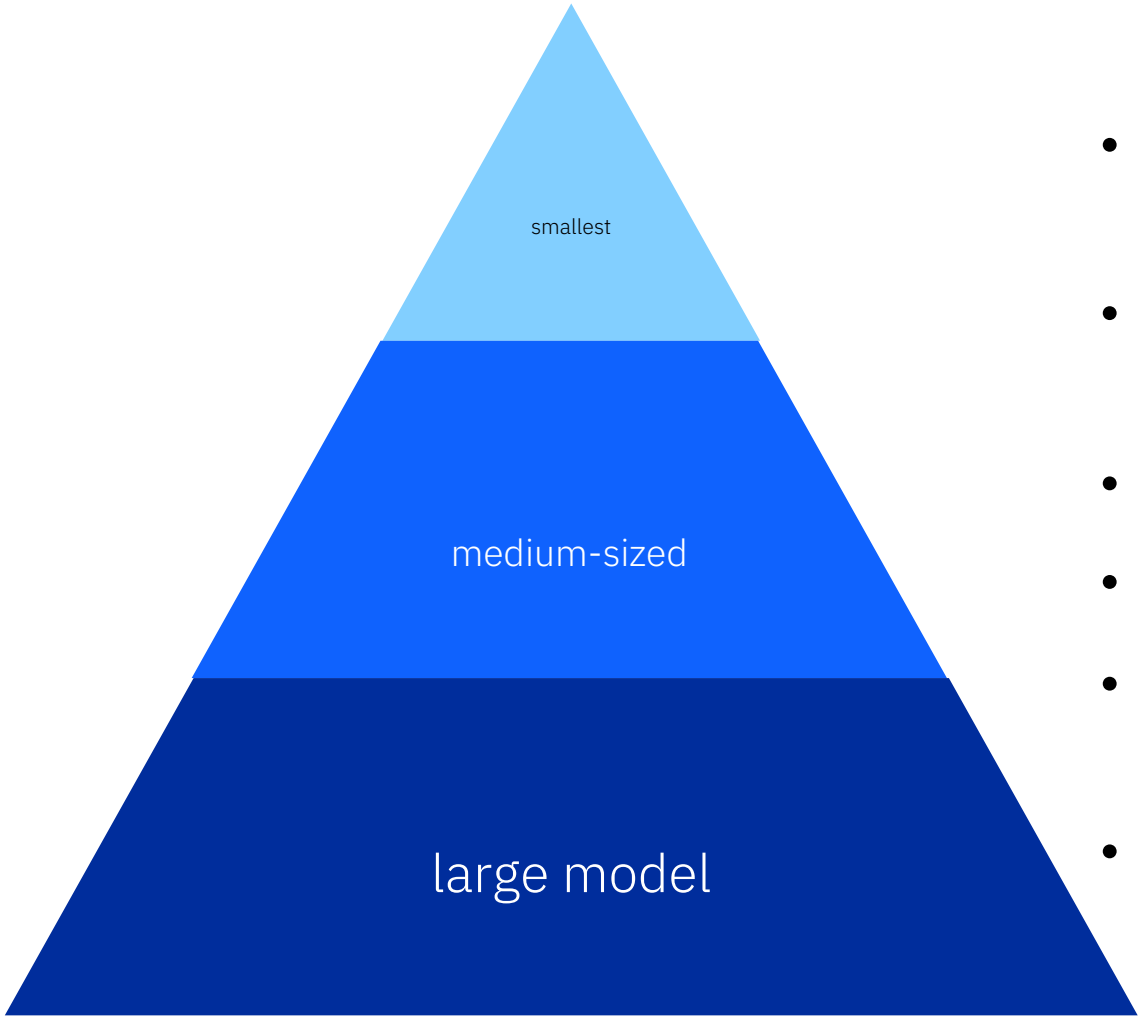
- Model performance
- Research
- Ethics
- Legal and data



Workflow

1. Review technical papers
2. Model Information
3. Performance Benchmark
4. Internal IBM use
5. Commercial Applicability
6. Licensing
7. Reputation
8. Use Case Alignment
9. Training Data
10. Infrastructure

Cost vs. performance considerations of LLMs

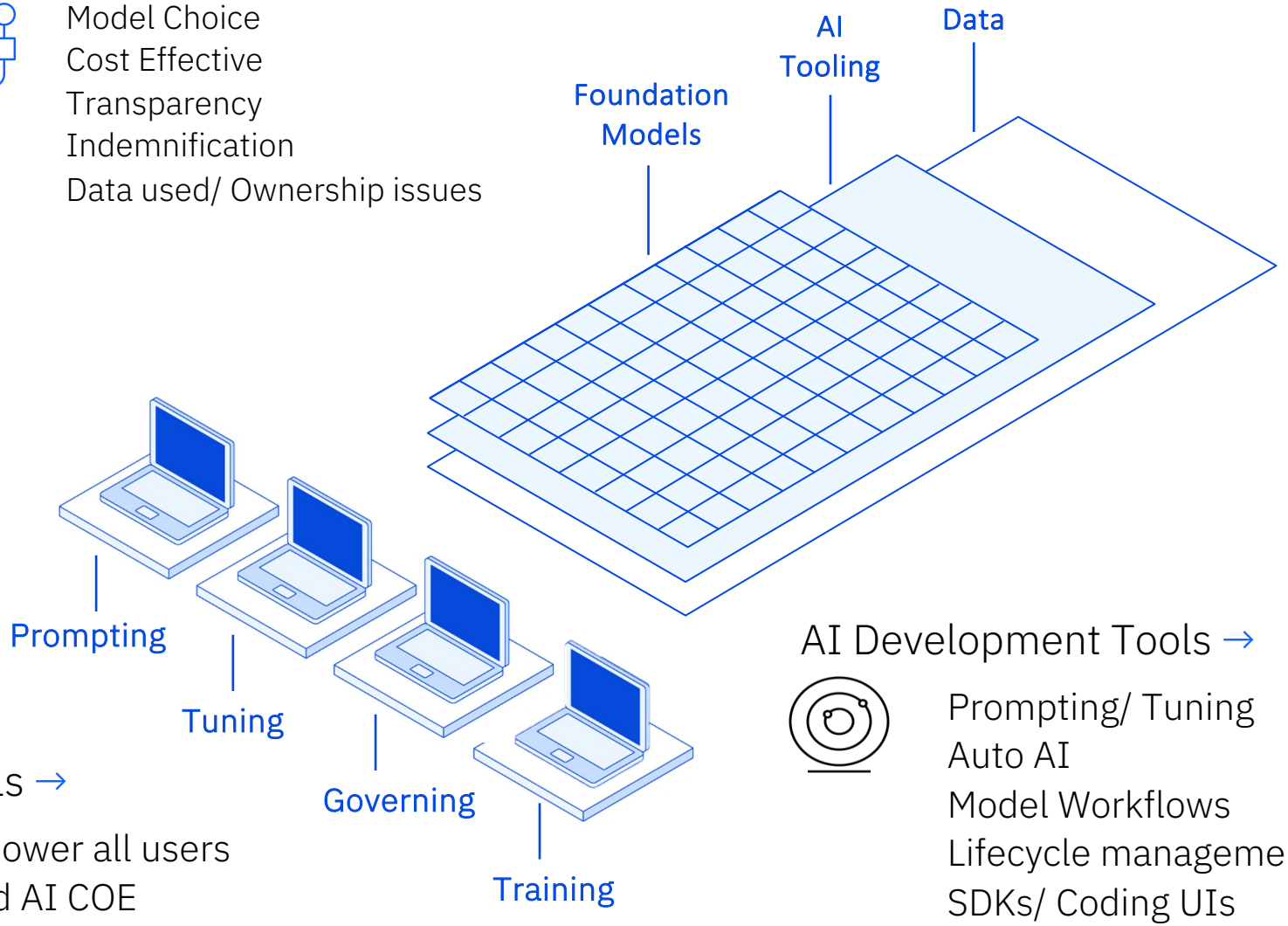


- Can you use a smaller model that produces similar results?
- Testing various task-specific, smaller ML models vs. 1 LLM
- LLMs that require more than 1 GPU
- Cost per inference or per hour
- Consider letting a vendor host the model for you vs. carrying costs
- Sharing the cost with multi-tenant

AI Factory/ Platform – Representative Functions

Pre-trained Foundation models →

Model Choice
Cost Effective
Transparency
Indemnification
Data used/ Ownership issues



Skills →

Empower all users
Build AI COE
Build Model Team

AI Development Tools →

Prompting/ Tuning
Auto AI
Model Workflows
Lifecycle management
SDKs/ Coding UIs
Deployment/ Monitoring

Platform capabilities →

Hybrid cloud
Privacy/ Security
Latency/throughput
Sustainability/ Scale
Flexible deployments
APIs

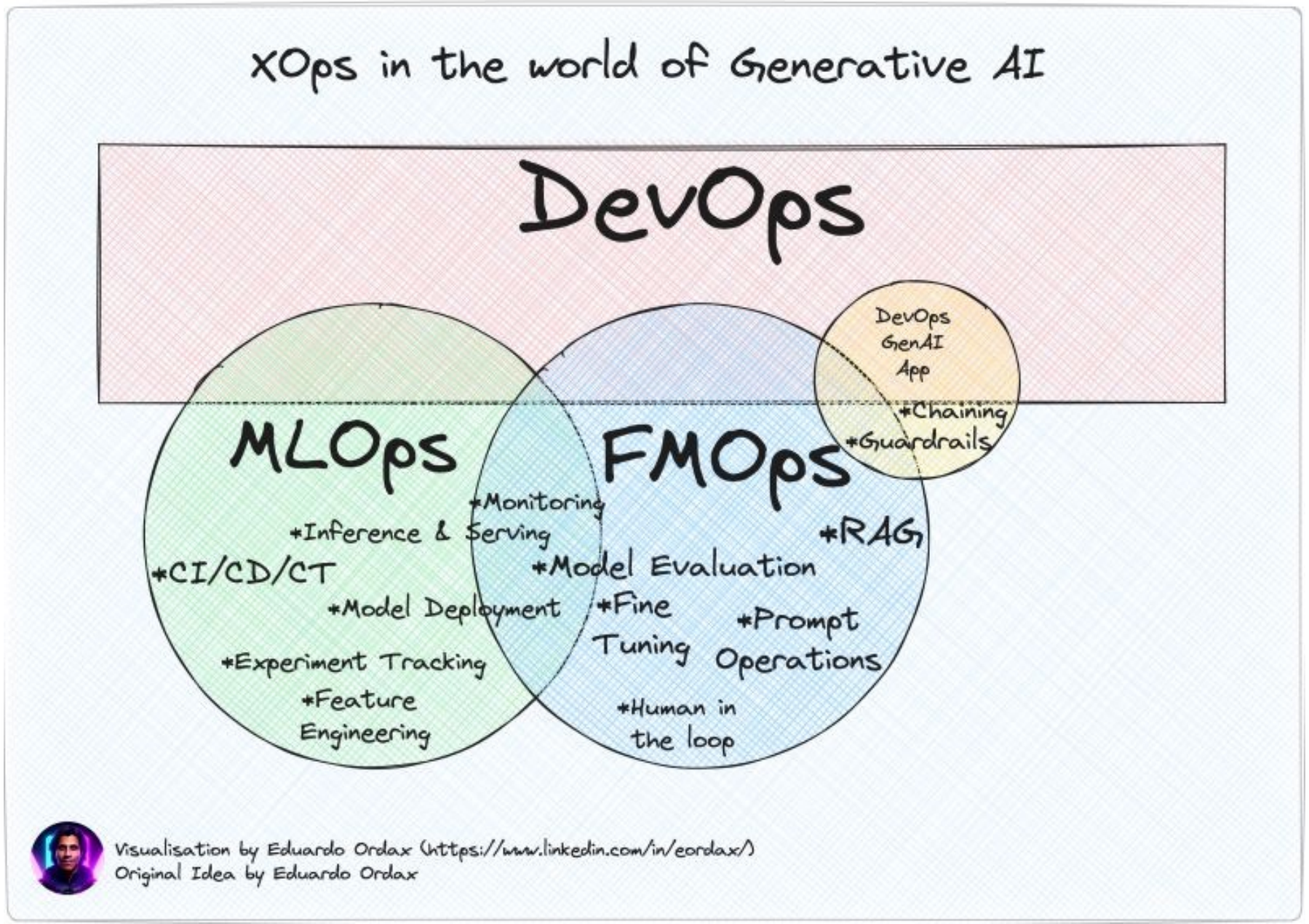
AI Governance Requirements →

Model Inventory for LLMs
Factsheets for LLMs
Evaluation Metrics for LLMs
Monitor LLM Health
Attribution for Q&A
Predictive ML & LLM Both

Data →

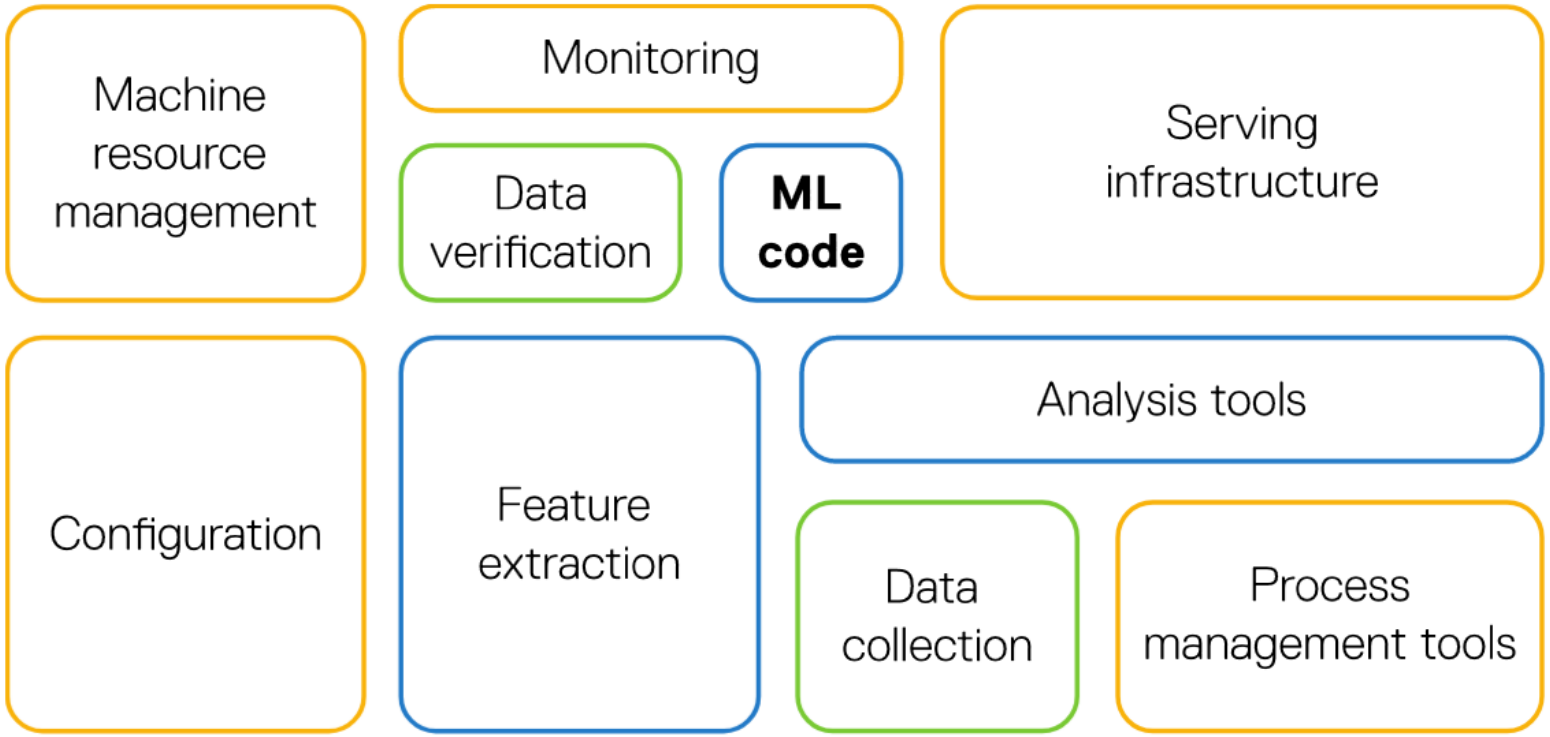
Leverage existing data investments
Build a data architecture to leverage AI
RAG
Synthetic Data

Automation,
Repeatability,
Scalability,
Stability



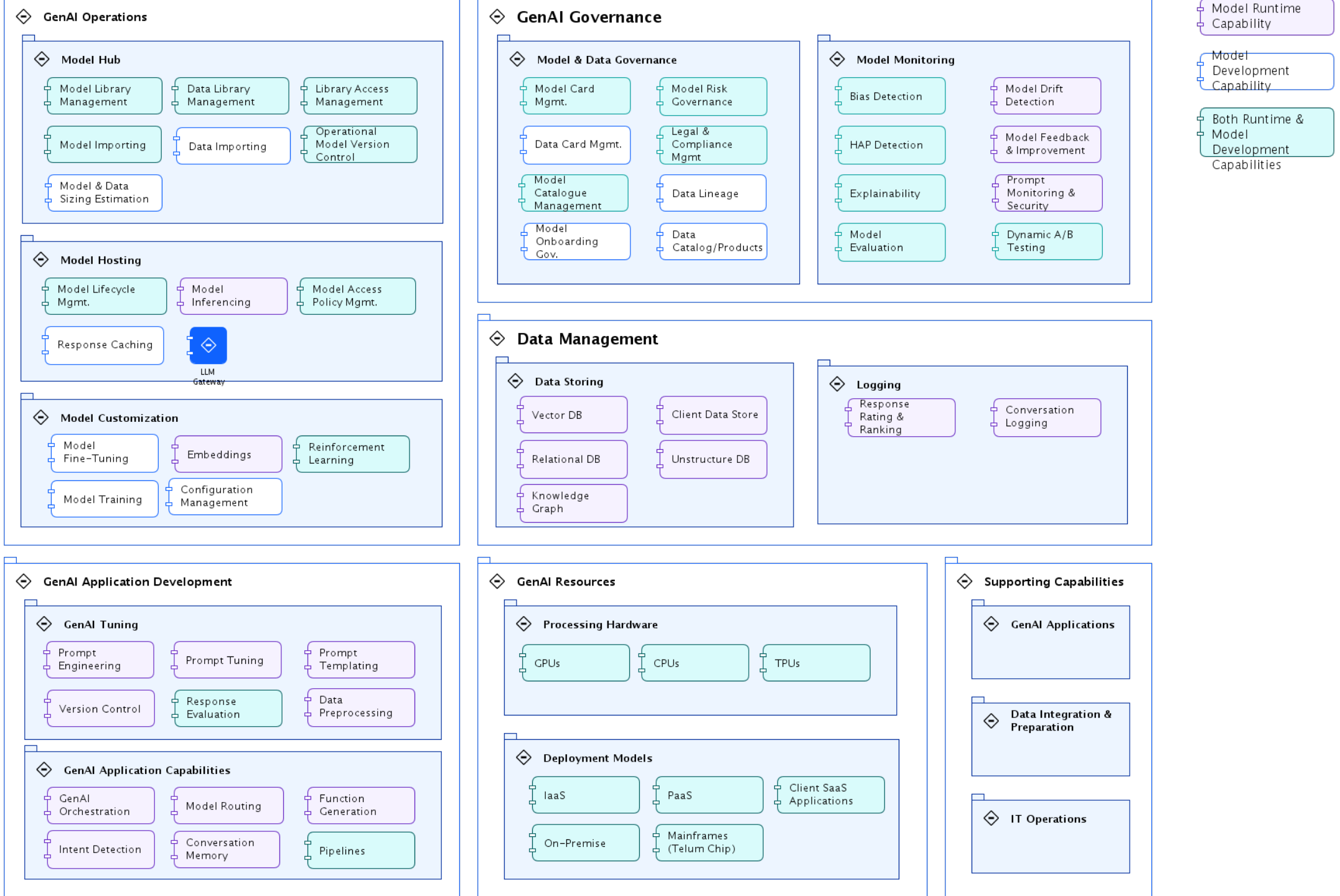
MLOps: DevOps Thought to ML Development

- Cloud Native: Containerized, Microservices, Immutability
- Open Source: Avoid Vendor Lock-In, State of the Art, Broad Community
- Scalable: Flexible with Training, Inferencing, and Hosting Needs
- Traceable: Audit and Lineage Tracking, Associate Assets Bundled
- Repeatable: Able to Replicate and Alter Dynamically



Dell MLOps Reference Architecture

FMOps


GenAI Operations
Model Hub
Model Library Management
Data Library Management
Library Access Management
Model Importing
Data Importing
Operational Model Version Control
Model & Data Sizing Estimation
Model Hosting
Model Lifecycle Mgmt.
Model Inferencing
Model Access Policy Mgmt.
Response Caching
LLM Gateway
Model Customization
Model Fine-Tuning
Embeddings
Reinforcement Learning
Model Training
Configuration Management
GenAI Governance
Model & Data Governance
Model Card Mgmt.
Model Risk Governance
Data Card Mgmt.
Legal & Compliance Mgmt
Model Catalogue Management
Data Lineage
Model Onboarding Gov.
Data Catalog/Products
Model Monitoring
Bias Detection
Model Drift Detection
HAP Detection
Model Feedback & Improvement
Explainability
Prompt Monitoring & Security
Model Evaluation
Dynamic A/B Testing
Data Management
Data Storing
Vector DB
Client Data Store
Relational DB
Unstructure DB
Knowledge Graph
Logging
Response Rating & Ranking
Conversation Logging
GenAI Application Development
GenAI Tuning
Prompt Engineering
Prompt Tuning
Prompt Templating
Version Control
Response Evaluation
Data Preprocessing
GenAI Application Capabilities
GenAI Orchestration
Model Routing
Function Generation
Intent Detection
Conversation Memory
Pipelines
GenAI Resources
Processing Hardware
GPUs
CPUs
TPUs
Deployment Models
IaaS
PaaS
Client SaaS Applications
On-Premise
Mainframes (Telum Chip)
Supporting Capabilities
GenAI Applications
Data Integration & Preparation
IT Operations
Model Runtime Capability
Model Development Capability
Both Runtime & Model Development Capabilities

ChatGPT Resources

GPT Parameters: *GPT-4 1-175 Trillion (Estimates), GPT-3 175 Billion*

GPT Conversation Memory: *GPT-4 64K Words, GPT-3 8K Words*

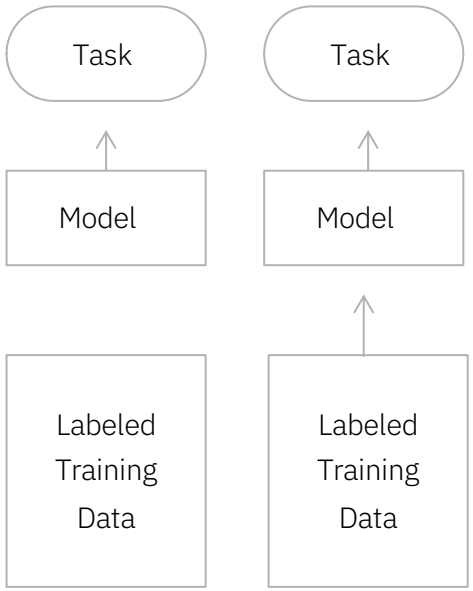
GPT RAM Requirement: *GPT-3 Small 16 GB System Memory 8 GB VRAM, GPT-3 175B 600 GB System Memory 250 GB VRAM*

Architecture model

Adapting Models with Data

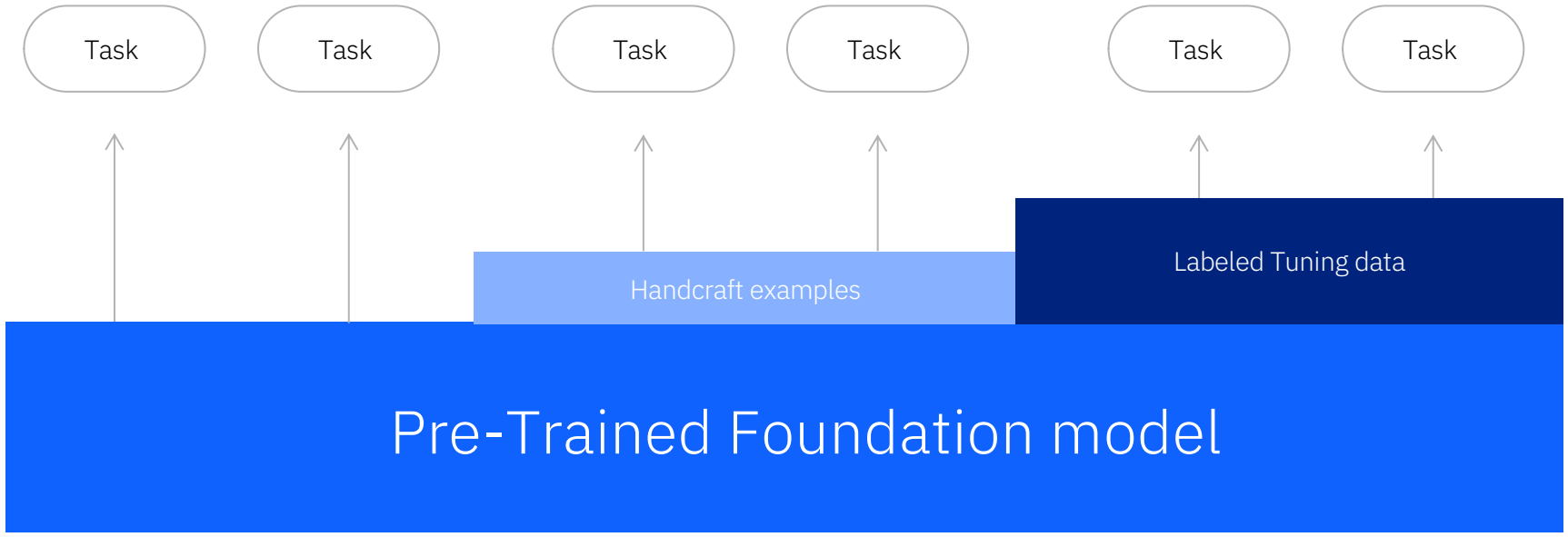
Traditional AI

1000s — 1000000s labeled data points / task



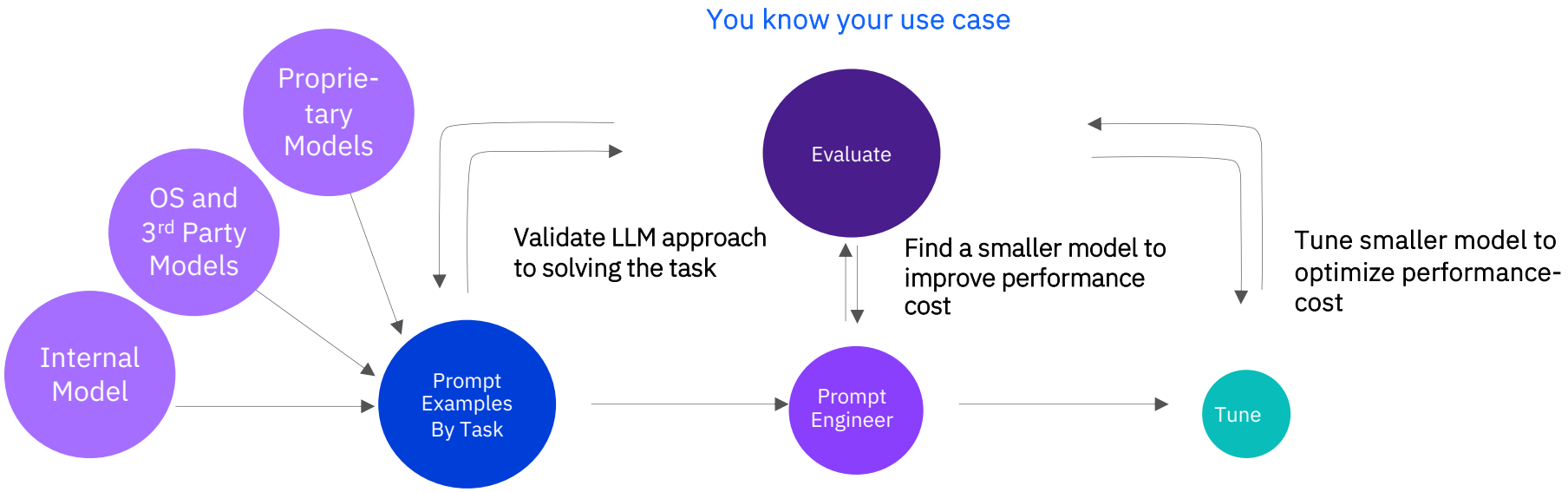
Generative AI

Zero-shot Prompting	Few-shot Prompting	Data-driven Tuning
No labeled data needed	1-10 labeled data points / task	100s - 1000s labeled data points / task



Supports both conventional NLP tasks and new human + AI, iterative workflows

Evaluation workflow



Proof-of-concept

Prove the use case using a large model with minimal labeled data. Evaluate.

Tuning

Use a medium-size model. Prompt engineer or prompt-tune with additional labeled data. Evaluate.

Selection

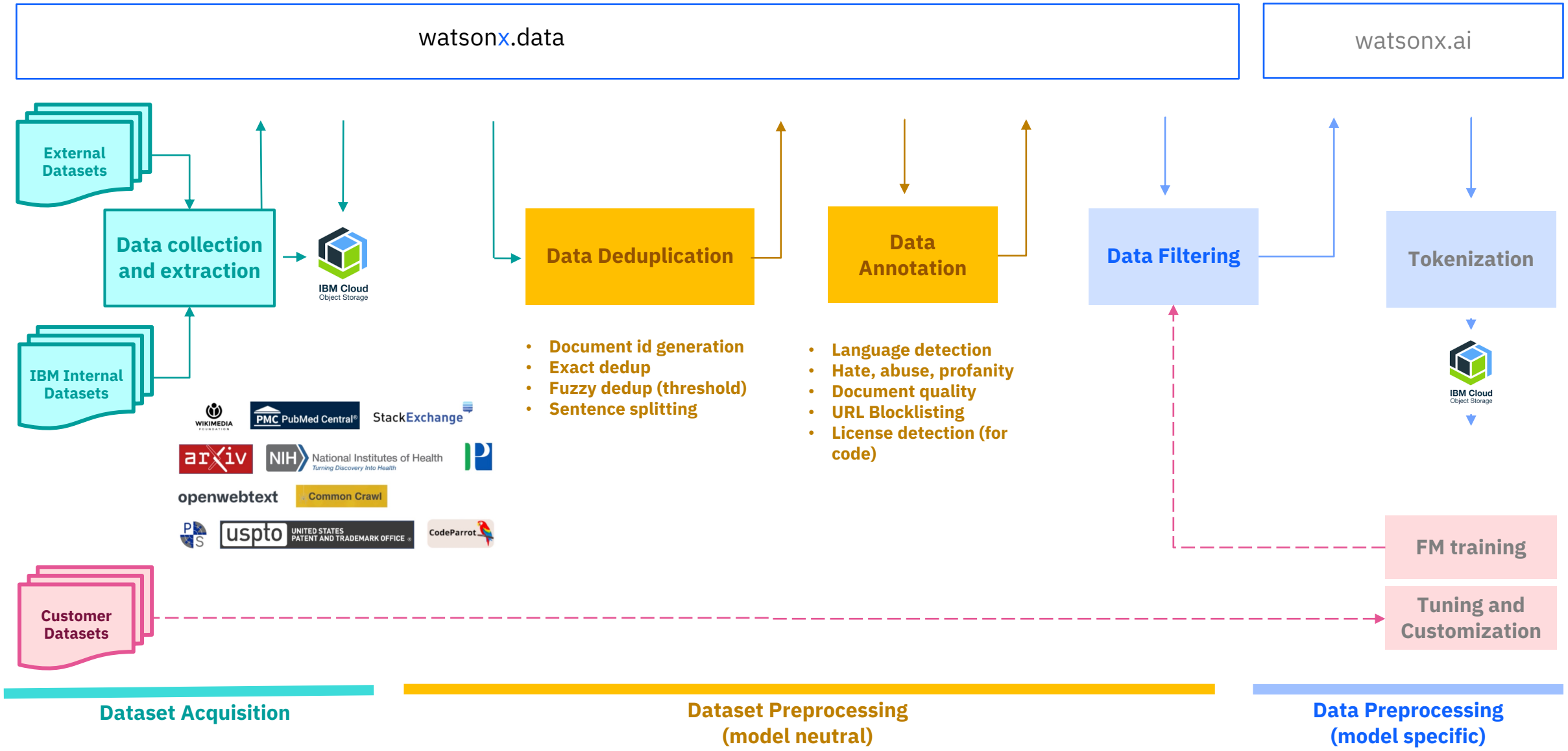
- Smallest model
- Additional data
- Fine-tune/prompt-tune
- Evaluate

Winning

Using the most accurate and cost-effective model

watsonx IBM Data Training Pipeline

Enterprise-ready data acquisition, curation, provenance, and governance



Two common issues with large language models

Lack of information source

“The bank offers 2.5% interest on accounts with a balance over $20,000.00.”

This sounds great – but where did the information come from?

How can a user verify that this is true?

Where is it documented?

Outdated information

“Who is the highest-scoring player in the NBA?”

The Llam2-70b model returns:

”Kareem Abdul-Jabbar is the highest-scoring player in the NBA”.

This is an outdated answer as Lebron James broke that record in 2023.

This means that llama2-70b was trained on pre-2023 data.

Retrieval augmented generation (RAG)

RAG addresses these issues:

- Where did the LLM get its answer?
- Is the answer based on updated material?

RAG does this by:

- Working with “external data” (data not used for training the LLM):
  - Source of answers? From curated, validated, and accurate data
  - Currency of data? As current as the source
- NO model retraining required

A “human interaction” analogy of RAG is providing an update document to a person and asking them to answer question based on the information in the document.

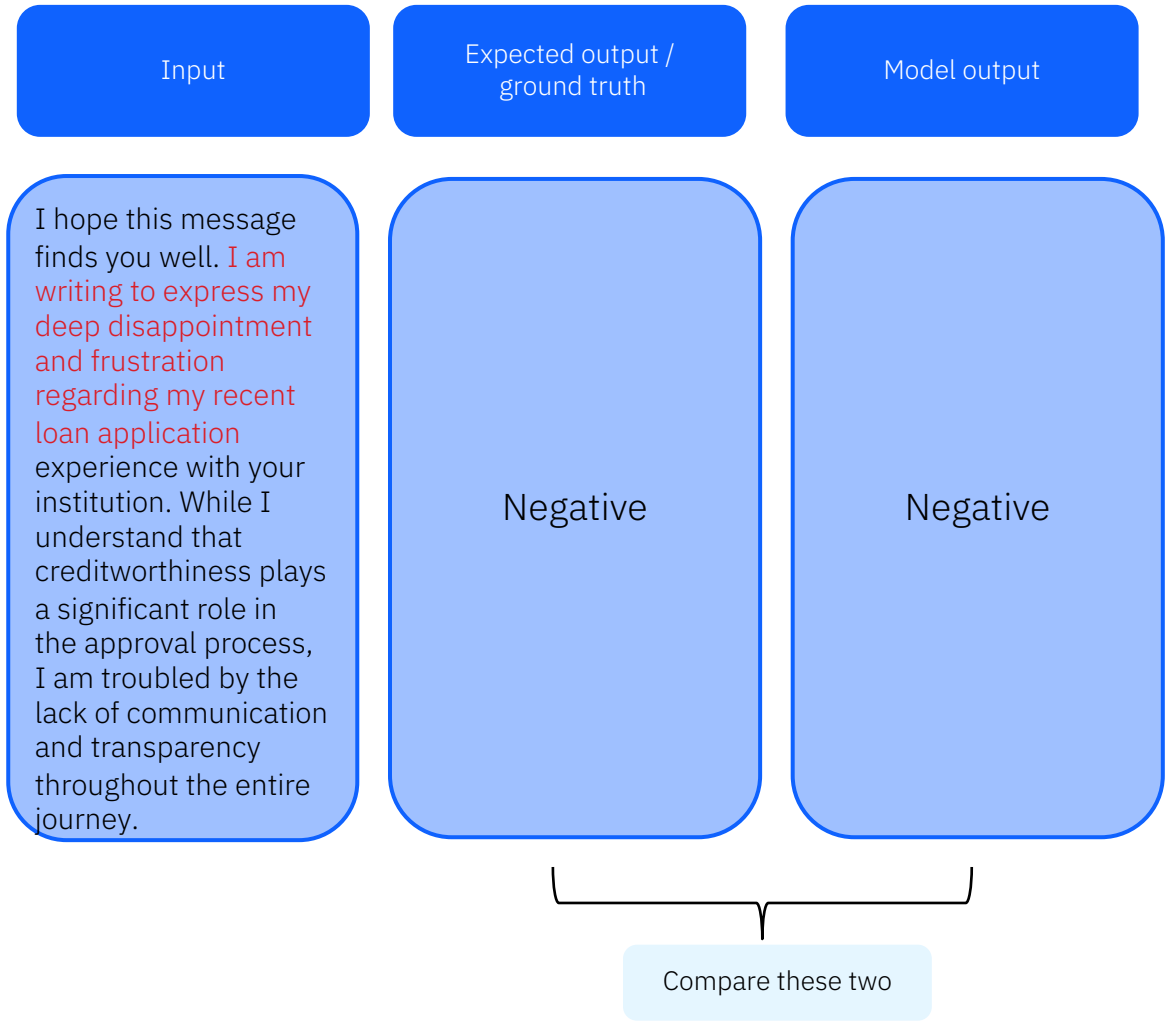
General process for model evaluation

1. Collect a list of test records
2. Gather expected output (ground truth)
3. Generate model output
4. Compare model output with the expected output
5. Use a metric to measure the overall performance

Freeform text

Examples of metrics - accuracy

Text classification use case



Read the customer review and classify the sentiment into **positive** or **negative**.

Model output = {“Negative”, “Negative”, “Negative”}

Expected output = {“Negative”, “Negative”, “Positive” }

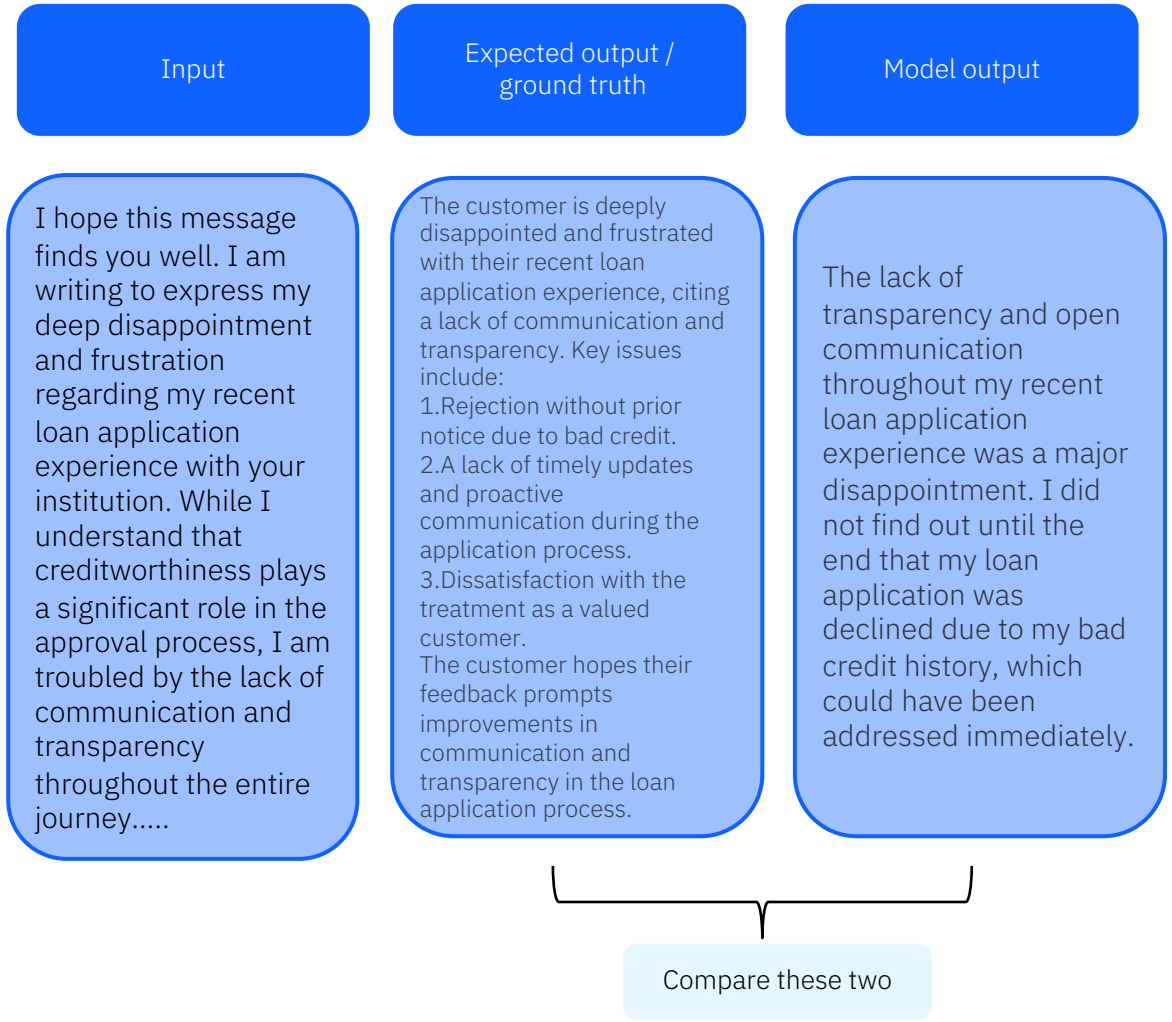
=> Accuracy = 2/3 = 66.67%

The accuracy metric is easy to understand and apply for outputs that are easy to compare as in classification.

Examples of metrics - rouge

Text summarization use case

Read the customer review and write a short summary.



Compare the Model Ouptut to the Ground Truth.

Rouge metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation.

Note that ROUGE is case insensitive, meaning that upper case letters are treated the same way as lower case letters.

$$Recall = \frac{\text{Number of matching n-grams}}{\text{Number of n-grams in the Reference}}$$

$$Precision = \frac{\text{Number of matching n-grams}}{\text{Number of n-grams in the Candidate}}$$

Other metrics commonly used

ROGUE

- Recall-Oriented Understudy for Gisting Evaluation
- Evaluate **summarization**
- Compare the output summary against a human-produced summary

BLEU

- Bilingual Evaluation Understudy
- Evaluate the **quality of translated** texts.
- Measure how close a model's output is to a set of good reference (human) translations.

Perplexity

- Evaluate **text generation**
- Evaluate the probabilities assigned to the next word by the model.
- Lower perplexity indicates better performance

F1

- Evaluate **question answering**
- The harmonic mean of the precision and recall value (both measure how frequently a model correctly identifies a true positive)

Metrics for evaluating Large Language Models

Text Summarization Metrics

- ROUGE
- SARI
- WIKI_SPLIT
- BLEURT
- METEOR
- Sentence Similarity - Jaccard Similarity
- Sentence Similarity - Cosine Similarity

Content Generation, Q&A Evaluation Metrics

- BLEU
- exact_match
- Perplexity**
- rl_reliability**

Text Classification Metrics

- Accuracy
- Precision
- Recall
- ROC AUC
- F1 Score
- Brier Score
- GLUE metrics
- Matthews Correlation Coefficient
- Label Skew

NLP Encoding Model

- HAP Detection**
- PII Detection**

** Uses Metrics Computation Model

Entity Extraction Metrics

- Seq eval

Language Translation Evaluation Metrics

- Character
- charcut_mt
- Chrf
- google_bleu
- super_glue
- TER
- nist_mt
- Poseval
- sacrebleu
- XTREME-S

Reference-Free Metrics

- Levenshtein distance based Diversity metrics
- Textstat toolkit based flesch metrics to determine readability, complexity, and grade level.
- blanchelp**
- Shannon**

Active Research Domains

- Stigma Detection**
- Social Bias/Values Detection**
- Faithfulness / Hallucination**

Governance necessities

Monitor and evaluate

- Monitor predictive models for fairness, accuracy, and drift
- Monitor generative models for PII and HAP, with additional monitors coming soon
- Explain model predictions and output

Track facts and metrics

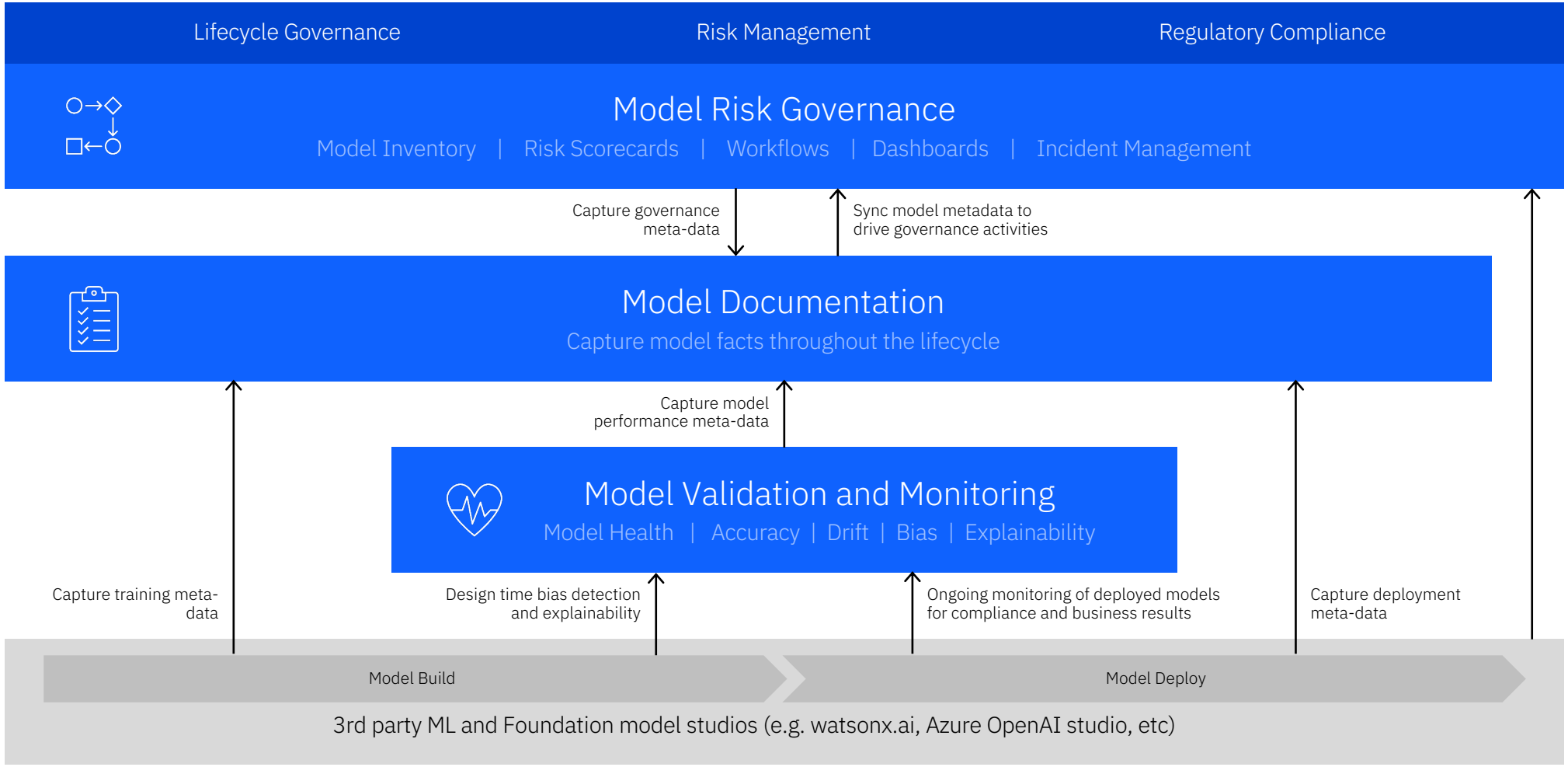
- Automatically gather model metrics and metadata
- Provide model information in a fully-managed, searchable catalog
- Track models throughout the entire lifecycle

Manage lifecycle and risk

- Fully customize model approval workflows, from initial request to production deployment
- Track risk for all models across the enterprise
- Configure dashboards and reporting for model performance

Governance Operational View

- Model Owners
- Model Validators
- Audit Teams
- Compliance Teams
- Risk Management Teams
- Data Privacy Teams
- Principal Data Scientists



- Data Engineers
- (Citizen) Data Scientists
- MLOps & LLMOps
- ML & LLM Engineer

Interesting Resources

LangChain - framework for developing applications powered by large language models (LLMs).

Ollama - run open-source large language models (LLMs) locally on your machine.

Ray – unified computing framework for training and scale.

CodeFlare - an open-source platform for AI and ML development.

Awesome-LLM – git repo of assorted LLM items.

Trusted-AI – git repo for AI fairness, explainability, adversarial robustness, and other trust/privacy bits.

minikube – local k8s cluster on mac, Linux,win.

watsonx.ai – IBM AI platform

Thank You