Lessons Learnt in **Productionization**

Venkata Pingali,
Co-Founder/ CEO Scribble Data

www.scribbledata.io

Scribble Data:

An LLM /ML Data product company since 2019

TORONTO NEW YORK BANGALORE

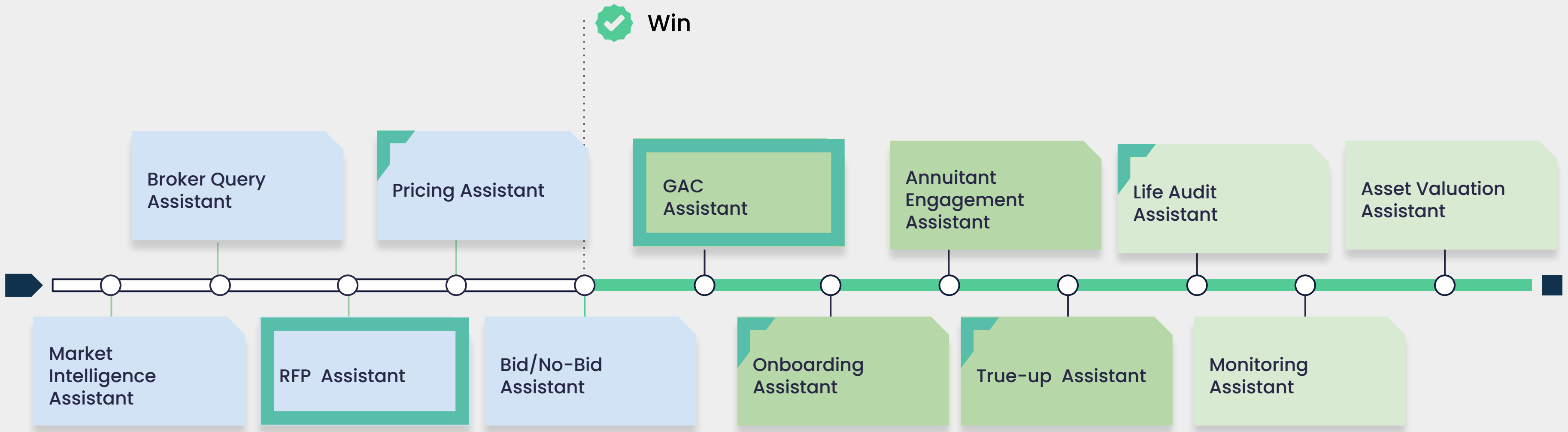
Focus	TECHNOLOGY: AI Platform Enablement INDUSTRY: Insurance/PRT/Benefits
Mission	We are committed to catalyzing a transformative journey for insurers, empowering them to thrive in the face of change.
Global Experience	US EU AFRICA INDIA | Insurance Fintech Ecommerce CPG Retail
Secure and Robust	SOC2 and GDPR Compliant

PRT Value Chain

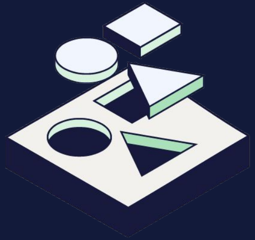
Plan Sponsor	Advisor	Insurer	Admin
Private company decides it needs to **de-risk** itself of its pension obligations	Structures and facilitates the transaction	Prices the risk and takes on the pension assets and liabilities	Handles payroll, true-ups, reconciliations. In-house (insurer) or third party
	BCG PENSION RISK CONSULTANTS PENBRIDGE	MassMutual	alight
	Milliman	AMERICAN NATIONAL INSURANCE	BUCK
	Mercer	ATHENE	Principal
	AON	Prudential	CONDUENT

scribble Data

AI Assistants



* Throughout 30 year plan administration phase



How Did We Get to **2024**

Evolution of Scribble and Domain

< 2017

Google/Facebook/Others

Mainly internal platforms, not publicly discussed, Some early platforms

2017

Uber's Michelangelo

Broad platform including Palette, Uber feature store

2019

V2 Implementations

Tecton, Feast, Hopsworks, Scribble, Splice, Kaskada, Molecula

2021

Hype & Correction

Everybody wants to build ML Platform

2023

V3 Start

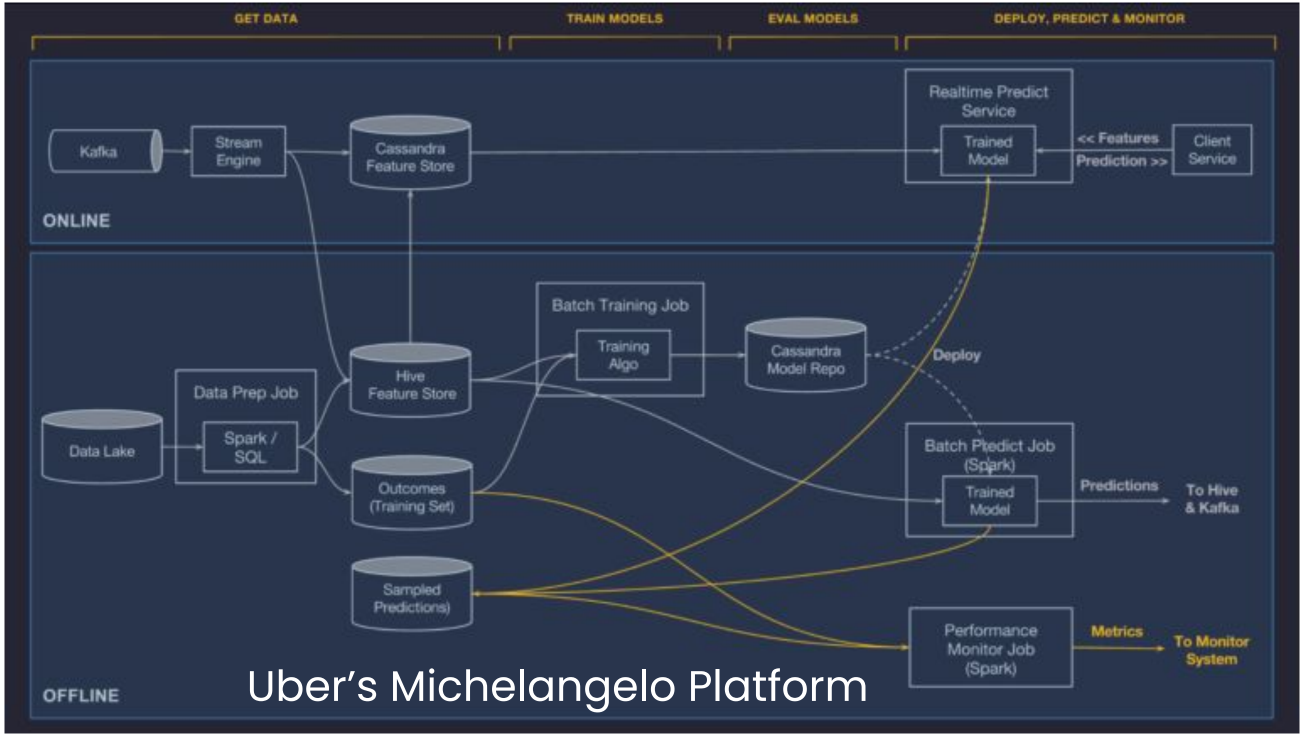
ML cooling off, consolidation of platforms, Shrinking teams, ChatGPT

2025

AI Applications

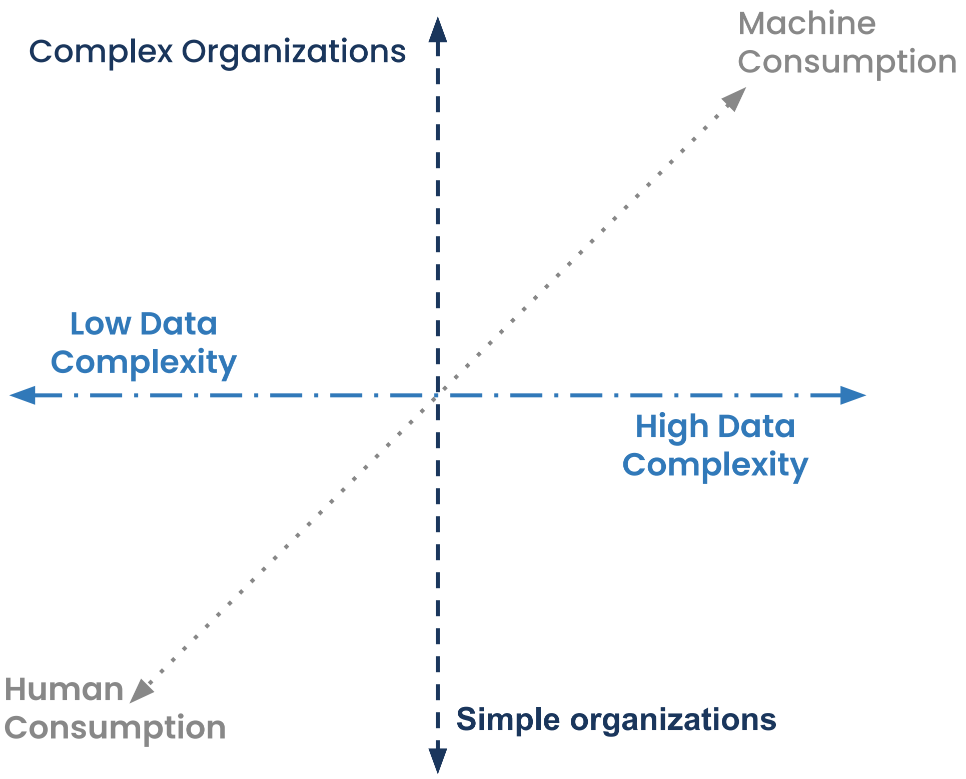
DL models, explosion of document processing

Original Sin: Projection

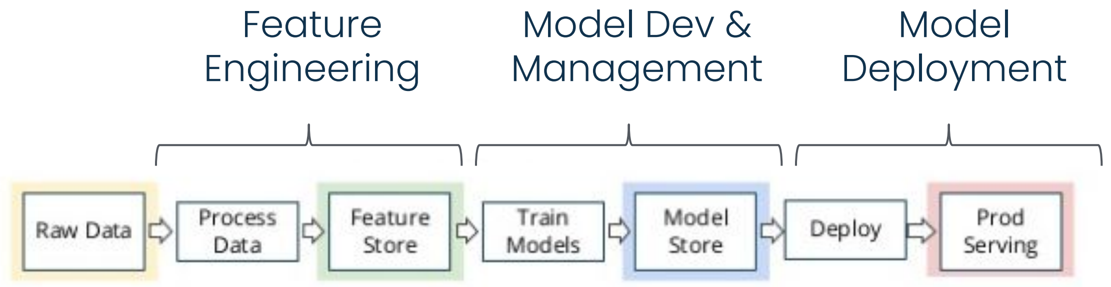


https://eng.uber.com/michelangelo-machine-learning-platform/

Large Design Space - No One-size Fits All



Simple Linear Flows are Misleading

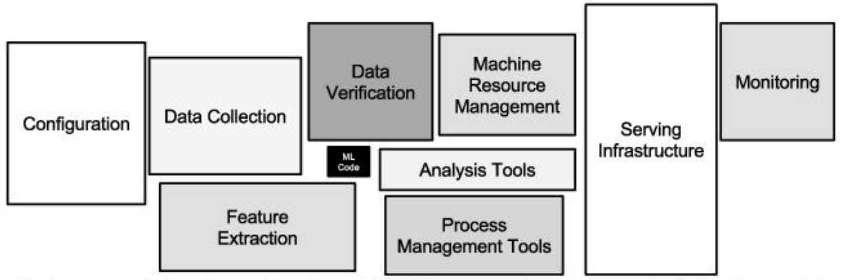


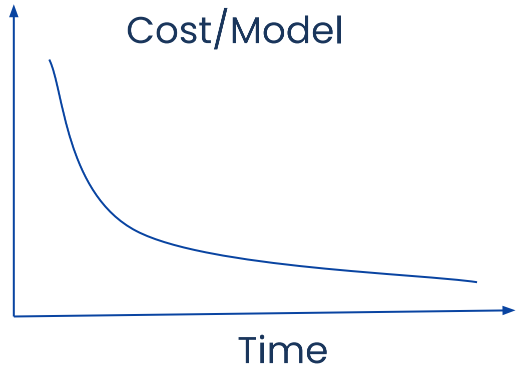
Simple but wrong model from Gojek, 2019

https://databricks.com/session/scaling-ride-hailing-with-machine-learning-on-mlflow

It is Complex and Iterative Process

- Expensive, error prone activity
- Complex implementations
- Iterative and evolutionary
- Growing need
- High impact on correctness

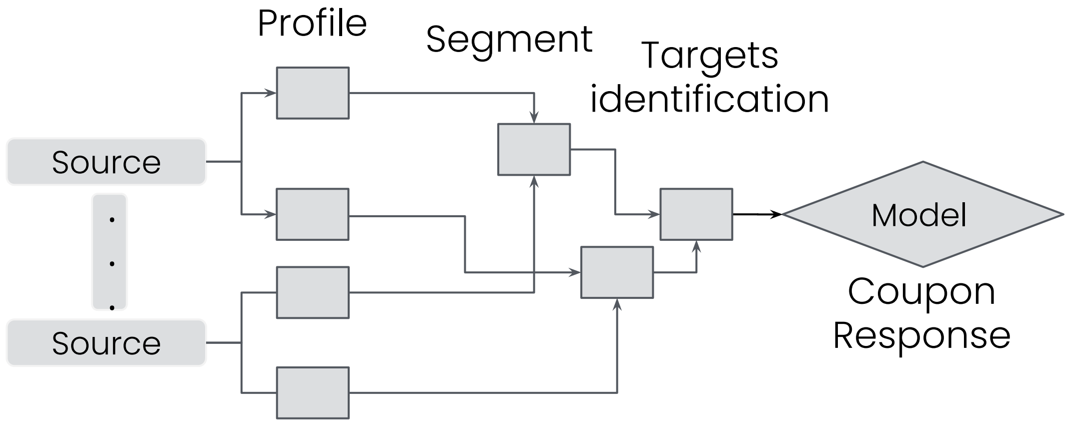




https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

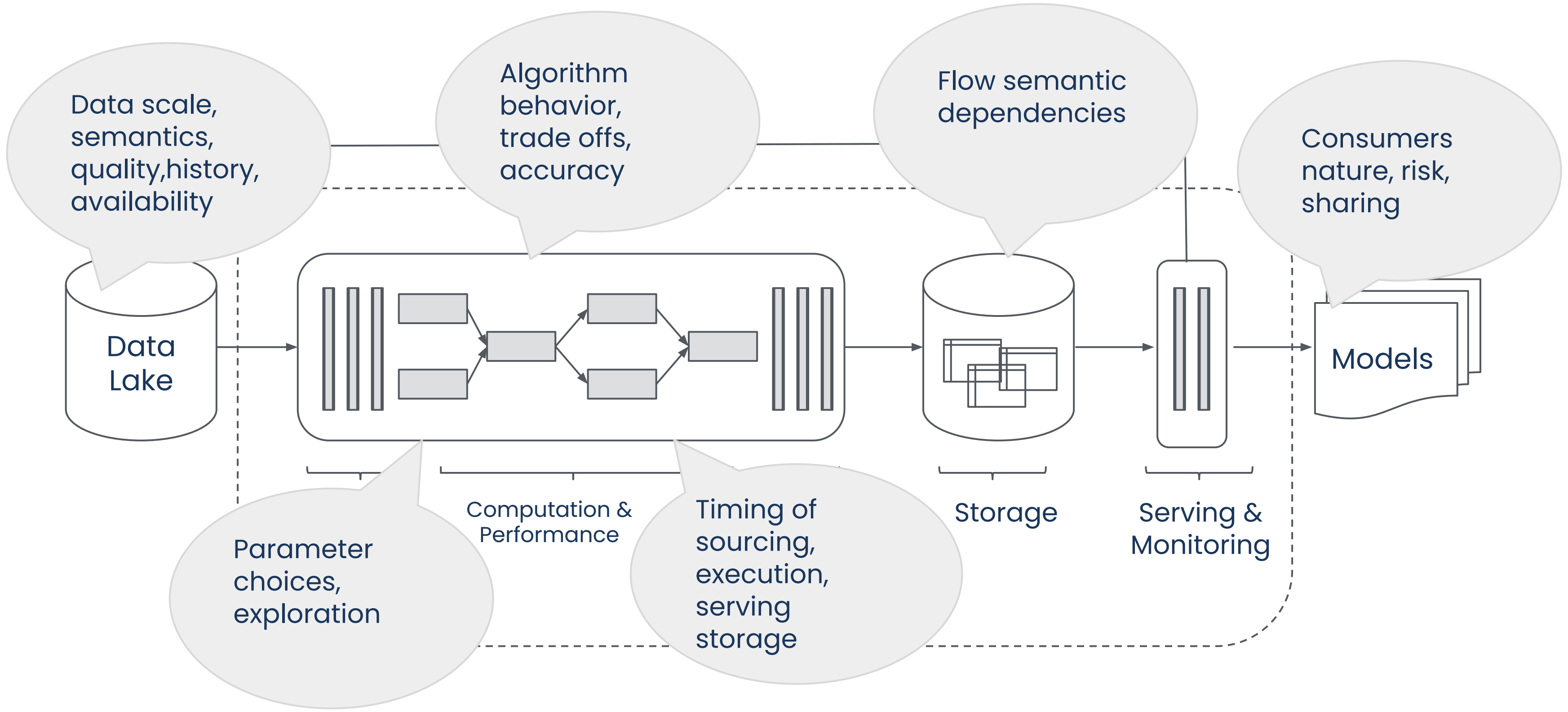
Example: Coupon Delivery Pipelines

- Can be broad and deep
- Continuous change
- Compute intensive
- Long execution times
- High volume of data
- Txns, emails, reviews



Sample: 50GB/day, 2M customers, 200 features, 10 pipelines, 4 hours execution time

Variations Possible at Every Step


Data scale, semantics, quality,history, availability
Algorithm behavior, trade offs, accuracy
Flow semantic dependencies
Consumers nature, risk, sharing
Data Lake
Models
Parameter choices, exploration
Computation & Performance
Timing of sourcing, execution, serving storage
Storage
Serving & Monitoring

Lessons: Trust is Table Stakes

- Can we bet a business KPI on the system?
  - Accuracy, reliability, security, safety, legality
- Our learnings:
  - Trust is end-to-end
  - High degree of control (limited use of auto*)
  - Deep semantic checks
  - Auditability/GDPR

Nature	Offering	Function
Extraction correctness	Galileo	AI observability
Data Drift/ Observability	Arize, Abacus AI	Notice structural changes and act
Data Quality/ Correctness	Monte Carlo, SODA	Expectations
Data Leaks/ Poisoning	Tecton	Point in time values
Reproducibility/ Deep Metadata	Scribble	Deep instrumentation
Compliance	Konfer	Validate process

Lessons: Efficiency Required for Effective Delivery

- What is the time to value?
- Stakeholder buying required through lifecycle
- 10x time in RCA than dev
- Lessons learnt
  - More usecases than bandwidth available
  - Reproducibility critical for RCA, checkpointing is plus
  - Dense code, deep auditability required
  - Conditional execution of transforms

Nature	Offering	Function
Frameworks	LangChain	Application structure
Development	Feast	SQL-like*
Solutioning	Scribble	Full-stack
Realtime	Tecton	Abstractions
Integration	Clouds	Platform
Error Propagation	Gantry	Understand dependencies
Knowledge	Molecula	Domain focus

Lessons: Flexibility Required for Value

- Will it fit into my context?
- Incorporate newer information
- Complexity due to org, tech stack, usecases, change
- Scale, cost, flexibility - Choose two
- Lessons learnt
  - Fewer assumptions about access, skill etc
  - Support multiple customers/ products/ workflows

Nature	**Offering**	**Function**
Low-level tooling	Opensource	Customize to need
Fast Deployment	Vercel	Speed to experience value
Distributed development	Featureform	Reduce friction to dev. Unified abstraction
Data Products	Full-stack solutions	AI Engg, AI call center agent

Scribble's Own Approach

Chosen Scope of Problems

- Regulated, mid-large fintech experience (payments, insurance etc)
  - High risk/low **trust**

- **Internal facing** use cases
  - Cooperative users (not adversarial)
  - Highly skilled individuals

- Controlled automation, **full-stack**
  - Human-supervision
  - **Incremental** integration into existing workflows
  - Short time to deploy

- Small to mid volume
  - Extraction, computational contracts, occasional ML

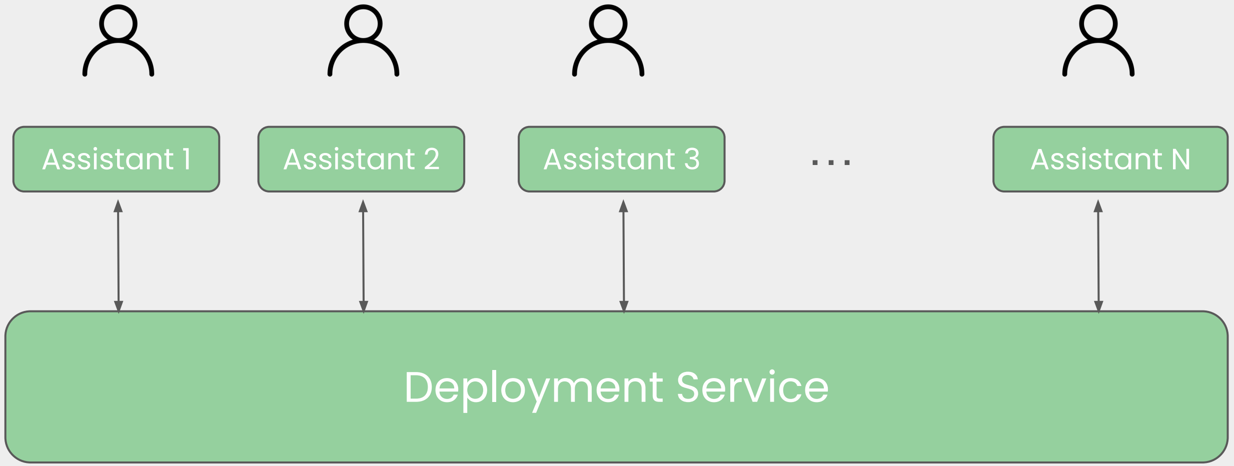
Chosen Approach

- Focus on **speed**
  - Simplify, simplify, simplify
  - Flexible lego-like system to rapidly assemble apps
  - Rapid recovery and rollout/rollback/backfill

- **Full-stack**
  - No dependencies and therefore speed
  - Integrations to be part of workflow

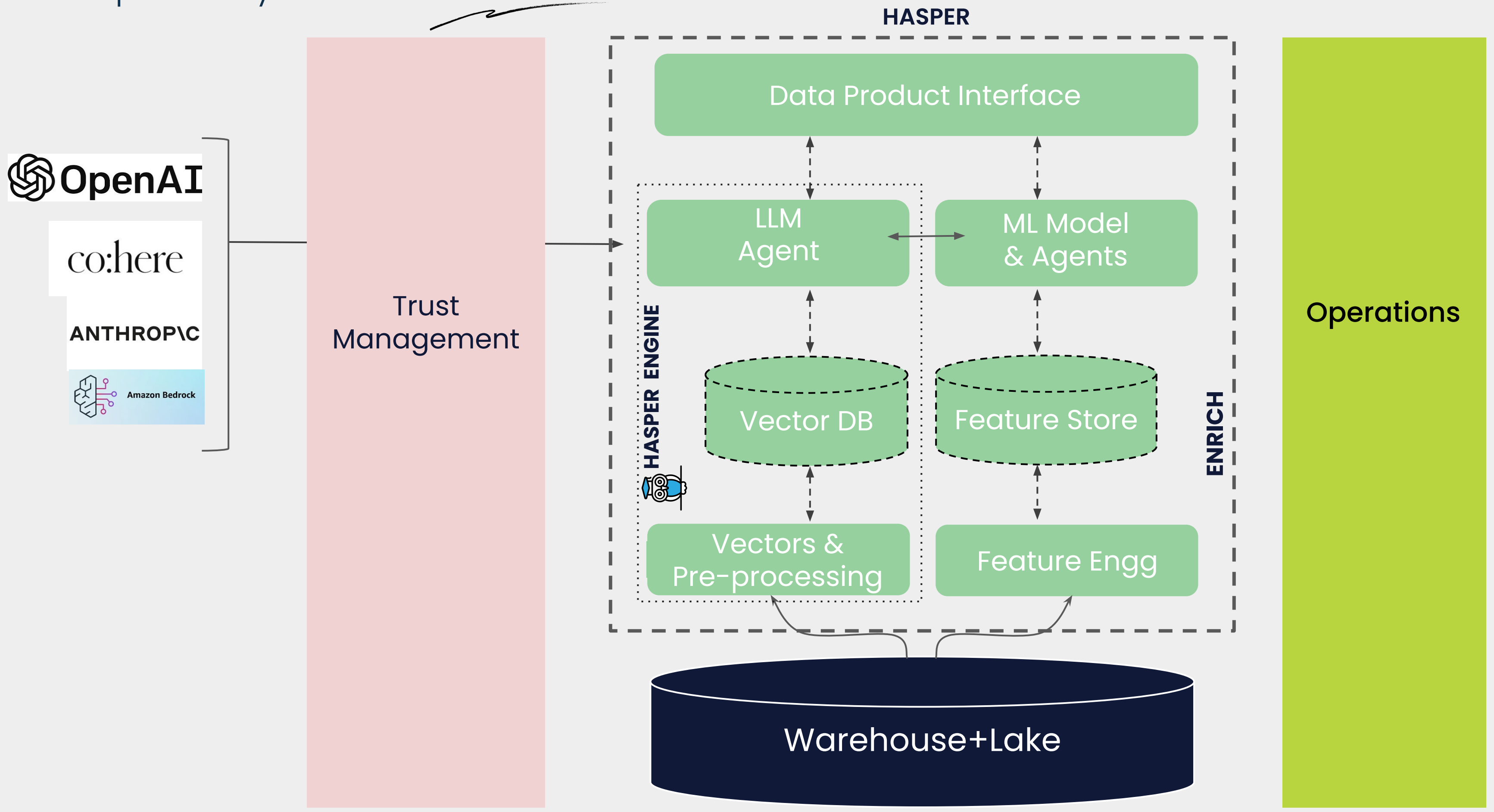
- Deep auditability and reproducibility
  - Audit every step of the way

Assistants are Task-Specific

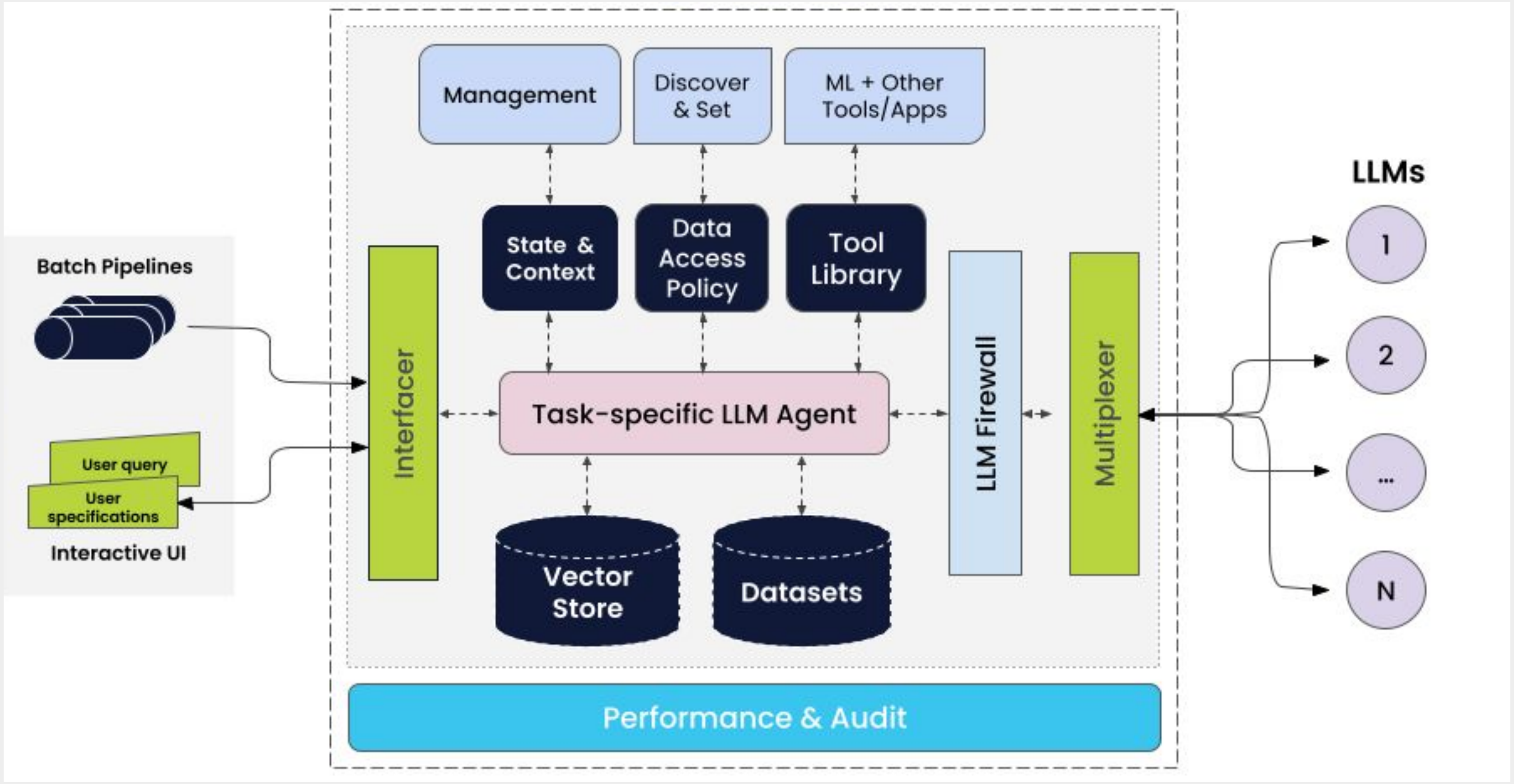


- Task specificity required for multiple reasons - adoption, feasibility, cost etc
- Performance has to cross a threshold for adoption
    - Accuracy, timing, auditability, completeness etc
    - Content and aesthetics are both important
- High value tasks are also tricky - deep domain knowledge, complex experiences etc

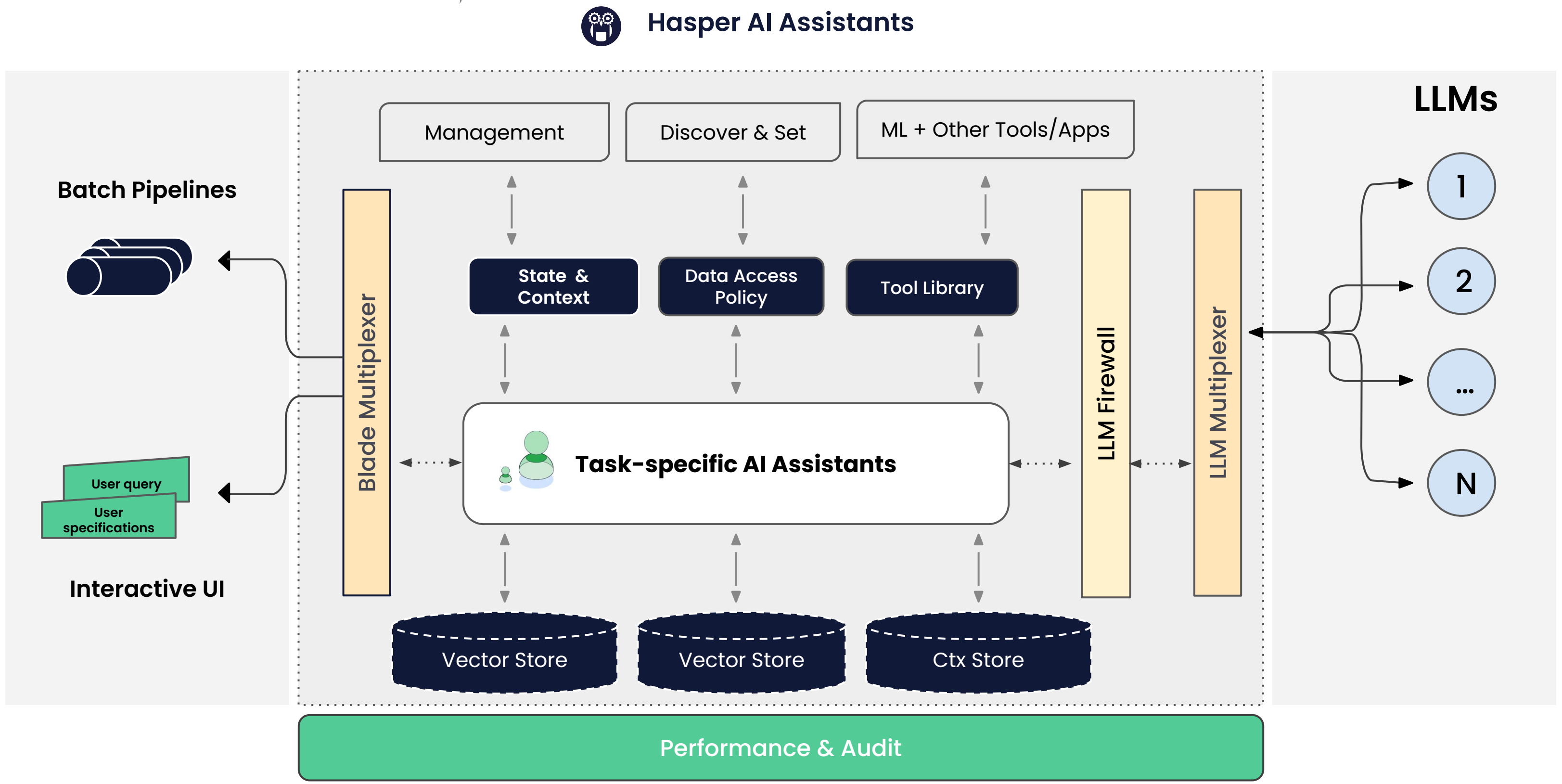
Hasper System Architecture



Hasper Task Blade



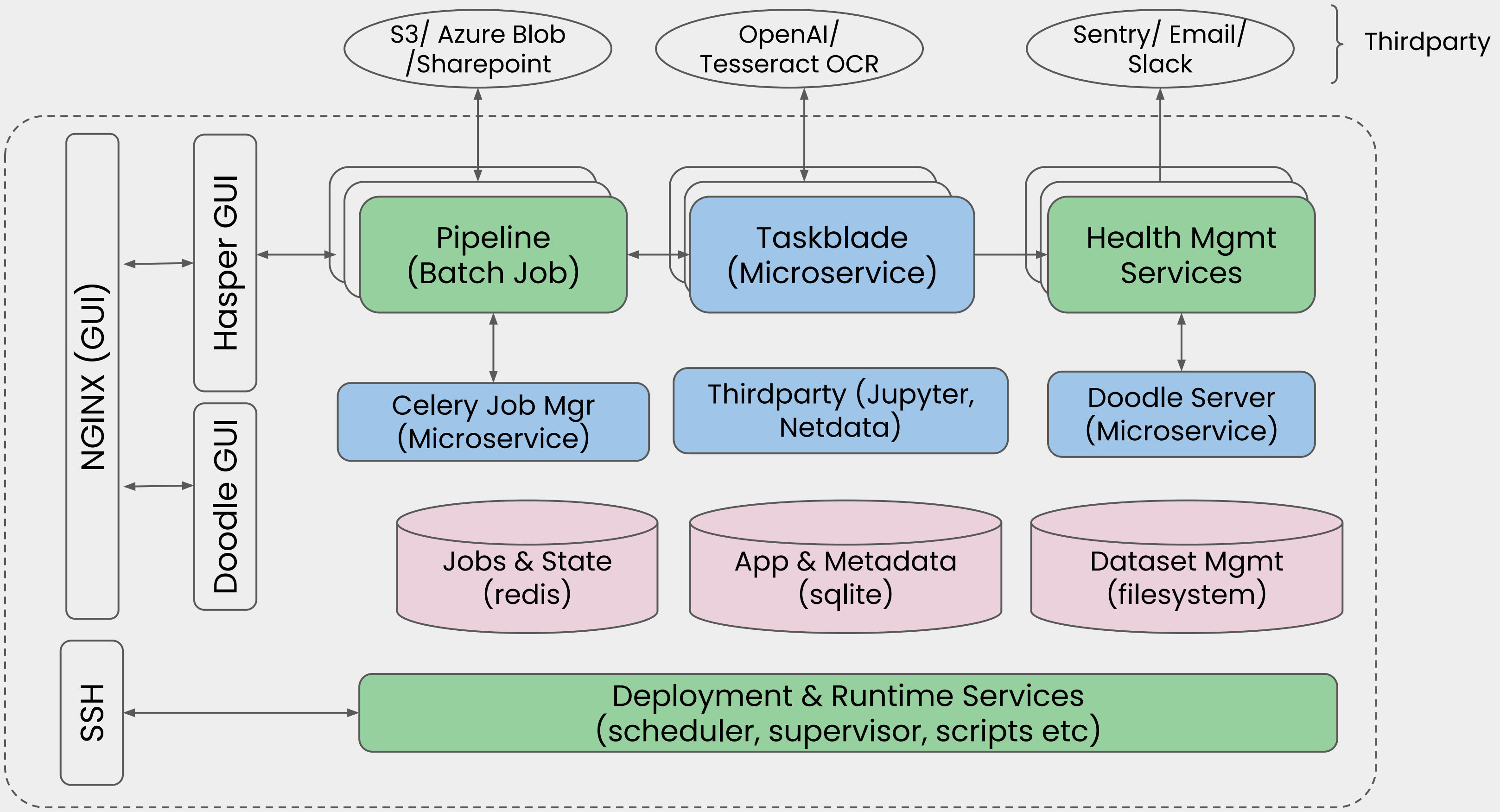
Hasper Task Blade



Takeaways from Our Experience

- Agents/Assistants' economics is real. Here to stay
- Shelf-life of any idea/system is short and shrinking
  - Fast build, fast evolution. Keep it simple
  - Build composable/lego-like system
- Decision makers and users are increasingly non-technical
  - Be usecase and value driven - Avoid trap of MLOps/MDS
  - Should be usable/valuable even with partial answers
- Collaboration with users is crucial
  - You can simplify/reframe hard problem
- Complex dynamics because jobs/reputation is involved
- Applications structure: No pure agents
  - 20% of work is LLM, 80% is software
  - Hallucination is real, not going away
  - Check for integrity/accuracy every step of the way
  - More DAGs than agentic world

Hasper System Components


S3/ Azure Blob /Sharepoint
OpenAI/ Tesseract OCR
Sentry/ Email/ Slack
Thirdparty
NGINX (GUI)
Hasper GUI
Doodle GUI
Pipeline (Batch Job)
Taskblade (Microservice)
Health Mgmt Services
Celery Job Mgr (Microservice)
Thirdparty (Jupyter, Netdata)
Doodle Server (Microservice)
Jobs & State (redis)
App & Metadata (sqlite)
Dataset Mgmt (filesystem)
SSH
Deployment & Runtime Services (scheduler, supervisor, scripts etc)

Thank you!

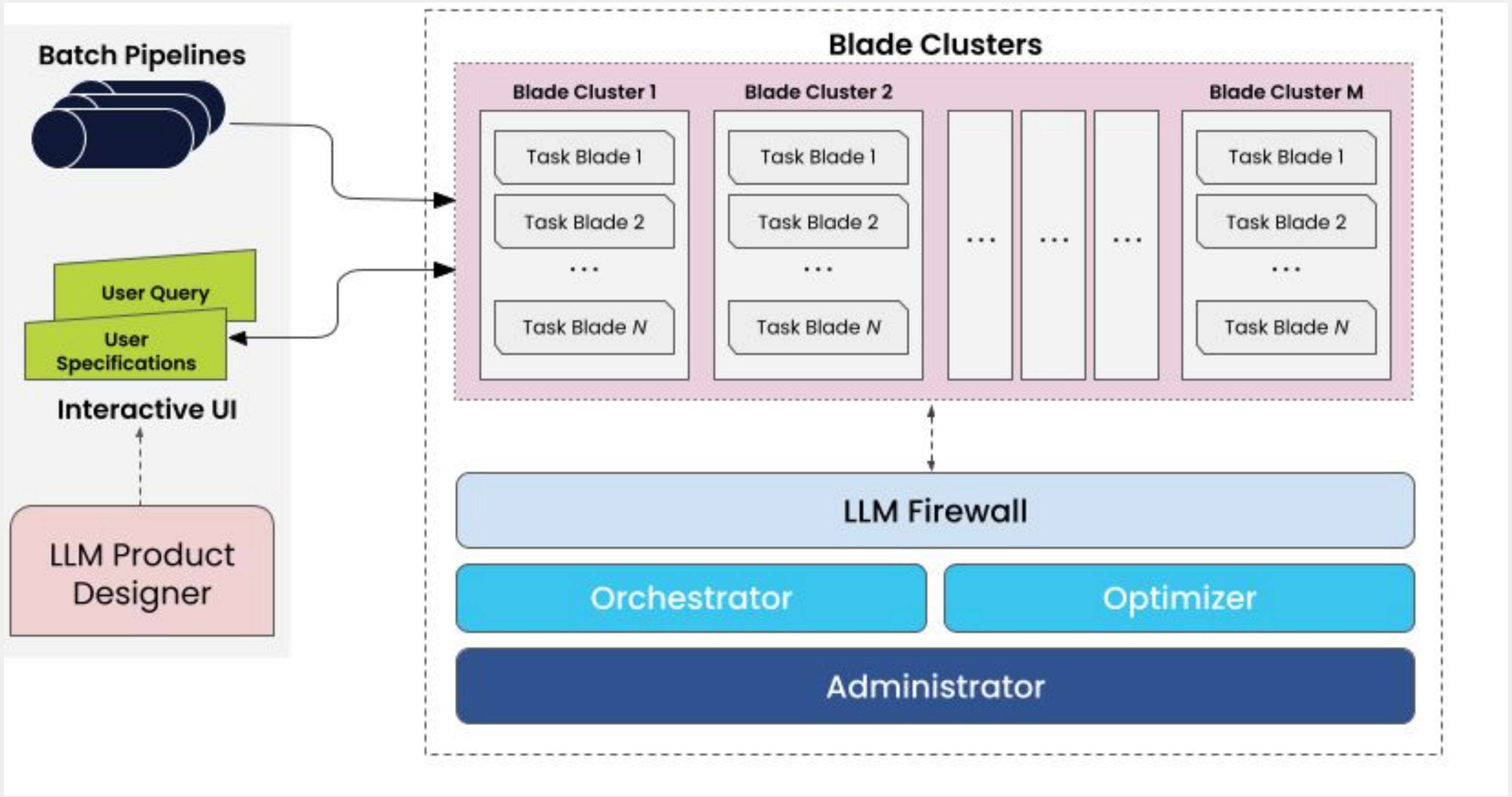
I am Venkata Pingali

pingali@scribbledata.io

www.scribbledata.io

New York Toronto Bangalore

Hasper System Architecture


Batch Pipelines
User Query
User Specifications
Interactive UI
LLM Product Designer
Blade Clusters
Blade Cluster 1
Blade Cluster 2
Blade Cluster M
Task Blade 1
Task Blade 2
...
Task Blade N
LLM Firewall
Orchestrator
Optimizer
Administrator