BUG TRIAGE AUTOMATION USING NLP

THE PROBLEM: MANUAL BUG TRIAGE

- Requires human effort, time, and domain expertise
- Inconsistent classifications
- Not scalable for large-scale projects

WHY NLP ?

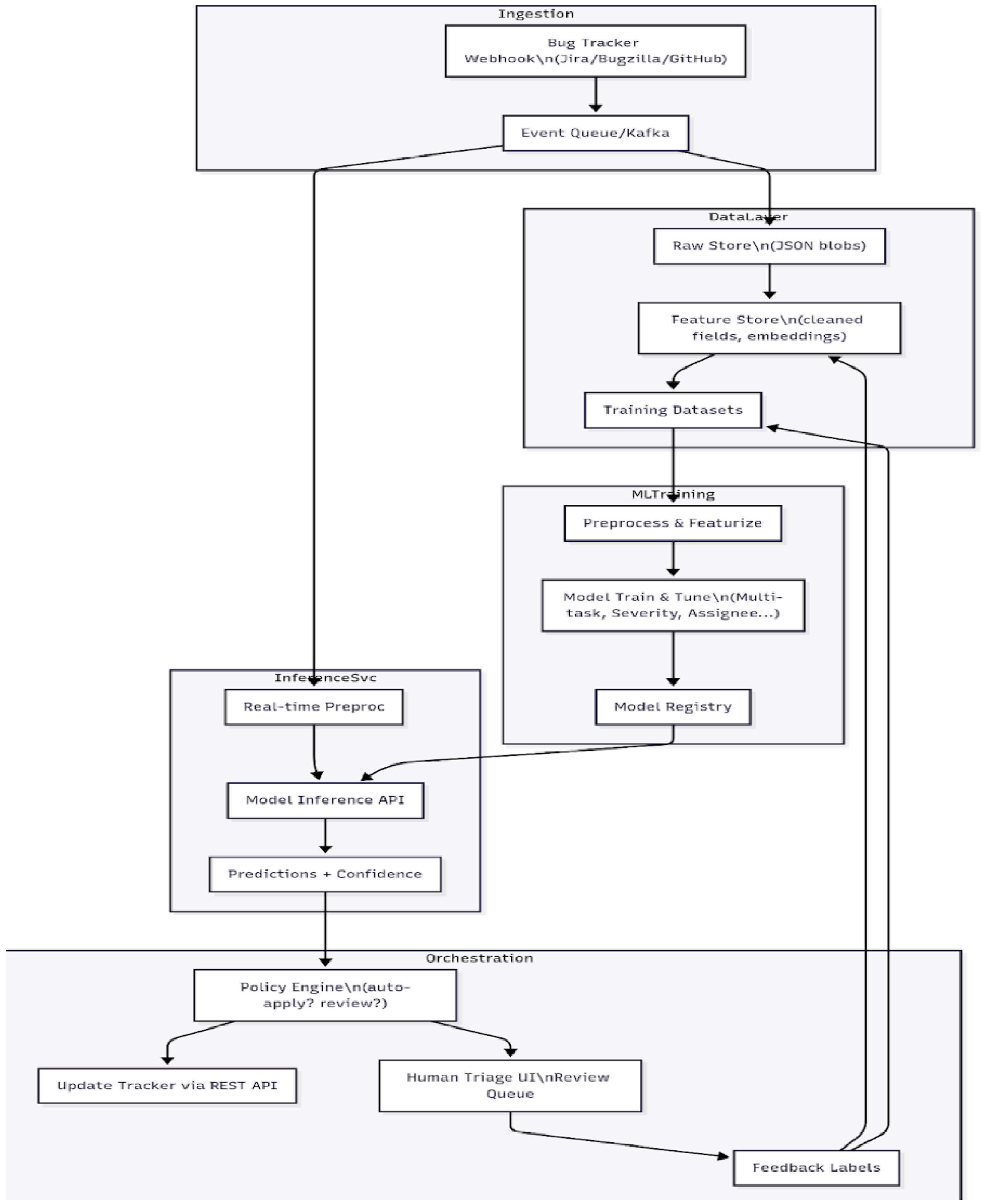
- Bug reports are unstructured textual data
- NLP enables machine understanding of this text

PROJECT OBJECTIVES

Predict:

- Bug Category (e.g., UI, Backend, Security)
- Bug Severity (e.g., Low, Medium, Critical)
- Suitable Assignee (developer)
- Integrate predictions with tools like JIRA

SYSTEM ARCHITECTURE :



Ingestion
Bug Tracker Webhook\n(Jira/Bugzilla/GitHub)
Event Queue/Kafka
DataLayer
Raw Store\n(JSON blobs)
Feature Store\n(cleaned fields, embeddings)
Training Datasets
MLTraining
Preprocess & Featurize
Model Train & Tune\n(Multi-task, Severity, Assignee...)
Model Registry
InferenceSvc
Real-time Preproc
Model Inference API
Predictions + Confidence
Orchestration
Policy Engine\n(auto-apply? review?)
Update Tracker via REST API
Human Triage UI\nReview Queue
Feedback Labels

PREDICTING CATEGORY

- Goal: Classify bug into types like UI, Backend, Performance, Security
- Steps:

1. Preprocess input text (lowercase, lemmatize, remove stopwords)
2. Convert to TF-IDF vectors
3. Use Random Forest, Logistic Regression, or Naive Bayes
4. Model outputs a single category label

PREDICTING BUG SEVERITY

- Use the same preprocessed text
- Train separate model to predict one of four severity levels : Low, Medium, High, Critical
- Approach:
- Word embeddings + gradient boosted trees (XGBoost)
- Fine-tuned BERT for ordinal regression
- Leverage domain-specific intensity phrases: crash, data loss, freeze

PREDICTING ASSIGNEE (DEVELOPER MAPPING)

Goal: Assign bug to the most likely developer

Method:

- Use past data: who solved which bugs
- Train classifier (Naive Bayes, SVM)
- Output developer ID or name

JIRA INTEGRATION (API LAYER)

- Use jira Python library or raw REST API calls
- Authenticate using token
- Push prediction results as:

1. Labels → Category
2. Priority → Severity
3. Assignee → Developer

- Auto-create or update bug tickets

CONCLUSION & FUTURE SCOPE

- Automating triage using NLP + ML for:
- Category classification (e.g., UI, Backend)
- Severity prediction (e.g., Critical, Medium)
- Developer assignment based on historical bug data
- Integrating seamlessly with Jira using REST APIs

Key Benefits

- Reduced triage time & reassignments
- More consistent and faster bug resolutions
- Frees up developers to focus on actual fixing