



From Anomaly Detection to Autonomous Diagnosis: A Multi-Agentic Framework for Large Scale Neutrino Detector Ops

Dikshant Sagar, Kaiwen Yu, Boyan Pan, Jiayi Liu, Pengfei Ding, Jianming Bian, Pierre Baldi

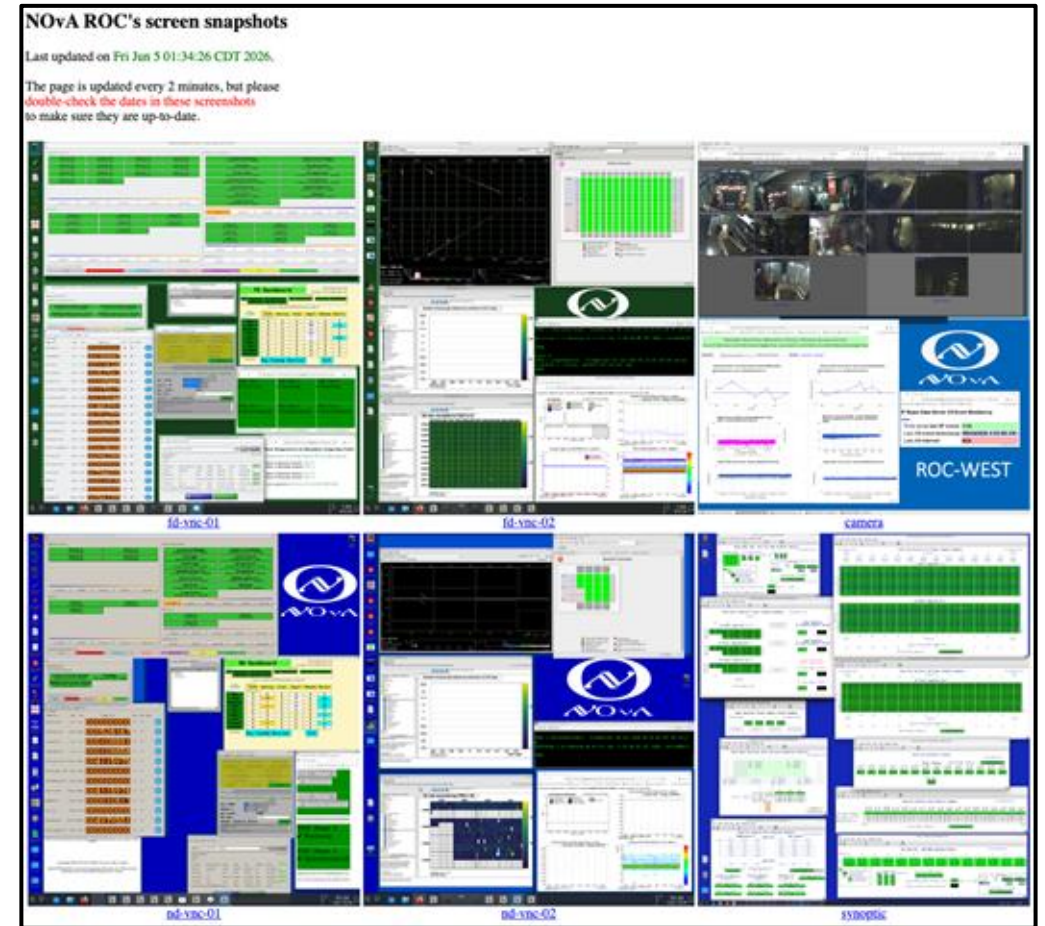
Project Overview

- Develop AI-based real-time anomaly detection and prediction for HEP DAQ systems
- Build an LLM-based General AI Shifter to diagnose issues and recommend mitigation actions
- Use NOvA's 10+ years of operational data for model development and validation ← We are here
- Leverage NERSC/Perlmutter GPU resources for large-scale AI training and deployment Integrate AI models with online DAQ systems for real-time monitoring, alerts, and shifter support
- Extend and generalize the framework from NOvA to ProtoDUNE/DUNE

Why DAQ Ops Need Agentic Assistance

Current Challenges

- DAQ systems generate enormous volumes of telemetry and logs
- Human shifters must monitor multiple dashboards simultaneously
- Root-cause analysis often requires searching:
 - ECL logs
 - detector images
 - telemetry streams
 - historical incidents
- Expertise is not always available **24/7**
- Mean Time To Resolution (MTTR) can be high.



The Vision

Goal

Create an Agentic AI-powered DAO Shifter capable of:



Detecting
Anomalies in
Real Time



Gathering Relevant
Context
Automatically



Diagnosing Likely
Causes



Recommending
Recovery Actions



Executing Approved
Actions Safely



Learning from
Previous Incidents

End Goal

Transition from:



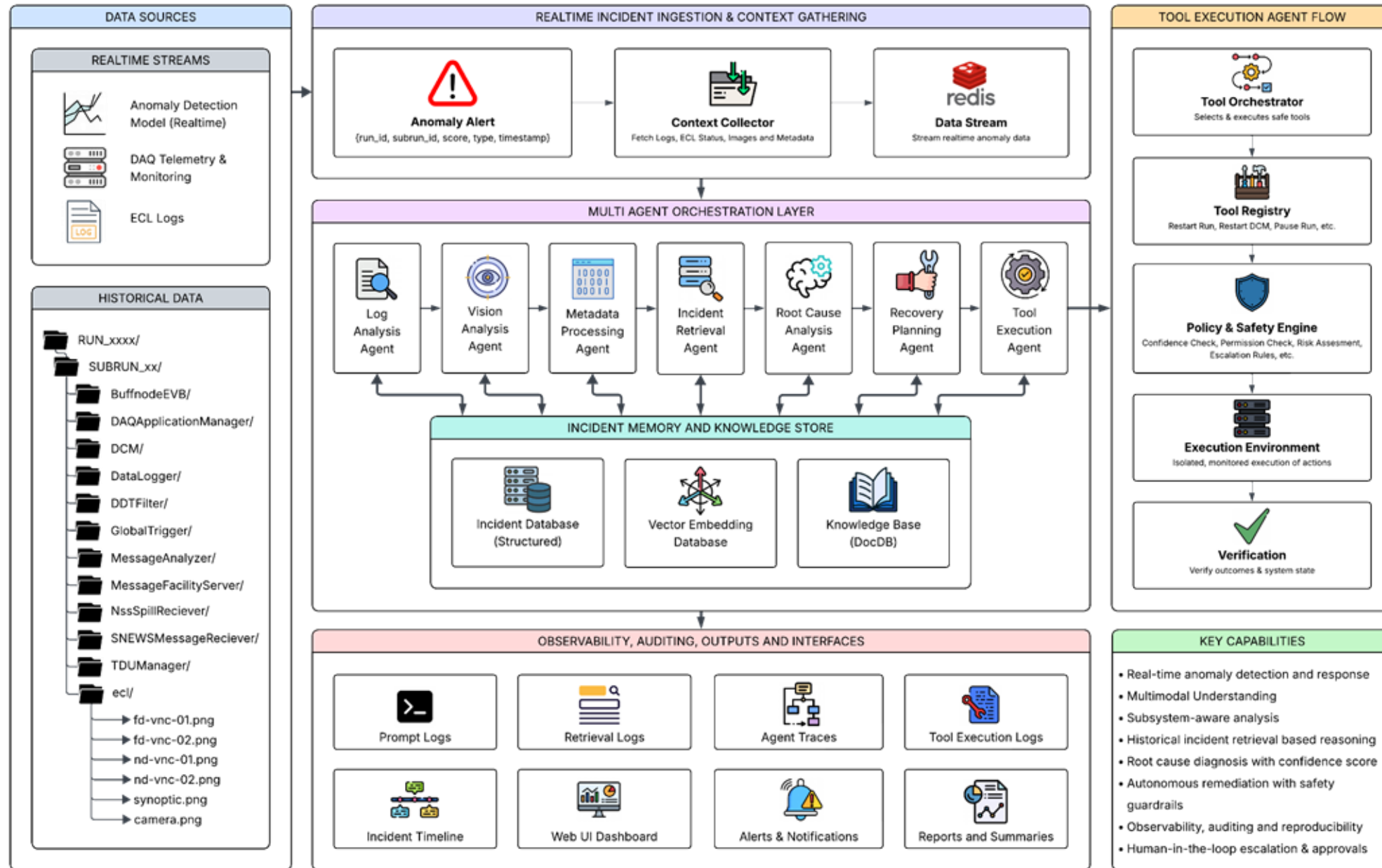
Human-centric Reactive
Operations

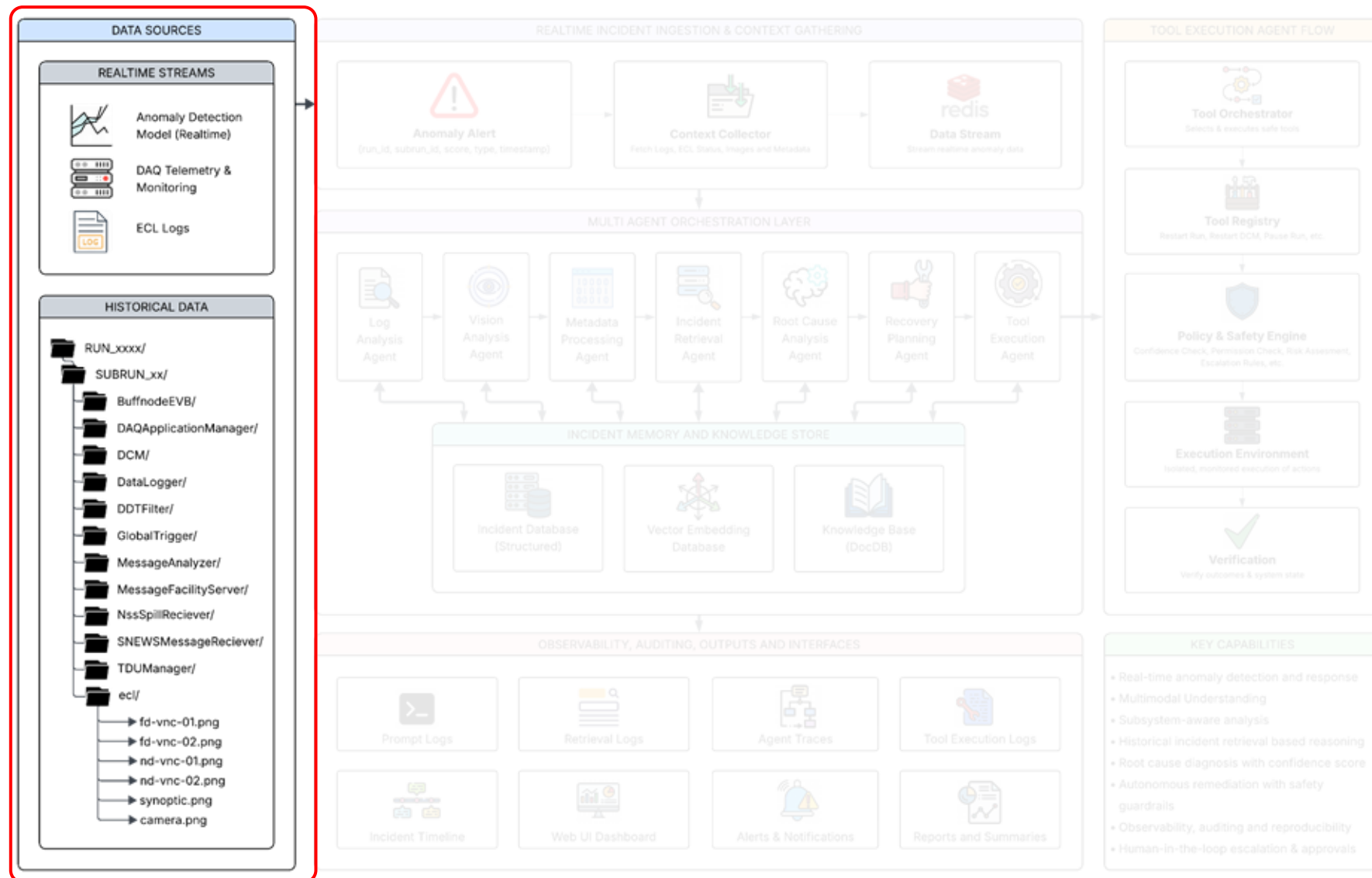


Agent-assisted Autonomous
Operations

for NOvA and future DUNE deployments.

High-Level Architecture





Data Sources & Inputs - NOvA

Operational Data Ecosystem

Telemetry Sources



- NOvA DAQ monitoring metrics
- Run status
- Performance counters

Visual Sources



- NOvA VNC screenshots
- NOvA Synoptic displays
- NOvA Detector monitoring images

NOvA Subsystem Logs



- DCM
- DataLogger
- GlobalTrigger
- BuffnodeEVB
- MessageFacility
- etc.

NOvA Subsystem Logs

Log files are generated by almost all DAQ subsystems. They are stored on a volume hosted by the message logger machines for each cluster and are archived to tape approximately 15 days after creation.

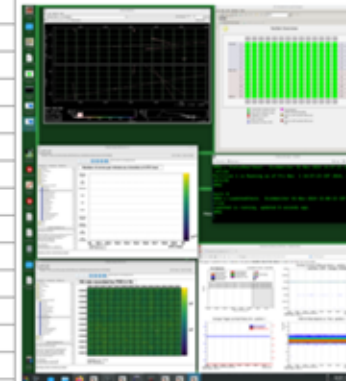
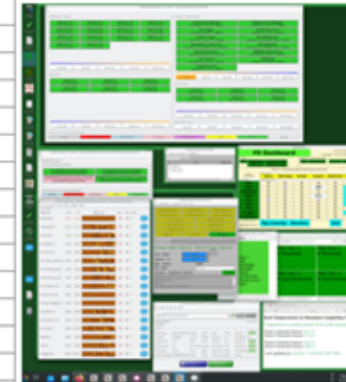
```
[novadaq@novadaq-near-mgr-01 ~]$ ls -lh /daqlogs/NearDet/Partition0
total 12K
drwxrwxrwx 14 novadaq nova 4.0K Jun  2  2014 BufferNodeEVB
drwxrwxrwx  7 novadaq nova  166 Dec 14  2023 ConfigurationEditor
drwxrwxrwx  3 novadaq nova   45 Jun  2  2014 ConfigurationManager
drwxrwxrwx  3 novadaq nova   45 Jun  2  2014 DAQApplicationManager
drwxrwxrwx  4 novadaq nova   76 Nov 22  2023 DataLogger
drwxrwxrwx 47 novadaq nova 4.0K Feb 18  2025 dds
drwxrwxrwx  3 novadaq nova   45 Jun  2  2014 DDTManager
drwxrwxrwx  4 novadaq nova 4.0K Jun  2 20:49 DowntimeLogger
drwxrwxrwx  3 novadaq nova   46 Jun  2  2014 EventDispatcher
drwxrwxrwx  3 novadaq nova   42 Jun  2  2014 GlobalTrigger
drwxrwxrwx  6 novadaq nova  137 Nov 13  2020 MessageFacility
drwxrwxrwx  3 novadaq nova   42 Jun  2  2014 NssSpillReceiver
drwxrwxr-x  2 novadaq nova   10 May 26 18:51 PedestalDataRunner
drwxrwxrwx  3 novadaq nova   45 Jun  2  2014 RunControl
drwxrwxrwx  4 novadaq nova   69 Aug 22  2023 TDUDelayMonitor
drwxrwxrwx  3 novadaq nova   45 Jun  2  2014 TDUManager
drwxrwxrwx  3 novadaq nova   46 Jun  2  2014 TriggerScalars
drwxrwxrwx  6 novadaq nova  137 Nov 13  2020 WatchDog
```

We separated and stored the data by run/subrun according to the timestamps stored in the log files.

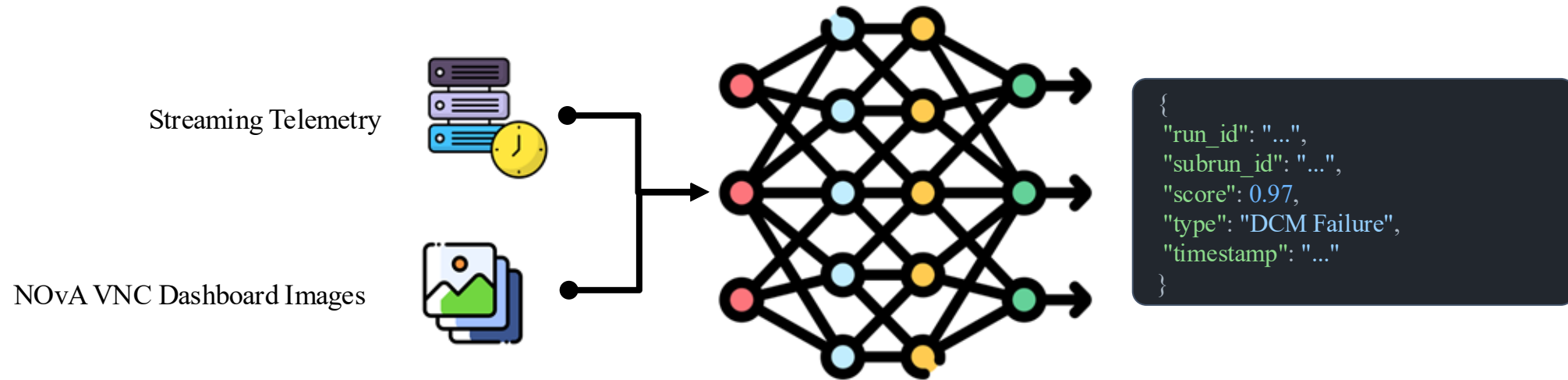
NOvA ECL Entries

Match the shifter checklists (updated every two hours) from ECL to the log files using run/subrun information, and apply a simple normal/anomalous ECL entry classification.

Form: 2023_Control-Room-checklist	
	:
	:
Last update time of FD-01 desktop?:	Within 5 minutes
Are all FD-01 windows present?:	Yes
Rack Temps OK?:	Yes
TCR page okay?:	Yes
Spill Server GUI green?:	No: but there is ongoing expert work
	:
Last update time of FD-02 desktop?:	Within 5 mins
Are all FD-VNC-02 windows present?:	Yes
DiskWatcher Heartbeat:	Ok: within last two hours
Loadshed Check:	Ok: with last 5 mins
	:
Last update time of synoptic desktop?:	Within 5 mins
Are all synoptic windows present?:	Yes
Time stamps okay?:	Yes
All boxes green?:	Yes
Are all dew points under threshold (13 C) for ND ?:	Yes
Are all dew points under threshold (55 F) for FD?:	Yes
	:
Are the cameras updating?:	Yes - all
	:
Last update time of ND-01 desktop?:	Within 5 mins
Are all ND-01 windows present?:	Yes
TCR page okay?:	Yes
Pre-supernova GUI okay?:	No: experts aware so no action taken
Spill server all green?:	Yes
	:
Last update time of ND-02 desktop?:	Within 5 minutes
Are all ND-02 windows present?:	Yes
Disk watcher heartbeat:	Ok: within 25 mins
Attach screenshots:	True
Additional information:	Near gateway networking/SNEWS issues still unresolved. For ROCs, using fardet gateway or VPN enables access to the VNC sessions



Real-Time Anomaly Detection



Anomaly Types

DCM failures

Trigger issues

Data logger stalls

Network degradation

Missing data streams

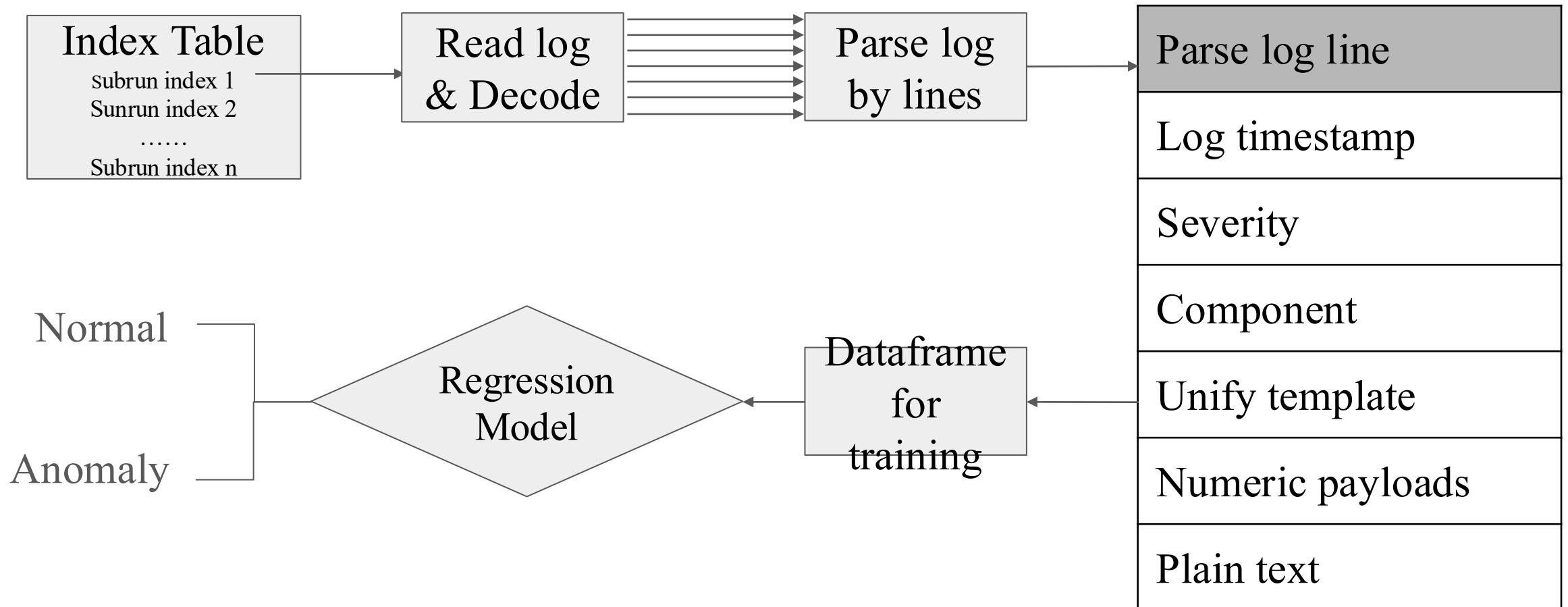
Unknown anomalies

⋮

⋮

⋮

Anomaly Detection Pipeline



Realtime Incident Ingestion & Context Gathering



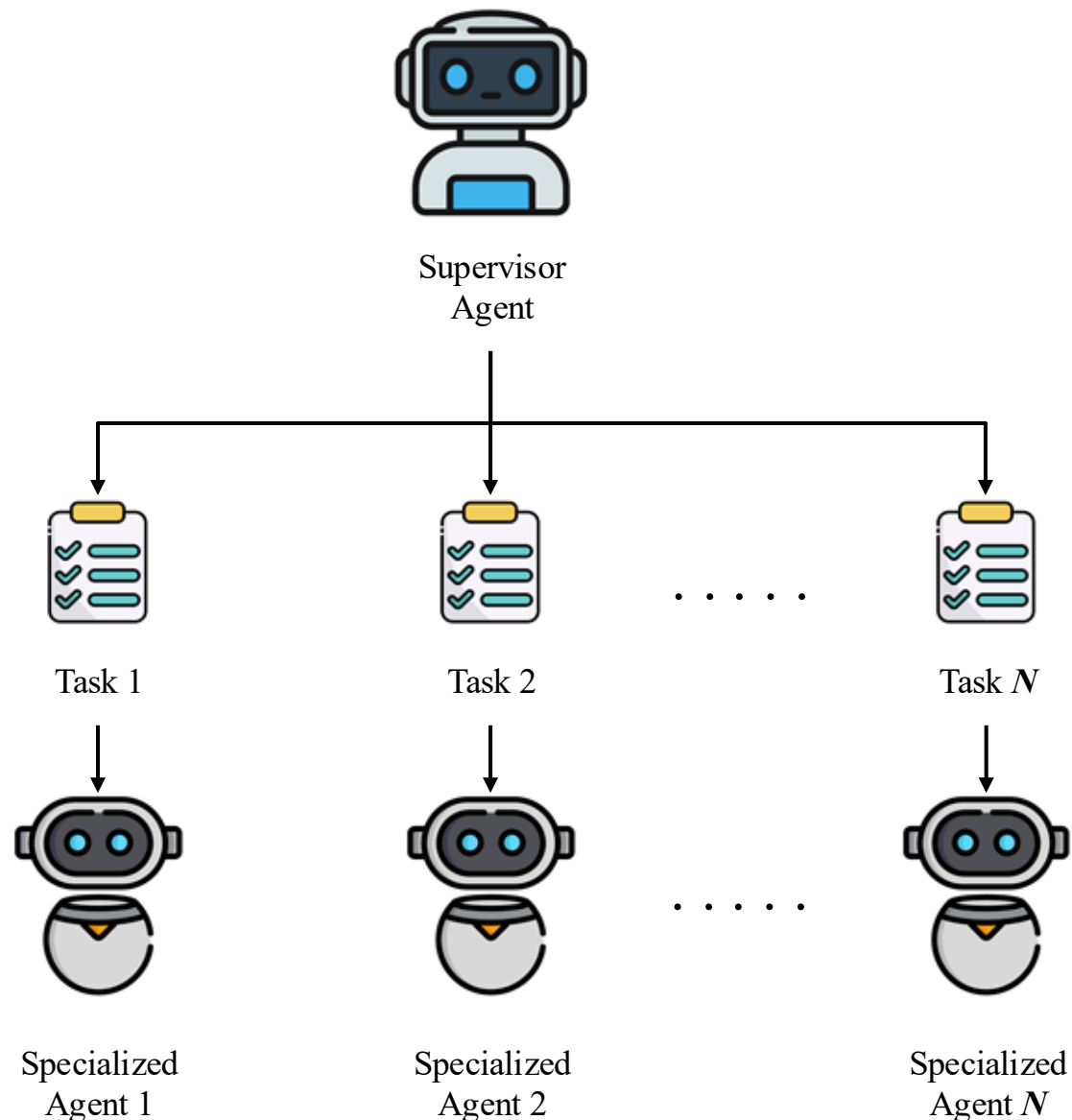
Multi-Agent Reasoning Framework

Single LLM Agent limitations:

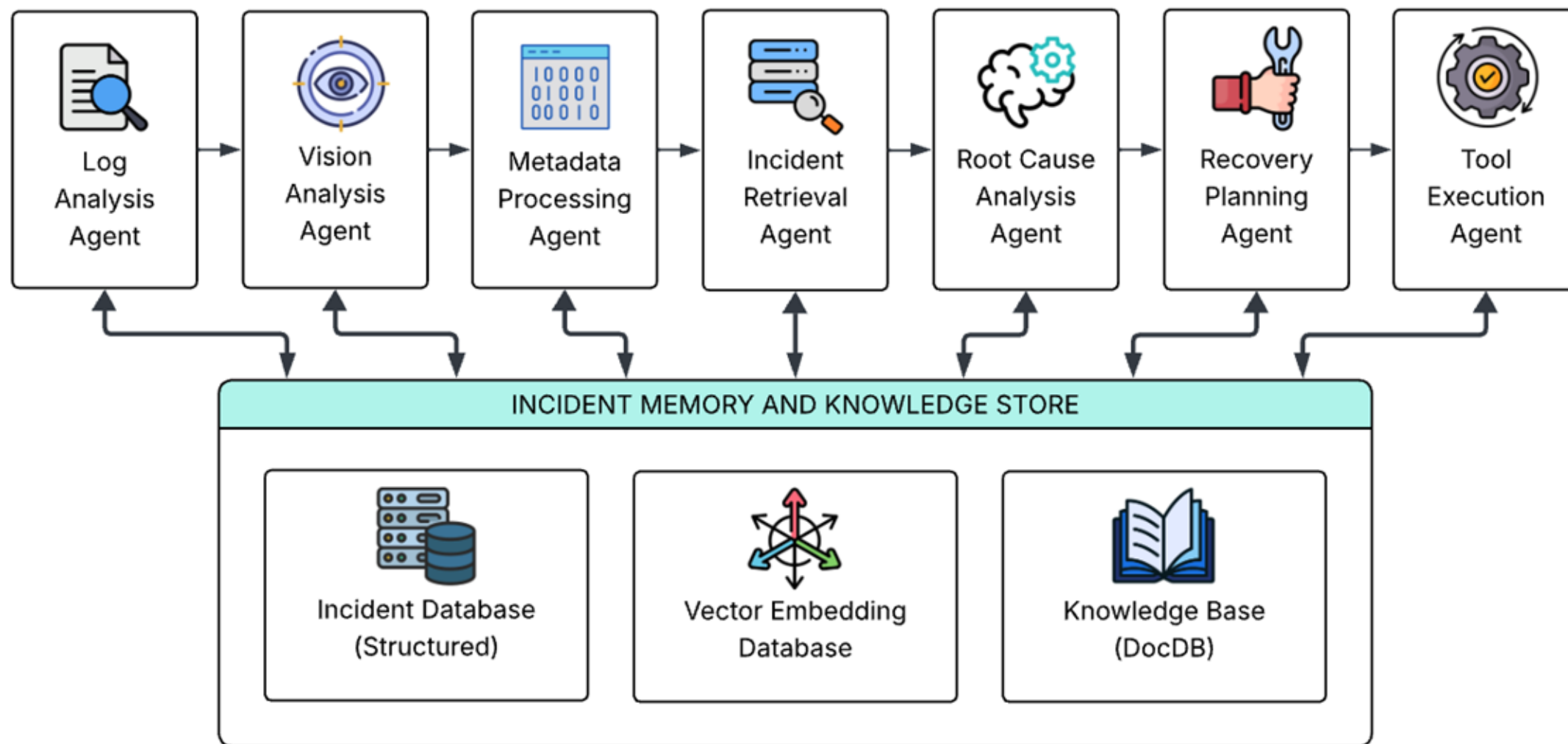
- Too much context
- Mixed responsibilities
- Difficult debugging

Multi-agent advantages:

- Specialized expertise
- Parallel reasoning
- Better traceability
- Easier maintenance



Multi-Agent Orchestration



Log Analysis Agent

Responsibilities:

- Parse ECL/Telemetry logs
- Detect errors/warnings
- Temporal correlation
- Extract subsystem failures

Output:

```
{  
  "suspected_component": "DCM",  
  "confidence": 0.89  
}
```

The image displays four screenshots of log files, likely from a system monitoring or debugging tool. Each screenshot shows a list of log entries with timestamps, thread IDs, and messages.

- Top Left:** Log file `bnevb001_20251125_072051.log`. It shows a sequence of events related to a buffer node, including connecting to shared memory, reading initial offsets, and adding pairs of data.
- Top Right:** Log file `appMgr_20251125_071941.log`. It shows a sequence of events related to a DAQ application manager, including valid starting, expires, and service principal information.
- Bottom Left:** Log file `ndl_20251125_072052.log`. It shows a sequence of events related to a data logger, including getting mutex, making new run configuration, and adding new.
- Bottom Right:** Log file `dcm-1-01-01_20251125_072103.log`. It shows a sequence of events related to a DCM (Data Control Module), including setting partition number, setting application name, and expanded msg config file.

Vision Analysis Agent

Inputs:

- VNC screenshots
- Synoptic displays
- Monitoring images

Tasks:

- Status interpretation
- Visual anomaly summarization
- Cross-validation with logs

Example findings:

- Offline nodes
- Red alarm indicators
- Frozen displays



SUMMARY

☐	_____
☐	_____
☐	_____

Incident Retrieval Agent

Uses:

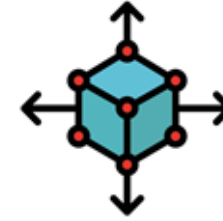
- A Vector Embedding Database

Retrieves:

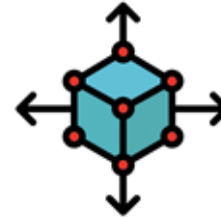
- Similar past failures and expert fixes from the ECL
- Relevant knowledge from DocDB and Internal Codebases
- Past Agent-resolved Incidents

Benefits:

- Faster diagnosis
- Consistent responses
- Knowledge preservation
- Creates a continuously improving system



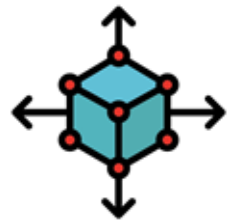
DocDB



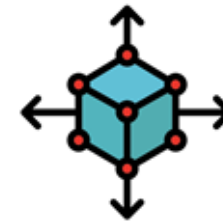
ECL Logs



Similarity Search



Codebase



Past
Incidents

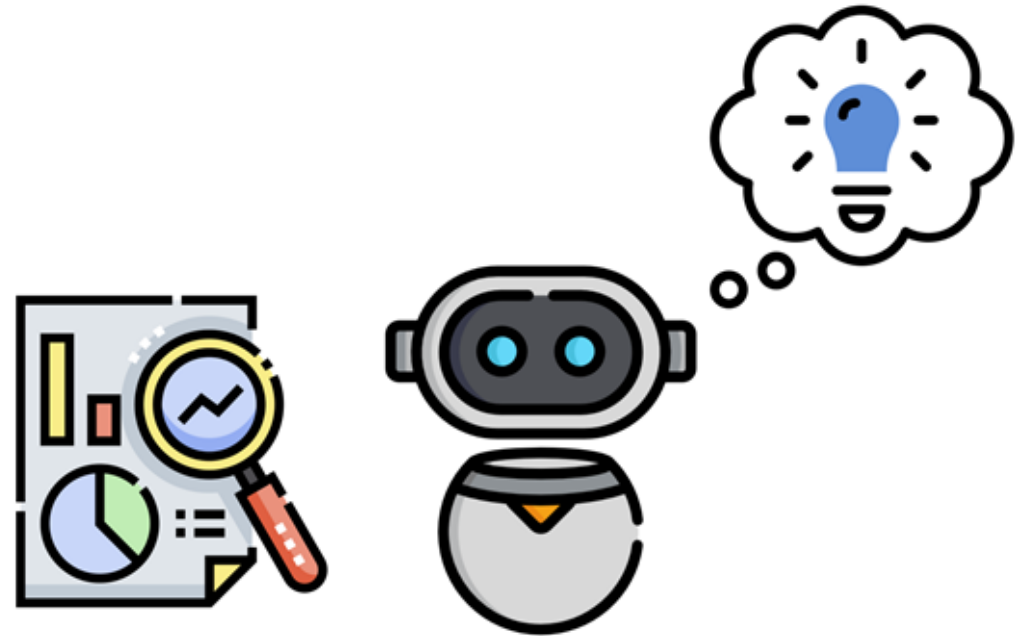
Root Cause Analysis Agent

Combines Context from:

- Logs
- Images
- Metadata
- Incident Retrieval

Produces:

```
{  
  "root_cause": "...",  
  "confidence": 0.93,  
  "evidence": [...]  
}
```

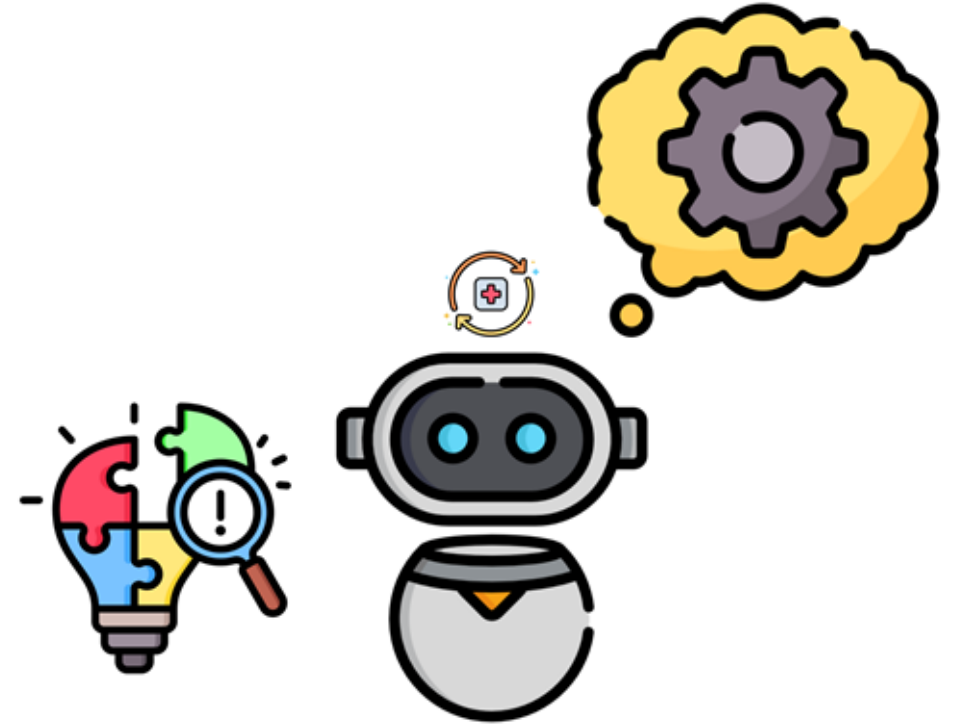
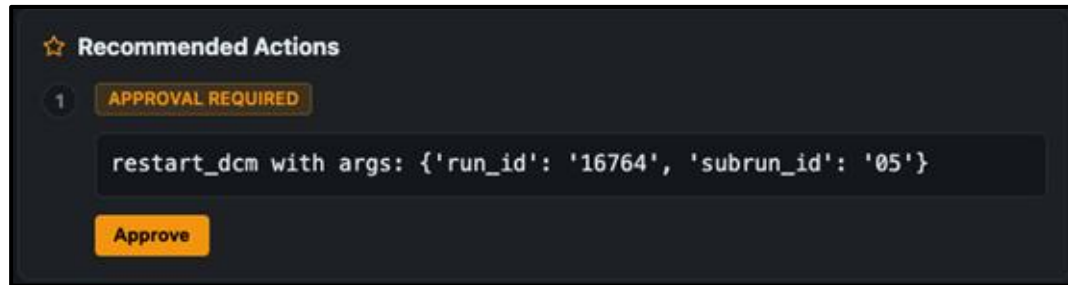


Recovery Planning Agent

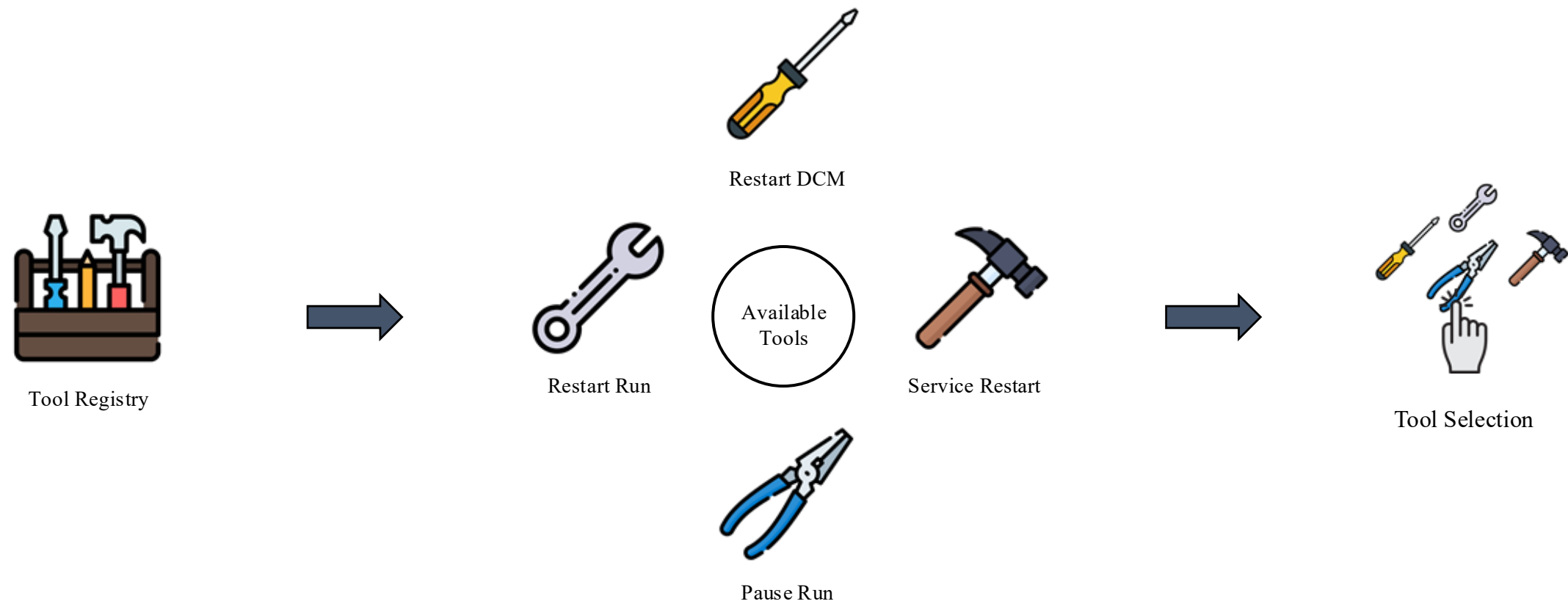
Generates:

- Recovery recommendations
- Risk assessment
- Expected outcomes

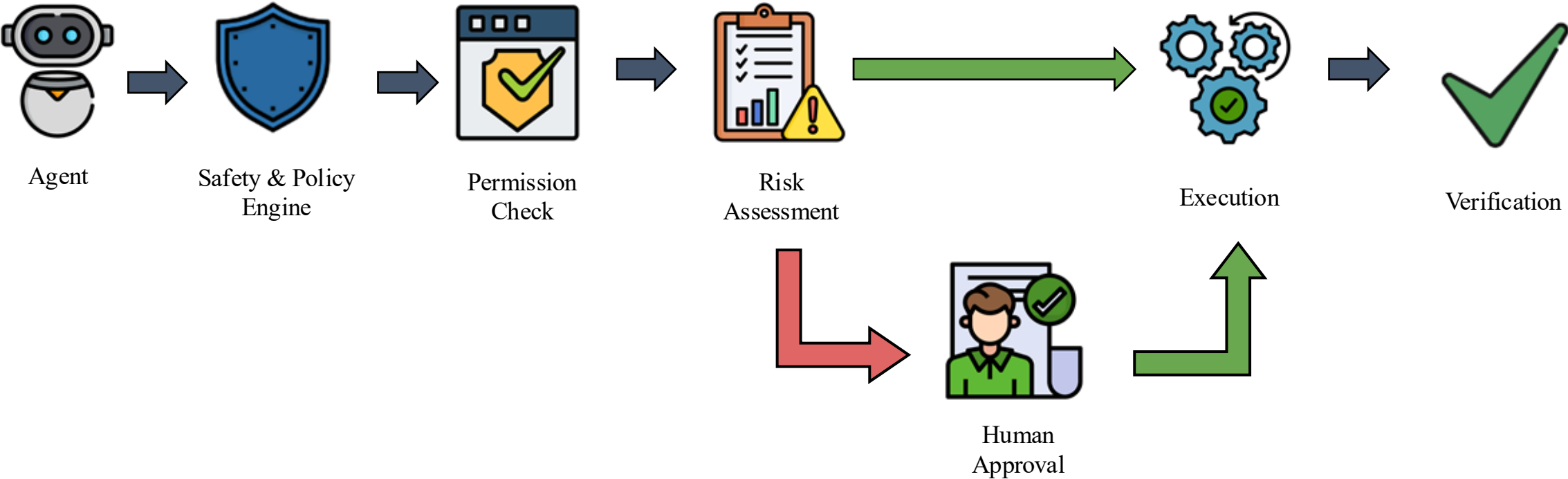
Example:



Tool Execution Framework



Tool Execution Safety Architecture



Safety mechanisms:

Confidence Thresholds	Approval Gates	Role-based Permissions	Escalation Rules
-----------------------	----------------	------------------------	------------------

Human-in-the-Loop Design

Cases requiring approval:

- High-risk actions
- Low-confidence diagnosis
- Novel anomalies

Benefits:

- Trust
- Safety
- Operational acceptance

The screenshot displays a HITL interface for incident management. At the top, the incident ID is `inc_09505980`. Action buttons include `Ask Agent`, `Escalate`, and `Close Incident`. Below this, a timeline of events is shown, starting with an anomaly detection at 17:32:17 and followed by various agent actions like log collection, analysis, and recovery planning. A section titled `Historically Similar Incidents (RAG)` lists two previous incidents: `DCM corruption due to timing desync` and `Buffer node crash caused by RDS network fault`. The `Root Cause Analysis` section provides a detailed explanation: "A timing synchronization failure caused the DCM (Data Concentrator Module) to lose alignment with the global trigger, leading to a burst of DCM corruption and cascading message facility/analyzer errors." This analysis is marked with `88% CONFIDENCE`. Finally, the `Recommended Actions` section shows a single action: `pause_run with args: {'run_id': '16764'}`, which requires approval.

Incident `inc_09505980` `Ask Agent` `Escalate` `Close Incident`

`Run 16764` `Subrun 05` `dcm_corruption_burst`

▼ Live Incident Timeline

```
17:32:17 Anomaly dcm_corruption_burst detected
17:32:17 Log Analysis Agent collecting subsystem logs...
17:32:17 Vision Analysis Agent collecting ECL snapshots...
17:32:17 Log Analysis Agent parsing telemetry...
17:32:18 Incident Retrieval Agent generating embeddings...
17:32:18 Incident Retrieval Agent retrieving historical context...
17:32:18 Vision Analysis Agent performing multimodal reasoning...
17:32:18 Root Cause Analysis Agent synthesizing findings...
17:32:49 Recovery Planning Agent proposing actions...
17:32:53 Recovery Planning Agent validating against safety policy...
17:32:53 Recovery Planning Agent executing safe tools...
17:32:53 Recovery Planning Agent verifying remediation outcome...
```

▼ Historically Similar Incidents (RAG)

`Run 16750` `Subrun 01` DCM corruption due to timing desync.

`Run 16755` `Subrun 04` Buffer node crash caused by RDS network fault.

Root Cause Analysis `88% CONFIDENCE`

A timing synchronization failure caused the DCM (Data Concentrator Module) to lose alignment with the global trigger, leading to a burst of DCM corruption and cascading message facility/analyzer errors.

☆ Recommended Actions

1 `APPROVAL REQUIRED`

`pause_run with args: {'run_id': '16764'}`

`Approve`

Observability & Explainability: Operational Transparency

Stored artifacts:

- Agent traces
- Prompts
- Retrieval results
- Tool executions
- Decisions

Provides:

- Auditability
- Reproducibility
- Debugging support

The screenshot displays a user interface for managing an incident, identified by the ID `inc_09505980`. At the top, there are buttons for `Ask Agent`, `Escalate`, and `Close Incident`. Below the incident ID, there are tabs for `Run 16764`, `Subrun 05`, and `dcm_corruption_burst`.

The main section is titled `Live Incident Timeline` and contains a list of events with timestamps and descriptions:

- 17:32:17 Anomaly `dcm_corruption_burst` detected
- 17:32:17 Log Analysis Agent collecting subsystem logs...
- 17:32:17 Vision Analysis Agent collecting ECL snapshots...
- 17:32:17 Log Analysis Agent parsing telemetry...
- 17:32:18 Incident Retrieval Agent generating embeddings...
- 17:32:18 Incident Retrieval Agent retrieving historical context...
- 17:32:18 Vision Analysis Agent performing multimodal reasoning...
- 17:32:18 Root Cause Analysis Agent synthesizing findings...
- 17:32:49 Recovery Planning Agent proposing actions...
- 17:32:53 Recovery Planning Agent validating against safety policy...
- 17:32:53 Recovery Planning Agent executing safe tools...
- 17:32:53 Recovery Planning Agent verifying remediation outcome...

Below the timeline, there is a section for `Historically Similar Incidents (RAG)` with two entries:

- `Run 16750` `Subrun 01`: DCM corruption due to timing desync.
- `Run 16755` `Subrun 04`: Buffer node crash caused by RDS network fault.

The `Root Cause Analysis` section shows a confidence level of `88% CONFIDENCE` and states: "A timing synchronization failure caused the DCM (Data Concentrator Module) to lose alignment with the global trigger, leading to a burst of DCM corruption and cascading message facility/analyzer errors."

The `Recommended Actions` section lists one action:

- 1 `APPROVAL REQUIRED`
`pause_run` with args: `{'run_id': '16764'}`
`Approve`

Detection Model Evaluation



Accuracy

Precision

Recall

F1 Score

ROC-AUC

Detection Latency

Agentic System Evaluation

Diagnostic Quality

- Root cause accuracy
- Recommendation accuracy

Retrieval Quality

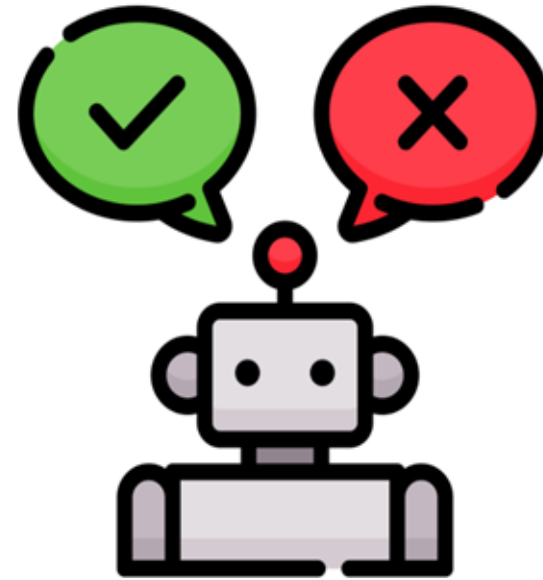
- Recall@K
- Mean Reciprocal Rank (MRR)

Operational Metrics

- MTTR reduction
- Time-to-diagnosis
- Human intervention rate

Safety Metrics

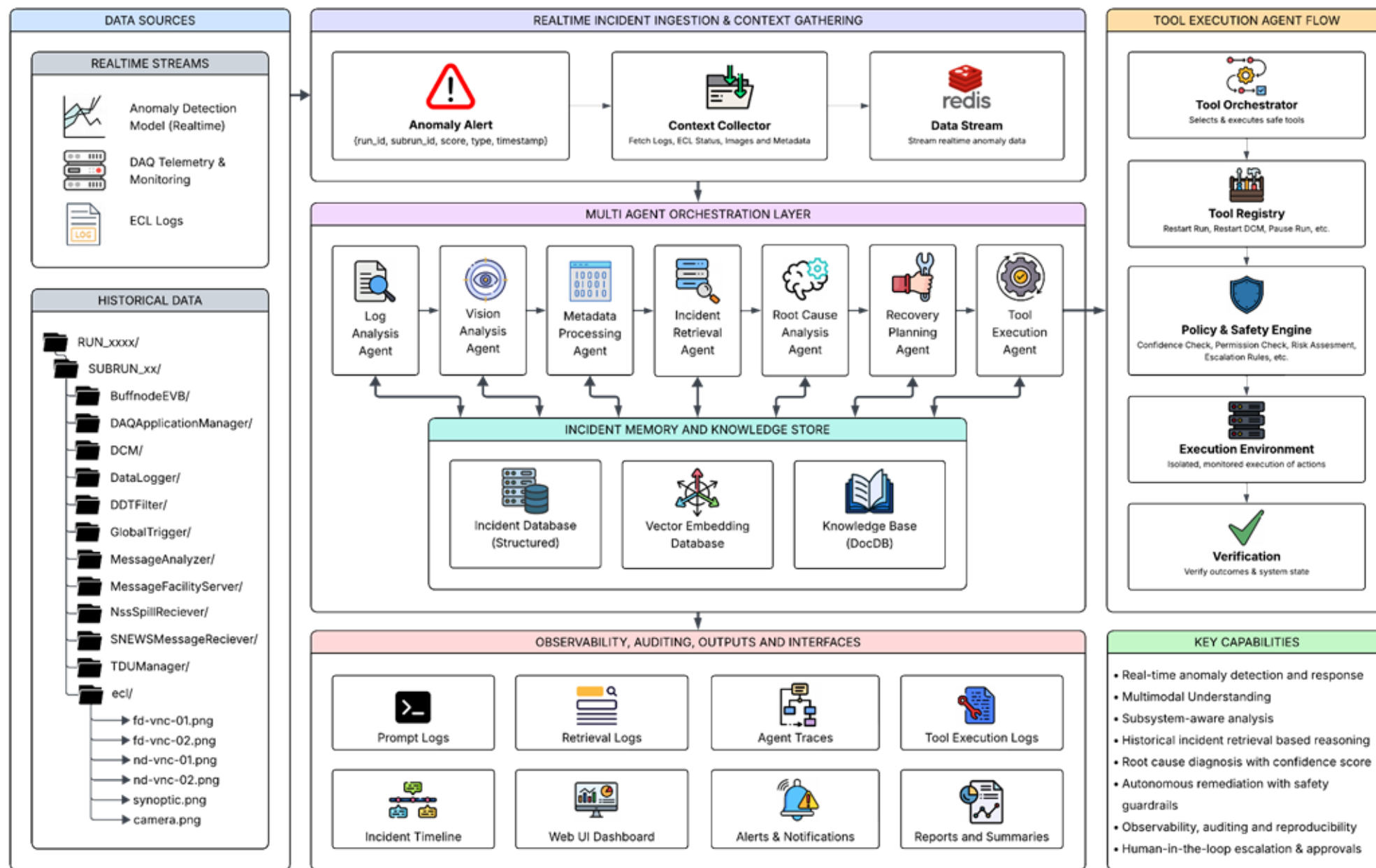
- Unsafe action rate
- Escalation accuracy





Workflow

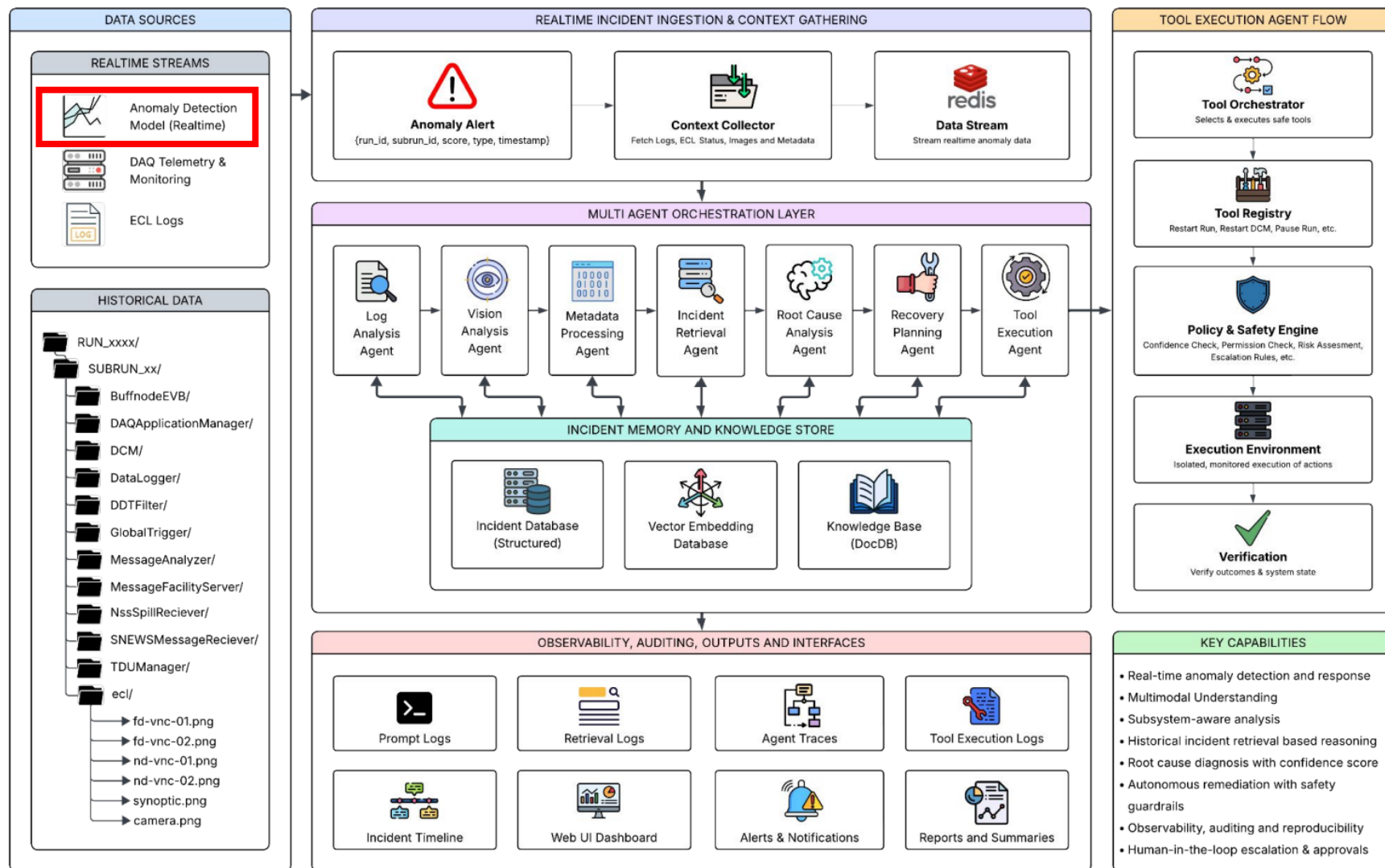
Example Incident Walkthrough



System Overview

Anomaly Detection Model:

- Subrun-level detection
- Lightweight: no LLM/GPU needed
- Fast: ~seconds per subrun
- Ensemble of multiple models
- Also outputs top features for AI agent to explain root causes and solutions



Data Inputs for Model Training

- **Training data:** archived log files divided into run/subrun level;
- **Truth labels (normal/anomaly):** determined based on ECL checklists filled by the shifters;
- Using all ND data from 2023-2025, ~20k subruns in total, of which ~14k have a corresponding ECL entry.

Manager Processes:Managers	
ConfigurationManager [novadaq-near-mgr-04: 2271303]	DAQApplicationManager [novadaq-near-mgr-04: 2241099]
DataLogger [novadaq-near-dd-03: 2937704]	EventDispatcher [novadaq-near-dd-03: 2937668]
GlobalTrigger [novadaq-near-mgr-04: 2271347]	MessageAnalyzer [novadaq-near-msglogger-a: 3930741]
MessageFacilityServer [novadaq-near-msglogger-a: 3930730]	MessageViewer [novadaq-near-msglogger-a: 3930740]
RunControlServer [novadaq-near-mgr-04: 2244394]	SNEWSMessageReceiver [novadaq-near-mgr-04: 2272619]
SpillServer [novadaq-near-mgr-04: 2272608]	TDUManager [novadaq-near-mgr-04: 2271368]
TriggerScalars [novadaq-near-dd-03: 2937683]	

Log files from each subsystem (training data)

```
ECL Status: ANOMALY
ECL Entry: #251280
ECL Timestamp: 2025-12-05T16:36:55.905168+00:00
ND Run: 16735/4
FD Run: 53463/31
Reasons: nd_feb_hit_rate: No - sections with high/low rate
```

Processed ECL checklists (truth label)

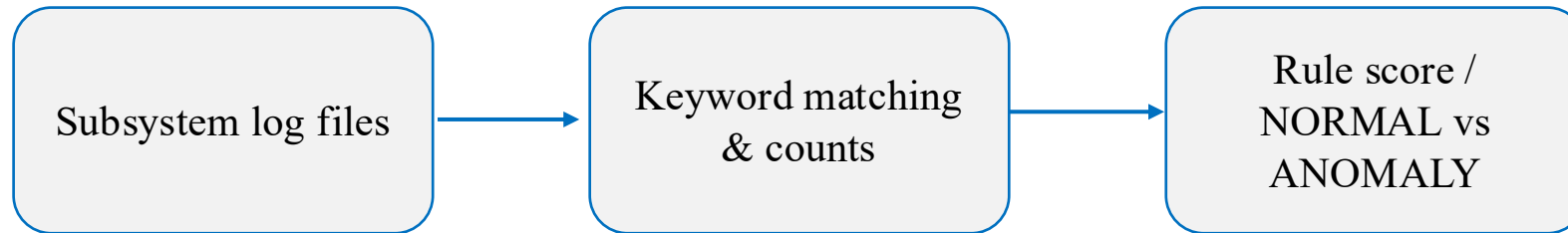
```
$ls -lh RUN_16735/SUBRUN_04
total 100K
-rwxr-xr-x 1 jiaxi jiaxi 1.6K May 31 17:23 anomaly_report.txt
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 15:55 BuffnodeEVB
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 17:09 ConfigurationManager
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 17:11 DAQApplicationManager
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 15:55 DataLogger
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 16:36 DCM
drwxr-xr-x 2 jiaxi jiaxi 40K May 31 16:37 DDTFilter
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 16:09 ecl
-rwxr-xr-x 1 jiaxi jiaxi 179 Jul 8 07:22 ecl_status.txt
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 16:53 GlobalTrigger
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 17:14 MessageAnalyzer
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 17:17 MessageFacilityServer
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 17:19 NssSpillReceiver
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 17:20 SNEWSMessageReceiver
drwxr-xr-x 2 jiaxi jiaxi 4.0K May 31 17:20 TDUManager
```

Data input for each subrun

Anomaly Detection Models

Rule-Based Detection model:

Scans subsystem logs for known error keywords, counts matches, then applies fixed thresholds to produce a rule score and subrun verdict.



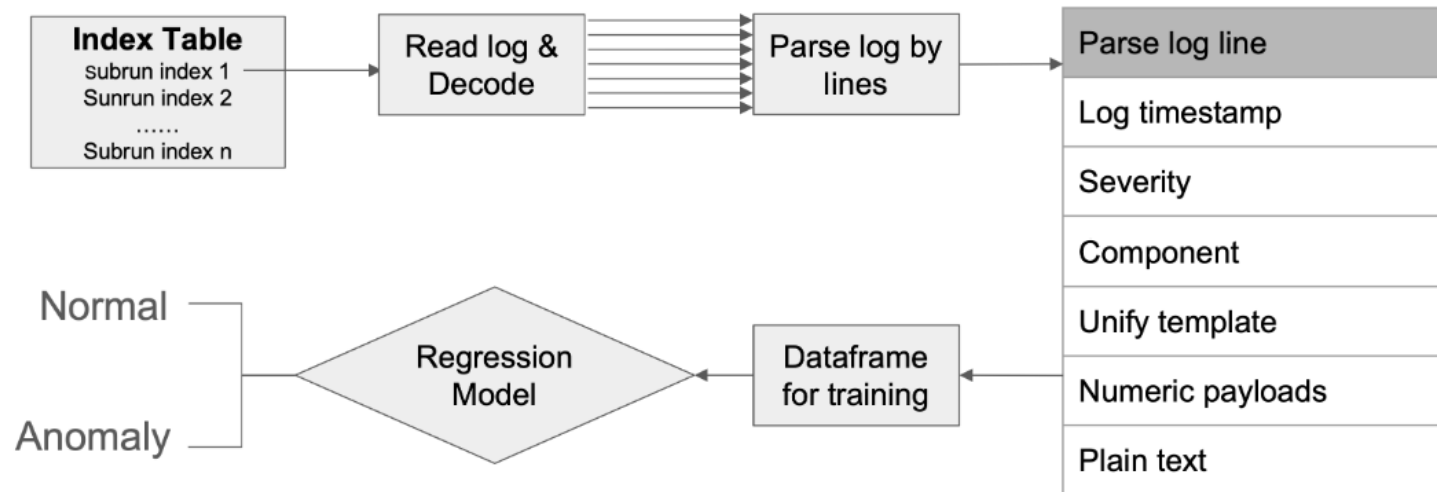
- Rule-based, fast deterministic detector, no training dataset required;
- Thresholds need to be recalibrated using real data.

Rule-Based Detection		NORMAL
Score		0
Type	—	
KEYWORD COUNTS		
Dcm Corruption		1, 120
Buffnode Rds Errors		0
Globaltrigger Conflicts		0
Datalogger Errors		0
Daqmanager Errors		0
Ddtfilter Errors		0
Messageanalyzer Errors		123
Messagefacility Errors		382
Nssspill Errors		0
Snews Errors		0
Tdumanager Errors		0
Timing Desync		false

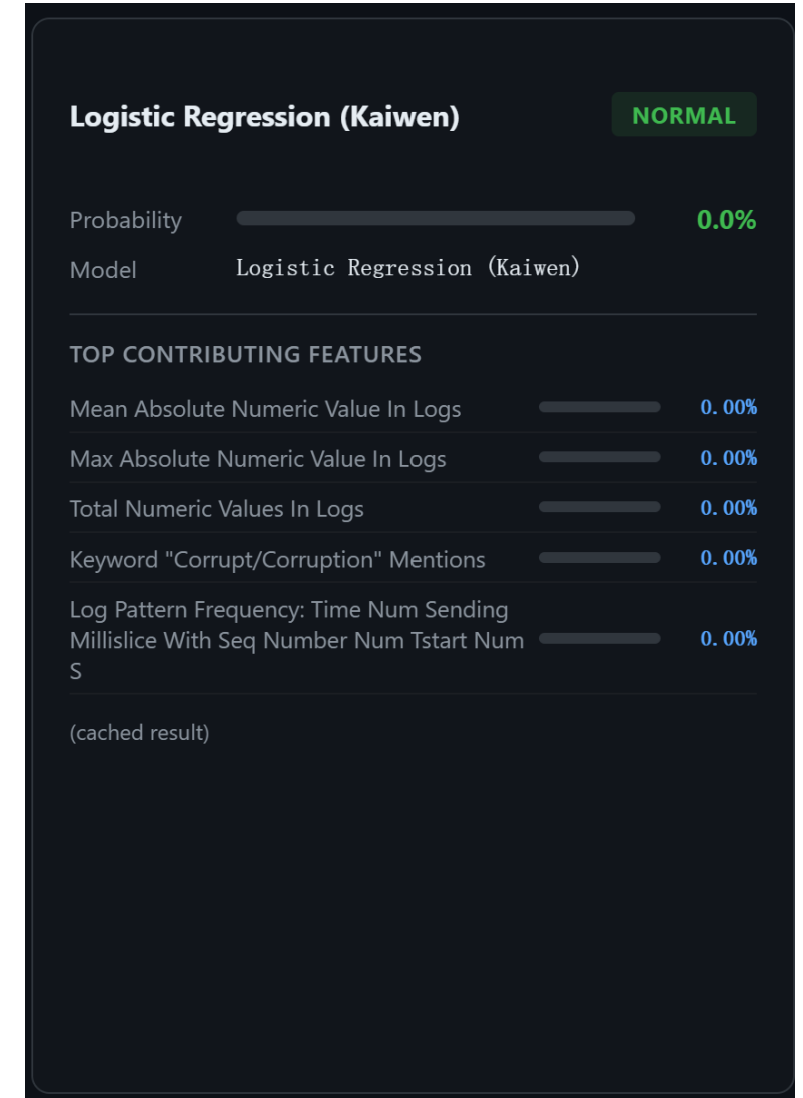
Anomaly Detection Models

Logistic Regression model A

Converts raw log files into numeric features and applies a scikit-learn pipeline of median imputation, standard scaling, and balanced-class logistic regression to output an anomaly probability for each subrun.



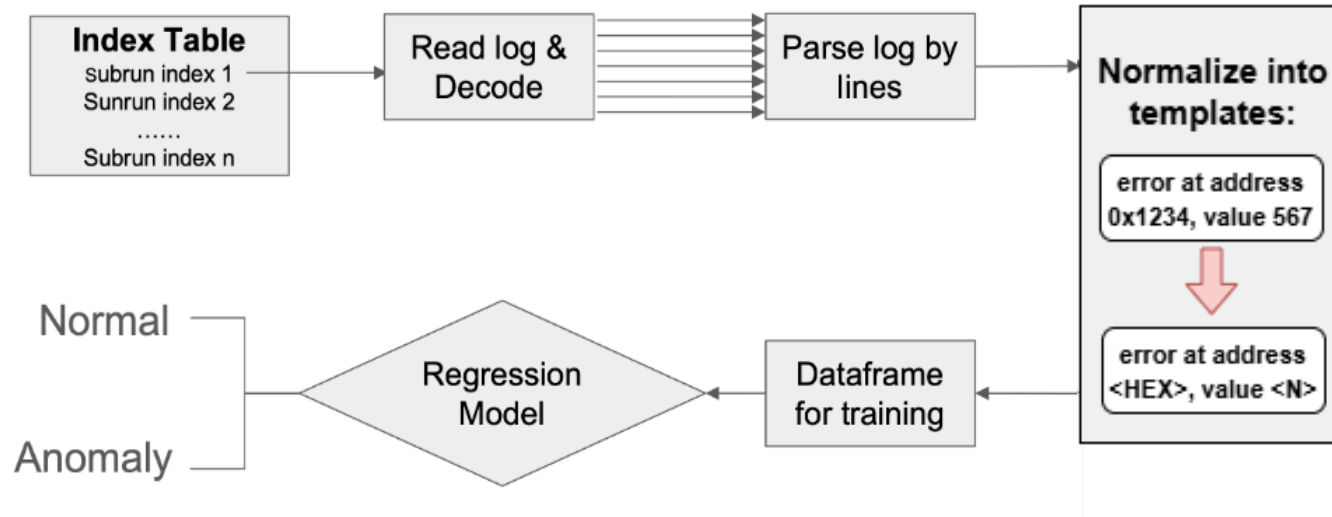
- Data-driven, more dependent on training data, but also easier to recalibrate;
- Less interpretable due to the feature extraction step.



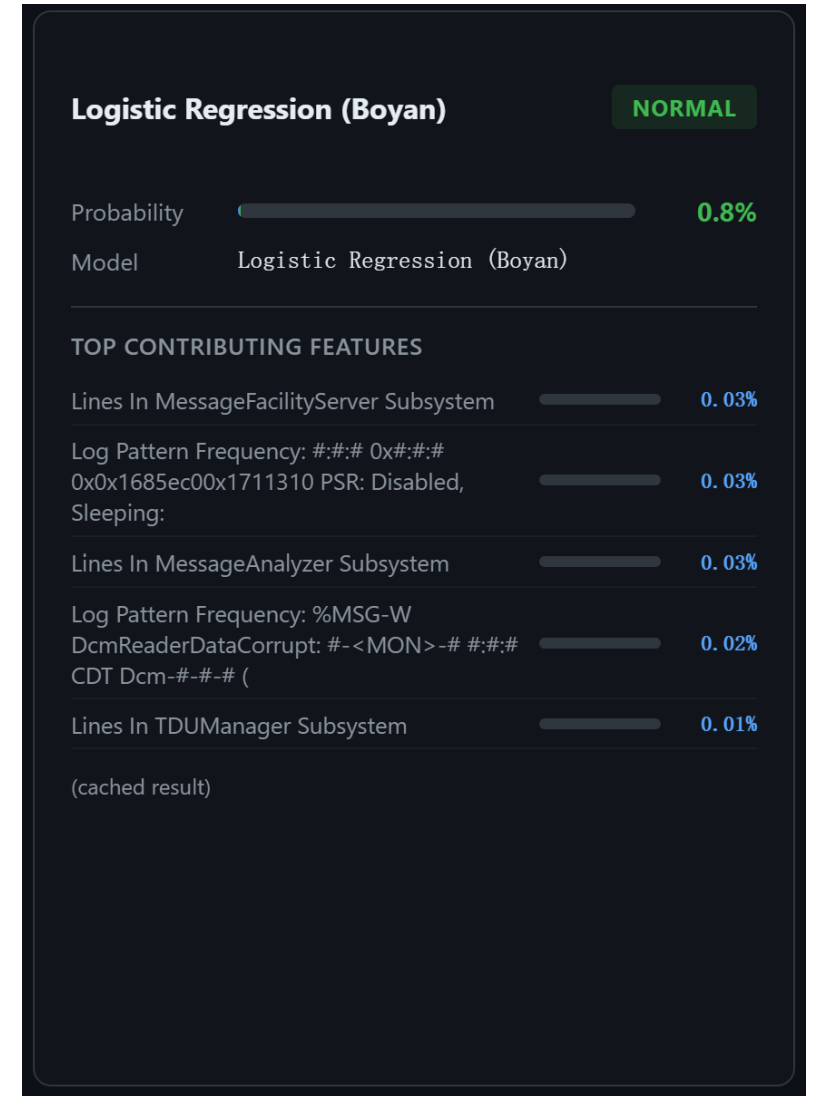
Anomaly Detection Models

Logistic Regression model B

Normalizes raw log files into templates by replacing variables into placeholders, and use a template-hash to automatically map them to feature space.

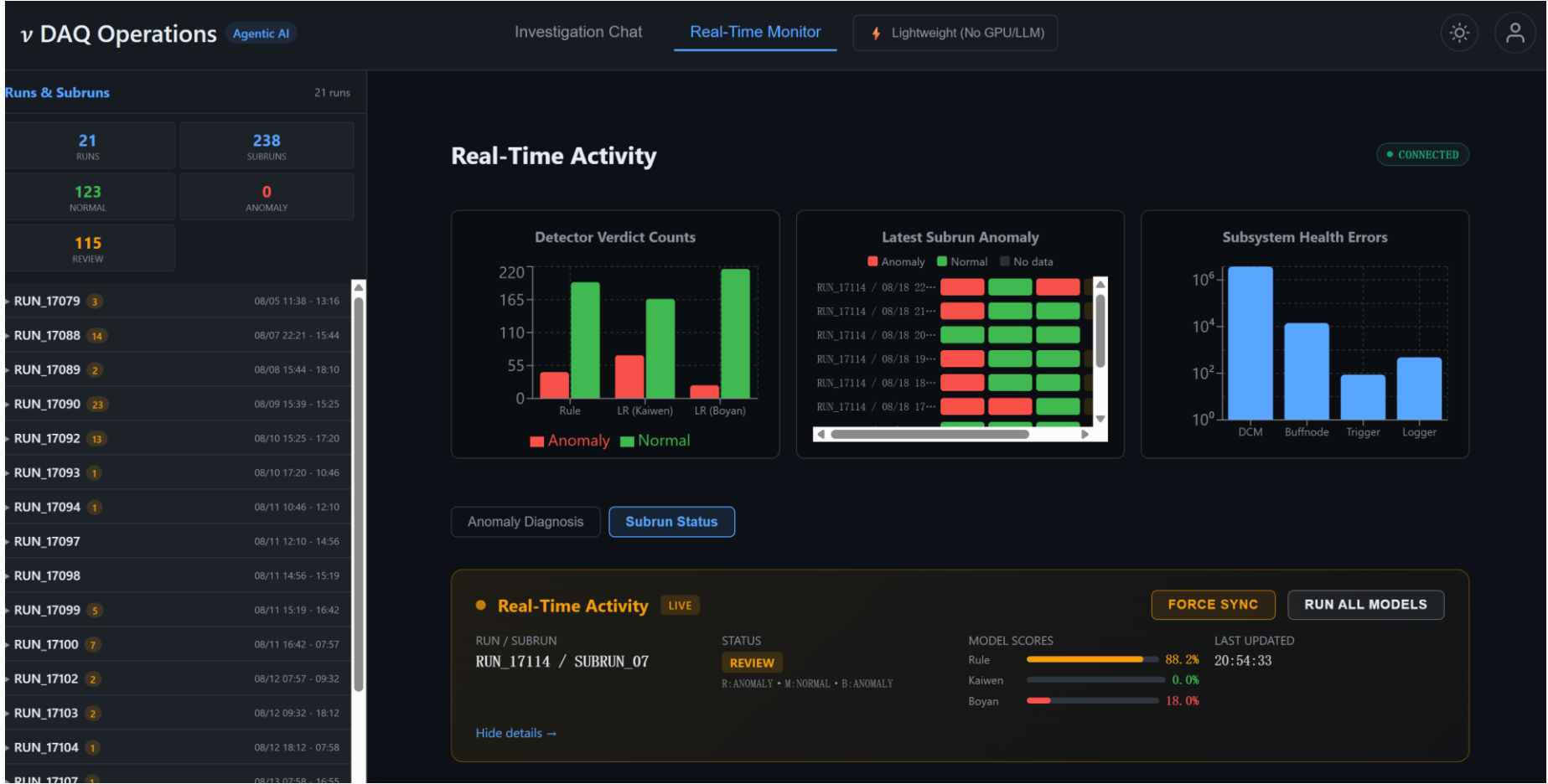


- Flexible auto-classification template, adaptability to changes in data formats;
- Also $5\times$ faster than Kaiwen's model, $\sim 3s$ per subrun.

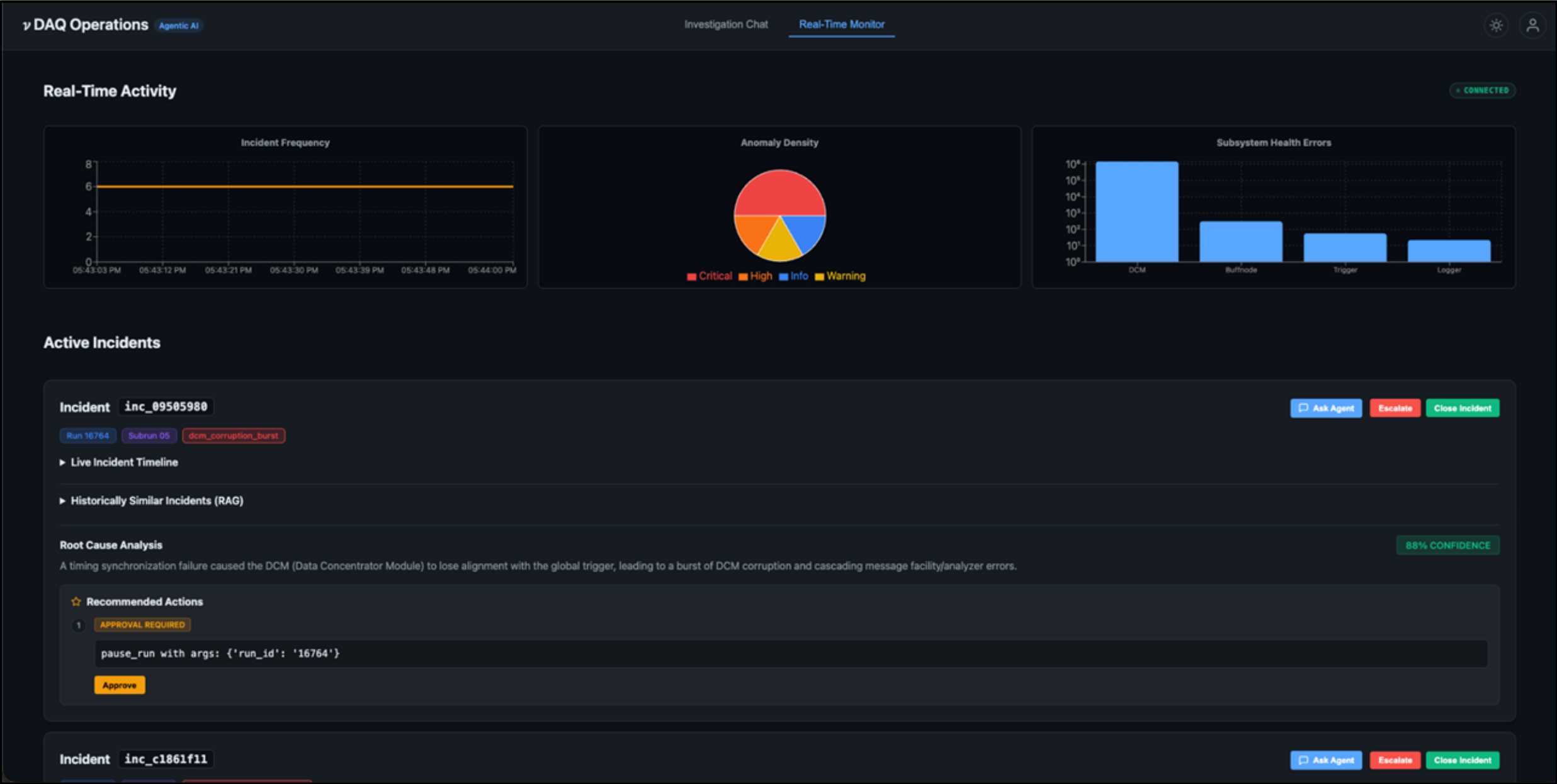


Realtime Anomaly Detection Pipeline

- Full pipeline runs successfully on UCI servers accessible to NOvA shifters, ready to be deployed on the Fermilab cluster
- Currently call a H200 GPU server at UCI for LLM inference
- Working on implementing it at NERSC to leverage GPU computing resources



Suggested Solution



Summary

- Key Accomplishments
 - Established NOvA data preparation workflow
 - Matched DAQ logs with ECL-based labels
 - Developed three anomaly detection approaches
 - Demonstrated real-time dashboard prototype
 - Available to NOvA shifter for testing
- Next Steps
 - Broaden testing with NOvA shifters and incorporate human operational feedback
 - Move LLM inference to NERSC GPU resources
 - Quantify model performance using operational metrics
 - Extend the framework toward ProtoDUNE and DUNE