LogPilot: Intent-aware and Scalable Alert Diagnosis for Large-scale Online Service Systems

Zhihan Jiang[1], Jinyang Liu[2], Yichen Li[2], Haiyu Huang[1], Xiao He[2],

Tieying Zhang[2], Jianjun Chen[2], Yi Li[2], Rui Shi[2], Michael R. Lyu[1]

[1]The Chinese University of Hong Kong,

[2]ByteDance

ARISE
Automated Reliable Intelligent Software Engineering

香港中文大學
The Chinese University of Hong Kong

ByteDance

Background

Online Services are Everywhere

Cloud services

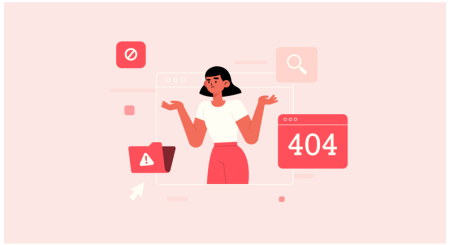
To-Business services

To-Customer services

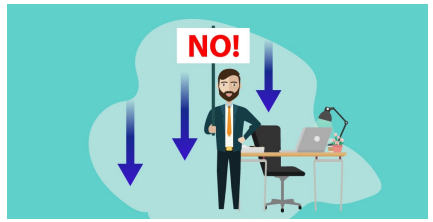
Service Reliability

Service reliability is critical for both service providers and users!

An error



Service Issues



User Dissatisfaction

Revenue Loss

News > Business

Amazon 'missed out on $34m in sales during internet outage'

The e-commerce giant generates $9,615 in sales per second – but not when it's website is down

Ben Chapman • Tuesday 08 June 2021 16:54 • 1 Comments

YOUTUBE · **Published** December 14, 2020 9:43am EST

Google lost $1.7M in ad revenue during YouTube outage, expert says

YouTube and other Google services, such as Gmail, suffered outage Monday morning

On October 29, a single configuration error crashed Microsoft Azure for eight hours, causing up to $16 billion in losses. The outage affected major companies worldwide and exposed how two providers control 55% of the cloud market, creating serious risks for businesses everywhere.

Service Monitoring and Alerting

Metric Instrumentation Statements

```
pod-a: manage.py:
self._check_initialized("http_requests_total")
self.client.emit_counter("http_requests_total", 1,
{"response_code": response_code, "instance": "pod-a",
"path": path, "account_id": account_id})   ...
```

pod-a:http_requests_total *pod-b:http_requests_total* *pod-c:http_requests_total*

Instance-Level Metrics

PromQL

```
sum by (gateway_id) ( rate ( http_requests
_total { account_id="aid1", path="/api/v1/
users", response_code=~"2.*" } [1m] ) )
```

api_users_endpoint:successful_throughput_1m

```
[2024-07-30T12:00:00Z,  17.1]
[2024-07-30T12:00:15Z,  17.5]
...
```

Service Level Indicators (SLIs)

Service Monitoring

Design SLOs and Alerting Rules

```
name: APIServiceSLOAlerts
rules:
- alert: APIServiceThroughputCritical
  expr:api_users_endpoint:successful_thro
ughput_1m : rate_1m <= 5
  for: 5m
  labels:
    severity: critical
    service: api
```

Triggered Alert

```
"promql": ... ,
"legend": ... ,
"alert_time": 1722350160000,
"Threshold": 5.0,
"current_value": 0.1
```

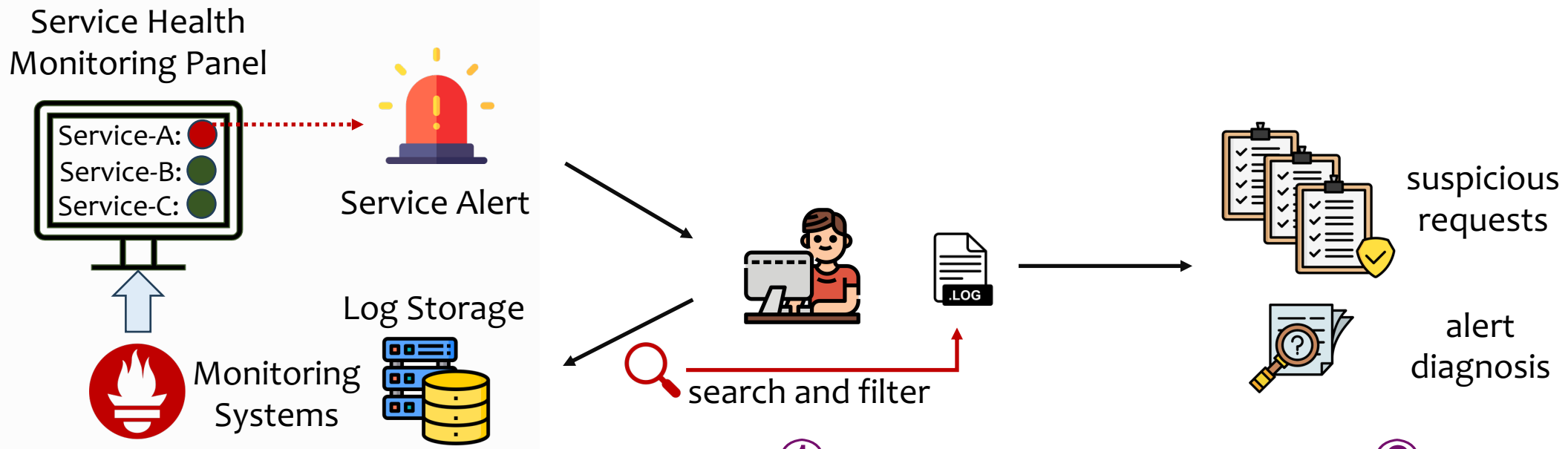
Analysis and Diagnosis by SREs

Service Alerting

Online Service Systems

Monitoring Dashboard

Alert Diagnosis Practice

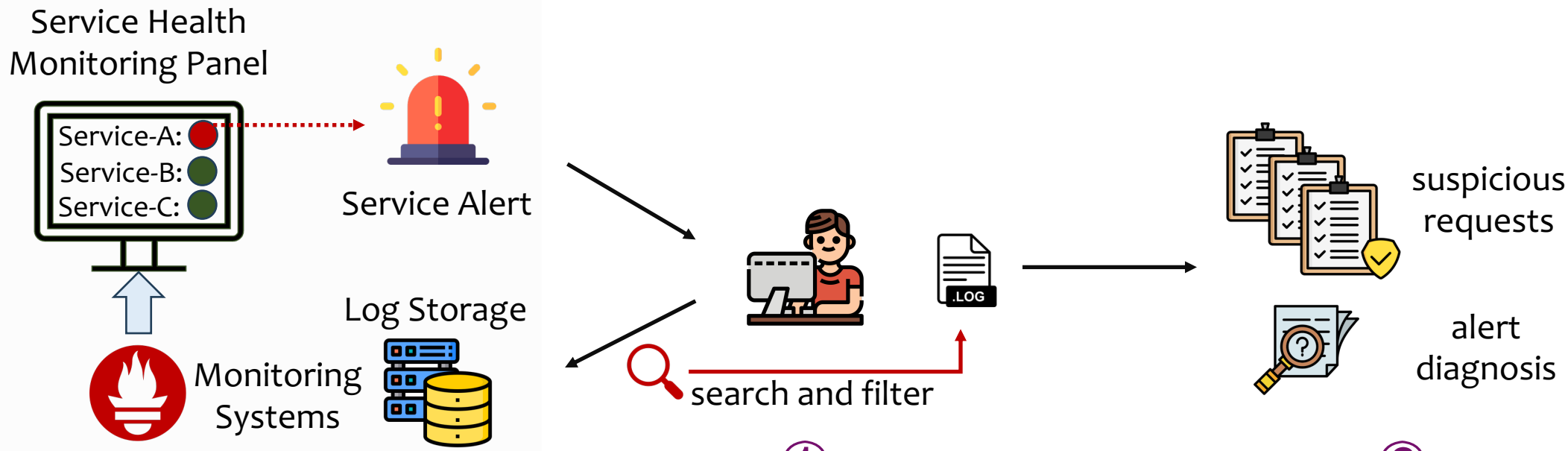


Service Alerting

① Alert-driven Log Scoping

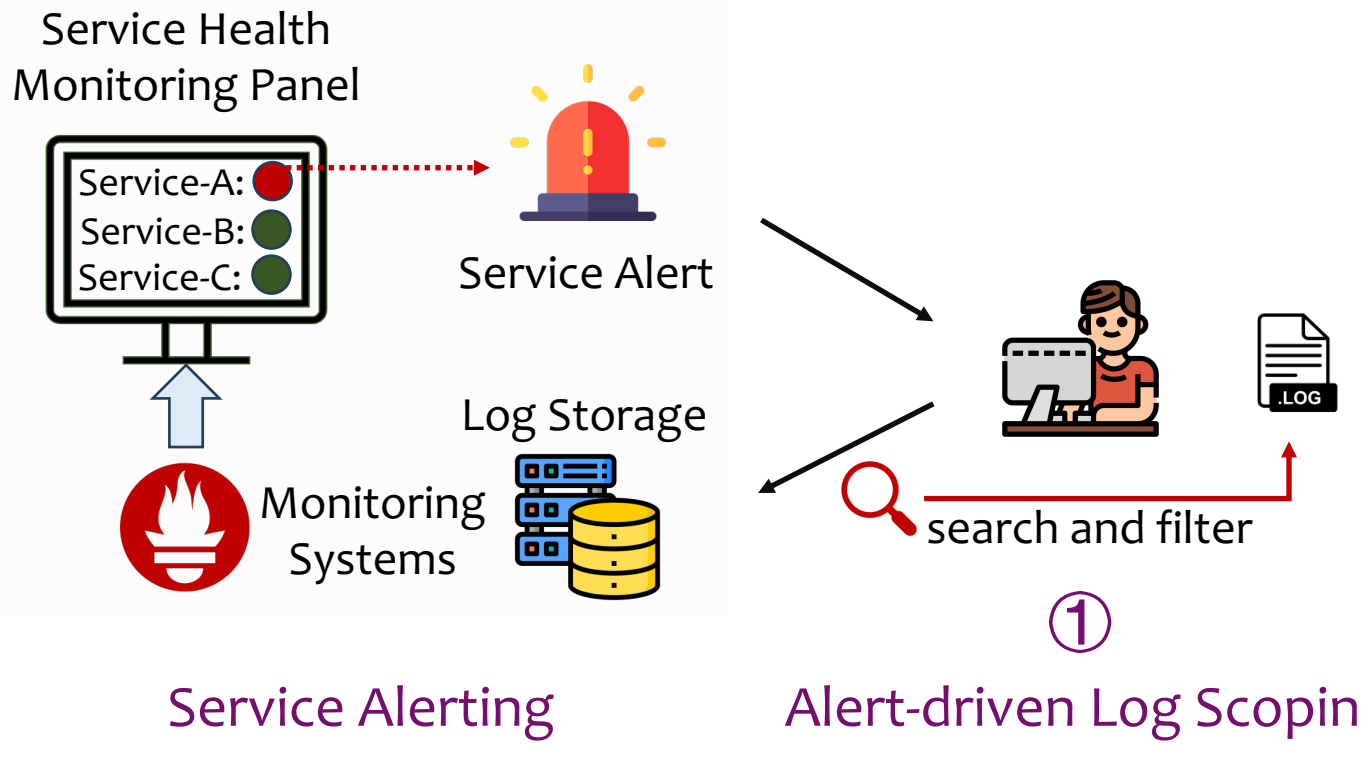
② Investigative Log RCA

Alert Diagnosis Practice



There are significant gaps between existing methods and the production requirements!

Motivation



Firing Alert

"**promql**": "sum by (gateway_id) (rate(requests_total{account_id=~"aid1", response_code!~"2.*|100|403"}[1m]))"
"**legend**": gateway_id=gid1
"**alert_time**": timestamp (ms)

{ "timestamp": "2025-07-30T15:42:17Z", "level": "ERROR", "log_path": "/var/logs/gateway/error.log", "gateway_id": "gid1", "x_account_id": "aid1", "status_code": "502", "x_request_id": "req-a1b2c3d4e5", "message": "Gateway timeout while routing request to backend service."}

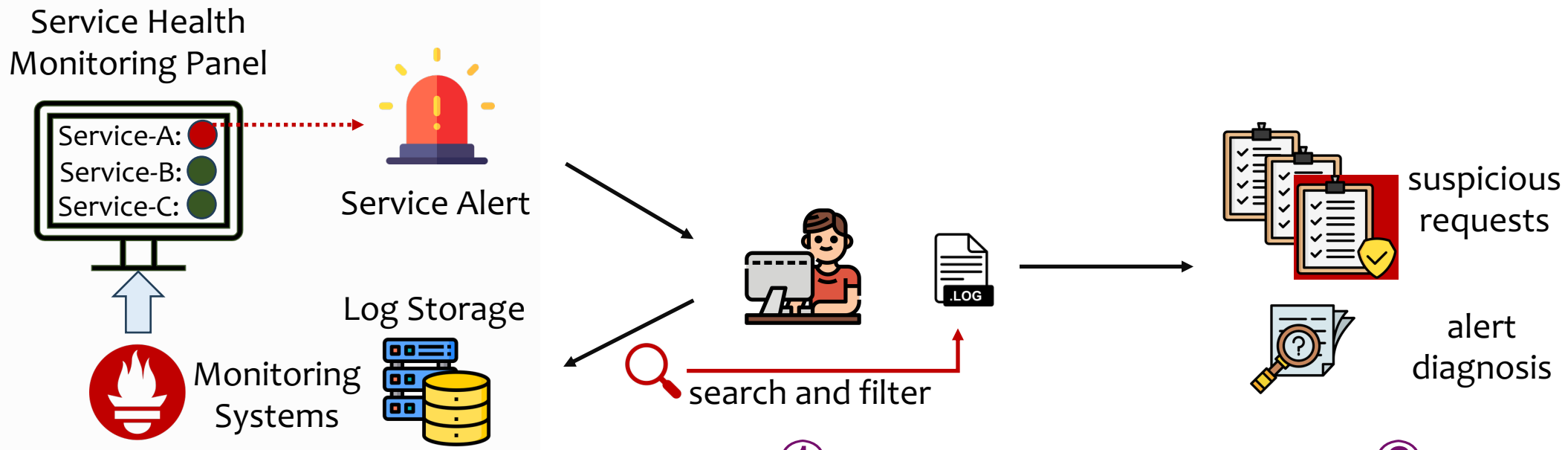
Phase ①:

Existing Approach:
- Keyword search
- Log anomaly detection

Limitation:
- Lack of alert intent awareness

Opportunity:
- Linking PromQL and log fields for correlation

Motivation



① Alert-driven Log Scoping

② Investigative Log RCA

Service Alerting

Phase ②:

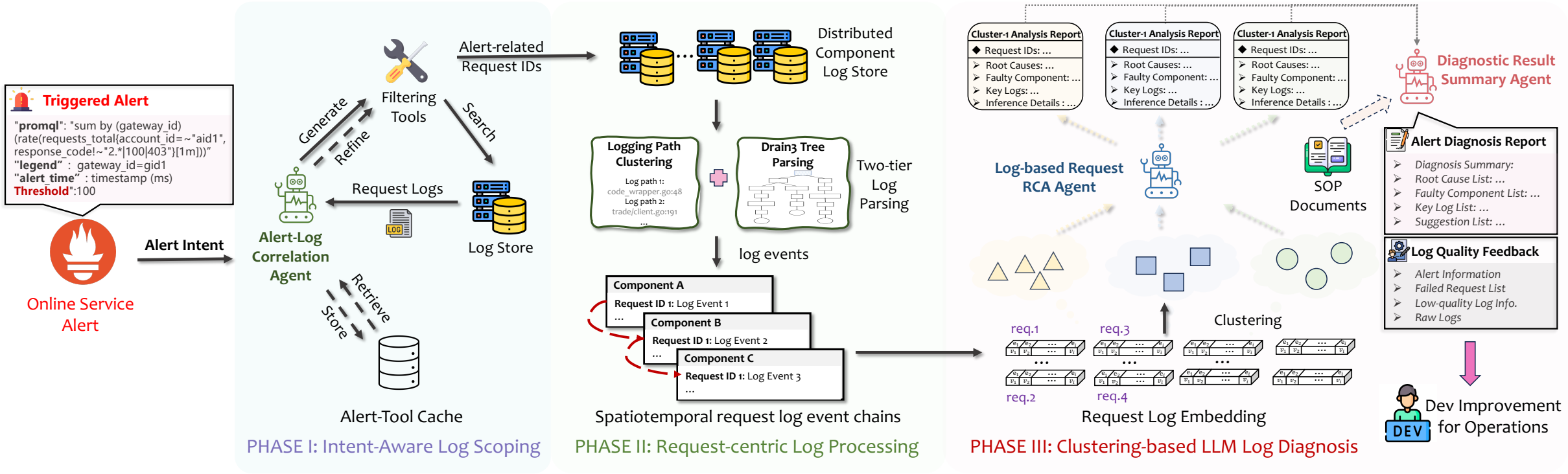
Existing Approach:
- Pattern learning
- Naïve LLM invocation

Limitation:
- Scale and complexity
- Context limitation

Opportunity:
- Similar request execution patterns

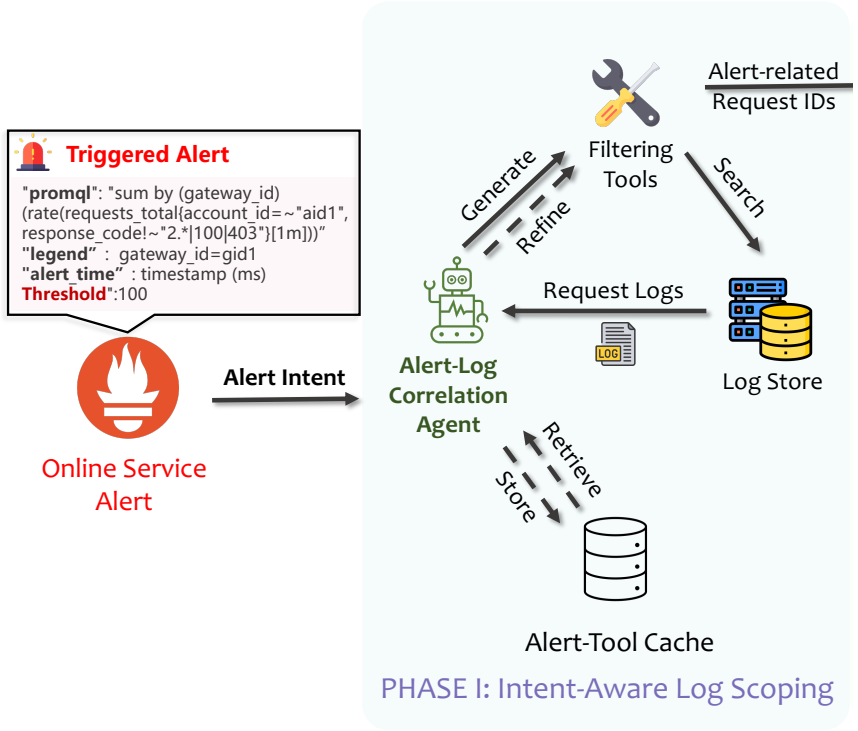
Methods

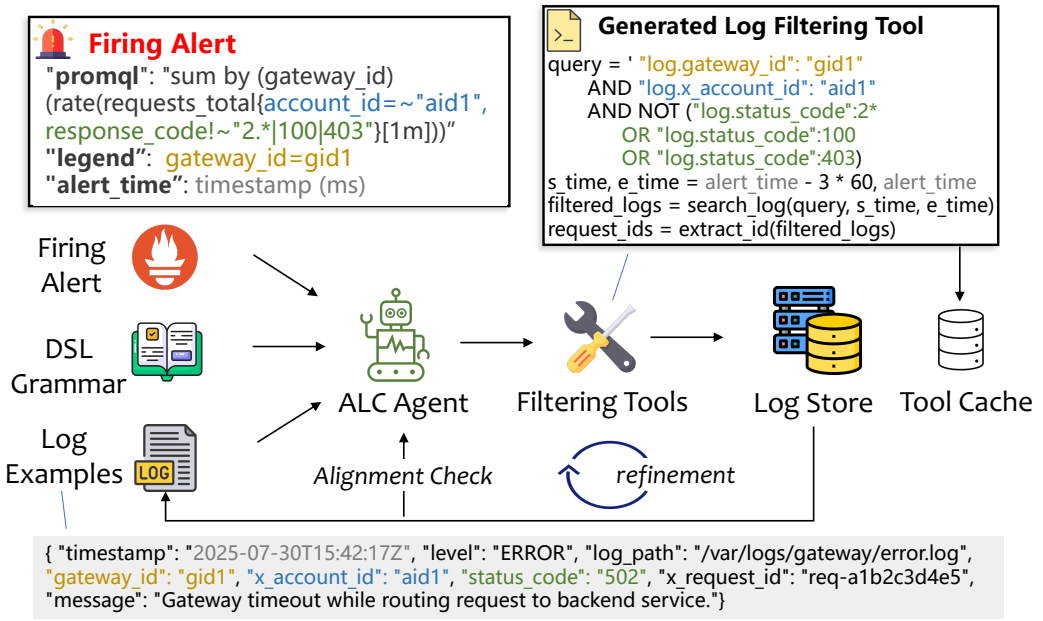
LogPilot: an intent-aware and scalable LLM-based framework for automated log-based service alert diagnosis



Methods: Phase I

LogPilot: *an intent-aware and scalable LLM-based framework for automated log-based service alert diagnosis*





① Tool Generation

Input:
- Alert definition
- Log Examples
- DSL Grammar

Output:
- Generated tool (a script with DSL query)

② Tool Refinement

Input:
- Alert definition
- Generated tool
- Execution results / errors

Iterative Output:
- Refined tool

③ Tool Caching

Methods: Phase II

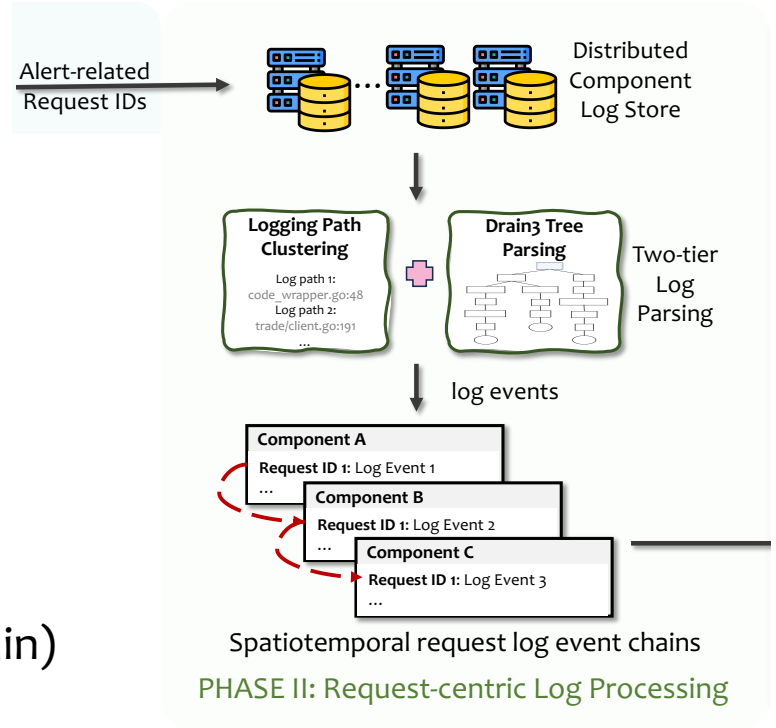
LogPilot: *an intent-aware and scalable LLM-based framework for automated log-based service alert diagnosis*

Request-centric Pre-processing

① Two-tier Log Parsing

Coarse-grained:
- Logging path clustering

Fine-grained:
- Heuristical clustering (Drain)



② Request Log Event Chain Construction

Spatiotemporal Log Event List
- Deduplicated by log templates
- Spanning multiple components
- Ordered by timestamp

Methods: Phase III

LogPilot: *an intent-aware and scalable LLM-based framework for automated log-based service alert diagnosis*

Log-based Request RCA Prompt

You are an expert specializing in log analysis and root cause diagnosis for online service systems.
Context
You are diagnosing a production alert based on the following data:
1. **Triggering Alerts**: *{detailed alert information}*
2. **Alert-related Request Log Chain**: *{request log chain across distributed components}*
3. **Service Component Information**: *{service component functionality}*
Instruction
1. **Analyze the Request Execution Path**: Analyze the Request Log Chain from start to finish. Find any abnormal behaviors like high latency, error status codes, or unexpected responses.
2. **Identify the Root Cause**: Pinpoint the specific service component that is the source of the failure. Isolate the key log entries that serve as direct evidence of the root cause.
3. **Handling Low Logging quality**: If the provided logs are insufficient to determine the root cause, or if there are clear inconsistencies, please provide the feedback with detailed information.
4. **Generate the Analysis Report**: Synthesize your findings into the RCA Report Format specified below. Your reasoning should be clear, concise, and directly supported by the provided log data.
Output Format
{analysis report format} and {feedback format}

Diagnostic Result Summary Prompt

You are an expert specializing in alert diagnosis for online service systems.
Context
You are diagnosing a production alert with the following information: *{detailed alert information}*
An automated analysis system has analyzed the root cause issues of the alert-related service requests, with several scattered request clusters: *{analysis reports from all request clusters}*
Here are also some related standard operation procedure documents: *{retrieved historical SOP}*
Task
1. **Review report and summarize root causes**: Review all analysis reports, merge root causes that are fundamentally identical in nature, and summarize the root causes for the alert.
2. **Generate final alert diagnosis report**: Synthesize your findings into the final diagnosis report for this alert following the format specified below.
3. **Generate log quality feedback**: If there are any low log quality issues in the analysis reports, please also give a summarized feedback for developer to improve.
Output Format
{alert diagnosis report format} and {log quality feedback format}

Example prompt for

- *log-based request RCA agent*
- *Diagnostic result summary agent*

① Request Log Embedding

- Convert each request's log chain into a vector representation

② Request Clustering

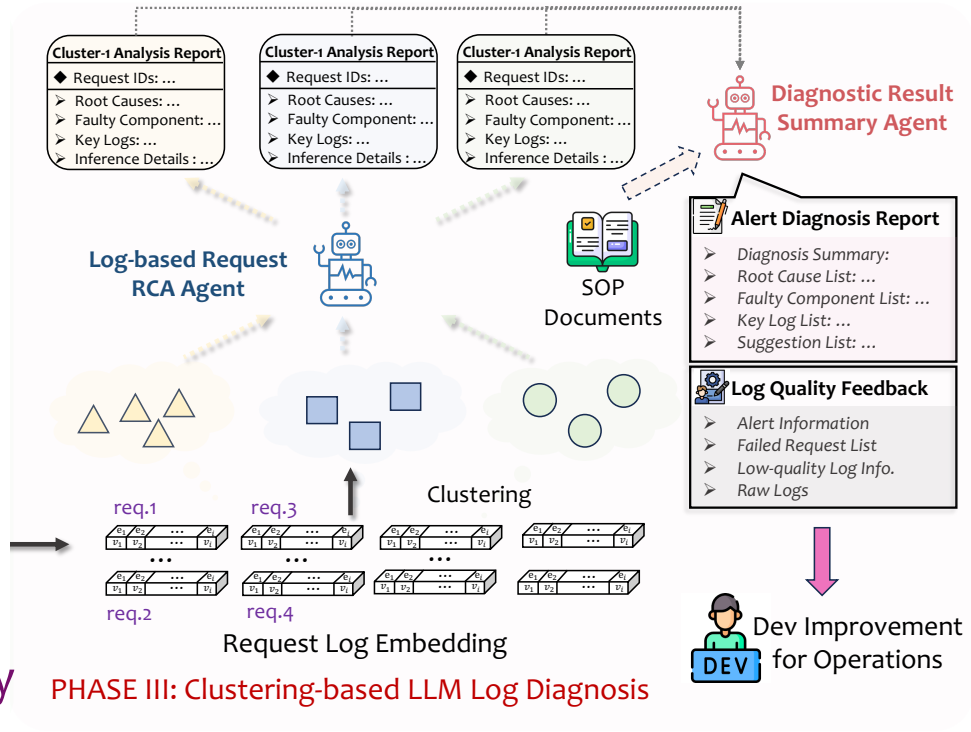
- Group requests with similar execution patterns

③ Log-based Request RCA

- Select one representative request per cluster for RCA

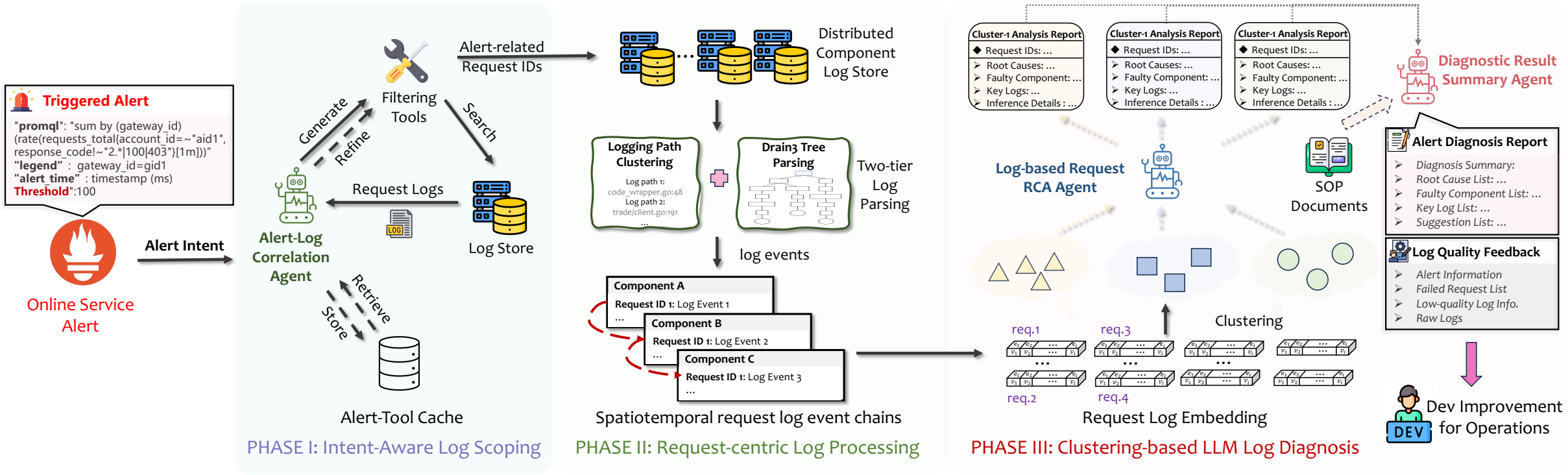
④ Diagnostic Result Summary

- Aggregate results to synthesize final comprehensive report



Methods

LogPilot: an intent-aware and scalable LLM-based framework for automated log-based service alert diagnosis



Evaluation

Research Questions

- RQ1: How **effective** is LogPilot in alert diagnosis?
- RQ2: How **robust** are the components of LogPilot?
- RQ3: What is the **efficiency** and **cost** of LogPilot?

Evaluation Metrics

- Root Cause Summarization
  - Automated Evaluation: **METEOR** and **ROUGE**
  - Human Evaluation: **Usefulness**
- Root Cause Localization
  - **Exact Match**
  - **Top-3 Accuracy**

Datasets

- Real-world data from June 15 to July 15, 2025
  - Total of 202 alerts, 161 unique root causes
- Manual annotation of alert root causes

TABLE I: Evaluated alerts and unique root causes (URCs)

Datasets	Service $\mathcal{A}$	Service $\mathcal{B}$	Service $\mathcal{C}$	Service $\mathcal{D}$	Total
# Alerts	35	40	52	75	202
# URCs	27	29	43	62	161

Baselines

- [Naïve] LLM with sampling
- [ICLR'25] RCA Agent
- [EuroSys'24] RCACopilot

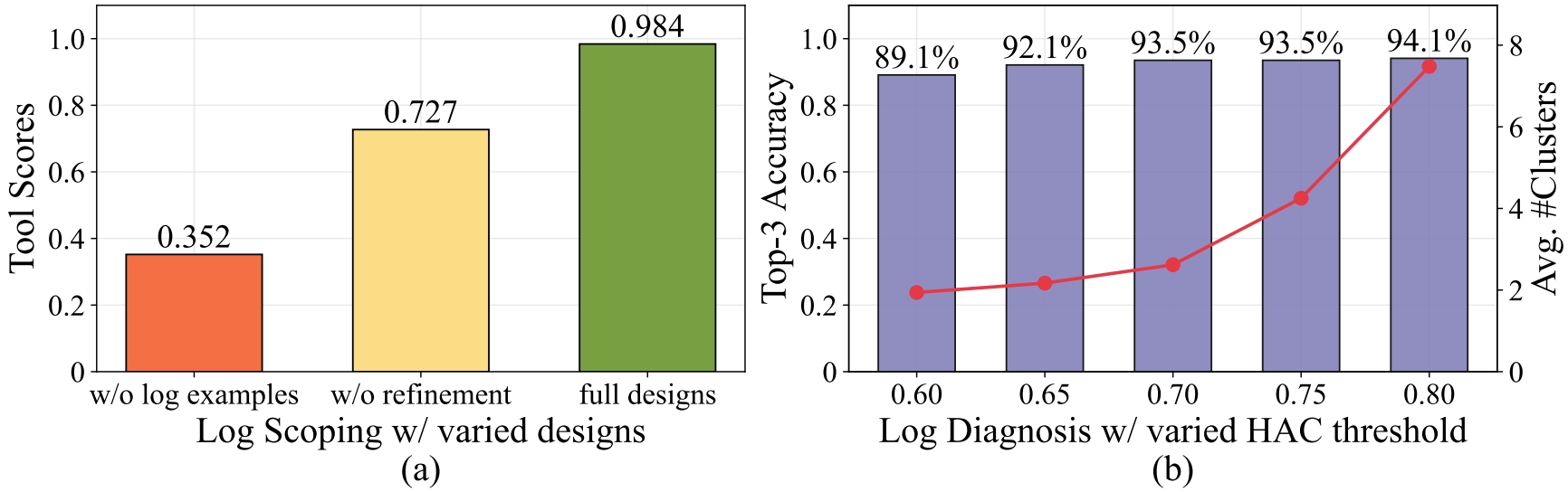
RQ1: Effectiveness

TABLE II: Root cause analysis results from both *summarization* and *localization* dimensions for each service.

System	Model	Root Cause Summarization				Root Cause Localization	
		ROUGE-1	METEOR	Semantics	Usefulness	Exact Match	Top-3 Acc.
Service $\mathcal{A}$	LLM w/ BS	0.381	0.303	0.713	0.454	0.343	0.543
	RCACopilot	0.468	0.374	0.795	0.563	0.429	0.686
	RCA Agent	0.475	0.369	0.812	0.640	0.457	0.743
	LogPilot	**0.587**	**0.522**	**0.894**	**0.823**	**0.743**	**0.943**
Service $\mathcal{B}$	LLM w/ BS	0.375	0.322	0.719	0.473	0.400	0.475
	RCACopilot	0.483	0.397	0.804	0.570	0.450	0.725
	RCA Agent	0.512	0.408	0.831	0.668	0.475	0.775
	LogPilot	**0.602**	**0.497**	**0.913**	**0.851**	**0.725**	**0.950**
Service $\mathcal{C}$	LLM w/ BS	0.352	0.309	0.688	0.427	0.346	0.481
	RCACopilot	0.413	0.346	0.755	0.494	0.423	0.673
	RCA Agent	0.432	0.367	0.808	0.588	0.442	0.747
	LogPilot	**0.538**	**0.460**	**0.852**	**0.785**	**0.712**	**0.885**
Service $\mathcal{D}$	LLM w/ BS	0.401	0.356	0.748	0.492	0.387	0.533
	RCACopilot	0.517	0.437	0.821	0.609	0.507	0.733
	RCA Agent	0.538	0.446	0.846	0.643	0.560	0.787
	LogPilot	**0.631**	**0.539**	**0.924**	**0.895**	**0.800**	**0.960**

- LogPilot outperforms all baselines in both *Root Cause Summarization and Localization*.

RQ2: Robustness

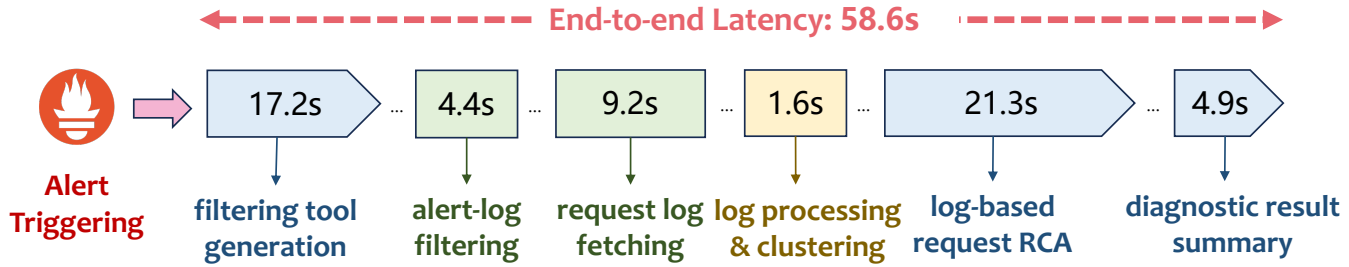


- LogPilot attains an average score of 0.984 in log-filtering quality evaluation.
  - Both *log examples* and *tool refinement* are important.

- LogPilot maintains a stable Top-3 root-cause acc. of ~94% across varying θ_{HAC} settings.
  - A high θ_{HAC} produces more fine-grained clusters, leading to additional LLM analysis.
  - A low θ_{HAC} results in fewer clusters and may cause a loss of diagnostic information.

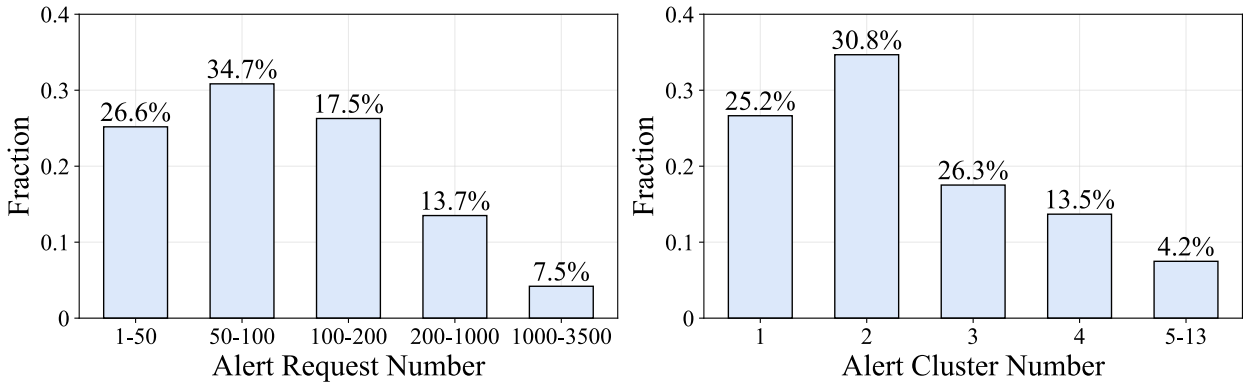
RQ3: Efficiency & Cost

Average stage-wise latency



- LogPilot maintains an end-to-end latency under one minute (58.6 s).

Distribution of request and cluster number

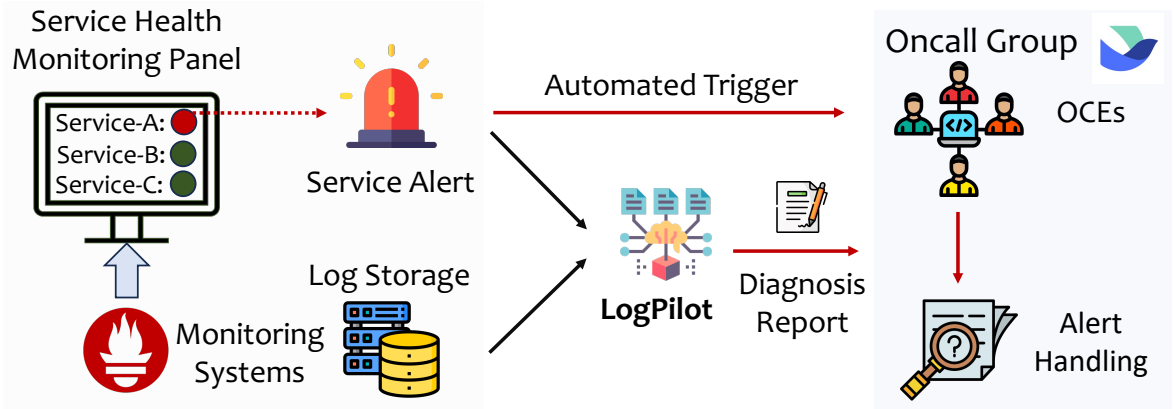


- LogPilot cuts the average number of analyzed requests from 198.65 to 2.56 (98.71% reduction).

- LogPilot uses 63.73K prompt and 8.08K response tokens on average, costing only $0.074 per alert.

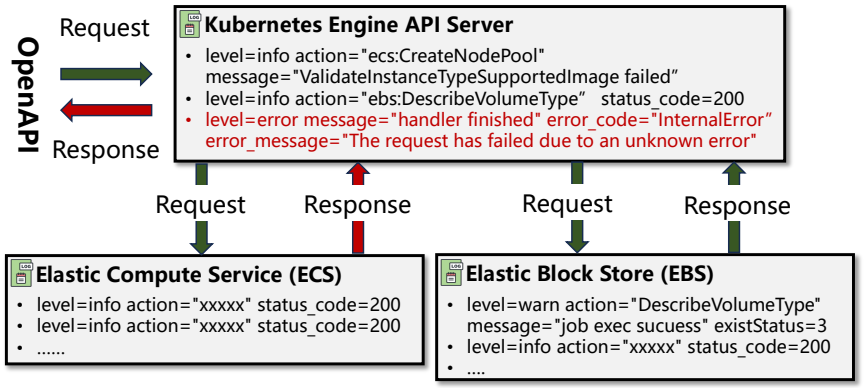
Deployment Experience

Production Deployment Workflow



- Deployed in **12 production services** within Volcano Engine Cloud.
- Analyzes 3500+ production alerts.
- Overall acceptance rate for the generated reports is **84.21%**.

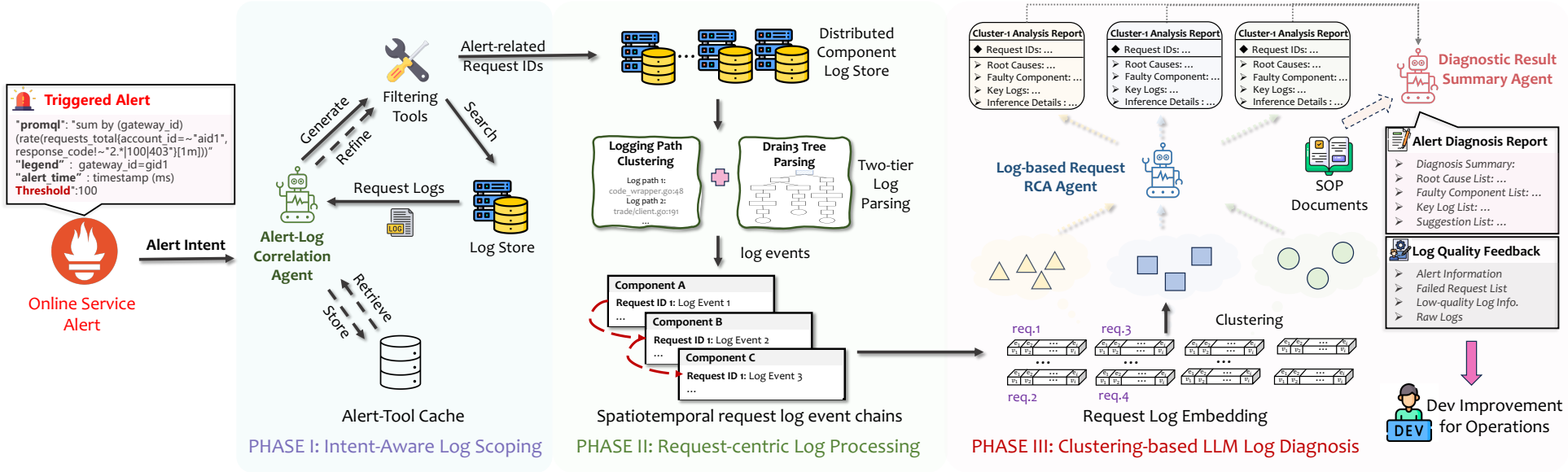
Identification of Inconsistent Logging Cases



- Improves log quality in production systems.
- Bridges the gap between **Development (Dev)** and **Operations (Ops)**.

Conclusion

- LLMs show strong potential for automated alert diagnosis, but a significant gap remains between existing log-based research and real-world production requirements.
- We design **LogPilot**, an intent-aware, scalable LLM-based framework that precisely correlates and efficiently analyzes massive logs for service alert RCA.
- Production deployment proves its practicality and offers guidance for future LLM-driven AIOps solutions.



Q&A

LogPilot: Intent-aware and Scalable Alert Diagnosis for Large-scale Online Service Systems

Thank you for listening

Q & A

Contact: zhjiang@link.cuhk.edu.hk

Pre-Print

ARISE Lab

ARISE Automated Reliable Intelligent Software Engineering

香港中文大學 The Chinese University of Hong Kong

ByteDance