

# From Plugin To Platform — Building an Observability Suite in OpenSearch Dashboards





**Prabhakar Vemparala**

Engineering Manager  
Observability



Freshworks

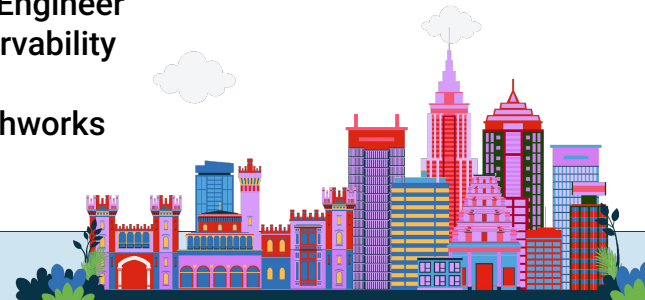


**Hari Krishnan Prabhakar**

Lead Engineer  
Observability



Freshworks



# Freshworks at a glance



**2010**

Founded



**FRSH**

IPO September 2021



**\$838M**

2025 Annual Revenue Guidance



**~4,500+**

Employees



**75,000**

Total Customers



**Recognition**

Winner of AI Excellence Awards, Best SaaS in AI  
and Winner of Customer Excellence Awards

# Freshworks Observability Platform



**Logs**



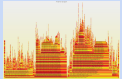
**Metrics**



**Traces**



**Frontend Observability**



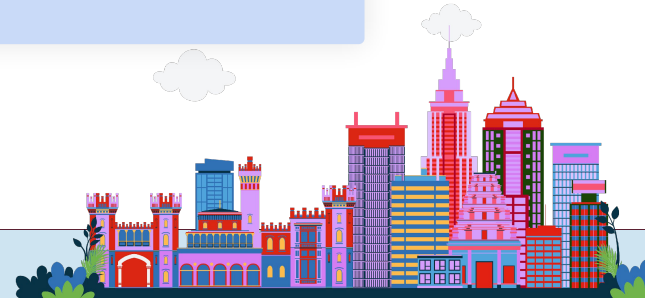
**Profiles**



**Synthetic Monitoring**



**Alerts**



# Upcoming Freshworks Talks @ OpenSearchCon 2026

## ODD is the new ELK - How ODD Powers Log Management at 400TB-Per-Day Scale - Rashmi Ramanathan & Buden Shaik, Freshworks

📅 Tuesday June 16, 2026 10:00am - 10:20am IST

📍 206 (Level 2)

### Speakers



#### Rashmi Ramanathan

Principal Engineer, Freshworks

Principal Engineer at Freshworks with 14 years of experience and built Freshworks' Distributed Tracing Platform. Eai



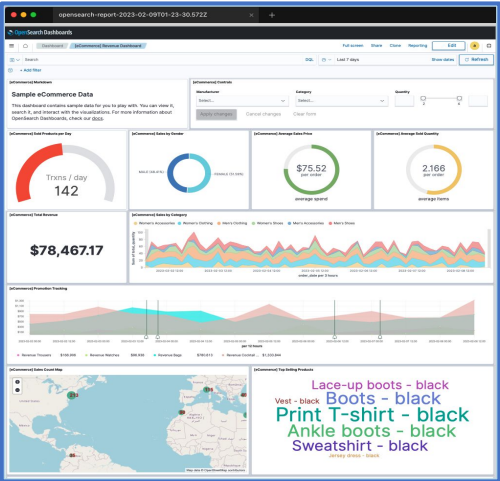
#### Buden Shaik

Staff Software Engineer, Freshworks

I'm a Staff Software Engineer with over 10 years of experience in native infrastructure. Throughout my career, I've focused on



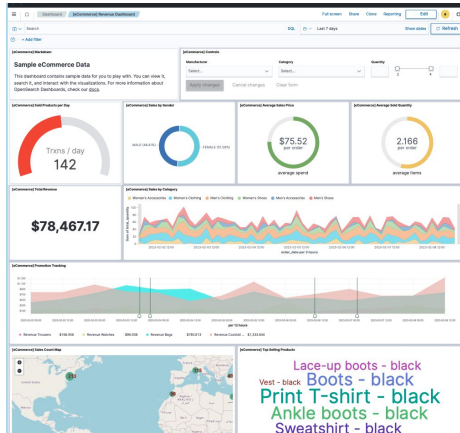
# Traditional Architecture



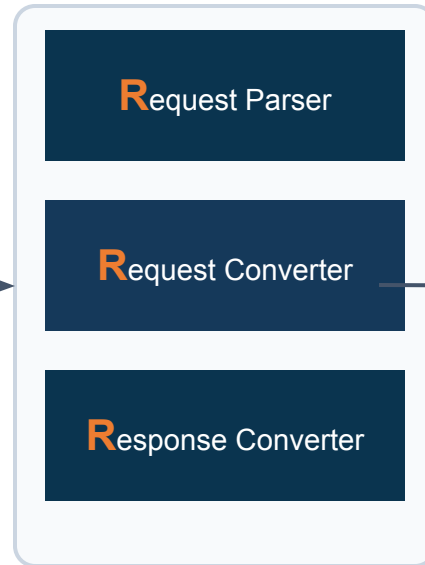
## OpenSearch Dashboards



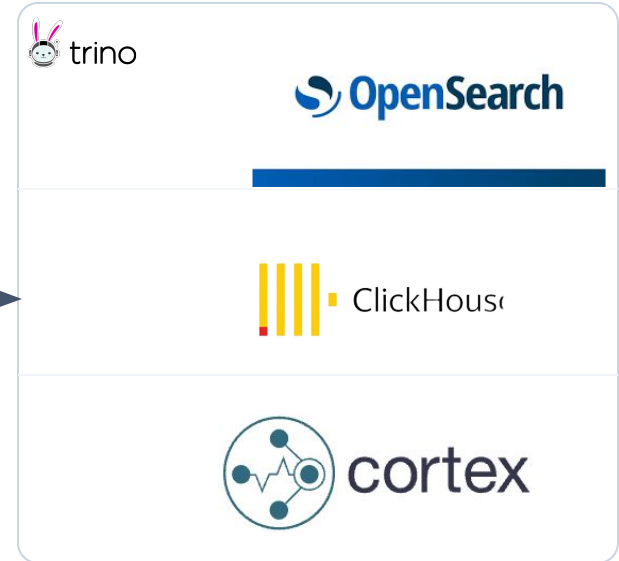
# Enhanced Architecture



OpenSearch Dashboards



Adapter Service



Products

Filter data using query syntax Eg., e\_an: 'MeaningfulPaint'

Quick Select Last 7 days

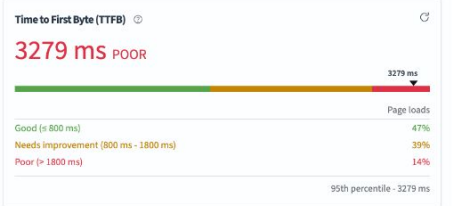
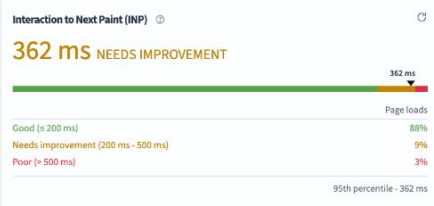
Refresh

+ Add filter

Definitions

Core Web Vitals

Aggregation p95



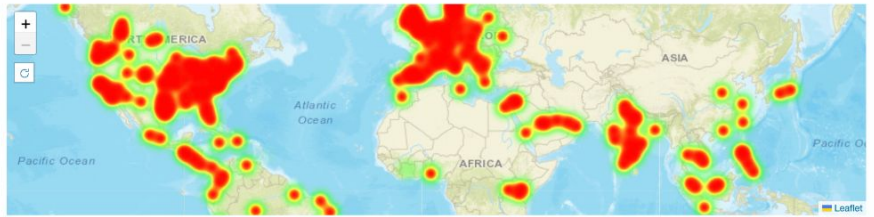
Route wise aggregated metrics

Route Name	Page Views ↓	LCP	CLS	TTFB	INP	Meaningful Paint
	8.81 M ↑ 1.10%	7.87s ↑ -0.67%	0.476 ↓ -0.24%	2.52s ↓ 1.95%	339.52ms ↓ 0.29%	14.47s ↓ 1.03%
	6.27 M ↓ -0.13%	13.50s ↓ -1.12%	0.446 ↑ -0.83%	2.23s ↓ 5.44%	483.51ms ↓ 0.24%	9.12s ↑ -1.59%
	4.02 M ↑ 2.24%	7.46s ↑ -2.14%	0.221 ↓ 0.25%	4.20s ↑ -0.95%	186.94ms ↑ 1.03%	7.46s ↑ -1.85%
	3.86 M ↑ 1.82%	10.43s ↑ -1.18%	0.358 ↑ -1.76%	4.82s ↓ 4.34%	369.32ms ↑ -0.45%	12.30s ↓ 2.81%
	671.36 K ↓ -1.50%	3.15s ↓ 1.08%	0.311 ↓ 1.54%	1.64s ↑ -0.50%	142.70ms ↑ -0.80%	3.64s ↓ 0.18%

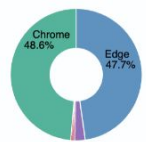
Rows per page: 5

Traffic & Demographics

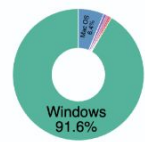
Traffic Distribution



Browser Usage



Operating System



### Steps Executed

Total Duration: 32 secs

Step name	Result	Duration
Login to freshservice account	Success	
Navigate to Developer Portal (AMP)	Success	
Validate whether amp is loading from gallery	Success	
Validate whether developer analytics is loading in AMP	Success	

## 2. Navigate to Developer Portal (AMP)

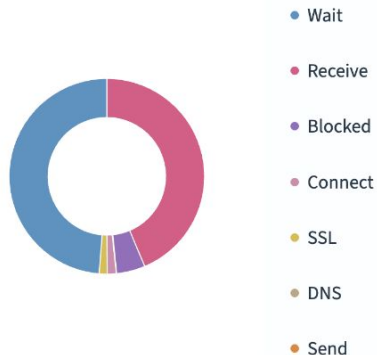
Step: Step 2 of 4  
 Status: Success  
 Location: frankfurt

### Console

```
[Report Only] Refused to connect to 'https://o113506.ingest.sentry.io/api/1828702/envelope/?sentry_key=ef29afcccb304c3ab243b5bde69f6374&sentry_version=7&sentry_client=sentry.javascript.vue%2F7.5.0' because it violates the following Content Security Policy directive: "connect-src 'self' http://localhost:10001".
```

```
[Report Only] Refused to evaluate a string as JavaScript because 'unsafe-eval' is not an allowed source of script in the following Content Security Policy directive: "script-src 'self' https://o113506.ingest.sentry.io/".
```

### Timing breakdown



### Metrics

Metric	Value	Definition
1.23 s	3.98 s	0.00316
FCP	LCP	CLS
1.94 s	14 s	4.33 s
DCL	Step Duration	Load

### Object weight

Total size: 8.24 MB

### Object count

Total: 609

Object Type	Weight	Count
JS	~1.5 MB	301
XHR	~0.5 MB	117
CSS	~0.5 MB	93
HTML	~0.1 MB	25
Image	~0.1 MB	42
Other	~0.1 MB	29
Font	~0.05 MB	2



## OSD Core

HTTP • Lifecycle • State Containers  
Saved Objects • Auth

## Plugins

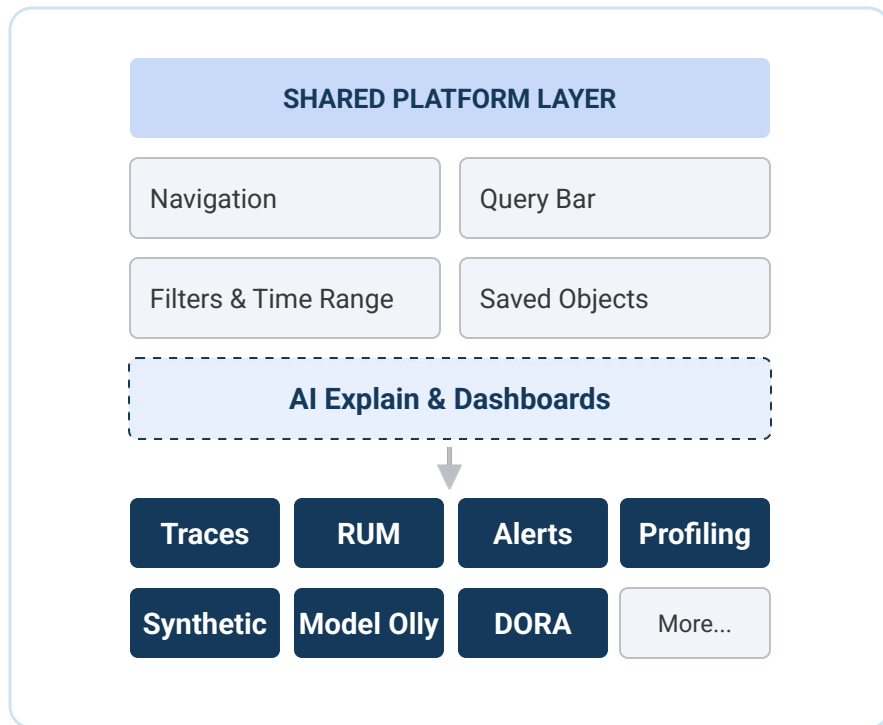
Discover • Dashboard • Dev Tools • Alerts

## Packages

Query Grammar • Types • Utilities  
Analytics • i18n



# Why One plugin ?



Many independent plugins would lead to "**plugin sprawl**."

Signals needed **shared context more than isolation**.

## The Decision

- Keep **observability plugin** as the platform shell; each signal is a module inside.



# How are we managing with Single Plugin ?

## Independent Modules: Each Module (Traces, RUM, Alerts, etc.)

- Own routes, components, and APIs
- Domain-based boundaries & lazy loading

## Shared Context = Unified Experience

- **State management in URL:** Filters, product, and time range stay in sync based on App key.
- **Auth:** Common auth & environment headers across modules via plugin lifecycle hook.
- **Reusable UI Blocks:** Gantt, React Flow, Flamegraphs in OSD packages.
- **Custom Query DSL:** Extended PEG grammar for autocomplete and suggestions.
- **Shared Contracts:** Common queries building pattern and request/response handlers.



## Shared Across Modules

Traces

RUM

Alerts

Discover

AI Explain Service

**Shared Plugin:** explainWithAi

Streaming Engine | UI Panel | Cache Logic

## Shared Services

### Problem:

AI explanation needed in Discover, Sherlock, and RUM.

### Solution:

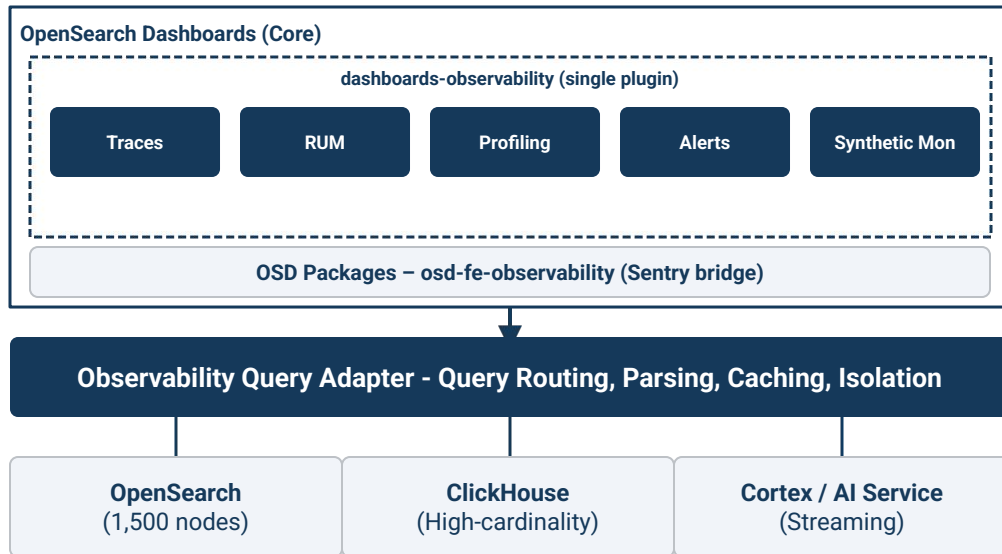
Shared plugin **explainWithAi** that owns the streaming engine, UI panel, and cache logic.

### Architecture Pattern:

*When functionality spans multiple products: Build a capability once, consume everywhere.*



The plugin architecture allows for an observability control plane, not just a frontend for OpenSearch.



## Key Points

- Data lives in multiple systems (OpenSearch, ClickHouse, Cortex).
- **External Client pattern abstracts backend differences.**
- Adapter service provides query federation and intelligent routing.
- Backend abstraction and flexibility to evolve storage.



## Lessons Learned

### Build Platforms, Not Features

- One plugin can host many products while preserving a unified experience.

### Leverage the OSD Platform

- Discover, Dashboards, and Embeddables accelerated development.
- Reusing platform services was more valuable than custom solutions.

### Design for Scale

- Shared capabilities scale better than duplicated implementations.

### Test Through Boundaries

- Cypress with stubbed responses provided reliable validation.

## Patterns That Scaled for Us

### Architectural Patterns

- One plugin, many modules; Lazy loading by route.
- Horizontal plugins (AI Explain) & External Client + Router.
- Embeddables & Saved Objects for composability.

### Path Forward

- Expand reusable workflows across observability modules.
- Explore workspace-aware observability experiences.
- Improve startup performance and platform scalability.

**The OSD plugin architecture is more than an extension mechanism. With the right patterns, a single plugin can evolve into a platform—bringing together multiple signals, backends, and shared workflows in one unified experience.**



# OpenSearchCon

INDIA

