



## SOLUTION BRIEF

# Chronosphere Cloud Native Observability



## The challenge

Cloud native environments fail without cloud native observability. Traditional monitoring tools used for older cloud infrastructure and monolithic applications fall short in addressing the dynamic and complex nature of cloud native systems. And building a DIY observability platform from disparate open source tools adds another operational burden to already stressed engineering teams. As a result, organizations face a new set of challenges:



Observability data growth and costs are growing unsustainably - Microservices-based architectures emit 10x more data than traditional cloud architectures. As developers and operations teams seek more data to fine-tune performance and exceed customer expectations, costs skyrocket, often outpacing the actual value derived from observability efforts.



Application performance and reliability are degrading - Current tools lack the sophistication to provide comprehensive visibility into mission-critical operating data for cloud native environments. MTTx is high and getting worse and you're failing to meet SLAs to internal or external customers.



Engineer productivity is declining - Engineers find themselves enmeshed in manual tasks and troubleshooting rather than focusing on new feature development. This is further complicated by an overreliance on 'hero developers' and disjointed observability tools that don't align with the developer's mental model, leading to increased cognitive load, frustration, and staff turnover.

The bottom line is companies struggle with unplanned cost overruns, customers react negatively to a poor user experience, and engineers spend all their time firefighting instead of developing and shipping new features that grow the business.

**Observability data is growing faster than infrastructure and the scale has created bottlenecks for reliability, performance and cost. Current approaches are failing to keep pace.**



### Open-source monitoring is time consuming and unreliable as you scale

Many companies find that once they reach a certain scale, they begin to struggle with the reliability and complexity of their homegrown OSS observability stack. This results in unreliable telemetry data that's hard to find quickly when needed. The situation is further exacerbated when observability platforms fail to align with the developer's mental model, increasing the cognitive load. As a result, valuable engineering resources are consumed in managing these complex stacks, taking focus away from shipping new features that directly impact business value.



### Traditional tools struggle to keep up with cloud native

As organizations transition to cloud native, they often find that their legacy infrastructure and application performance management (APM) tools can't keep pace. The cost of the solution is outpacing the value it delivers. With the cost of these solutions often exceeding their utility, developers are compelled to drop data to manage expenses. This leads to staff wasting time on managing data overages rather than focusing on high-value tasks

# Solution:

Chronosphere is a cloud native observability platform, providing deep insights into every layer of your stack – from the infrastructure to the applications to the business. Site reliability, devops, and engineering teams worldwide rely on Chronosphere to provide them with real-time visibility and alerts to help them operate scalable, highly available, and resilient applications.

## A few ways we're different from other monitoring and observability solutions:



### Control costs

Maximize business value of observability in a predictable, cost-efficient manner through visibility and control



### Exceed customer expectations

Optimize performance and availability of critical services and provide real-time business insights



### Improve developer productivity

Ensure developers spend time on high value work by reducing troubleshooting toil

### Step 1. Send telemetry data to Chronosphere.

This can be done by deploying either the Chronosphere collector, or the OpenTelemetry collector. You can also send data agentlessly via OpenTelemetry Protocol (OTLP). The Chronosphere collector is compliant with all major open source standards – Prometheus, OpenTelemetry, and older formats like Graphite/StatsD.

### Step 2. Analyze, refine, and operate telemetry data in the Chronosphere Control Plane.

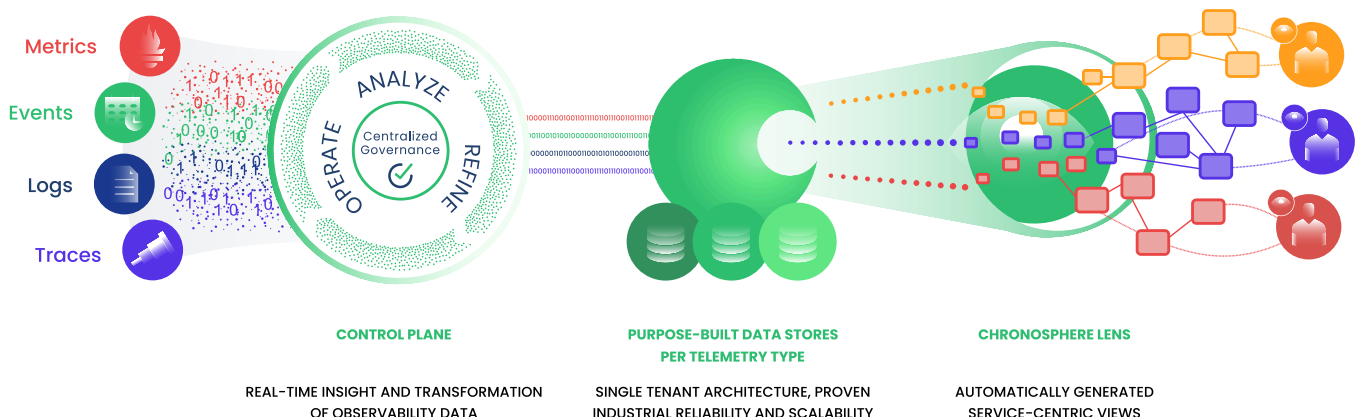
Determine how observability data should be aggregated, dropped, or rolled up based on environment and business needs. Tune ingest as needed to control for cardinality spikes. Set your metrics retention by application or environment.

### Step 3. Navigate, investigate, and remediate more efficiently with the Chronosphere Lens.

The Chronosphere Lens offers a dynamically generated, service-centric perspective on your data, providing more depth and context for your observability needs. Everything is compatible with the open source formats they already know: Grafana for dashboarding/visualization, Prometheus Alertmanager for alerts, and PromQL for queries.

### Step 4. Take back control.

With an end-to-end managed SaaS offering, you'll not only save engineering time, but will gain a solution that is more reliable than self hosted open-source tools, other SaaS offerings, and APM tools. On top of that, Chronosphere's unique centralized governance capabilities put you in control of your observability data. Set Quotas by teams, services, etc. and drive responsibility for data growth to the people that know the data best, cutting volumes by an average of 60% for customers.



Ready to learn more?  
Book your demo  
today by visiting  
[chronosphere.io](https://chronosphere.io)