DTCC

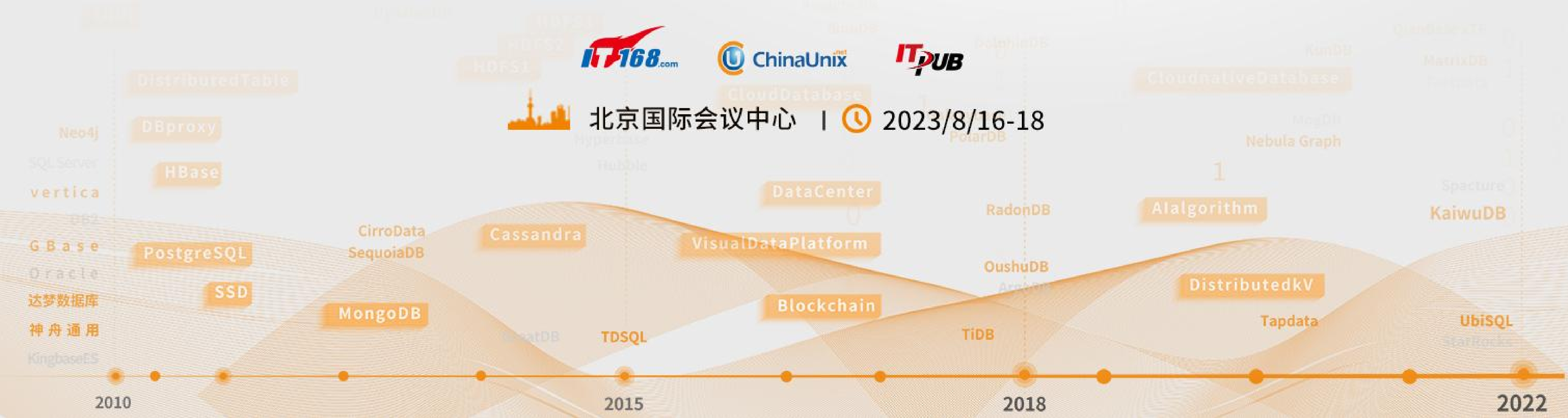
第十四届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA

数智赋能 共筑未来

IT168.com ChinaUnix ITPUB

北京国际会议中心 | 2023/8/16-18



可观测数据融合处理平台探索

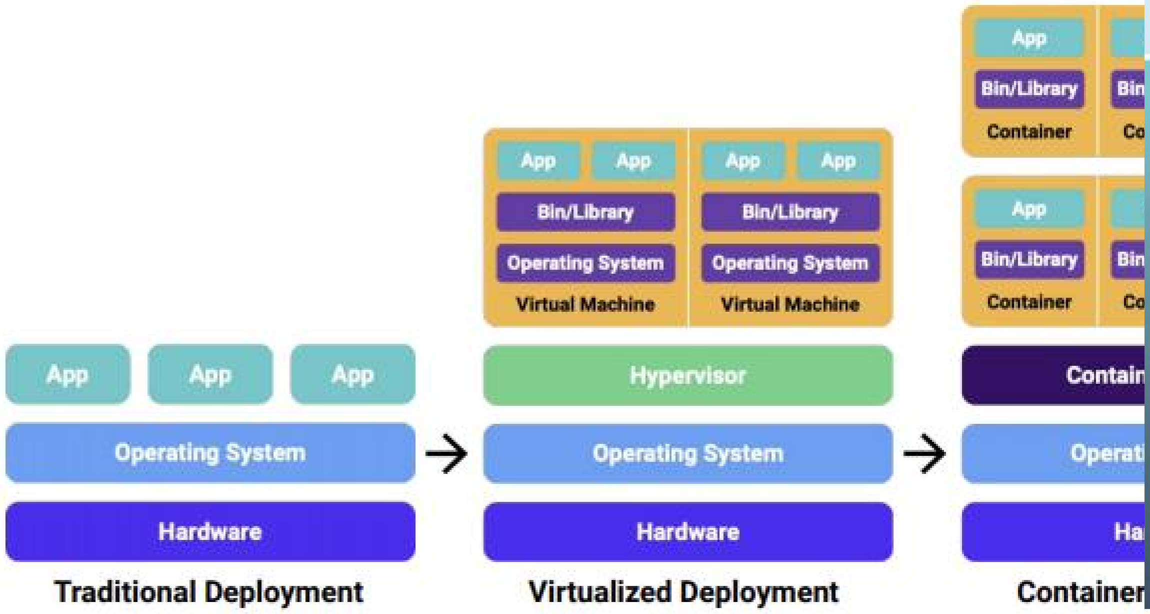
唐恒建

OPPO 高级后端工程师

- 1、可观测要素
- 2、opentelemetry现状
- 3、各类可观测方案对比
- 4、可观测数据融合平台
- 5、总结

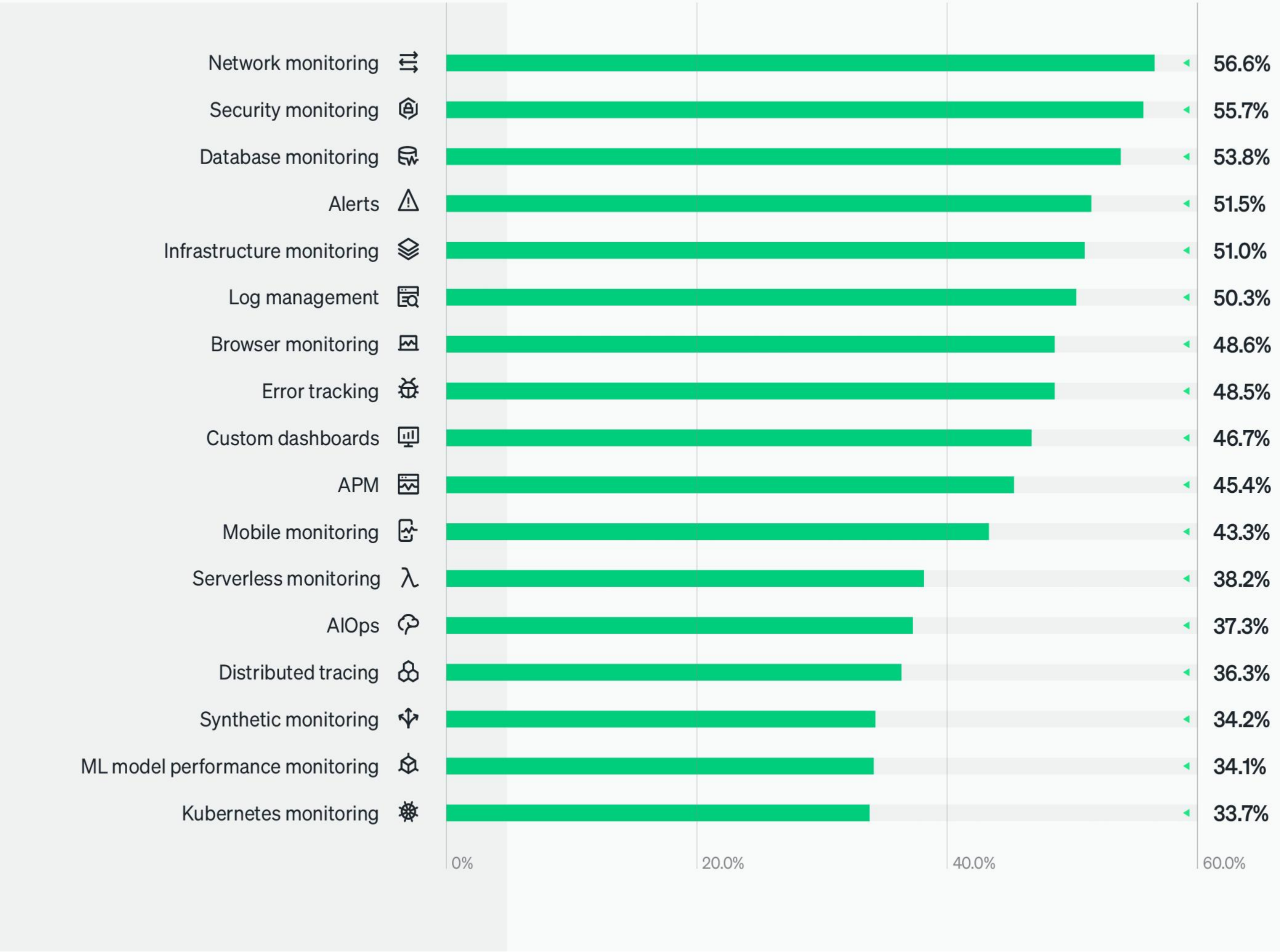
可观测要素

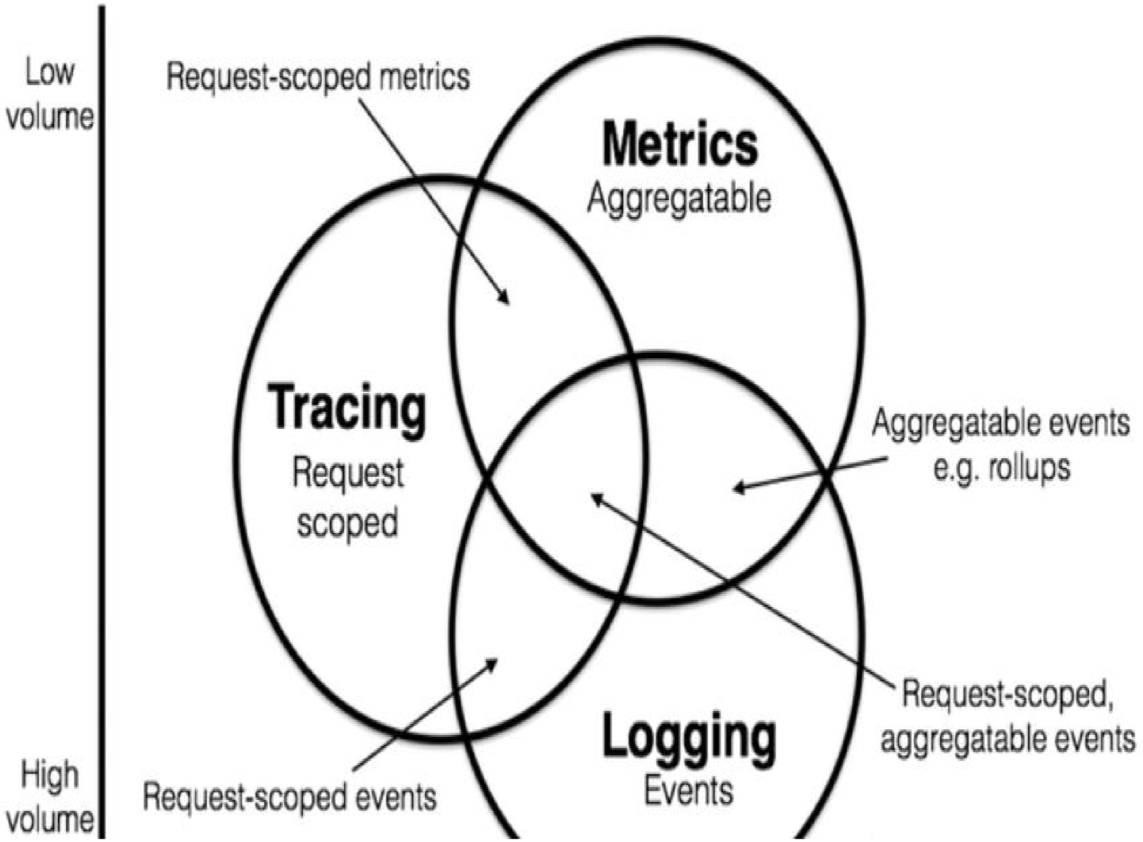
Increasingly complex software deployments



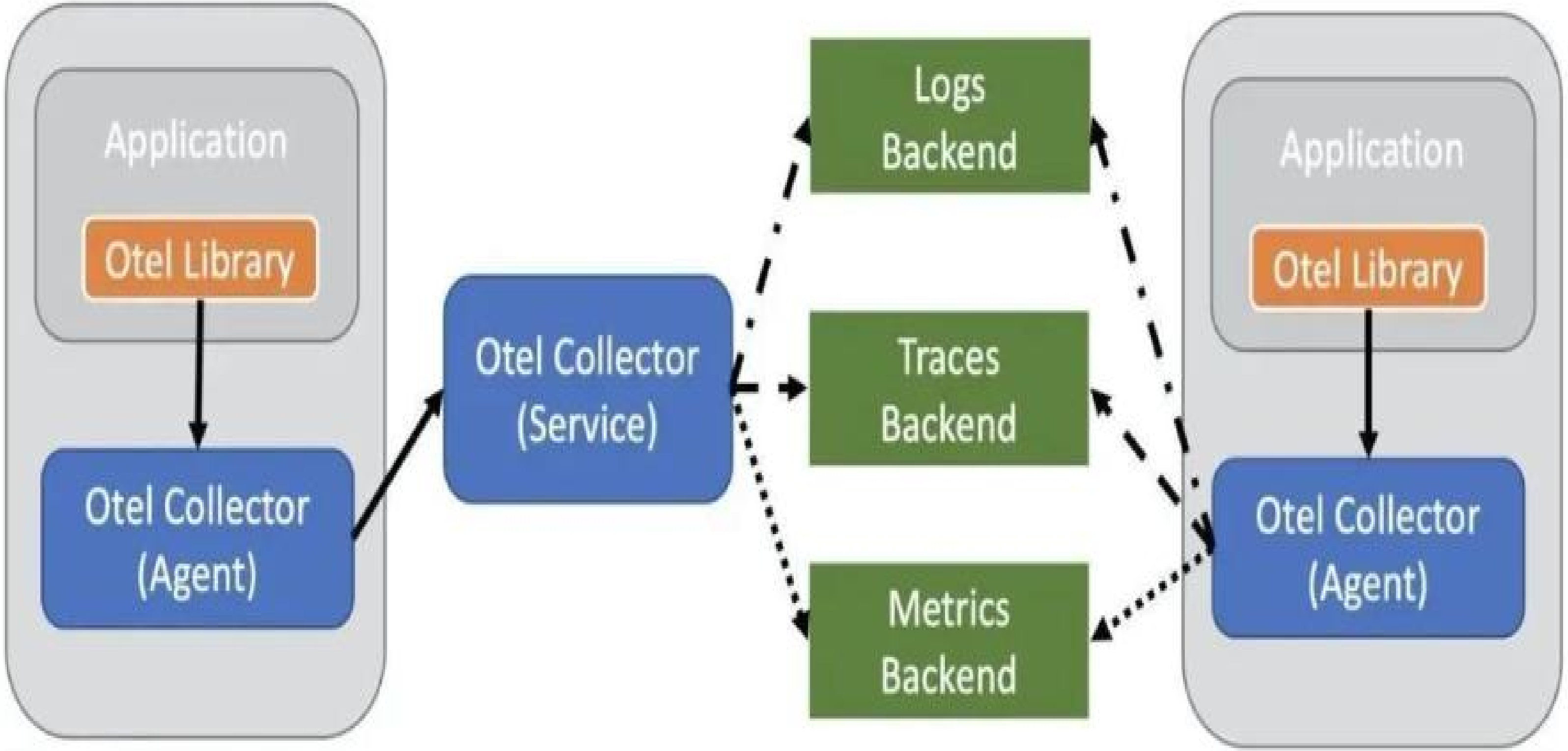
Monitoring
Observability

可观测要素


Network monitoring 56.6%
Security monitoring 55.7%
Database monitoring 53.8%
Alerts 51.5%
Infrastructure monitoring 51.0%
Log management 50.3%
Browser monitoring 48.6%
Error tracking 48.5%
Custom dashboards 46.7%
APM 45.4%
Mobile monitoring 43.3%
Serverless monitoring 38.2%
AIOps 37.3%
Distributed tracing 36.3%
Synthetic monitoring 34.2%
ML model performance monitoring 34.1%
Kubernetes monitoring 33.7%
0%
20.0%
40.0%
60.0%

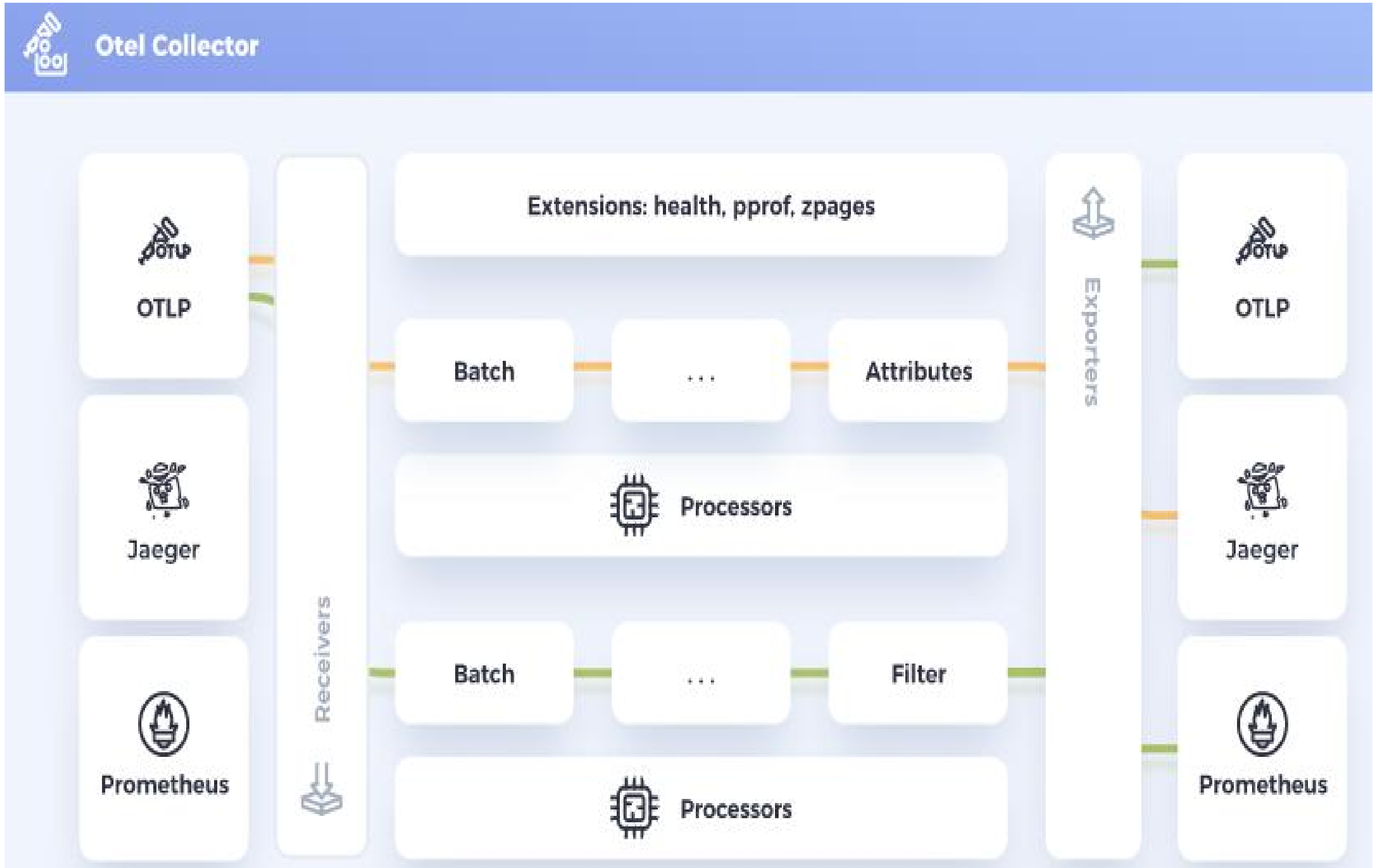

Low volume
High volume
Request-scoped metrics
Metrics
Aggregatable
Tracing
Request scoped
Aggregatable events
e.g. rollups
Request-scoped, aggregatable events
Logging
Events
Request-scoped events

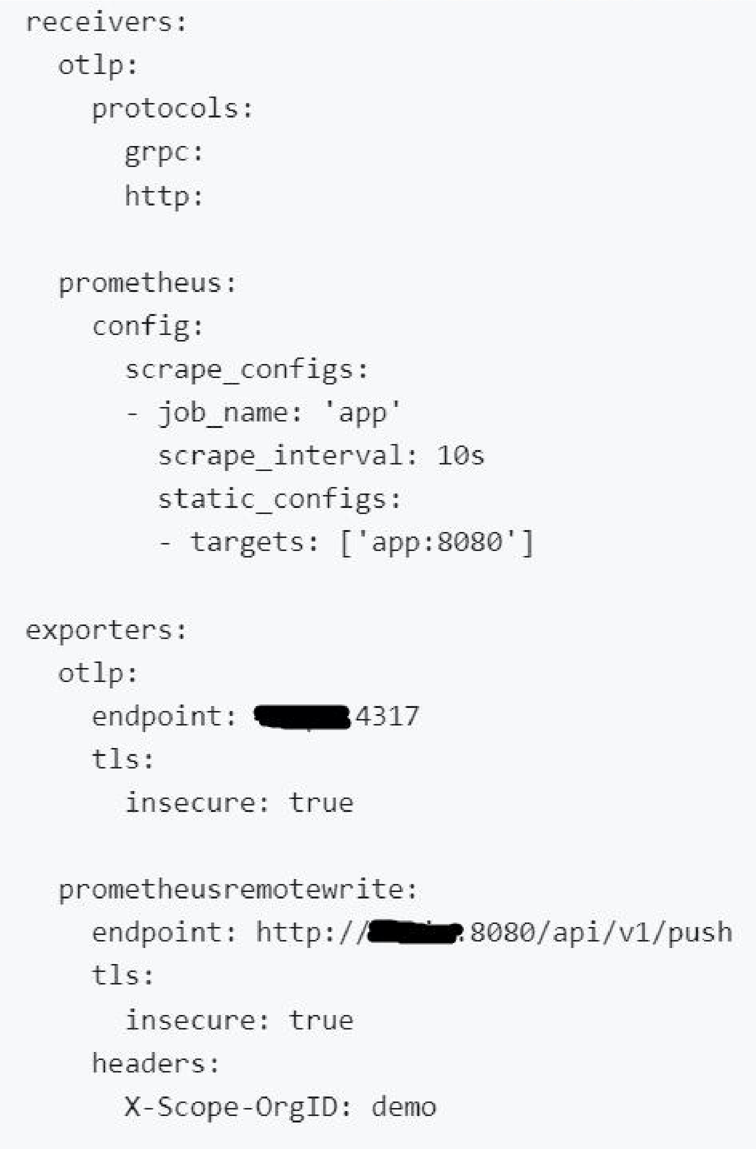
opentelemetry现状



- 维度：metric、trace、log
- 语言：JIT：Java、Rust、Python
  非JIT类：C++、Go
- agent、collect（batch）
- Multi protocol: receiver、exporter

opentelemetry现状

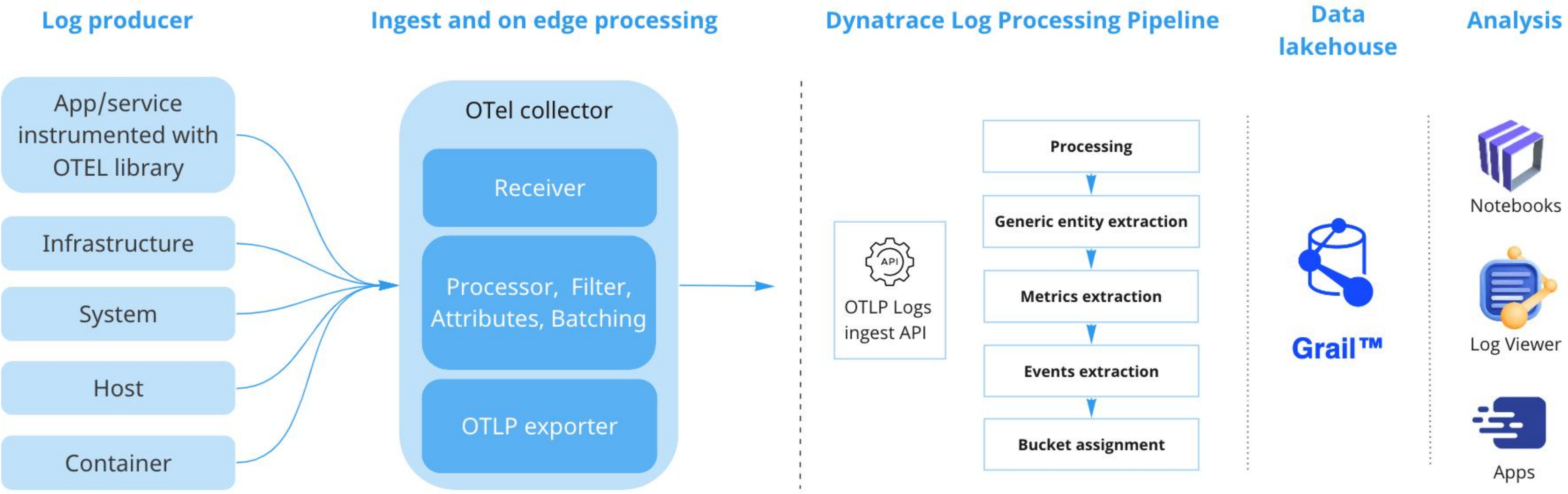




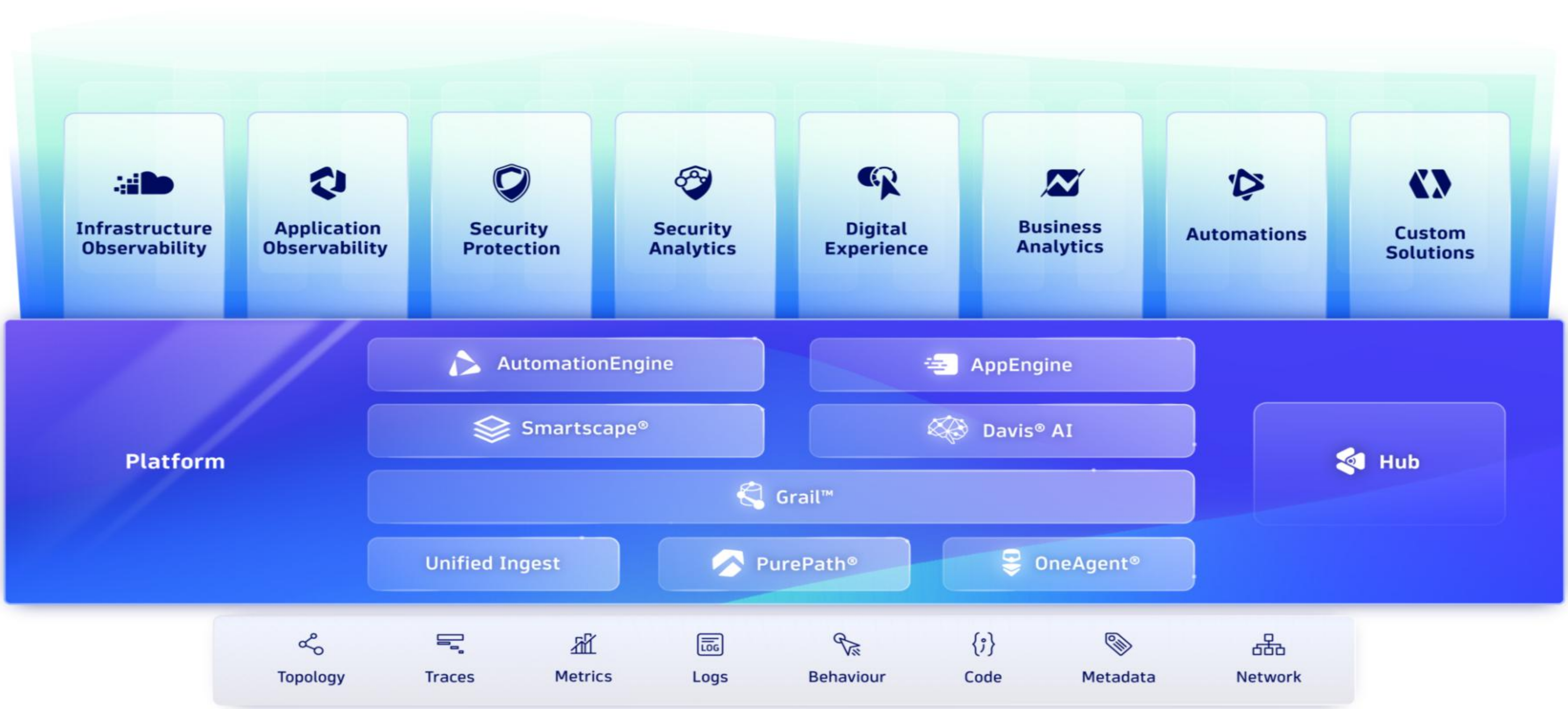
```
receivers:
  otlp:
    protocols:
      grpc:
      http:

  prometheus:
    config:
      scrape_configs:
      - job_name: 'app'
        scrape_interval: 10s
        static_configs:
        - targets: ['app:8080']

exporters:
  otlp:
    endpoint: [redacted]4317
    tls:
      insecure: true

  prometheusremotewrite:
    endpoint: http://[redacted]:8080/api/v1/push
    tls:
      insecure: true
    headers:
      X-Scope-OrgID: demo
```

各类可观测方案对比

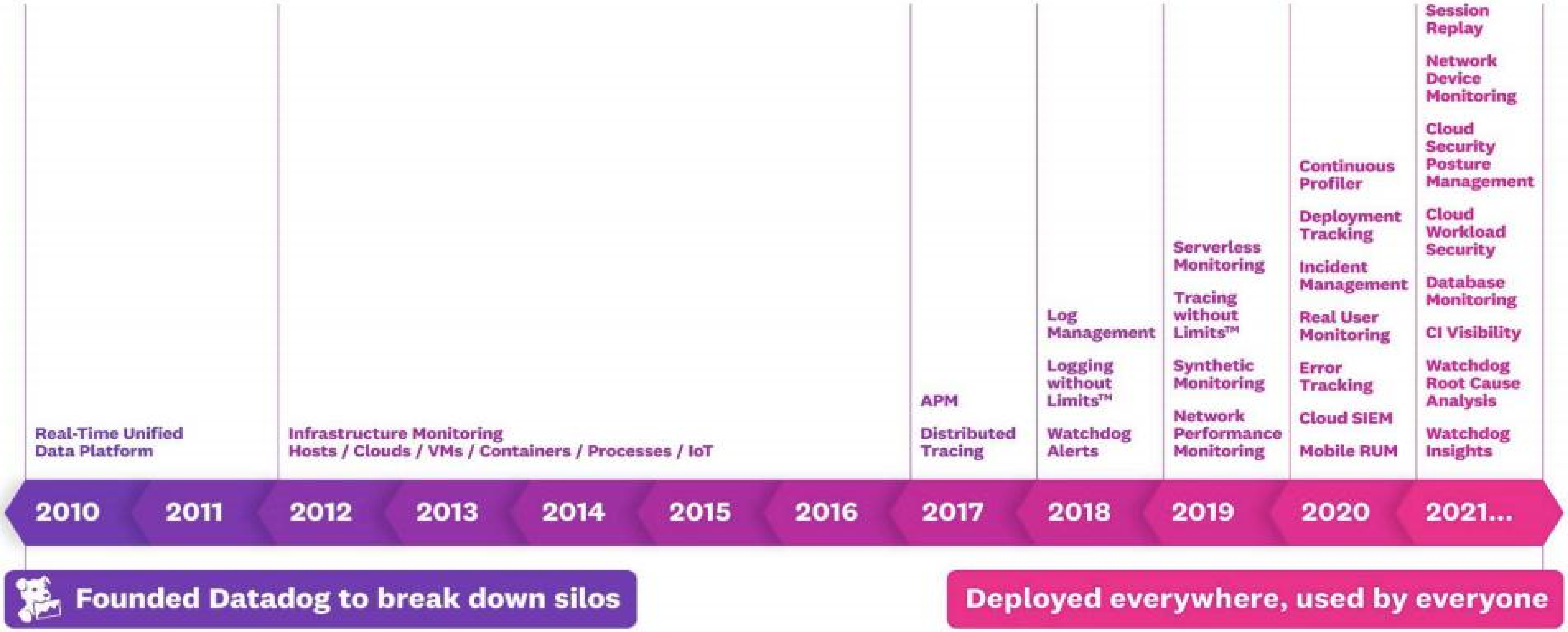
- Dynatrace



- Dynatrace


Infrastructure Observability
Application Observability
Security Protection
Security Analytics
Digital Experience
Business Analytics
Automations
Custom Solutions
Platform
AutomationEngine
AppEngine
Smartscape®
Davis® AI
Grail™
Unified Ingest
PurePath®
OneAgent®
Hub
Topology
Traces
Metrics
Logs
Behaviour
Code
Metadata
Network

- Datadog



2010: Real-Time Unified Data Platform

2012: Infrastructure Monitoring Hosts / Clouds / VMs / Containers / Processes / IoT

2017: APM

Distributed Tracing

2018: Log Management

Logging without Limits™

Watchdog Alerts

2019: Serverless Monitoring

Tracing without Limits™

Synthetic Monitoring

Network Performance Monitoring

2020: Continuous Profiler

Deployment Tracking

Incident Management

Real User Monitoring

Error Tracking

Cloud SIEM

Mobile RUM

2021...: Session Replay

Network Device Monitoring

Cloud Security Posture Management

Cloud Workload Security

Database Monitoring

CI Visibility

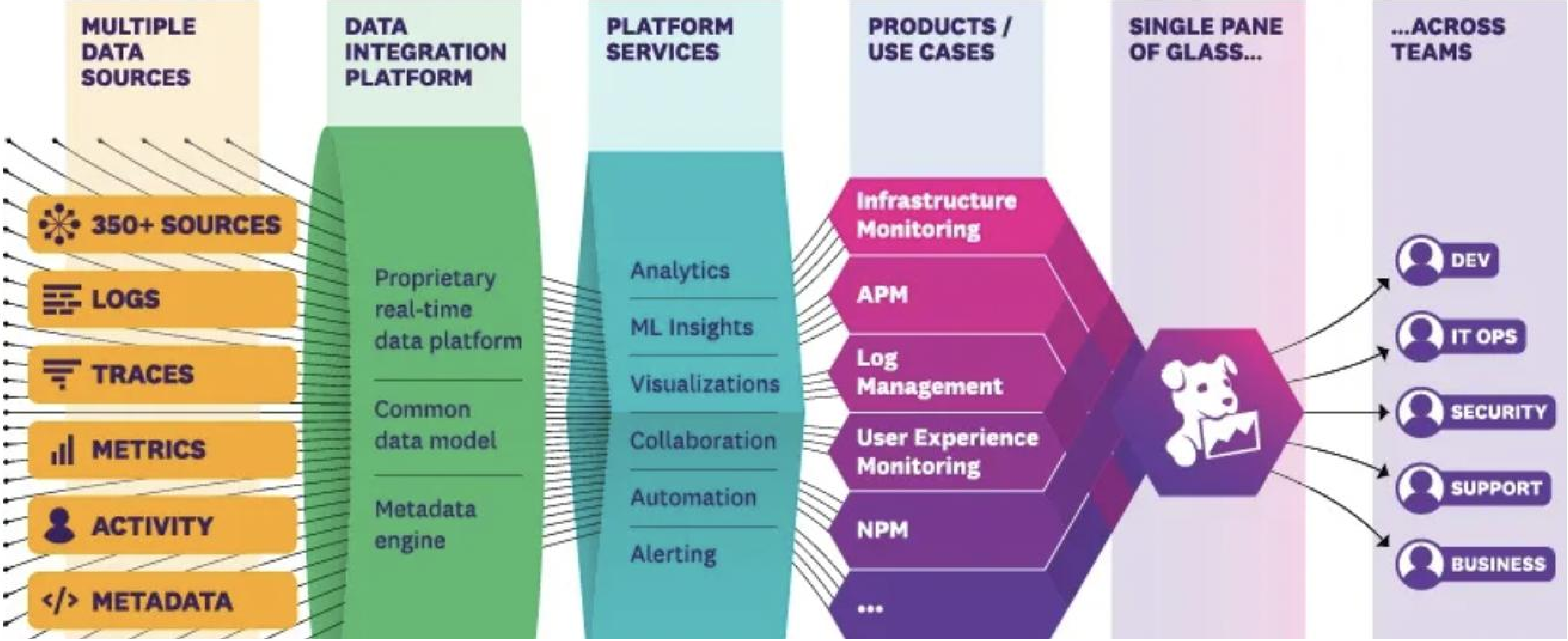
Watchdog Root Cause Analysis

Watchdog Insights

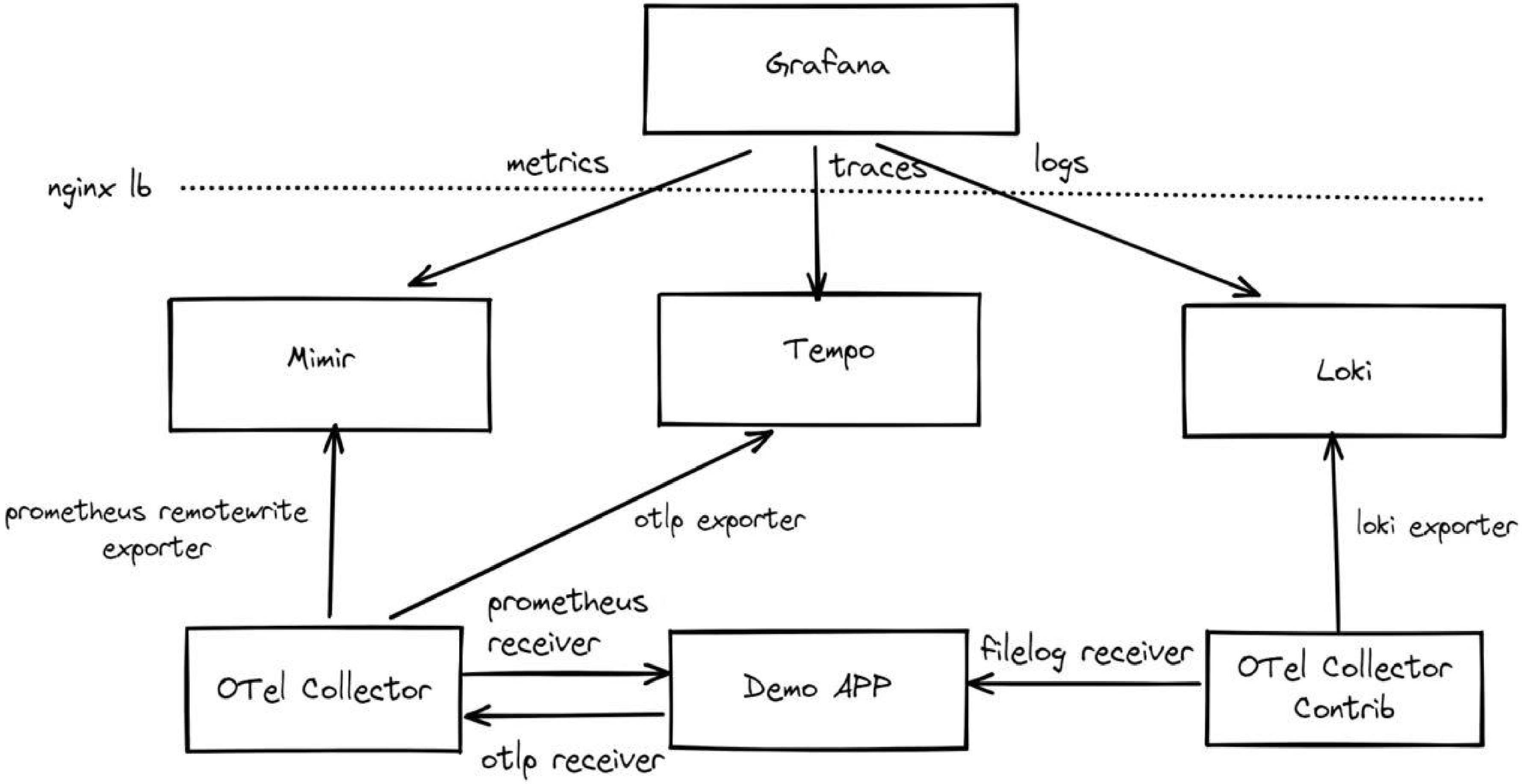
Founded Datadog to break down silos

Deployed everywhere, used by everyone

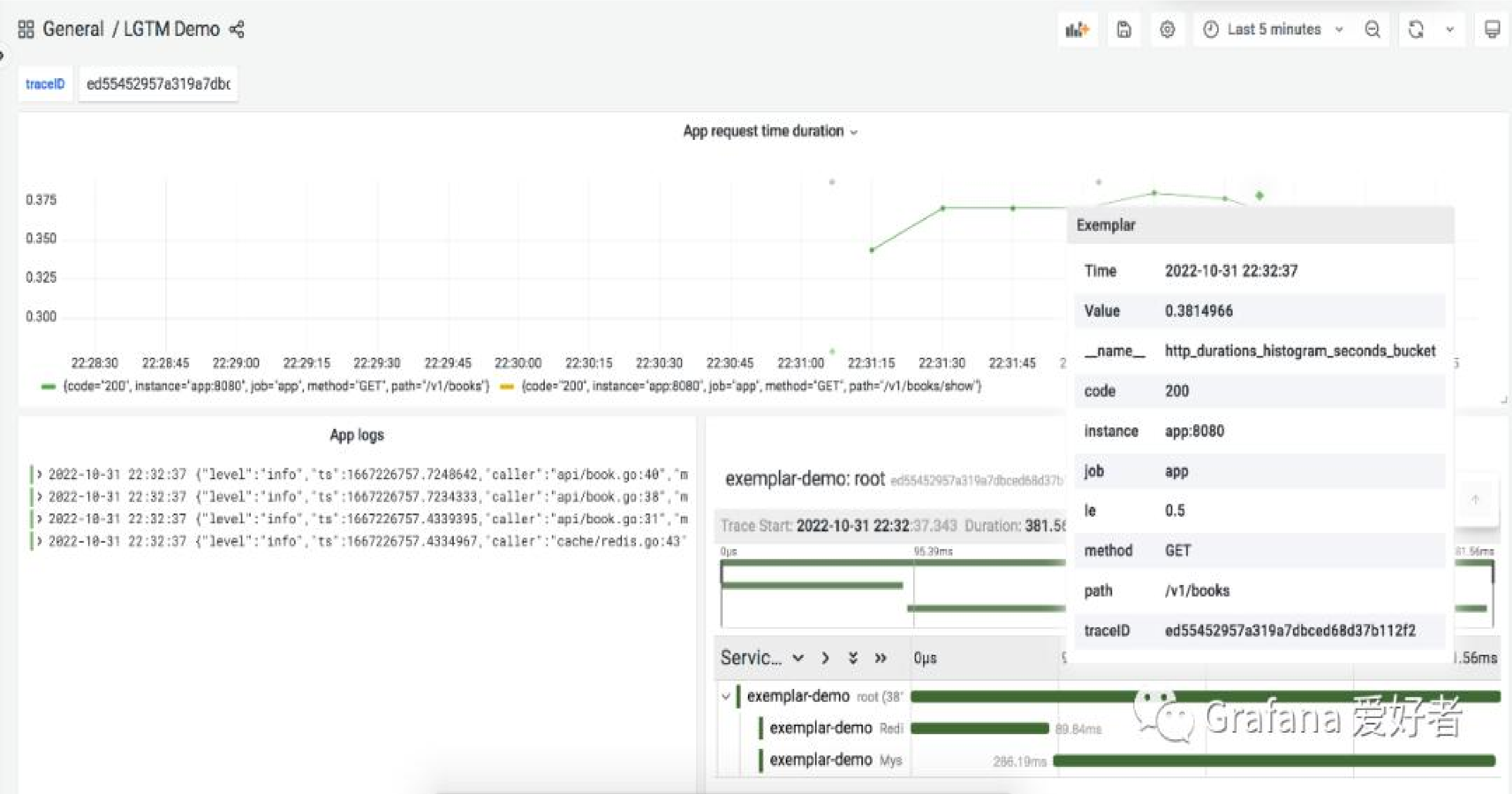
- Datadog


MULTIPLE DATA SOURCES
350+ SOURCES
LOGS
TRACES
METRICS
ACTIVITY
</> METADATA
DATA INTEGRATION PLATFORM
Proprietary real-time data platform
Common data model
Metadata engine
PLATFORM SERVICES
Analytics
ML Insights
Visualizations
Collaboration
Automation
Alerting
PRODUCTS / USE CASES
Infrastructure Monitoring
APM
Log Management
User Experience Monitoring
NPM
...
SINGLE PANE OF GLASS...
...ACROSS TEAMS
DEV
IT OPS
SECURITY
SUPPORT
BUSINESS

- Grafana


Grafana
metrics
traces
logs
nginx lb
Mimir
Tempo
Loki
prometheus remotewrite exporter
otlp exporter
loki exporter
prometheus receiver
filelog receiver
OTel Collector
Demo APP
OTel Collector Contrib
otlp receiver

- Grafana



Exemplar

Time	2022-10-31 22:32:37
Value	0.3814966
__name__	http_durations_histogram_seconds_bucket
code	200
instance	app:8080
job	app
le	0.5
method	GET
path	/v1/books
traceID	ed55452957a319a7dbced68d37b112f2

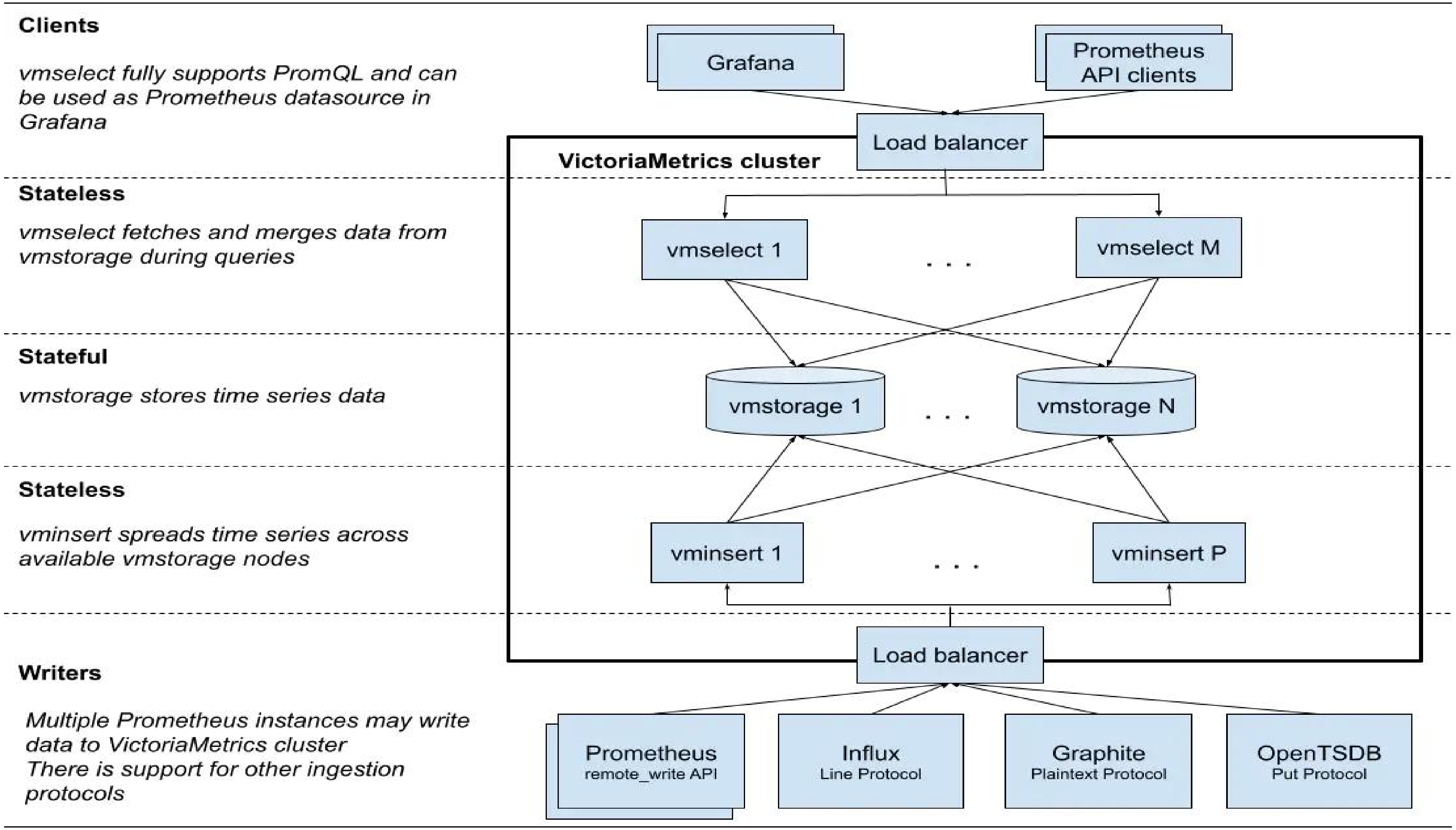
```
> 2022-10-31 22:32:37 {"level":"info","ts":1667226757.7248642,"caller":"api/book.go:40","m
> 2022-10-31 22:32:37 {"level":"info","ts":1667226757.7234333,"caller":"api/book.go:38","m
> 2022-10-31 22:32:37 {"level":"info","ts":1667226757.4339395,"caller":"api/book.go:31","m
> 2022-10-31 22:32:37 {"level":"info","ts":1667226757.4334967,"caller":"cache/redis.go:43"
```

exemplar-demo: root ed55452957a319a7dbced68d37b

Trace Start: 2022-10-31 22:32:37.343 Duration: 381.5

- 以exemplar为连接
- 联动metrics、log、
  trace
- Metrics–>logs
- metrics–>trace

- Victoriametrics



Clients

vmselect fully supports PromQL and can be used as Prometheus datasource in Grafana

Stateless

vmselect fetches and merges data from vmstorage during queries

Stateful

vmstorage stores time series data

Stateless

vminsert spreads time series across available vmstorage nodes

Writers

Multiple Prometheus instances may write data to VictoriaMetrics cluster
There is support for other ingestion protocols

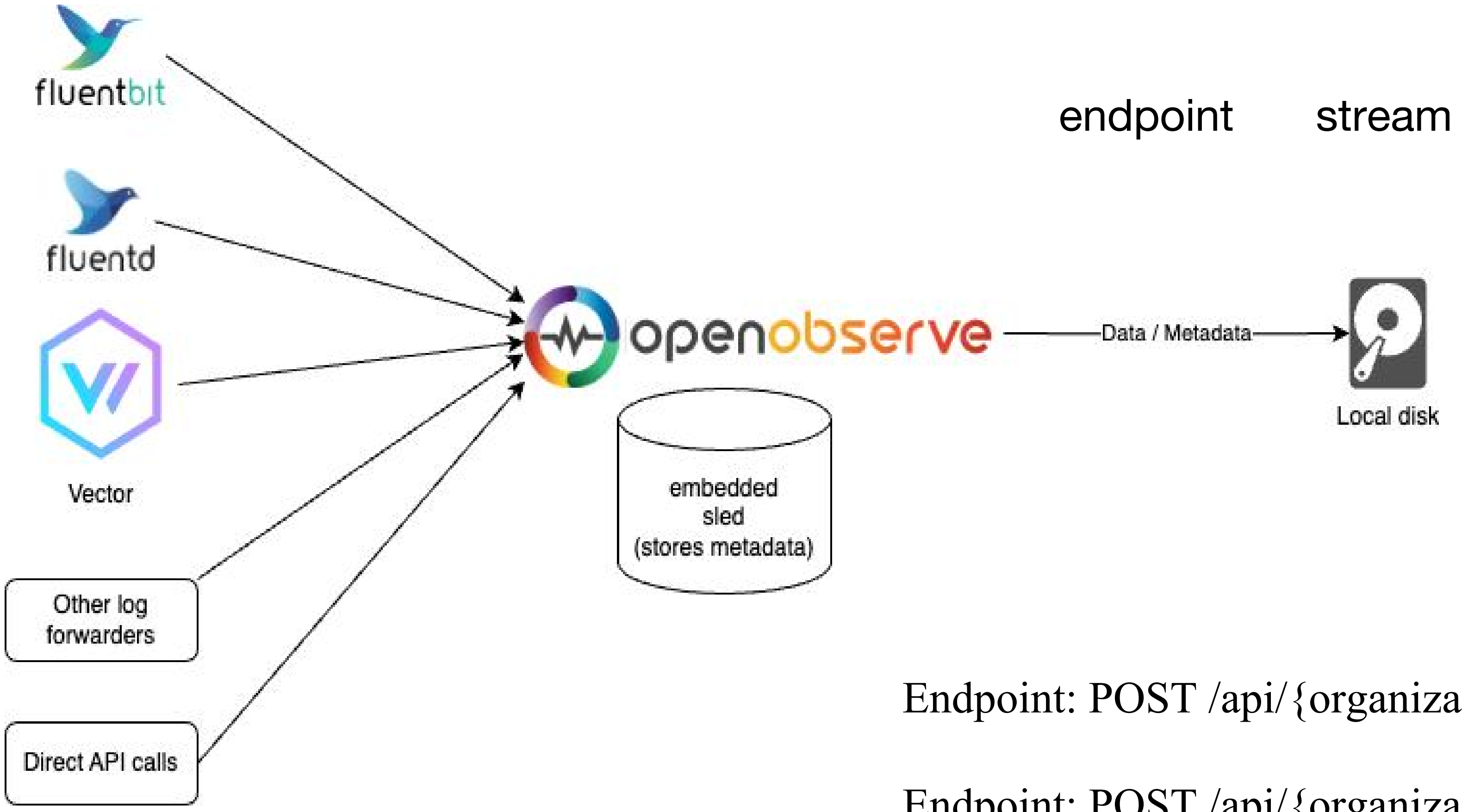
- Less CPU Usage
- Less Disk Usage

- Victoriametrics

VictoriaLogs --- Preview stage

- inspired by ClickHouse architecture
- It uses bloom filters
- encoding and compression for fields with different data types
- logs for the same log stream close to each other.
- maintains sparse index for log timestamps
- high cardinalitymetrics added.

- Openobserve



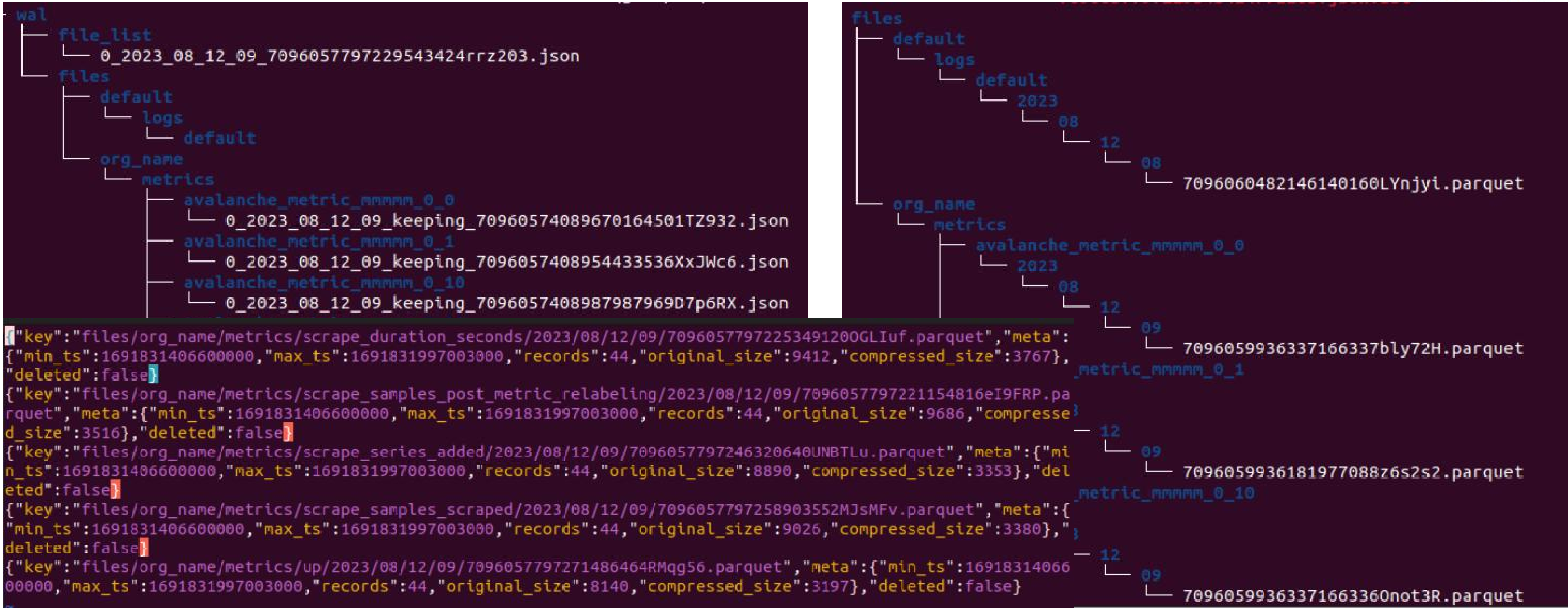
endpoint stream kvs parquet

Endpoint: POST /api/{organization}/{stream}/_json

Endpoint: POST /api/{organization}/prometheus/api/v1/write

Endpoint: POST /api/{organization}/traces

- Openobserve



```
wal
├── file_list
│   └── 0_2023_08_12_09_7096057797229543424rrz203.json
└── files
    ├── default
    │   └── logs
    │       └── default
    └── org_name
        └── metrics
            ├── avalanche_metric_mmmmm_0_0
            │   └── 0_2023_08_12_09_keeping_7096057408967016450lTZ932.json
            ├── avalanche_metric_mmmmm_0_1
            │   └── 0_2023_08_12_09_keeping_7096057408954433536XxJWc6.json
            ├── avalanche_metric_mmmmm_0_10
            │   └── 0_2023_08_12_09_keeping_7096057408987987969D7p6RX.json
```

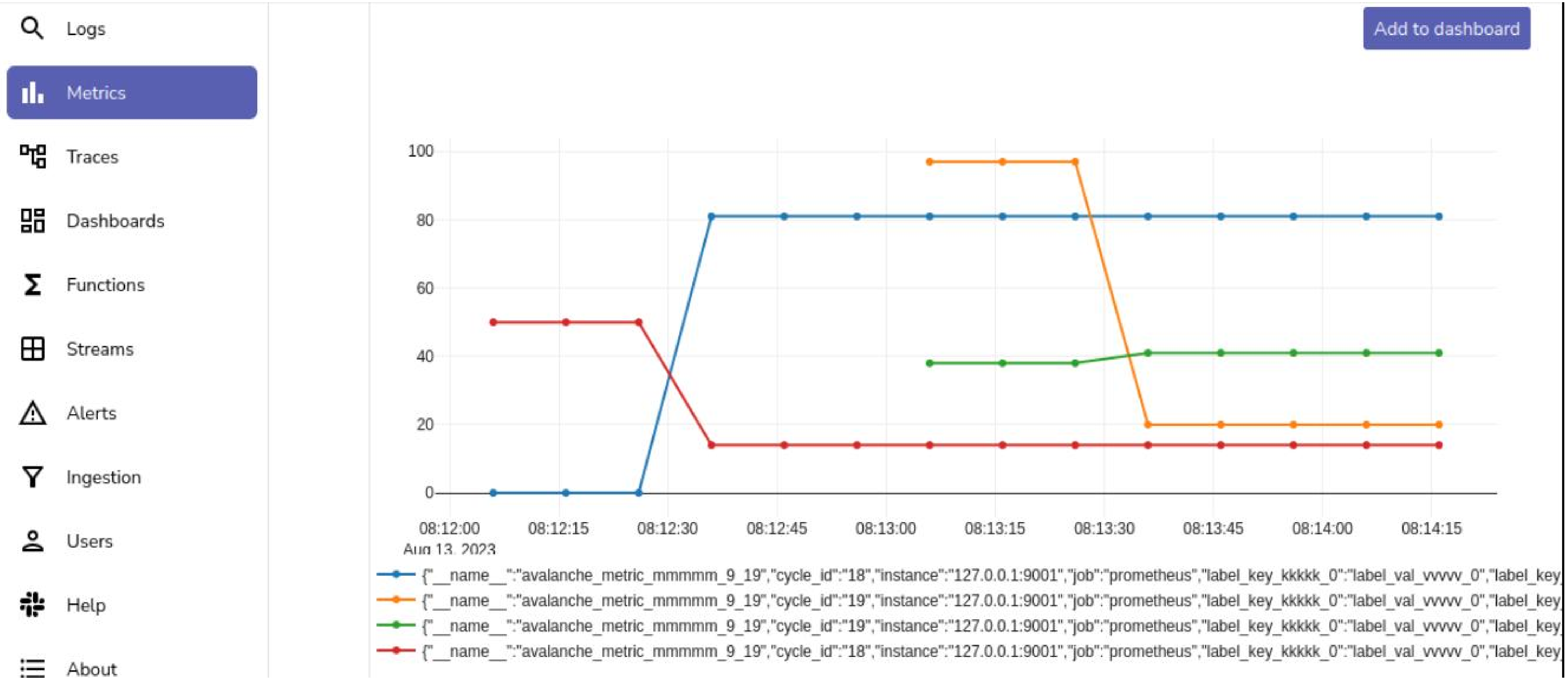
```
{"key":"files/org_name/metrics/scrape_duration_seconds/2023/08/12/09/7096057797225349120OGLIuf.parquet","meta":
{"min_ts":1691831406600000,"max_ts":1691831997003000,"records":44,"original_size":9412,"compressed_size":3767},
"deleted":false}
{"key":"files/org_name/metrics/scrape_samples_post_metric_relabeling/2023/08/12/09/7096057797221154816eI9FRP.pa
rquet","meta":{"min_ts":1691831406600000,"max_ts":1691831997003000,"records":44,"original_size":9686,"compresse
d_size":3516},"deleted":false}
{"key":"files/org_name/metrics/scrape_series_added/2023/08/12/09/7096057797246320640UNBTLu.parquet","meta":{"mi
n_ts":1691831406600000,"max_ts":1691831997003000,"records":44,"original_size":8890,"compressed_size":3353},"del
eted":false}
{"key":"files/org_name/metrics/scrape_samples_scraped/2023/08/12/09/7096057797258903552MJsMFv.parquet","meta":{
"min_ts":1691831406600000,"max_ts":1691831997003000,"records":44,"original_size":9026,"compressed_size":3380},"
deleted":false}
{"key":"files/org_name/metrics/up/2023/08/12/09/7096057797271486464RMqg56.parquet","meta":{"min_ts":16918314066
00000,"max_ts":1691831997003000,"records":44,"original_size":8140,"compressed_size":3197},"deleted":false}
```

```
files
├── default
│   └── logs
│       └── default
│           └── 2023
│               └── 08
│                   └── 12
│                       └── 08
│                           └── 7096060482146140160LYnjyi.parquet
└── org_name
    └── metrics
        ├── avalanche_metric_mmmmm_0_0
        │   └── 2023
        │       └── 08
        │           └── 12
        │               └── 09
        │                   └── 7096059936337166337bly72H.parquet
         metric_mmmmm_0_1

                ── 12
                    └── 09
                        └── 7096059936181977088z6s2s2.parquet
         metric_mmmmm_0_10

                ── 12
                    └── 09
                        └── 7096059936337166336Onot3R.parquet
```

WAL

blocks

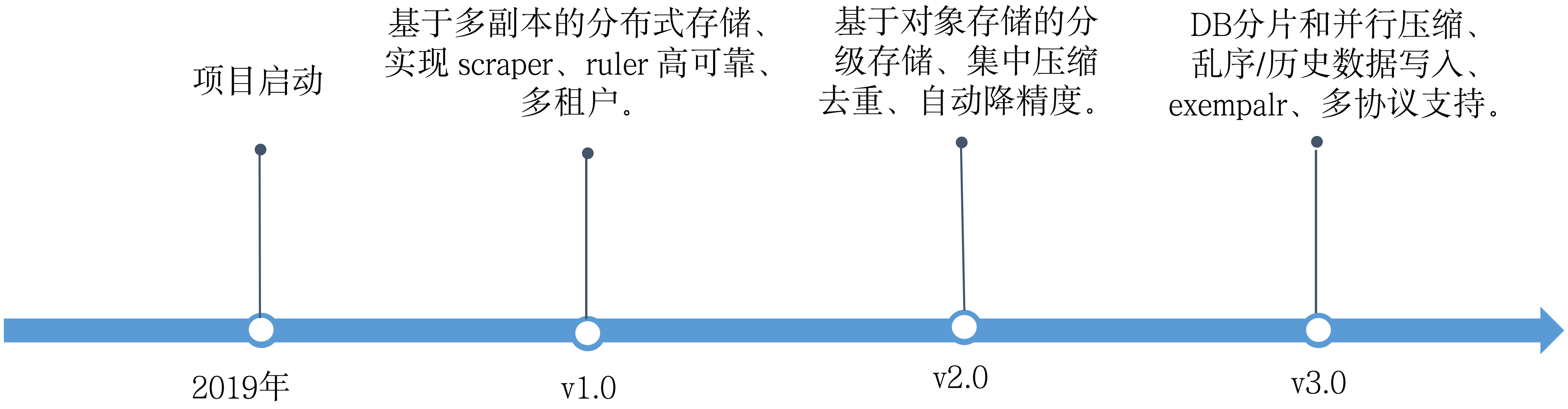
- Openobserve



```
{"__name__":"avalanche_metric_mmmmm_9_19","cycle_id":"18","instance":"127.0.0.1:9001","job":"prometheus","label_key_kkkkk_0":"label_val_vvvvv_0","label_key
{"__name__":"avalanche_metric_mmmmm_9_19","cycle_id":"19","instance":"127.0.0.1:9001","job":"prometheus","label_key_kkkkk_0":"label_val_vvvvv_0","label_key
{"__name__":"avalanche_metric_mmmmm_9_19","cycle_id":"19","instance":"127.0.0.1:9001","job":"prometheus","label_key_kkkkk_0":"label_val_vvvvv_0","label_key
{"__name__":"avalanche_metric_mmmmm_9_19","cycle_id":"18","instance":"127.0.0.1:9001","job":"prometheus","label_key_kkkkk_0":"label_val_vvvvv_0","label_key
```

可观测数据融合平台

夯实metrics能力



项目启动

2019年

基于多副本的分布式存储、实现 scraper、ruler 高可靠、多租户。

v1.0

基于对象存储的分级存储、集中压缩去重、自动降精度。

v2.0

DB分片和并行压缩、乱序/历史数据写入、exempalr、多协议支持。

v3.0

抽象平台层

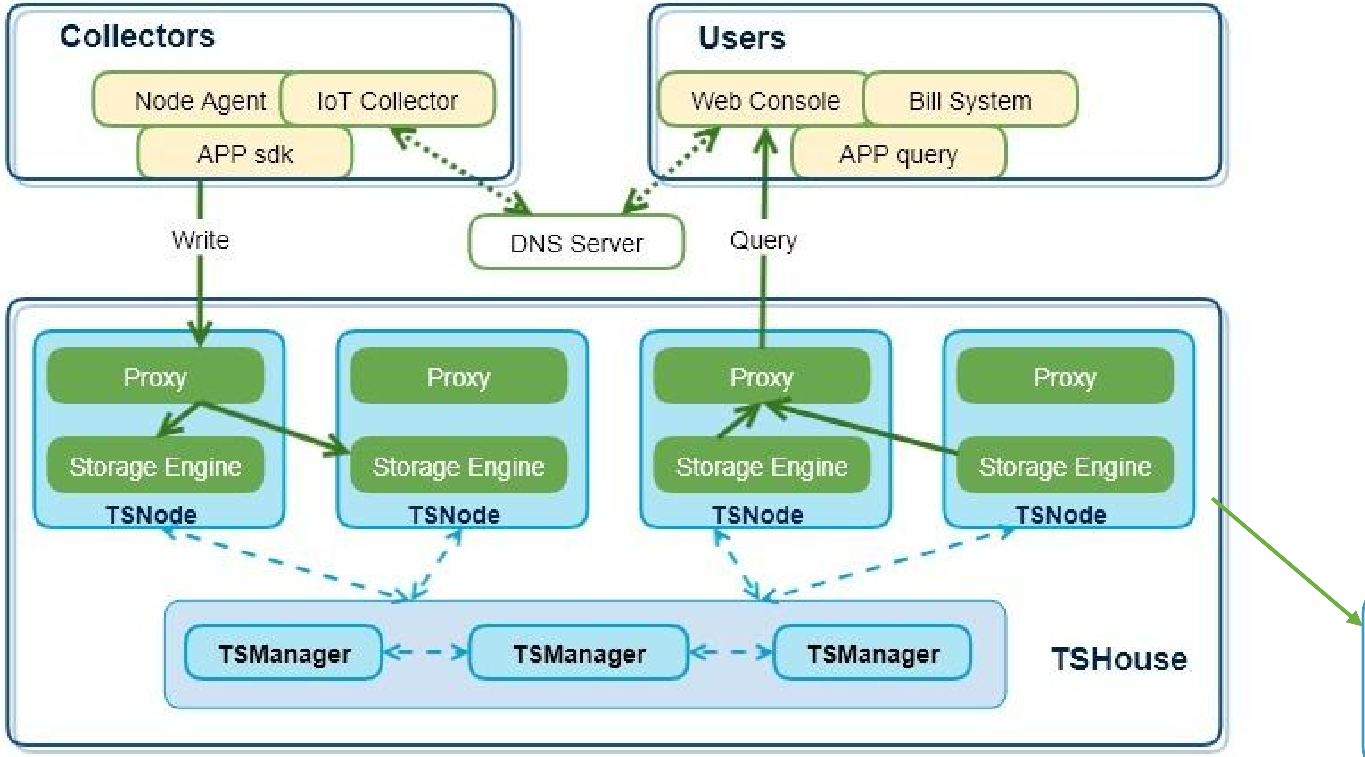
分布式、高可用、高基数

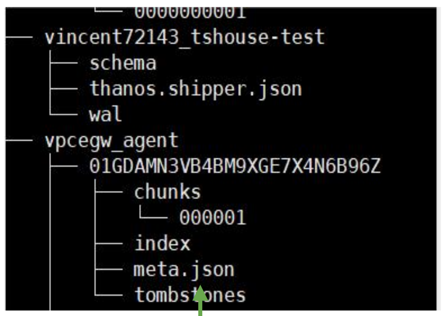
数据融合平台

扩展log、trace底座能力，作为数据存储层

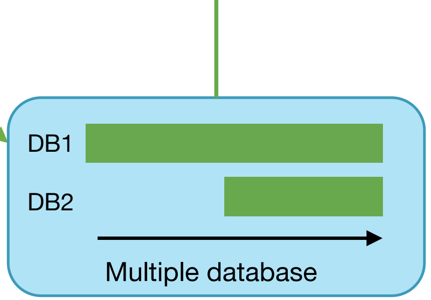
夯实metrics能力

v1.0 - 基于多副本实现存储高可靠





```
    └── 0000000001
├── vincent72143_tshouse-test
│   ├── schema
│   ├── thanos.shipper.json
│   └── wal
├── vpcegw_agent
│   ├── 01GDAMN3VB4BM9XGE7X4N6B96Z
│   │   ├── chunks
│   │   │   └── 000001
│   │   ├── index
│   │   ├── meta.json
│   │   └── tombstones
```



v1.0 – 小结

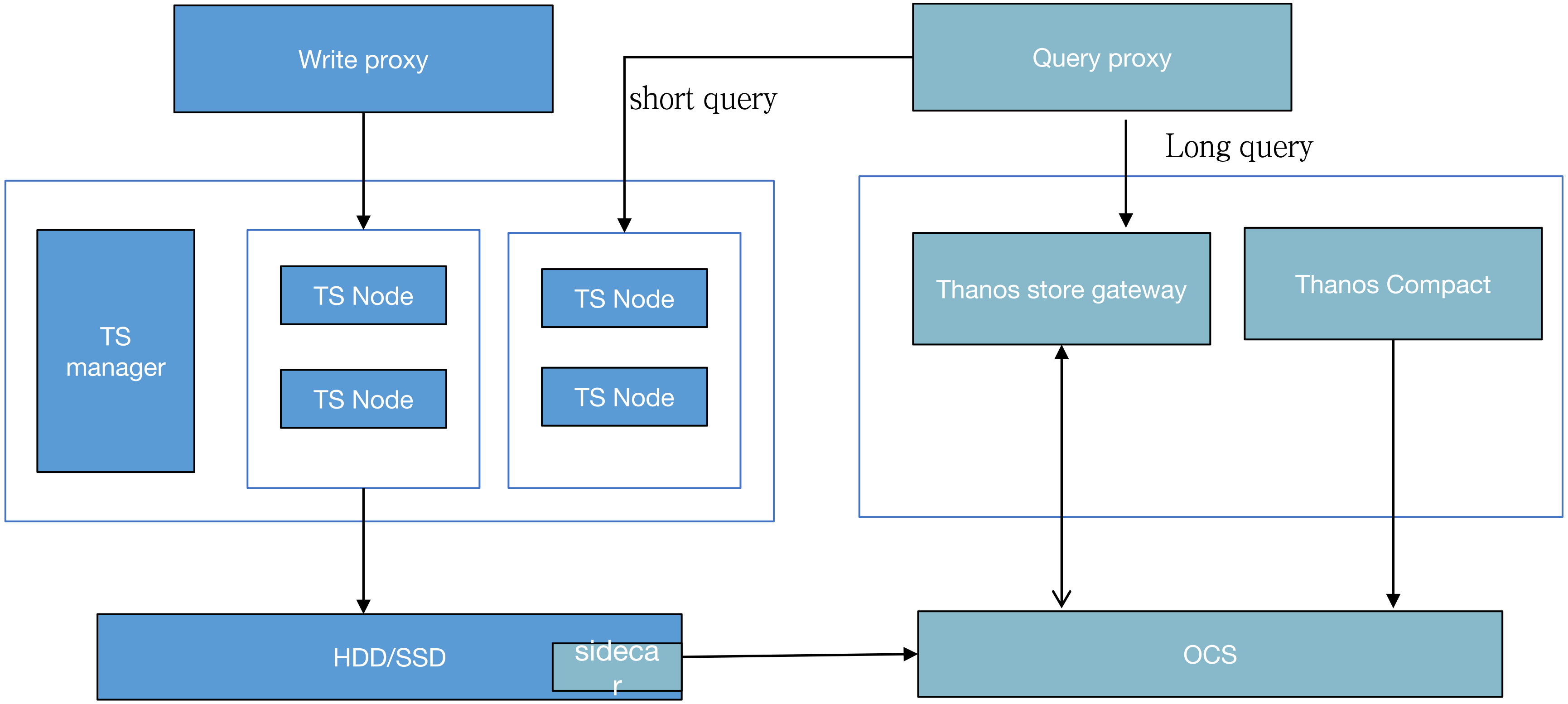
实现：

1. 基于 Prometheus TSDB 构建多租户的分布式存储服务。
2. 租户之间可以配置不同的读写QPS 限制和存储周期。
3. 分布式 scraper 和 ruler 实现数据抓取和 rule 评估的高可靠。

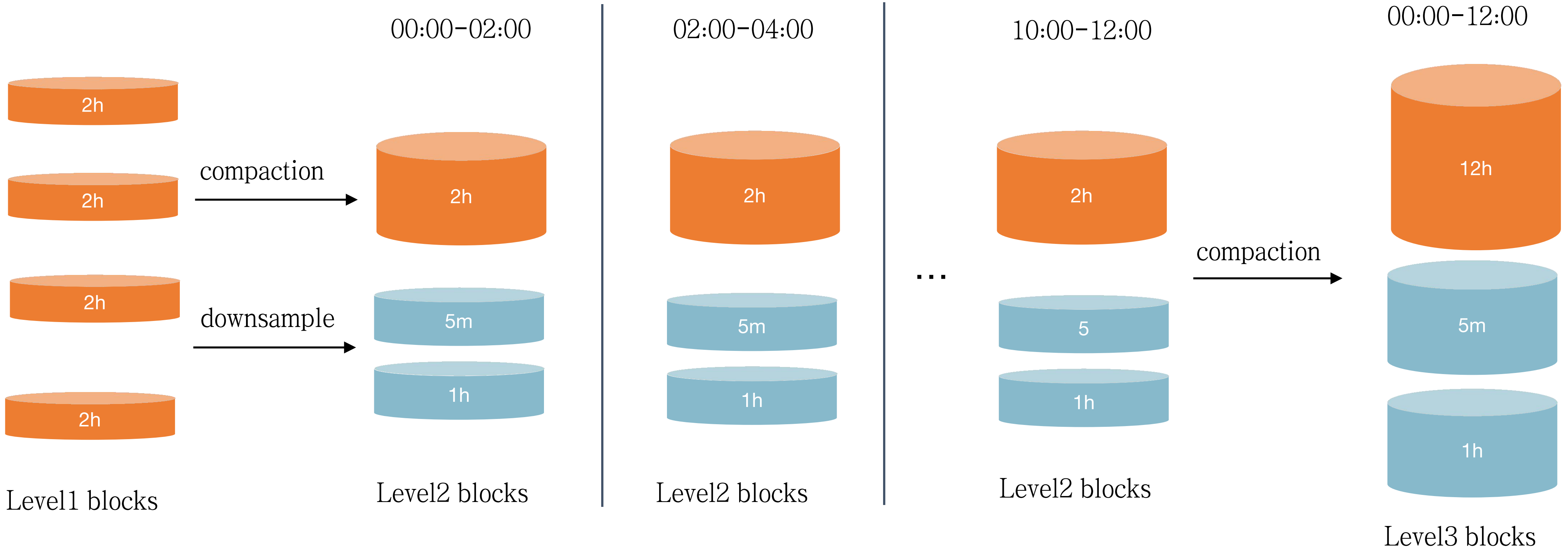
不足：

1. 长时间存储集群运维压力大，扩容和坏盘修复较复杂。
2. 读写没有分离，相互影响，尤其长时间范围查询。
3. 本地存储多副本，对于冷数据（一个月前）没有进行去重，造成存储浪费。

v2.0 - 基于对象存储的分级存储



v2.0 - 集中压缩和自动降精度



v2.0 – 小结

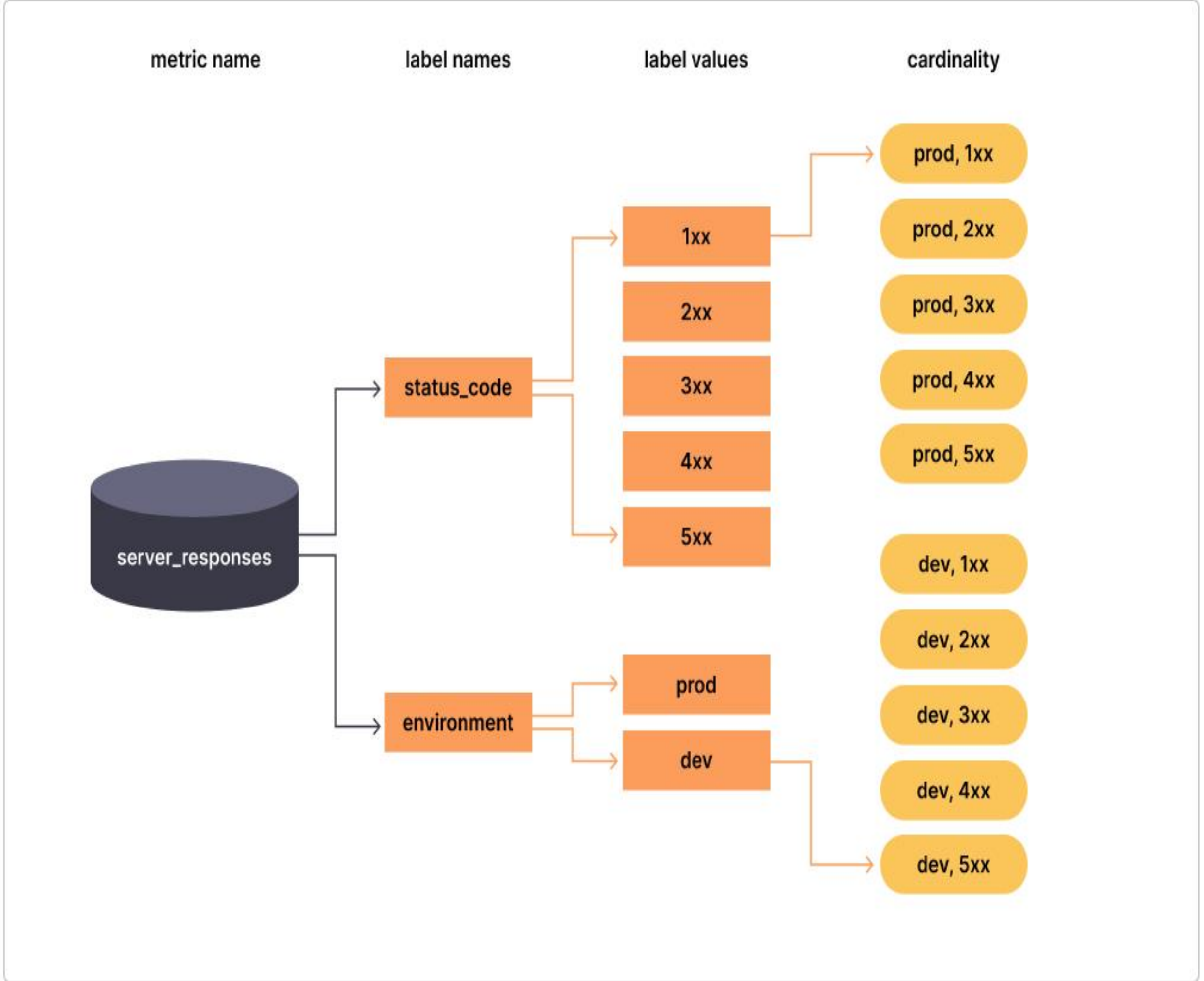
实现：

1. 集中压缩，数据去重，降低成本。
2. 自动降精度，长时间范围查询自动转化，提升查询效率。
3. 依赖对象存储实现长时间数据集中存储。

不足：

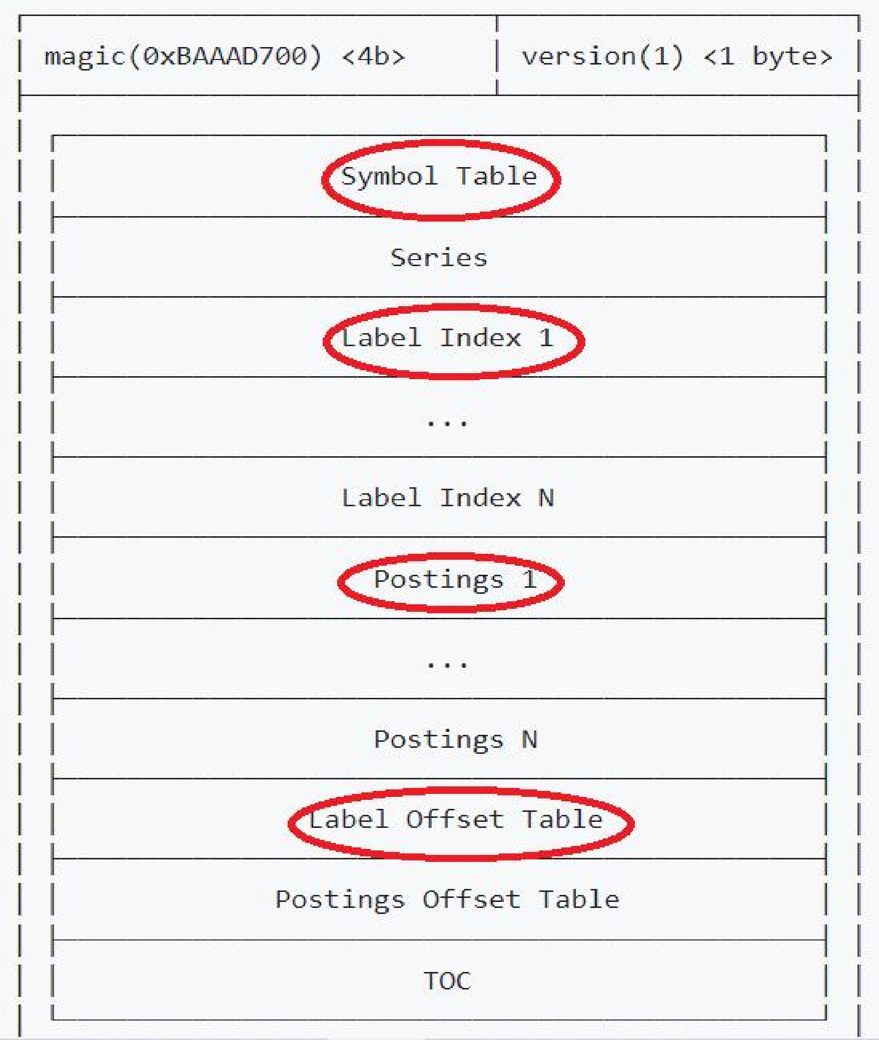
1. 超大规模数据压缩时间长，占用磁盘和内存大，受限 TSDB 索引中 symbols 长度 64GB 限制，高 level 压缩会导致失败。
2. 历史数据和乱序数据写入支持较差。
3. 仅支持 OpenMetrics/Prometheus 数据格式写入。

v3.0 - 高基数和大规模数据存储挑战



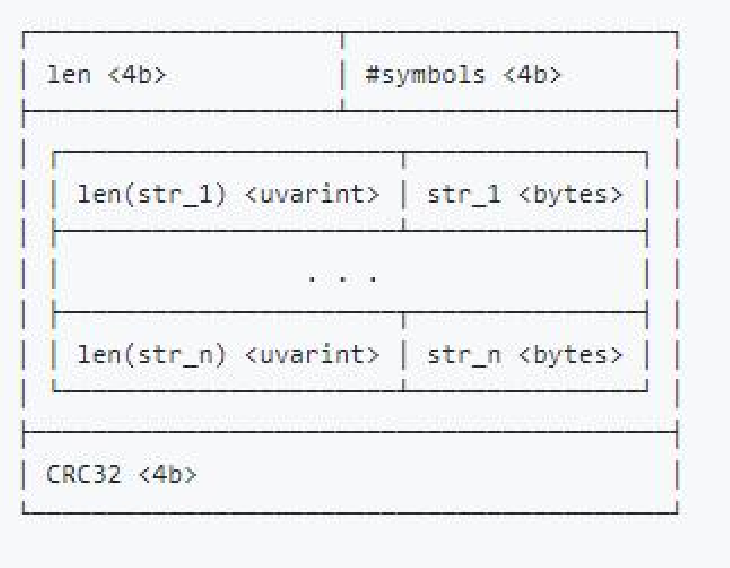
metric name
label names
label values
cardinality

server_responses
status_code
environment
1xx
2xx
3xx
4xx
5xx
prod
dev
prod, 1xx
prod, 2xx
prod, 3xx
prod, 4xx
prod, 5xx
dev, 1xx
dev, 2xx
dev, 3xx
dev, 4xx
dev, 5xx

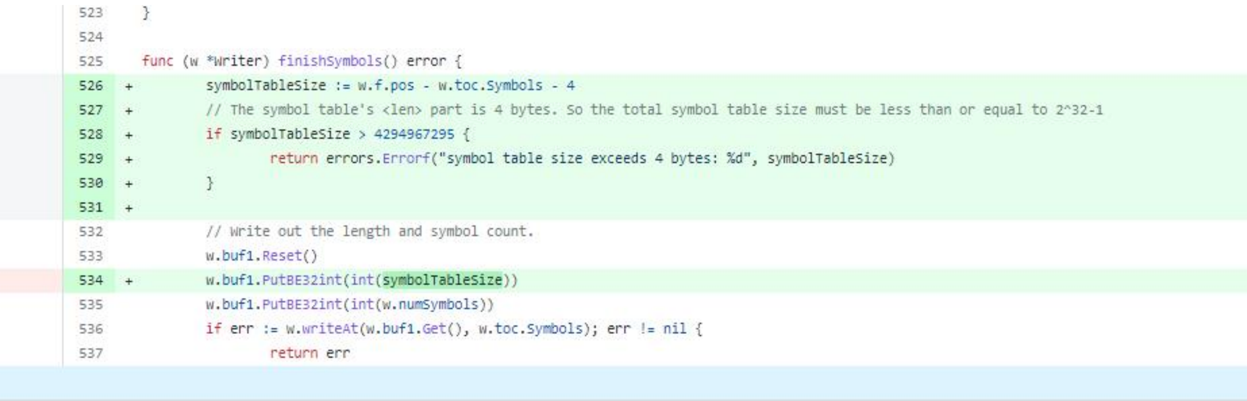


```
magic(0xBAAAD700) <4b> | version(1) <1 byte>
Symbol Table
Series
Label Index 1
...
Label Index N
Postings 1
...
Postings N
Label Offset Table
Postings Offset Table
TOC
```

v3.0 - 高基数和大规模数据存储挑战

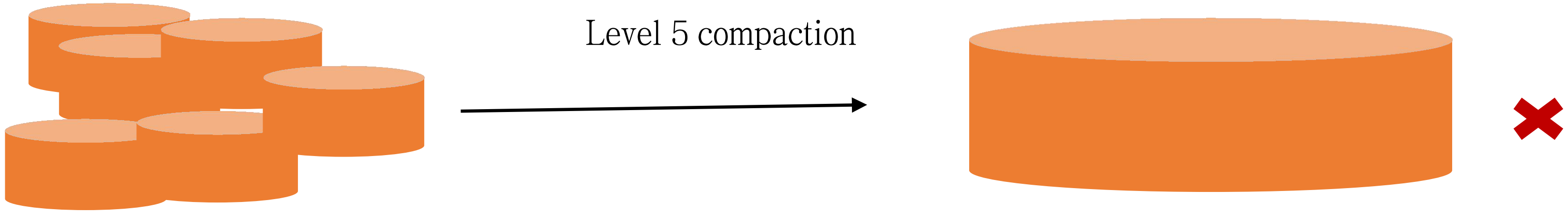


```
┌────────────────────┬─────────────────────┐
│ len <4b>           │ #symbols <4b>       │
├────────────────────┴─────────────────────┤
│ ┌──────────────────────┬───────────────┐ │
│ │ len(str_1) <uvarint> │ str_1 <bytes> │ │
│ ├──────────────────────┴───────────────┤ │
│ │                . . .                 │ │
│ ├──────────────────────┬───────────────┤ │
│ │ len(str_n) <uvarint> │ str_n <bytes> │ │
│ └──────────────────────┴───────────────┘ │
├──────────────────────────────────────────┤
│ CRC32 <4b>                               │
└──────────────────────────────────────────┘
```

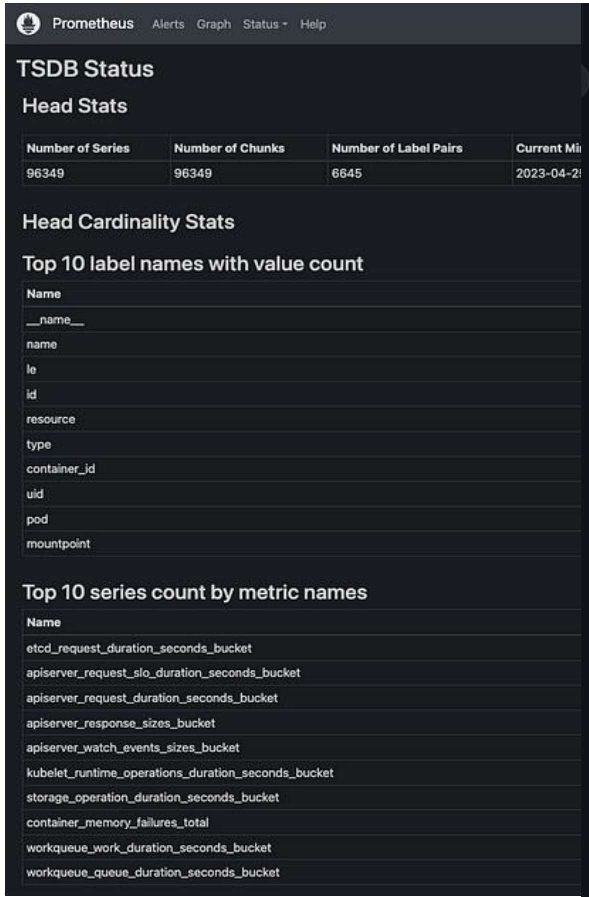


```
  }

  func (w *Writer) finishSymbols() error {
+         symbolTableSize := w.f.pos - w.toc.Symbols - 4
+         // The symbol table's <len> part is 4 bytes. So the total symbol table size must be less than or equal to 2^32-1
+         if symbolTableSize > 4294967295 {
+                 return errors.Errorf("symbol table size exceeds 4 bytes: %d", symbolTableSize)
+         }
+
          // Write out the length and symbol count.
          w.buf1.Reset()
+         w.buf1.PutBE32int(int(symbolTableSize))
          w.buf1.PutBE32int(int(w.numSymbols))
          if err := w.writeAt(w.buf1.Get(), w.toc.Symbols); err != nil {
                  return err
```

https://github.com/prometheus/prometheus/pull/9104



v3.0 - 高基数和大规模数据存储挑战



TSDB Status

Head Stats

Number of Series	Number of Chunks	Number of Label Pairs	Current Mi
96349	96349	6645	2023-04-2

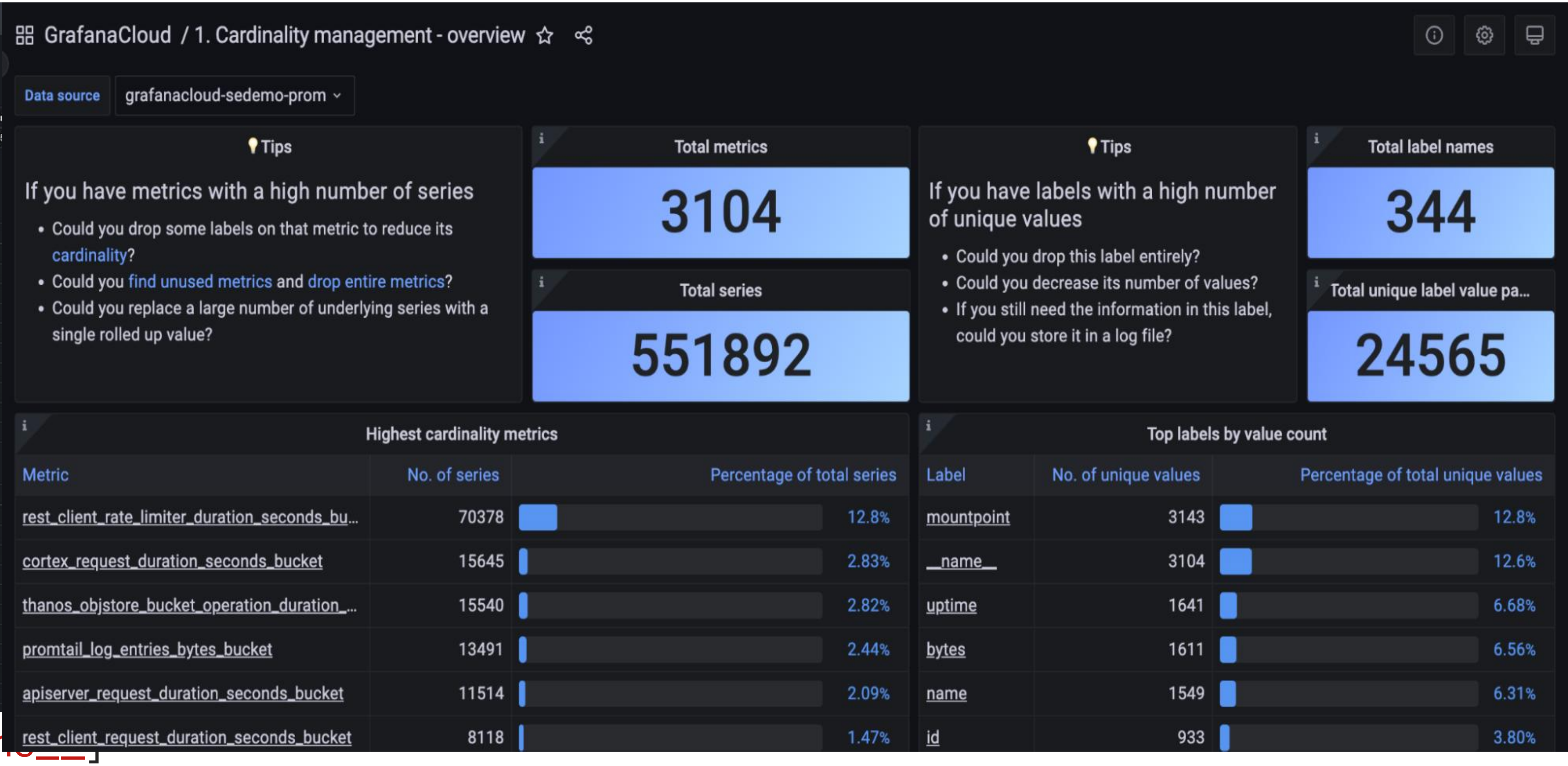
Head Cardinality Stats

Top 10 label names with value count

Name
__name__
name
le
id
resource
type
container_id
uid
pod
mountpoint

Top 10 series count by metric names

Name
etcd_request_duration_seconds_bucket
apiserver_request_slo_duration_seconds_bucket
apiserver_request_duration_seconds_bucket
apiserver_response_sizes_bucket
apiserver_watch_events_sizes_bucket
kubelet_runtime_operations_duration_seconds_bucket
storage_operation_duration_seconds_bucket
container_memory_failures_total
workqueue_work_duration_seconds_bucket
workqueue_queue_duration_seconds_bucket



💡Tips

If you have metrics with a high number of series

- Could you drop some labels on that metric to reduce its cardinality?
- Could you find unused metrics and drop entire metrics?
- Could you replace a large number of underlying series with a single rolled up value?

Total metrics: 3104

Total series: 551892

💡Tips

If you have labels with a high number of unique values

- Could you drop this label entirely?
- Could you decrease its number of values?
- If you still need the information in this label, could you store it in a log file?

Total label names: 344

Total unique label value pa...: 24565

Highest cardinality metrics

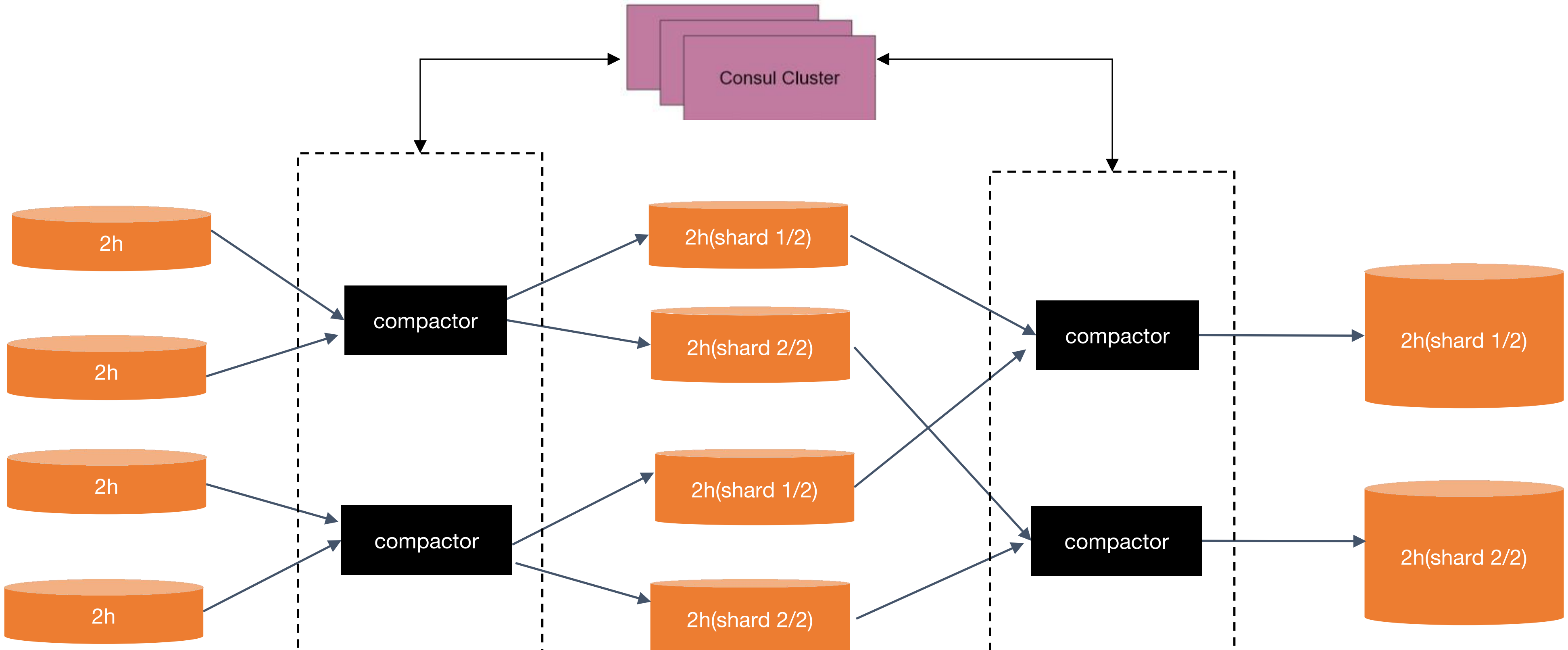
Metric	No. of series	Percentage of total series
rest_client_rate_limiter_duration_seconds_bu...	70378	12.8%
cortex_request_duration_seconds_bucket	15645	2.83%
thanos_objstore_bucket_operation_duration_...	15540	2.82%
promtail_log_entries_bytes_bucket	13491	2.44%
apiserver_request_duration_seconds_bucket	11514	2.09%
rest_client_request_duration_seconds_bucket	8118	1.47%

Top labels by value count

Label	No. of unique values	Percentage of total unique values
mountpoint	3143	12.8%
__name__	3104	12.6%
uptime	1641	6.68%
bytes	1611	6.56%
name	1549	6.31%
id	933	3.80%

```
- sourceLabels: [__name__]
action: drop
regex: 'node_(nf_conntrack_stat|
```

v3.0 - DB 分片与并行压缩

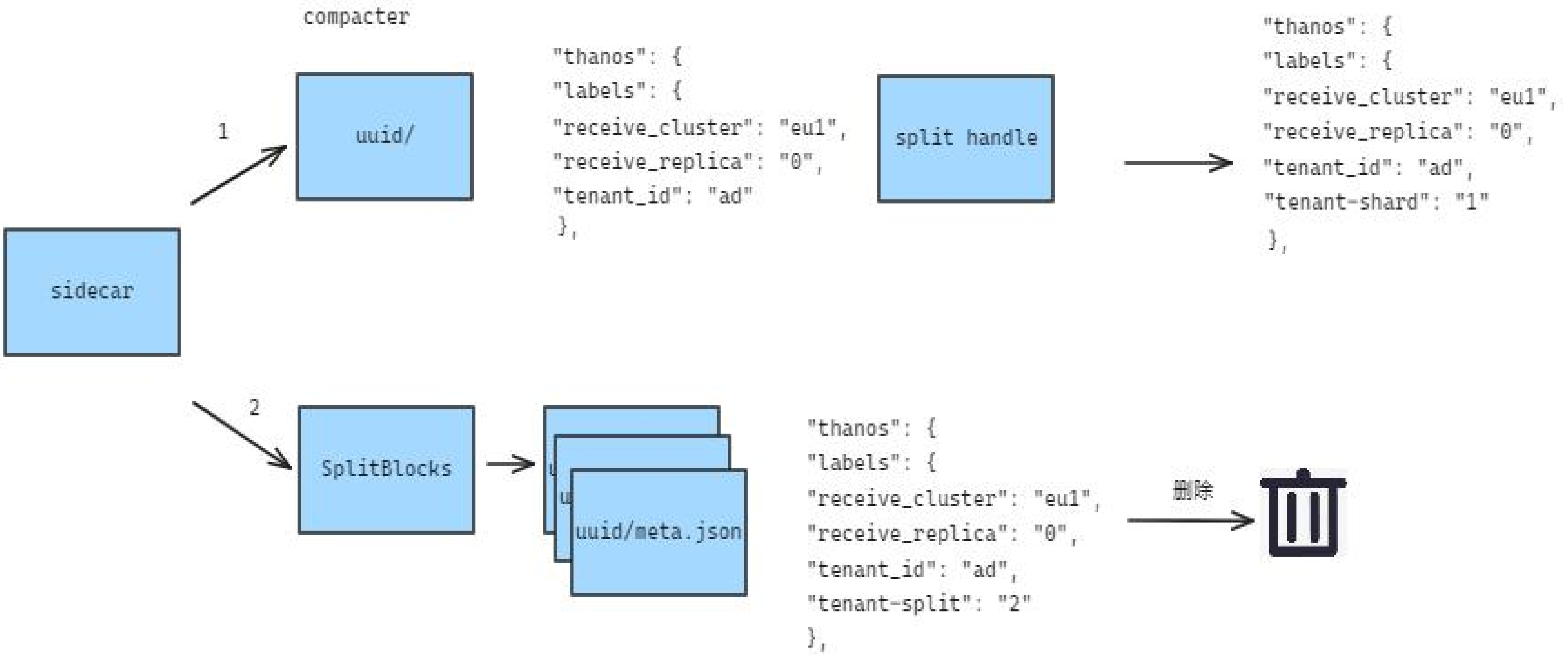


2h raw blocks

2h split blocks

2h compacted blocks

v3.0 - DB 分片与并行压缩

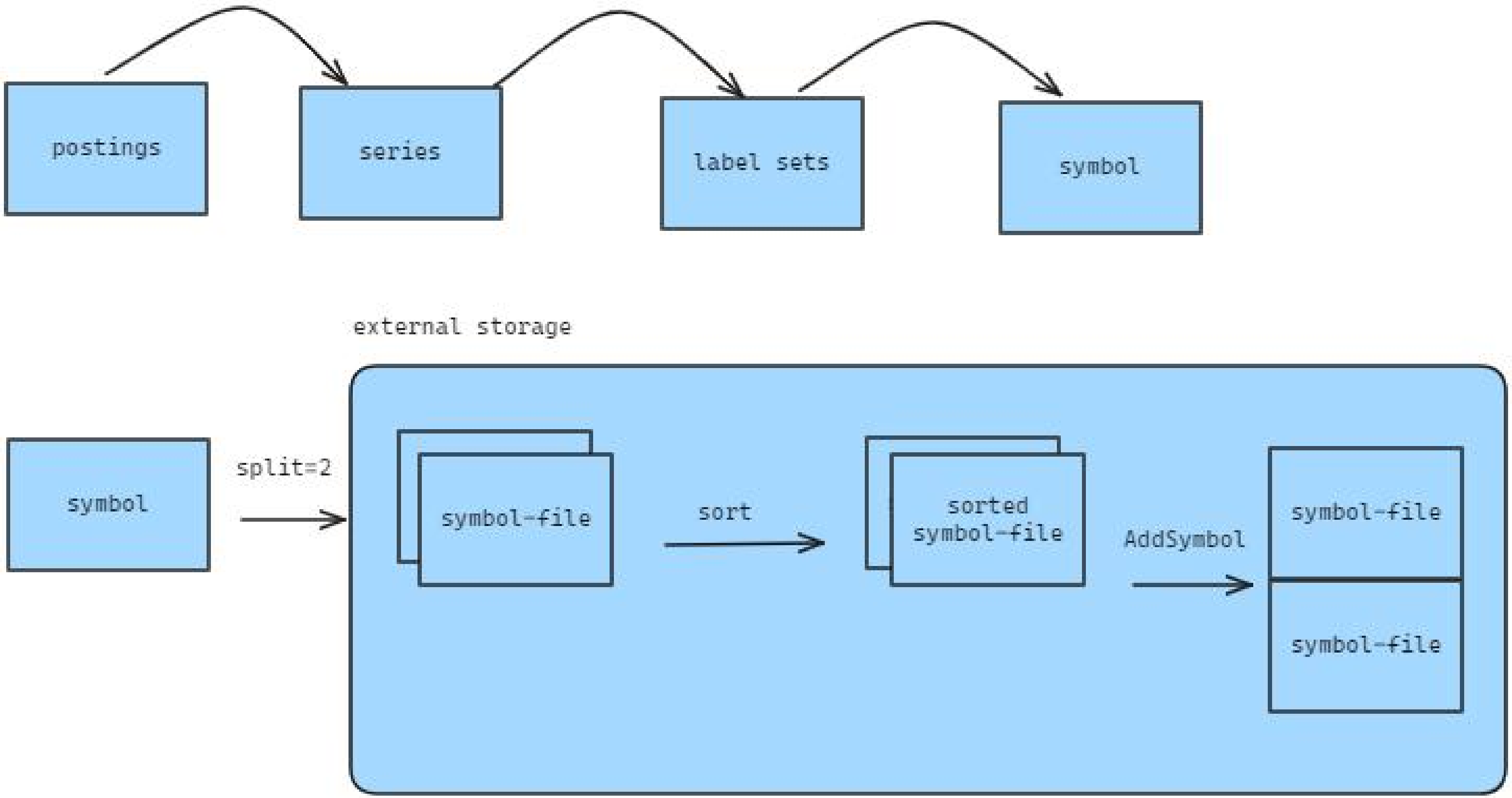


```
"thanos": {
"labels": {
"receive_cluster": "eu1",
"receive_replica": "0",
"tenant_id": "ad"
},
```

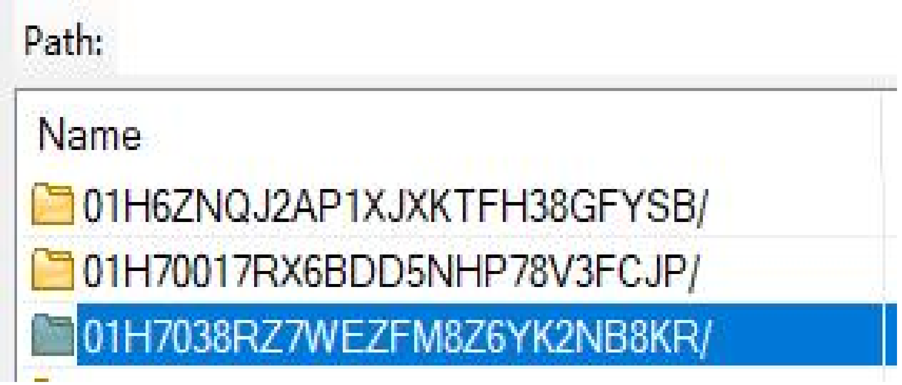
```
"thanos": {
"labels": {
"receive_cluster": "eu1",
"receive_replica": "0",
"tenant_id": "ad",
"tenant-shard": "1"
},
```

```
"thanos": {
"labels": {
"receive_cluster": "eu1",
"receive_replica": "0",
"tenant_id": "ad",
"tenant-split": "2"
},
```

v3.0 - DB 分片与并行压缩



v3.0 - DB 分片与并行压缩

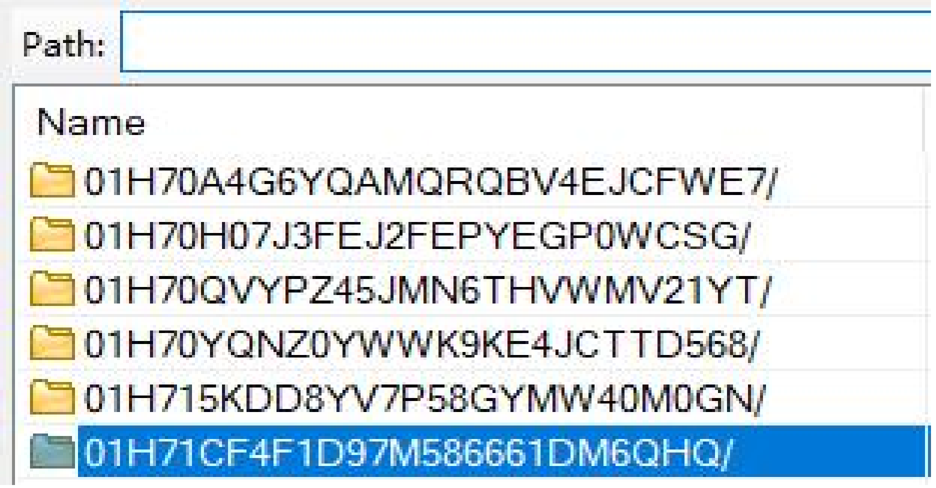


Path:

Name

```
01H6ZNQJ2AP1XJXKTFH38GFYSB/
01H70017RX6BDD5NHP78V3FCJP/
01H7038RZ7WEZFM8Z6YK2NB8KR/
```

```
"thanos": {
            "labels": {
          "receive_cluster": "eu1",
          "receive_replica": "0",
          "tenant_id": "ad",
          "tenant-split": "2",
         },
```



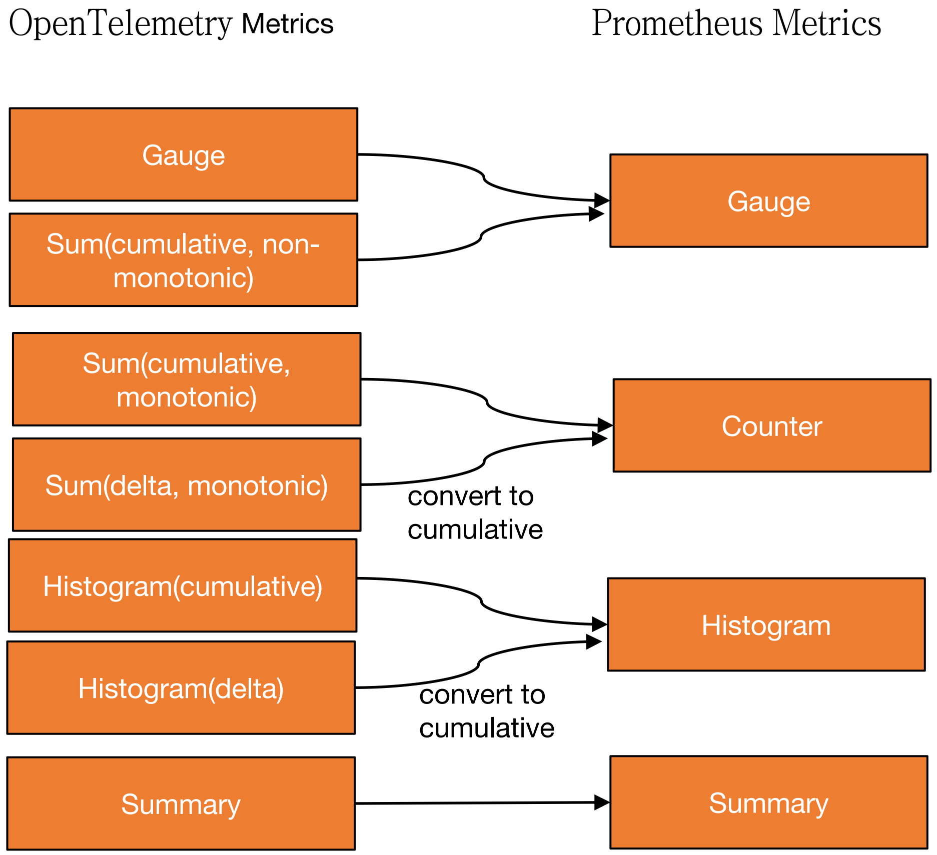
Path:

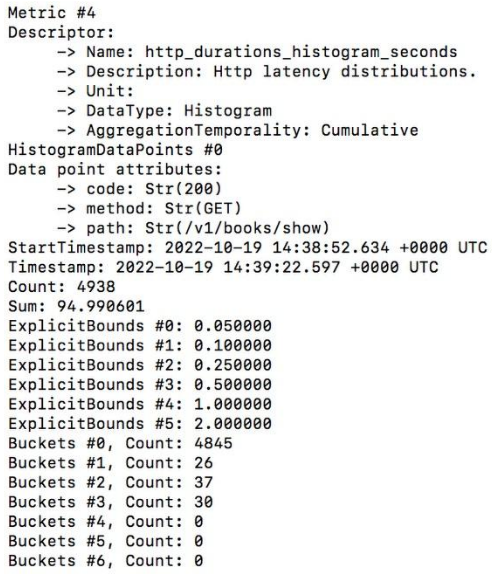
Name

```
01H70A4G6YQAMQRQBV4EJCFWE7/
01H70H07J3FEJ2FEPYEGP0WCSG/
01H70QVYPZ45JMN6THVWMV21YT/
01H70YQNZ0YWWK9KE4JCTTD568/
01H715KDD8YV7P58GYMW40M0GN/
01H71CF4F1D97M586661DM6QHQ/
```

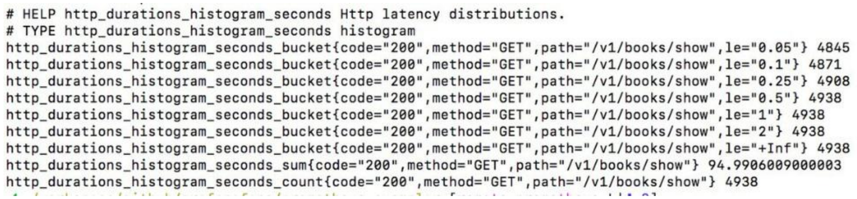
```
"thanos": {
            "labels": {
          "receive_cluster": "eu1",
          "receive_replica": "0",
          "tenant_id": "ad",
            "tenant-shard": "1",
            "tenant-split": "2",
         },
```

v3.0 - OpenTelemetry 协议支持





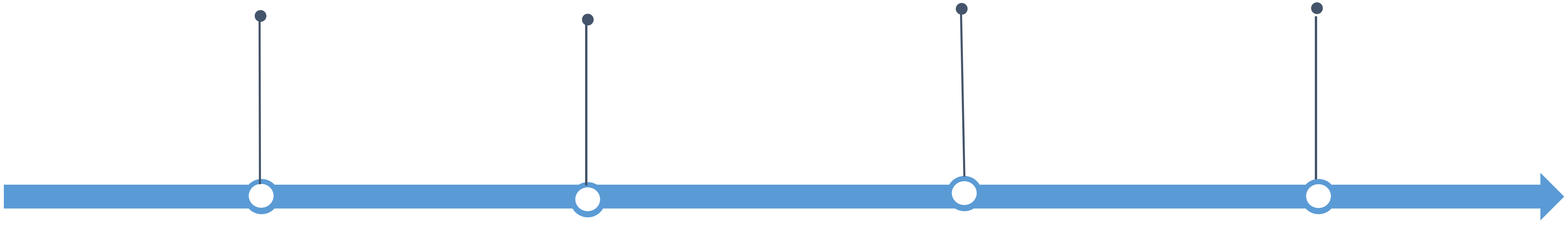
```
Metric #4
Descriptor:
     -> Name: http_durations_histogram_seconds
     -> Description: Http latency distributions.
     -> Unit:
     -> DataType: Histogram
     -> AggregationTemporality: Cumulative
HistogramDataPoints #0
Data point attributes:
     -> code: Str(200)
     -> method: Str(GET)
     -> path: Str(/v1/books/show)
StartTimestamp: 2022-10-19 14:38:52.634 +0000 UTC
Timestamp: 2022-10-19 14:39:22.597 +0000 UTC
Count: 4938
Sum: 94.990601
ExplicitBounds #0: 0.050000
ExplicitBounds #1: 0.100000
ExplicitBounds #2: 0.250000
ExplicitBounds #3: 0.500000
ExplicitBounds #4: 1.000000
ExplicitBounds #5: 2.000000
Buckets #0, Count: 4845
Buckets #1, Count: 26
Buckets #2, Count: 37
Buckets #3, Count: 30
Buckets #4, Count: 0
Buckets #5, Count: 0
Buckets #6, Count: 0
```



```
# HELP http_durations_histogram_seconds Http latency distributions.
# TYPE http_durations_histogram_seconds histogram
http_durations_histogram_seconds_bucket{code="200",method="GET",path="/v1/books/show",le="0.05"} 4845
http_durations_histogram_seconds_bucket{code="200",method="GET",path="/v1/books/show",le="0.1"} 4871
http_durations_histogram_seconds_bucket{code="200",method="GET",path="/v1/books/show",le="0.25"} 4908
http_durations_histogram_seconds_bucket{code="200",method="GET",path="/v1/books/show",le="0.5"} 4938
http_durations_histogram_seconds_bucket{code="200",method="GET",path="/v1/books/show",le="1"} 4938
http_durations_histogram_seconds_bucket{code="200",method="GET",path="/v1/books/show",le="2"} 4938
http_durations_histogram_seconds_bucket{code="200",method="GET",path="/v1/books/show",le="+Inf"} 4938
http_durations_histogram_seconds_sum{code="200",method="GET",path="/v1/books/show"} 94.9906009000003
http_durations_histogram_seconds_count{code="200",method="GET",path="/v1/books/show"} 4938
```

可观测数据融合平台

夯实metrics能力



项目启动

2019年

基于多副本的分布式存储、
实现 scraper、ruler 高可靠、
多租户。

v1.0

基于对象存储的分
级存储、集中压缩
去重、自动降精度。

v2.0

DB分片和并行压缩、
乱序/历史数据写入、
exempalr、多协议支持。

v3.0

抽象平台层

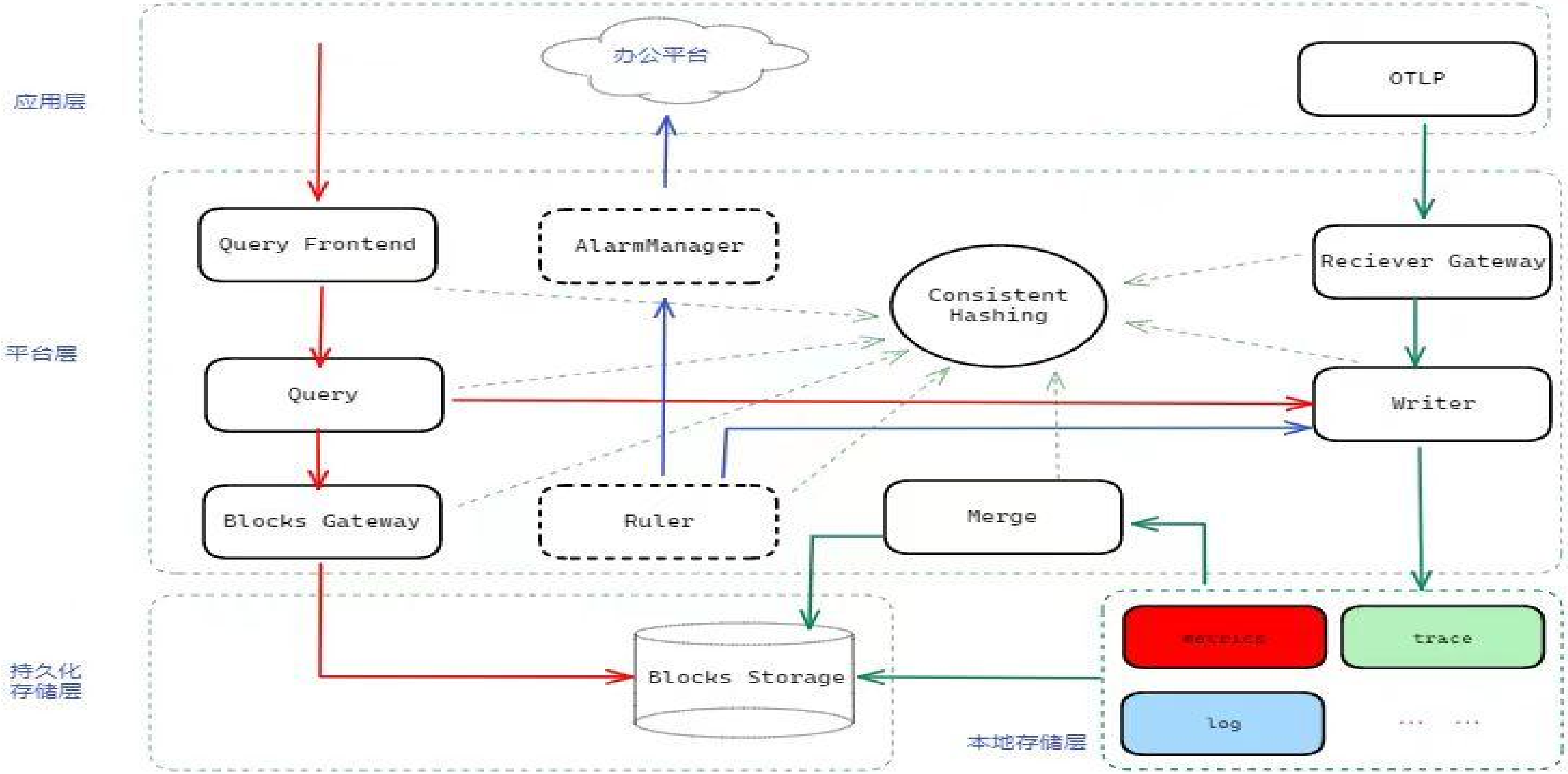
分布式、高可用、高基数

数据融合平台

扩展log、trace等底座能力，
作为数据存储层

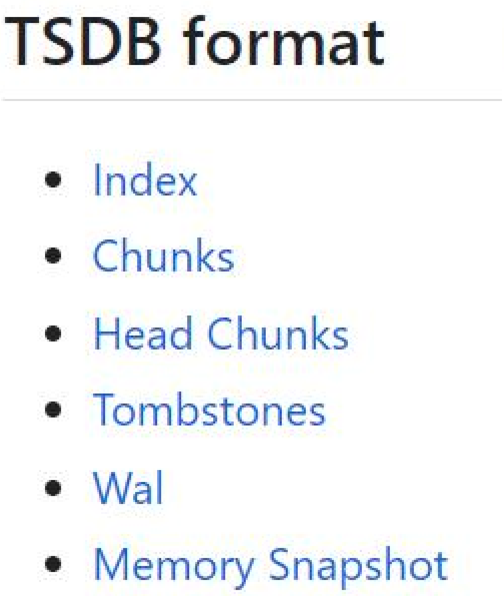
可观测数据融合平台

抽象平台层



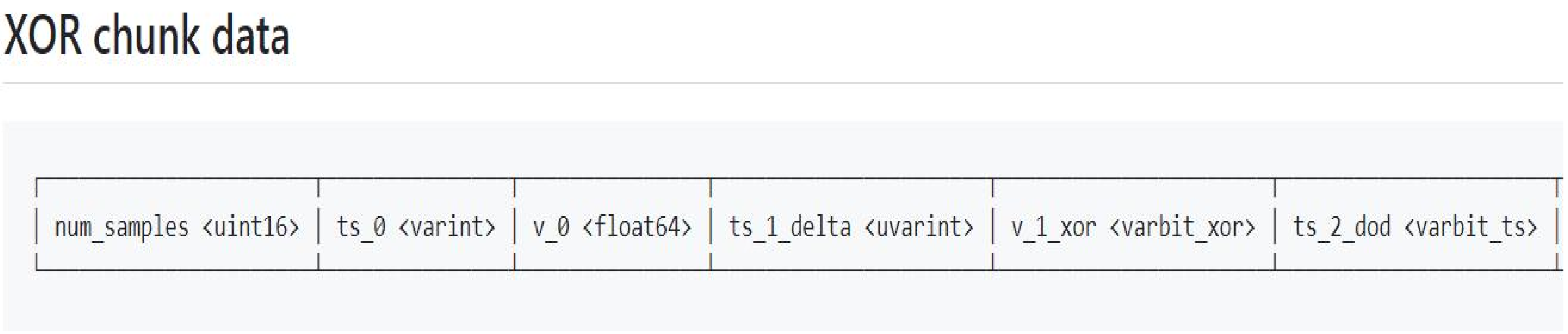
可观测数据融合平台

扩展log、trace能力



TSDB format

- Index
- Chunks
- Head Chunks
- Tombstones
- Wal
- Memory Snapshot



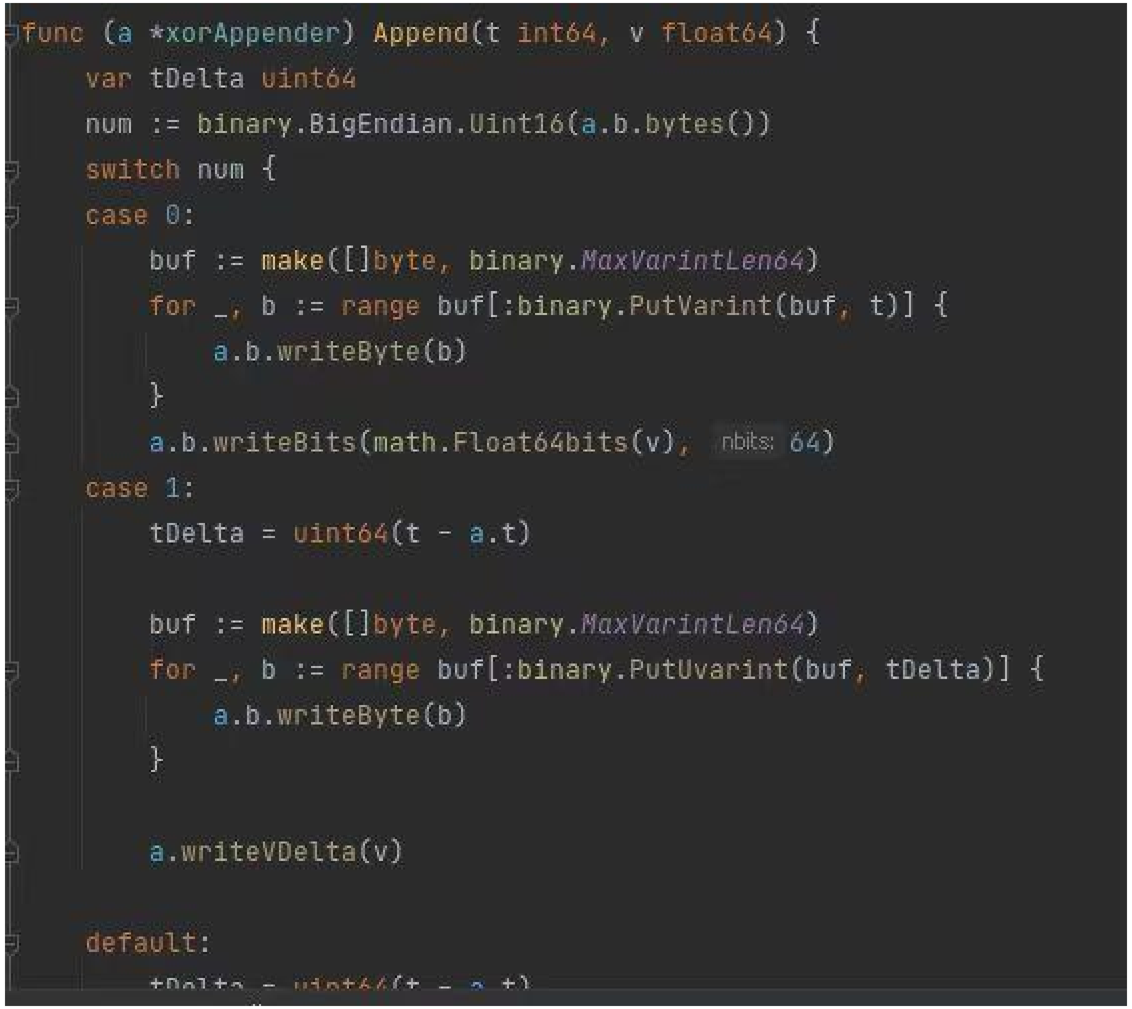
XOR chunk data

```
┌──────────────────────┬───────────────┬───────────────┬──────────────────────┬──────────────────────┬──────────────────────┐
| num_samples <uint16> | ts_0 <varint> | v_0 <float64> | ts_1_delta <uvarint> | v_1_xor <varbit_xor> | ts_2_dod <varbit_ts> |
└──────────────────────┴───────────────┴───────────────┴──────────────────────┴──────────────────────┴──────────────────────┘
```

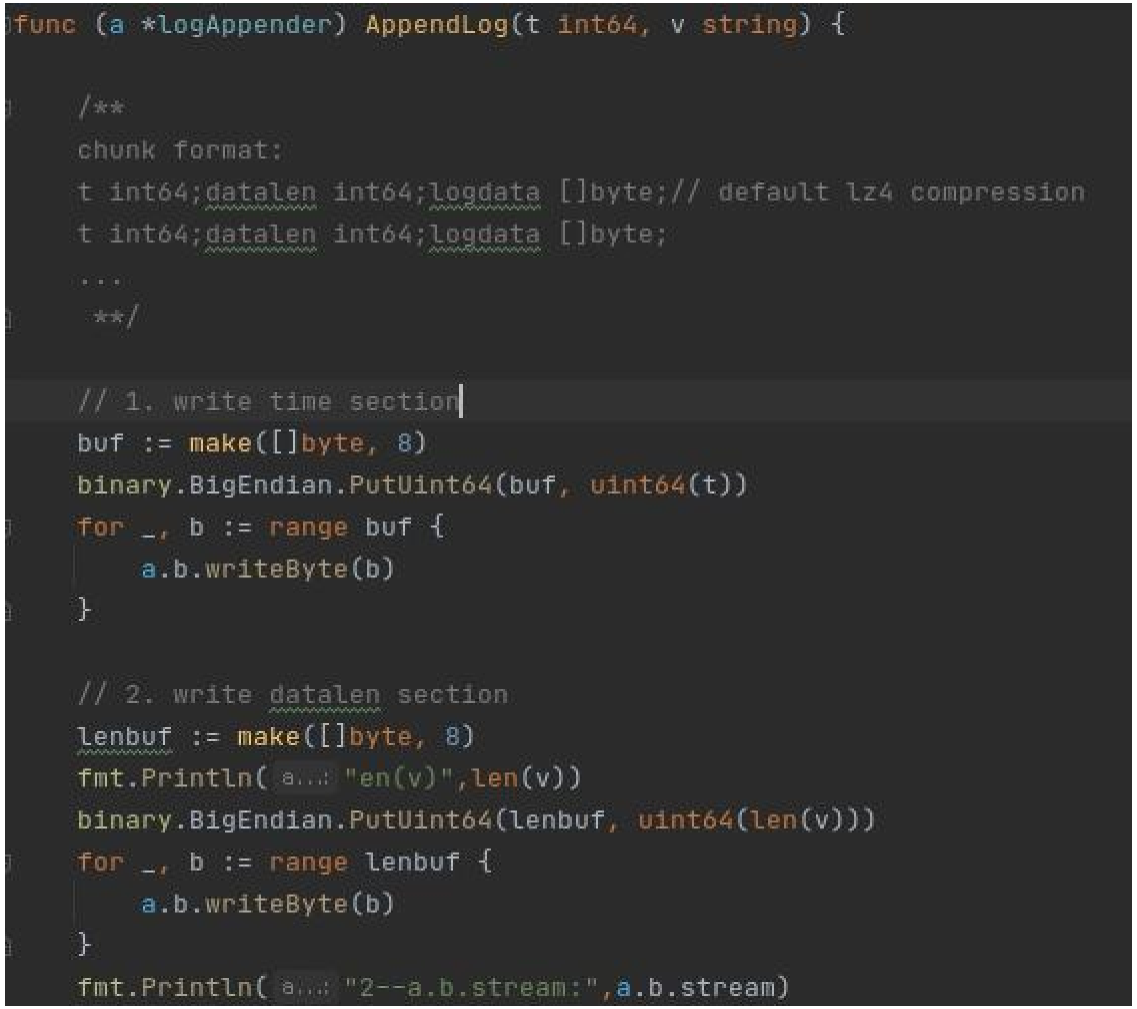
log chunk data

num_samples <uint16>	ts_0 <int64>	data_len <int64>	log_data <varint>	ts_1 <int64>

扩展log 能力



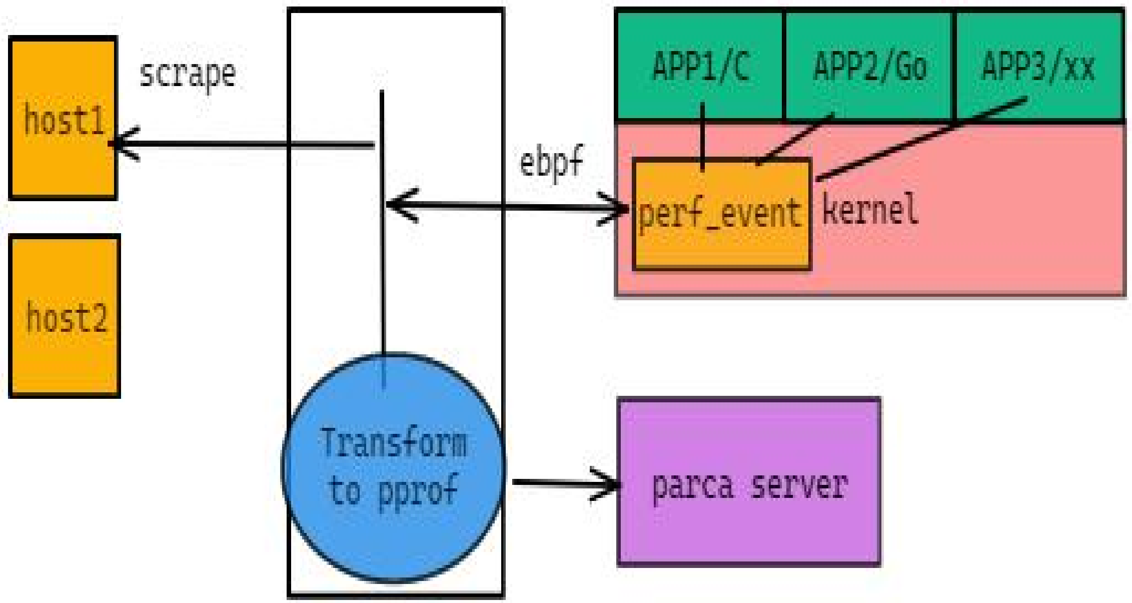
```
func (a *xorAppender) Append(t int64, v float64) {
    var tDelta uint64
    num := binary.BigEndian.Uint16(a.b.bytes())
    switch num {
    case 0:
        buf := make([]byte, binary.MaxVarintLen64)
        for _, b := range buf[:binary.PutVarint(buf, t)] {
            a.b.writeByte(b)
        }
        a.b.writeBits(math.Float64bits(v),  nbits: 64)
    case 1:
        tDelta = uint64(t - a.t)

        buf := make([]byte, binary.MaxVarintLen64)
        for _, b := range buf[:binary.PutUvarint(buf, tDelta)] {
            a.b.writeByte(b)
        }

        a.writeVDelta(v)

    default:
        tDelta = uint64(t - a.t)
```

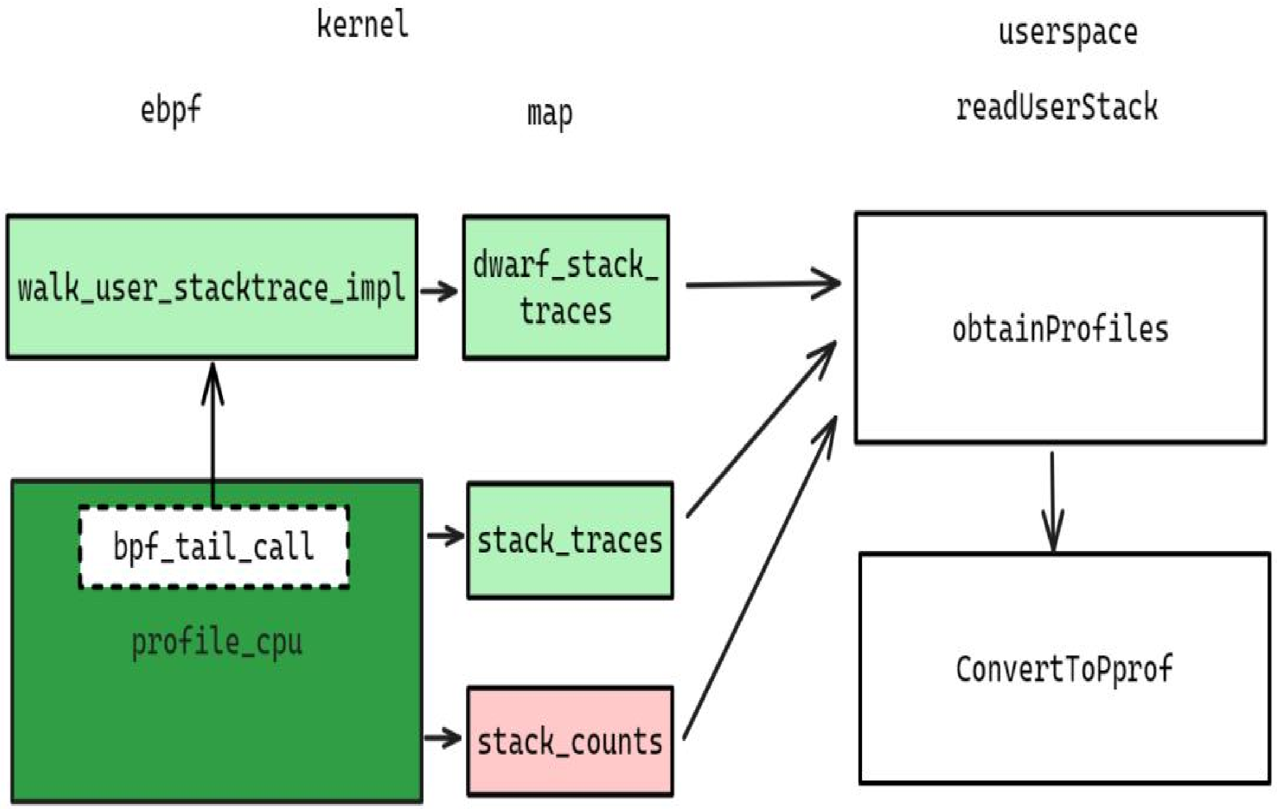


```
func (a *logAppender) AppendLog(t int64, v string) {

    /**
    chunk format:
    t int64;datalen int64;logdata []byte;// default lz4 compression
    t int64;datalen int64;logdata []byte;
    ...
     **/

    // 1. write time section
    buf := make([]byte, 8)
    binary.BigEndian.PutUint64(buf, uint64(t))
    for _, b := range buf {
        a.b.writeByte(b)
    }

    // 2. write datalen section
    lenbuf := make([]byte, 8)
    fmt.Println( a...: "en(v)",len(v))
    binary.BigEndian.PutUint64(lenbuf, uint64(len(v)))
    for _, b := range lenbuf {
        a.b.writeByte(b)
    }
    fmt.Println( a...: "2--a.b.stream:",a.b.stream)
```

eBPF

Parca

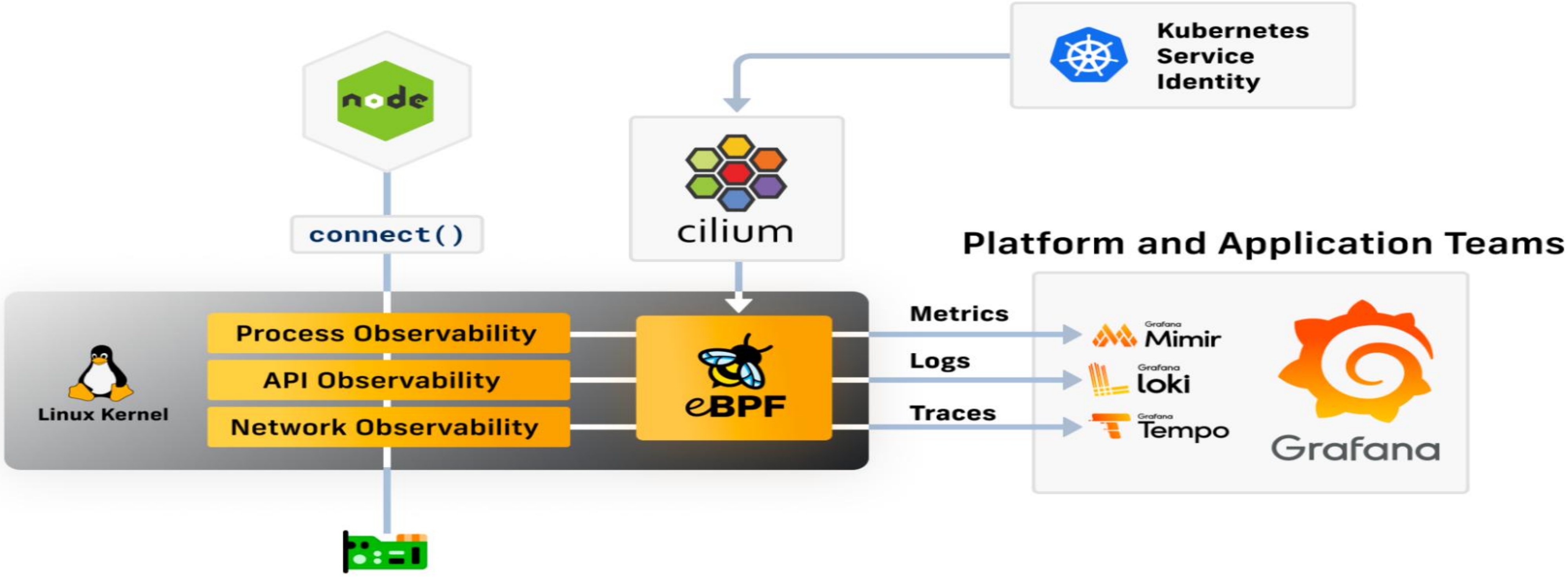




[bpf profile_cpu] "has_unwind_information" works rather than "has_fp" even though a porgrame(C) compiled with frame pointers https://github.com/parca-dev/parca-agent/issues/1657

OpenTelemetry Auto Instrumentation using eBPF
https://github.com/open-telemetry/opentelemetry-go-instrumentation/pull/149

eBPF


node
connect()
cilium
Kubernetes
Service
Identity
Platform and Application Teams
Linux Kernel
Process Observability
API Observability
Network Observability
eBPF
Metrics
Logs
Traces
Grafana Mimir
Grafana loki
Grafana Tempo
Grafana

总结

- 1、各类方案百花齐放，多调研和灰度验证
- 2、平台聚焦服务业务为主
- 3、找到适合业务的技术方案
- 4、慎重叠加解决方案
- 5、尽量抽象平台能力，扩展底层能力

谢谢

欢迎关注公众号~

Grafana fans

欢迎关注公众号~

THANKS