My Homelab Multitool

The OpenTelemetry Collector

Josh Lee • Open Source Advocate @ Altinity

Josh Lee

Open Source Advocate

@ Altinity

What are we monitoring?

What are we monitoring?

- MiniPC-Powered Home Lab
- *Some* critical services
- Proxmox, NixOS
- Lots and lots of containers...

“Domains” / “Users”

Management & Control Plane

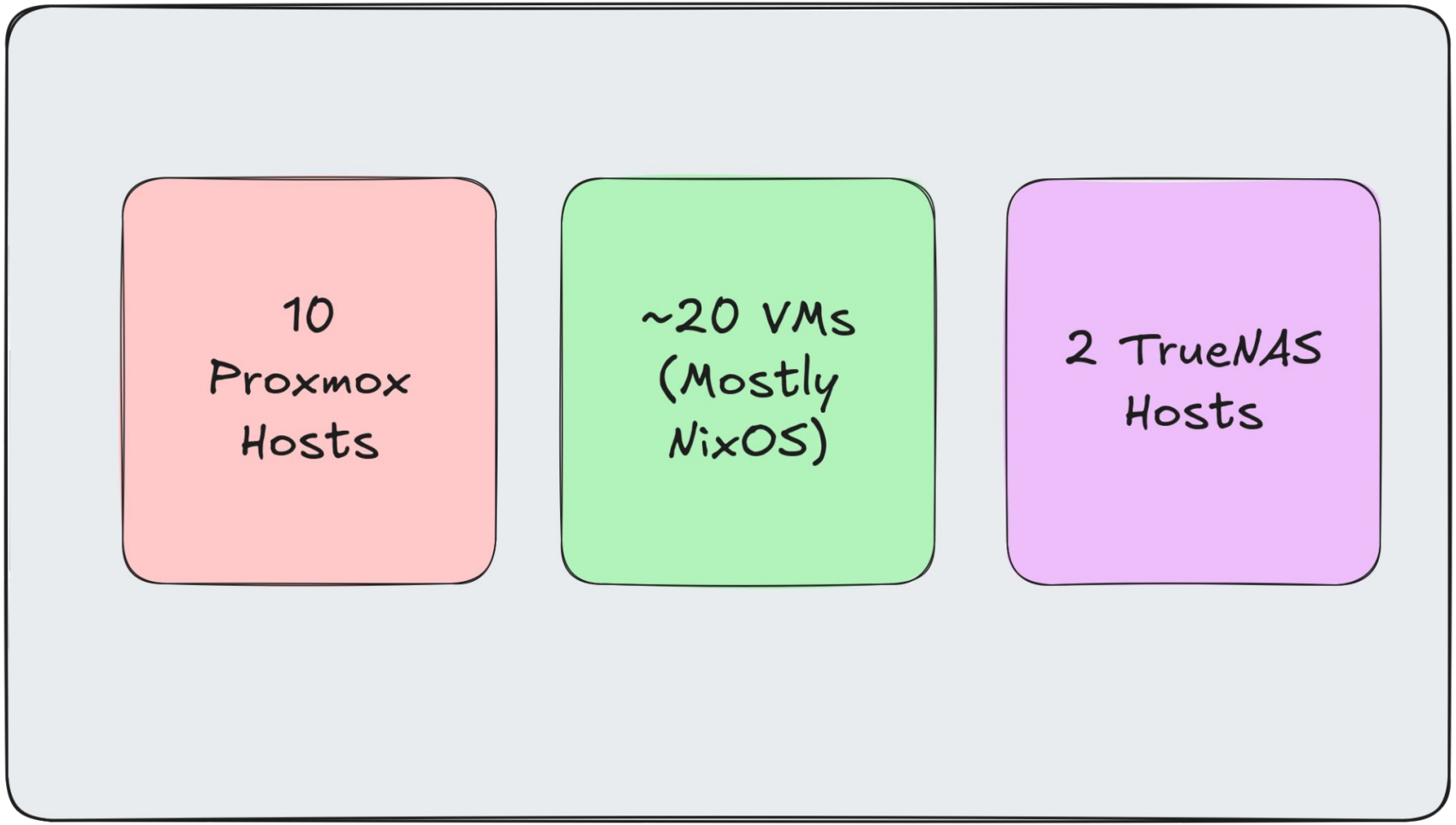
- Proxmox DCM
- Proxmox Backup
- Containers for management & monitoring

Personal Home Lab Services

- Data shares
- Containerized Applications (3rd Party)
- Production & staging environments

Development Sandbox

- Workstation environments
- Build servers
- Containerized services
- Ephemeral K8s clusters


10
Proxmox
Hosts
~20 VMs
(Mostly
NixOS)
2 TrueNAS
Hosts

Managing the Menagerie

- Ansible for Proxmox
- NixOS for VMs
- OpenTofu for “cloud stuff”
- GitOps + LLMs = 😁
- Manual VM Creation = 😢


Monitoring Goals

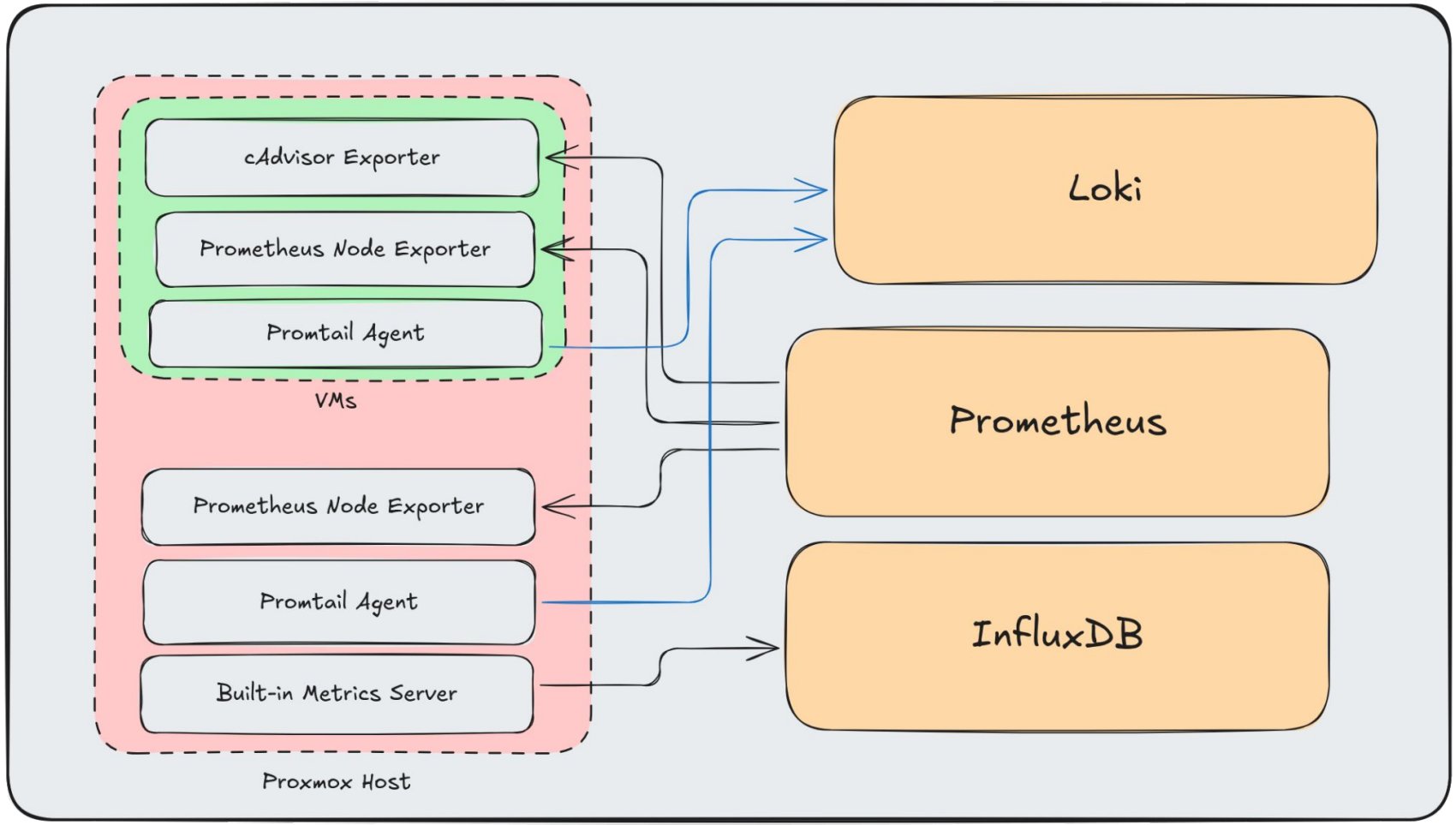
Monitoring Goals

Are my proxmox nodes healthy?

Monitoring Goals

- Is memory or CPU over-saturated?
- Am I ok on disk space?
- Are any VMs misbehaving?
- Container issues: ssh and journald
- (with LLM assistants = 😁)

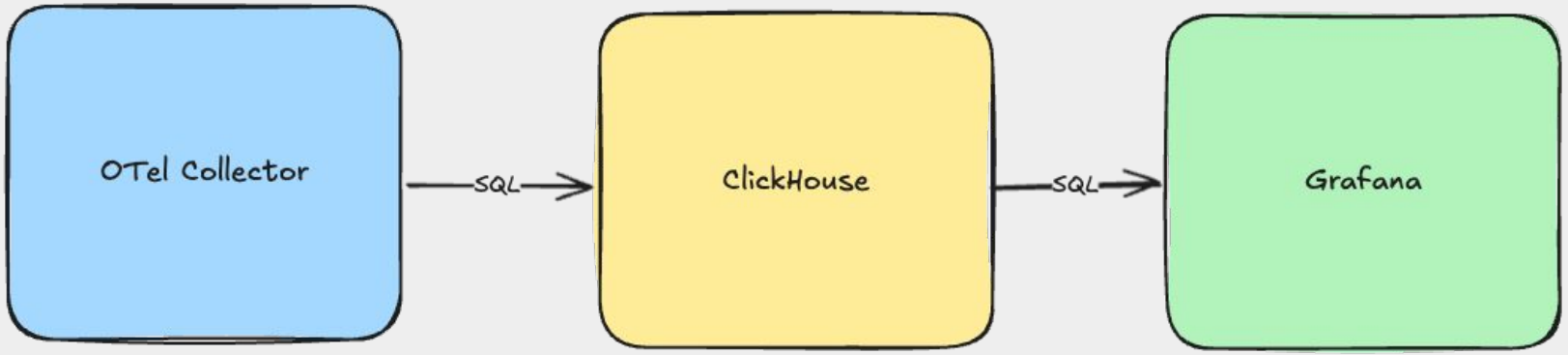

A Common Start


cAdvisor Exporter
Prometheus Node Exporter
Promtail Agent
VMs
Prometheus Node Exporter
Promtail Agent
Built-in Metrics Server
Proxmox Host
Loki
Prometheus
InfluxDB


Introducing ClickHouse

What is ClickHouse?

- Apache 2.0
- Single binary
- SQL Compatible
- Columnar
- *Really really* fast
- Laptop to exabyte scale
- Eats cardinality for breakfast



https://www.youtube.com/watch?v=wGBQU9sykL0

“O11y in One”

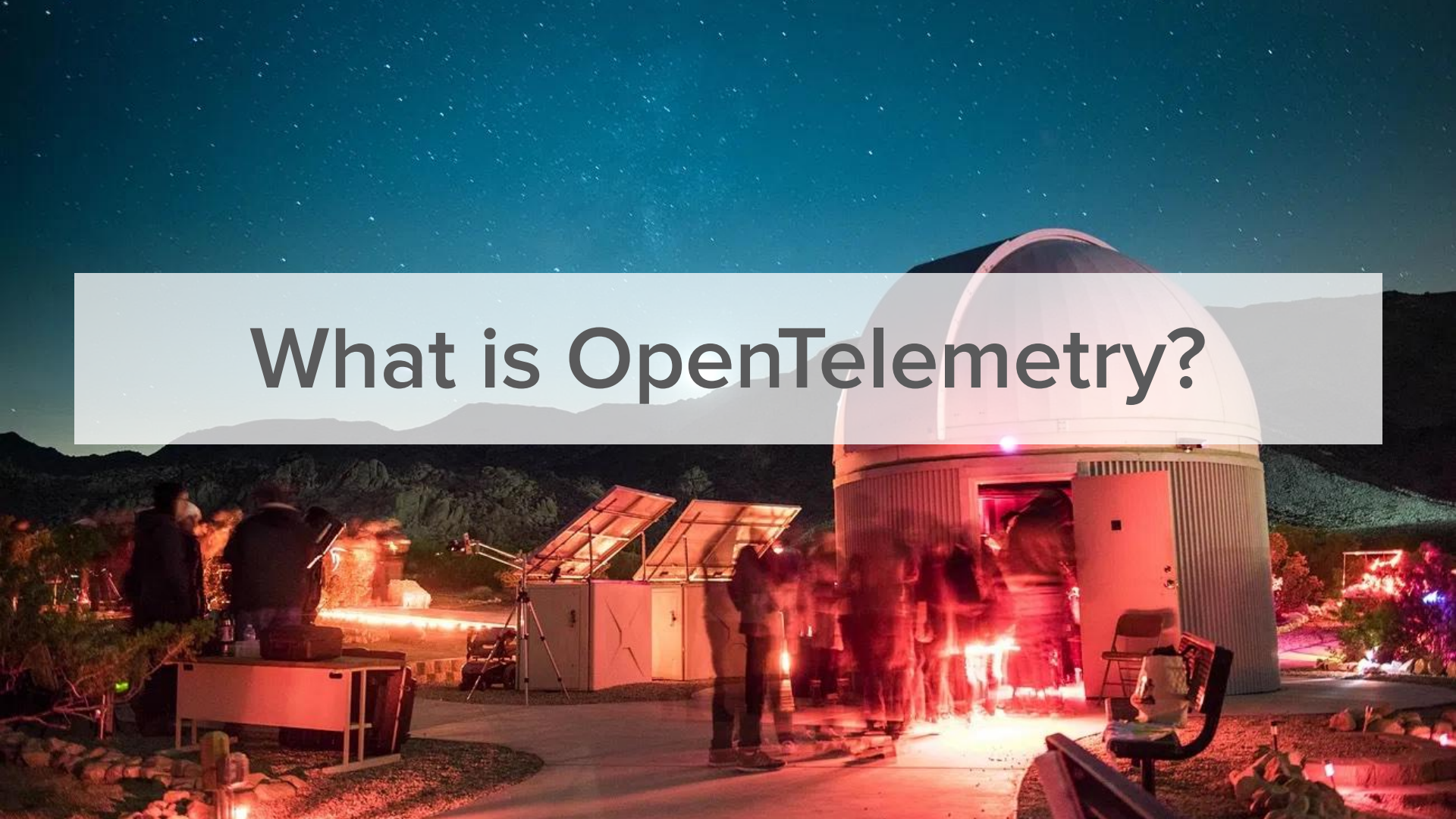
```
SHOW CREATE TABLE system.opentelemetry_span_log;
---
CREATE TABLE system.opentelemetry_span_log
(
    `hostname` LowCardinality(
String) COMMENT 'The hostname where this span was captured.',
    `trace_id` UUID COMMENT
'ID of the trace for executed query.',
    `span_id` UInt64 COMMENT
'ID of the trace span.',
    `parent_span_id` UInt64 COMMENT
'ID of the parent trace span.',
    `operation_name` LowCardinality(
String) COMMENT 'The name of the operation.',
    `kind` Enum8(
'INTERNAL' = 0, 'SERVER' = 1, 'CLIENT' = 2, 'PRODUCER' = 3, 'CONSUMER' = 4) COMMENT 'The SpanKind of the span. INTERNAL — Indicates that the
span represents an internal operation within an application. SERVER — Indicates that the span covers server-side handling of a synchronous RPC
or other remote request. CLIENT — Indicates that the span describes a request to some remote service. PRODUCER — Indicates that the span
describes the initiators of an asynchronous request. This parent span will often end before the corresponding child CONSUMER span, possibly
even before the child span starts. CONSUMER - Indicates that the span describes a child of an asynchronous PRODUCER request.',
    `start_time_us` UInt64 COMMENT
'The start time of the trace span (in microseconds).',
    `finish_time_us` UInt64 COMMENT
'The finish time of the trace span (in microseconds).',
    `finish_date`
Date COMMENT 'The finish date of the trace span.',
    `attribute` Map(LowCardinality(
String), String) COMMENT 'Attribute depending on the trace span. They are filled in according to the recommendations in the OpenTelemetry
standard.',
    `attribute.names` Array(LowCardinality(
String)) ALIAS mapKeys(attribute),
    `attribute.values` Array(
String) ALIAS mapValues(attribute)
)
ENGINE = MergeTree
PARTITION
BY toYYYYMM(finish_date)
ORDER BY (finish_date, finish_time_us, trace_id)
SETTINGS index_granularity =
8192
COMMENT
'Contains information about trace spans for executed queries.'


1 row in set. Elapsed: 0.034 sec.
```

Built-in Spans Table

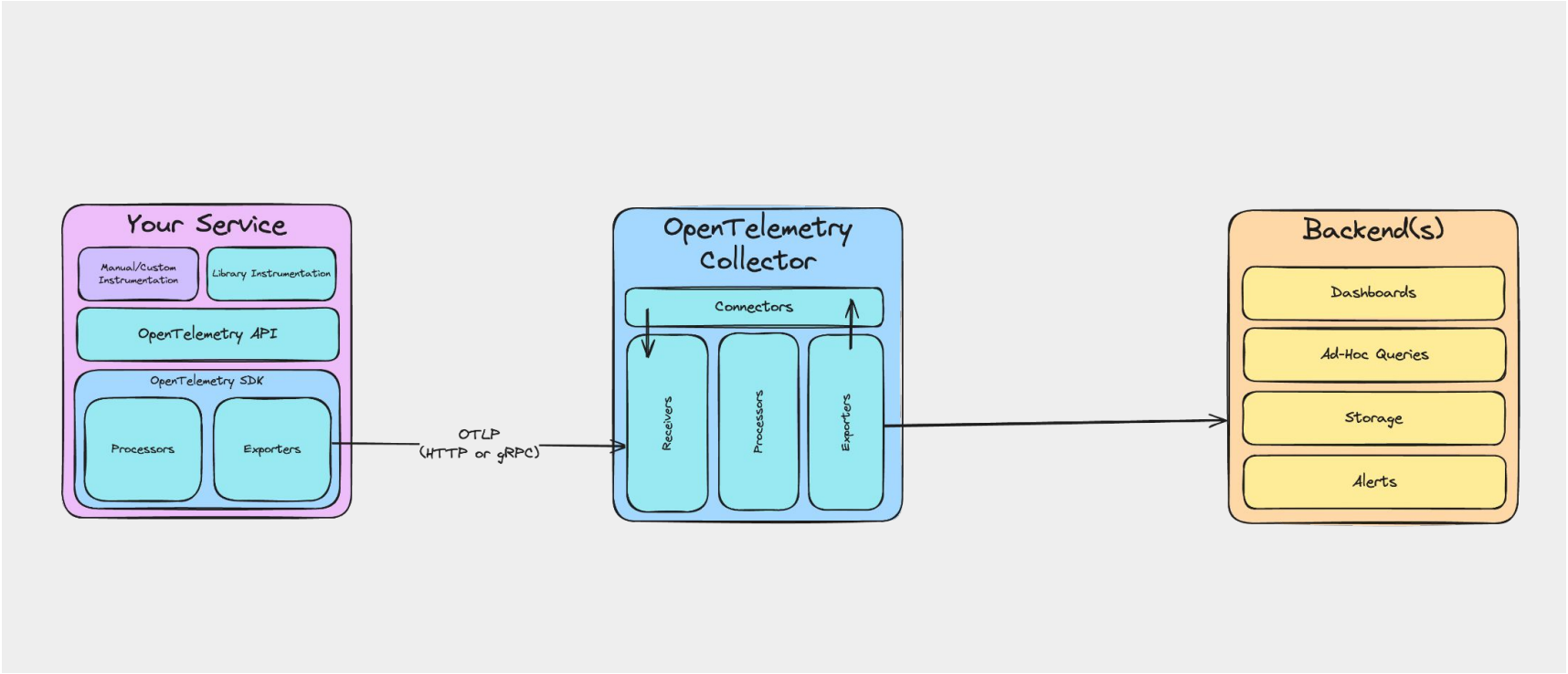
System table created dynamically

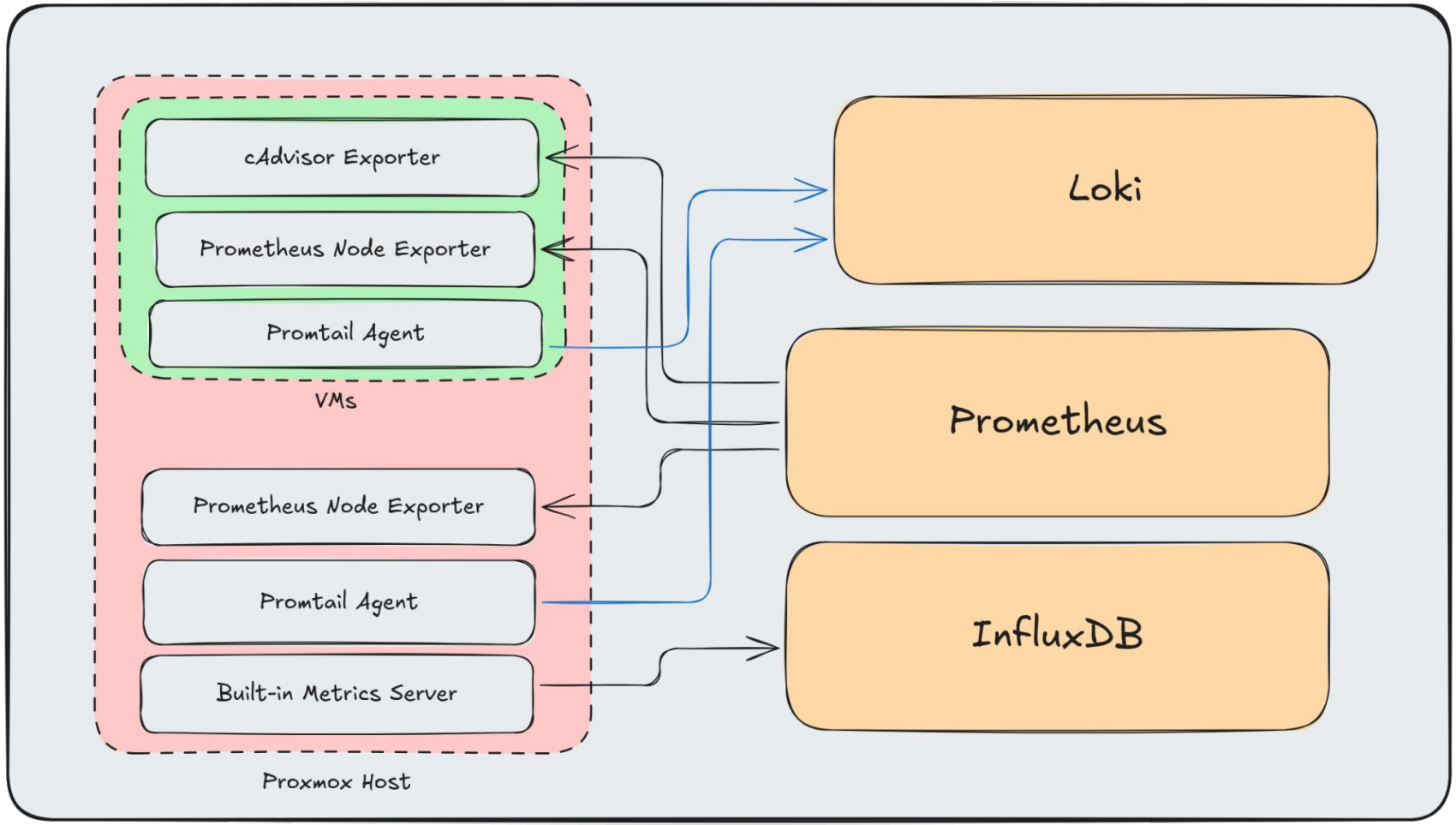
Enable tracing via setting (on query or user profile) or request headers

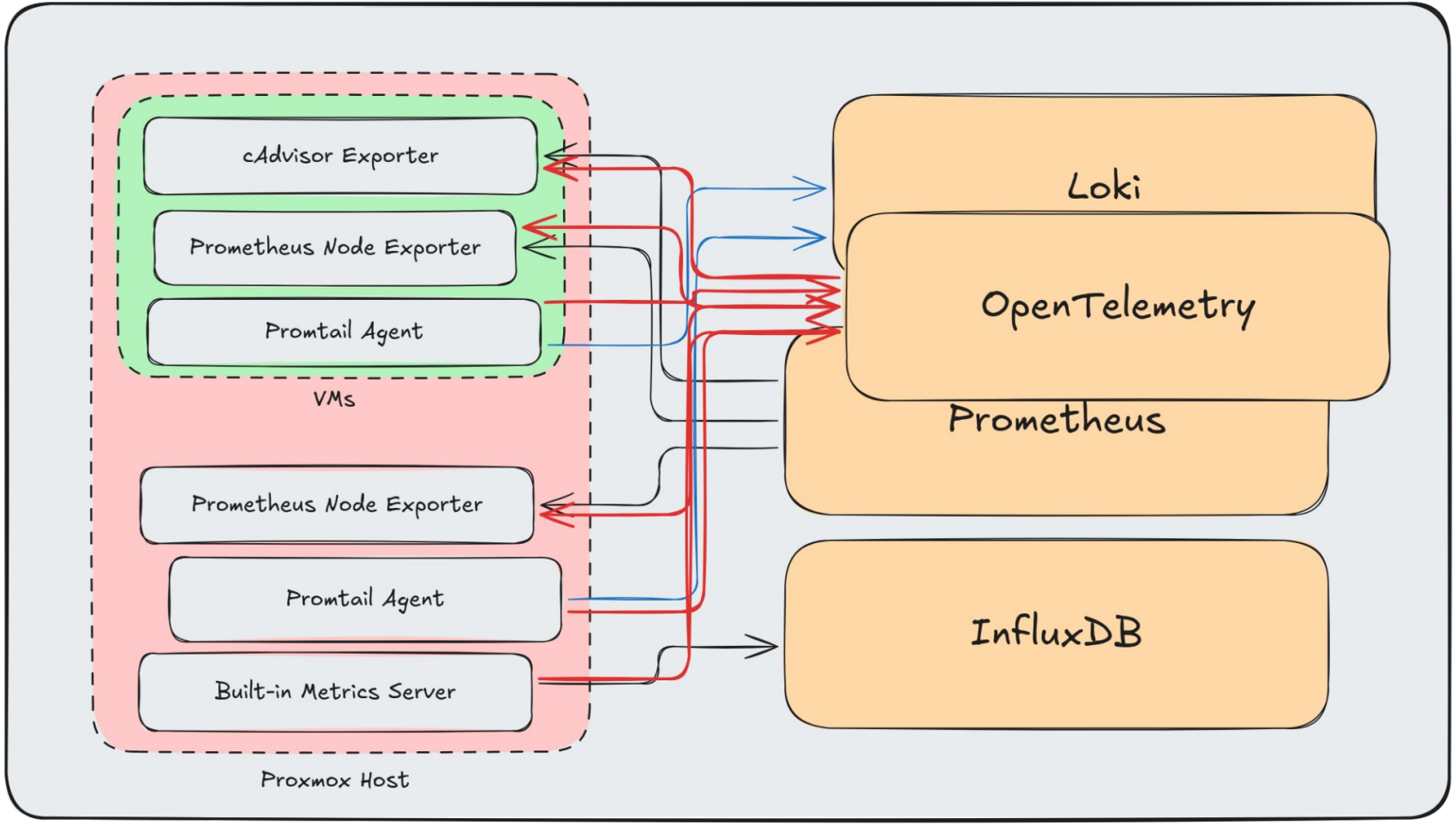
What is OpenTelemetry?



What is OpenTelemetry?



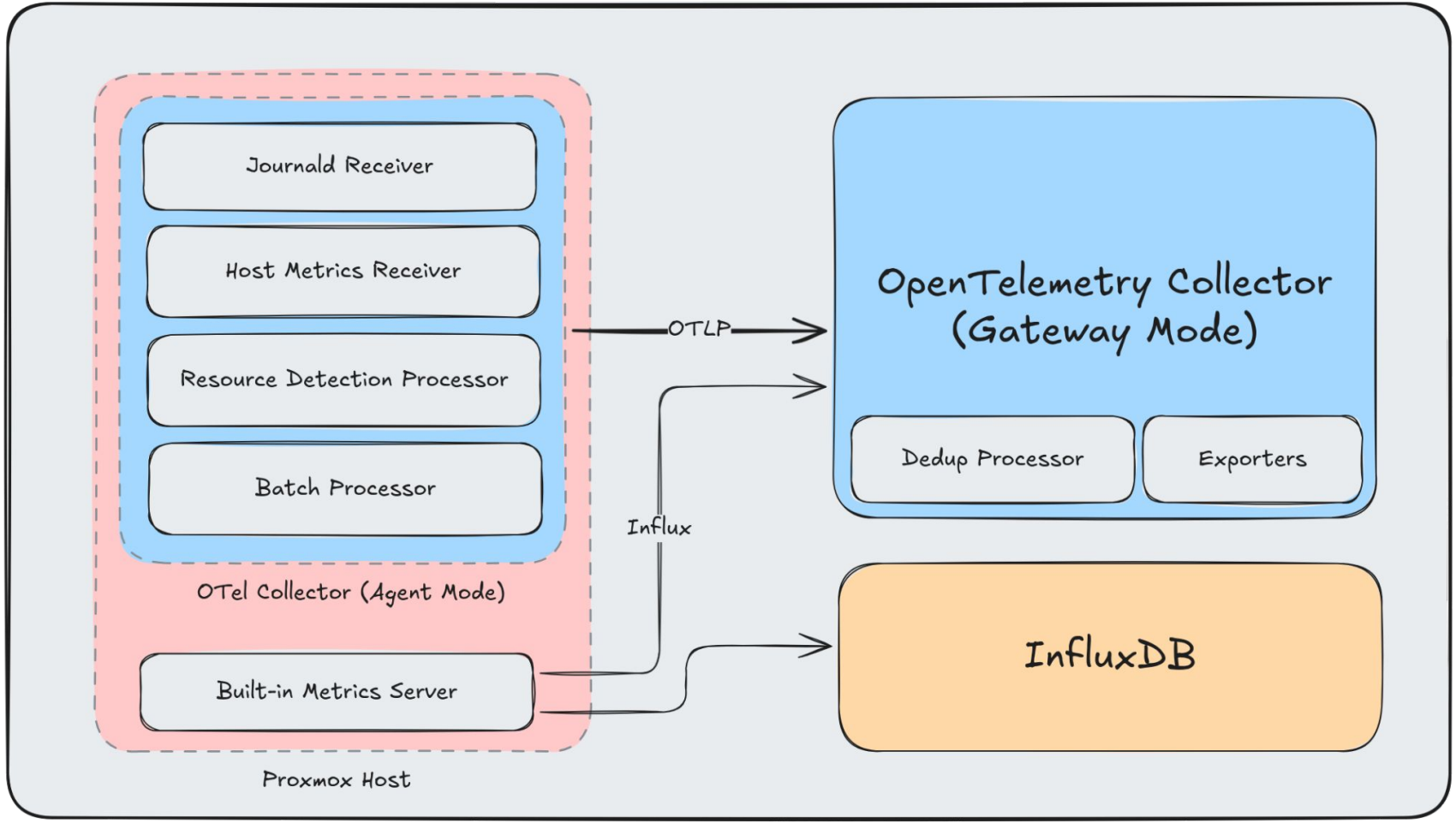

cAdvisor Exporter
Prometheus Node Exporter
Promtail Agent
VMs
Prometheus Node Exporter
Promtail Agent
Built-in Metrics Server
Proxmox Host
Loki
Prometheus
InfluxDB


cAdvisor Exporter
Prometheus Node Exporter
Promtail Agent
VMs
Prometheus Node Exporter
Promtail Agent
Built-in Metrics Server
Proxmox Host
Loki
OpenTelemetry
Prometheus
InfluxDB

Well that’s clean...

Well that’s clean...

Let’s do it the OTel way


Journald Receiver
Host Metrics Receiver
Resource Detection Processor
Batch Processor
OTel Collector (Agent Mode)
Built-in Metrics Server
Proxmox Host
OTLP
Influx
OpenTelemetry Collector
(Gateway Mode)
Dedup Processor
Exporters
InfluxDB

The OpenTelemetry Collector

Agent or Gateway

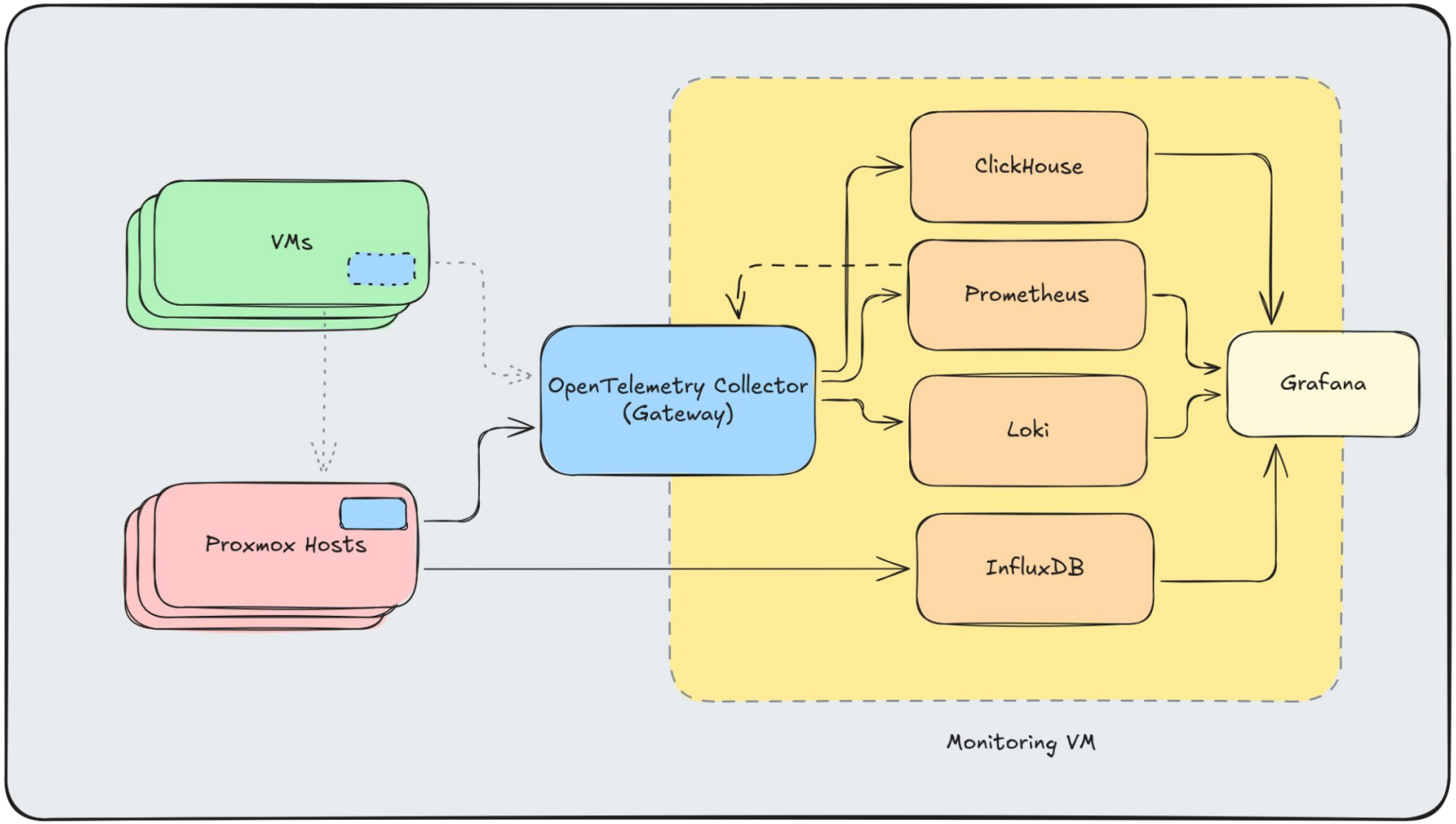
Container or Binary

Built-in Monitoring

Don’t use Contrib!

Collector Extensions

- OTLP Receiver
- Influx Receiver
- Journald Receiver
- Syslog Receiver
- Host Metrics Receiver
- Resource Detection Processor
- Batch Processor
- Log Dedup Processor
- OTLP Exporter
- ClickHouse Exporter
- Prometheus Exporter
- ...


VMs
Proxmox Hosts
OpenTelemetry Collector
(Gateway)
ClickHouse
Prometheus
Loki
InfluxDB
Grafana
Monitoring VM

Benefits of “Prometheus Native”

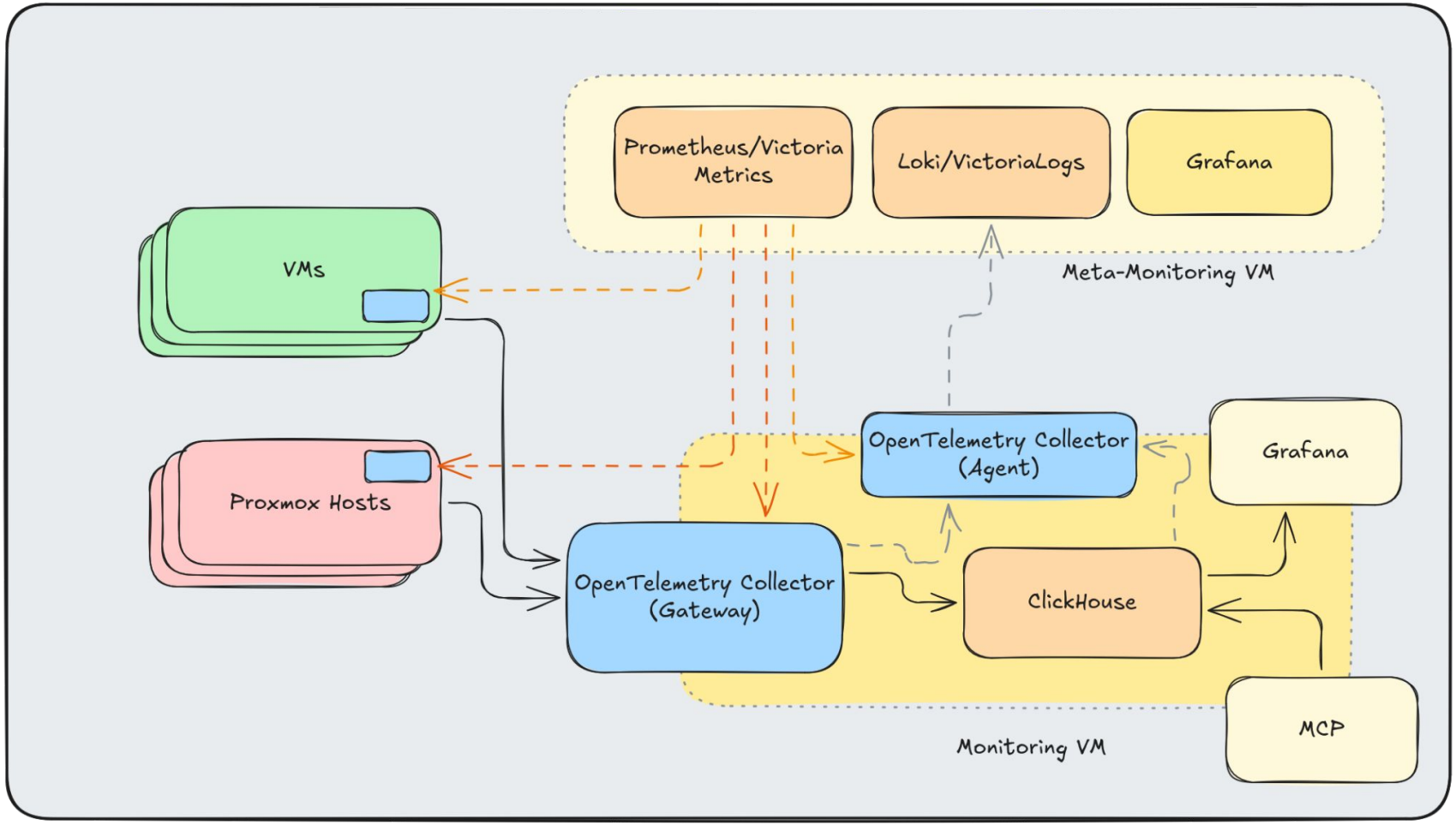
- Push or pull methodologies
- Huge library of off-the-shelf dashboards
- For other data sources (like Influx) as well…

But... MCP!

- Grafana MCP
- ClickHouse MCP
- OpenTelemetry MCP
- With Ansible inventory, NixOS Configurations, and an LLM = 😁😁😁

Future Plans

- OpenTelemetry Agents for VMs (container logs & metrics)
- High-Availability with Kubernetes
- Eliminate dual writes
- Meta-monitoring stack
- Application-specific metrics:
  - **Immich: no. of photos archived**
  - **Paperless: no. of documents stored**
  - **Proxmox Backup Server: successful backups in past 24 hrs**


Prometheus/Victoria Metrics
Loki/VictoriaLogs
Grafana
Meta-Monitoring VM
VMs
Proxmox Hosts
OpenTelemetry Collector (Agent)
Grafana
OpenTelemetry Collector (Gateway)
ClickHouse
MCP
Monitoring VM

Thank You!

🦋 @joshleecreates.bsky.social

🐘 @joshleecreates@hachyderm.io

🧳 linkedin.com/in/joshuamlee

💬 altinity.com/slack

Blog post & resources