



NETDATA

# Decentralized Monitoring, and why it matters

Shyam Sreevalsan



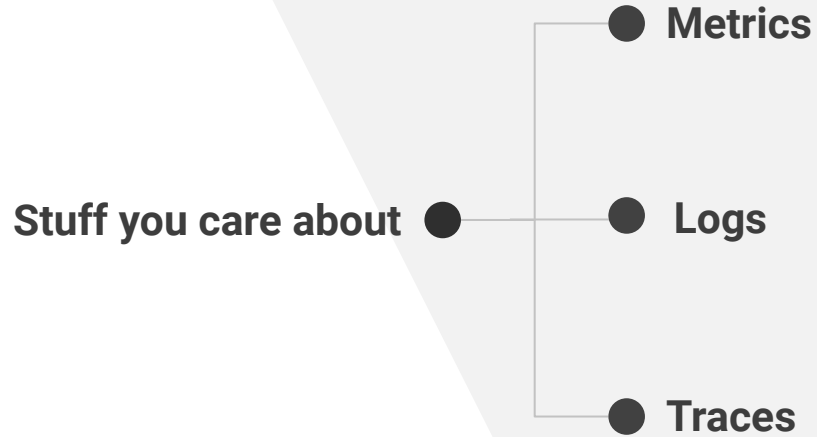


# Observability - in a nutshell

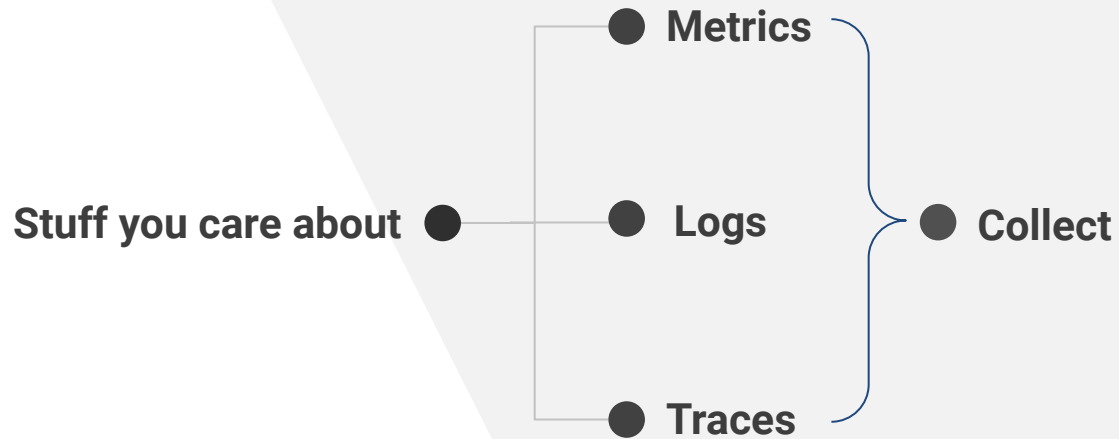
---

**Stuff you care about ●**

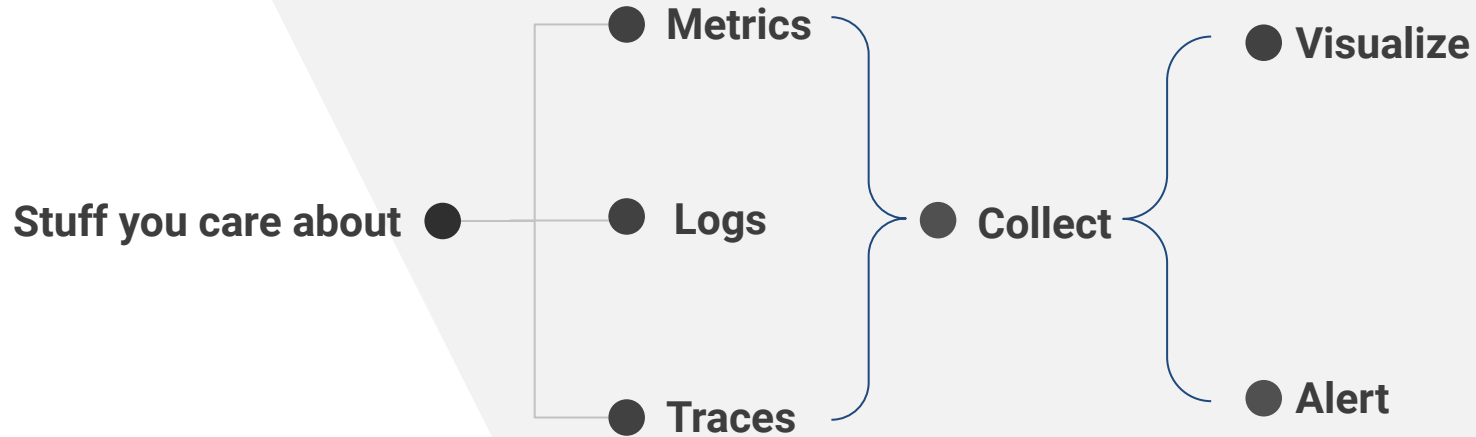
# Observability - in a nutshell



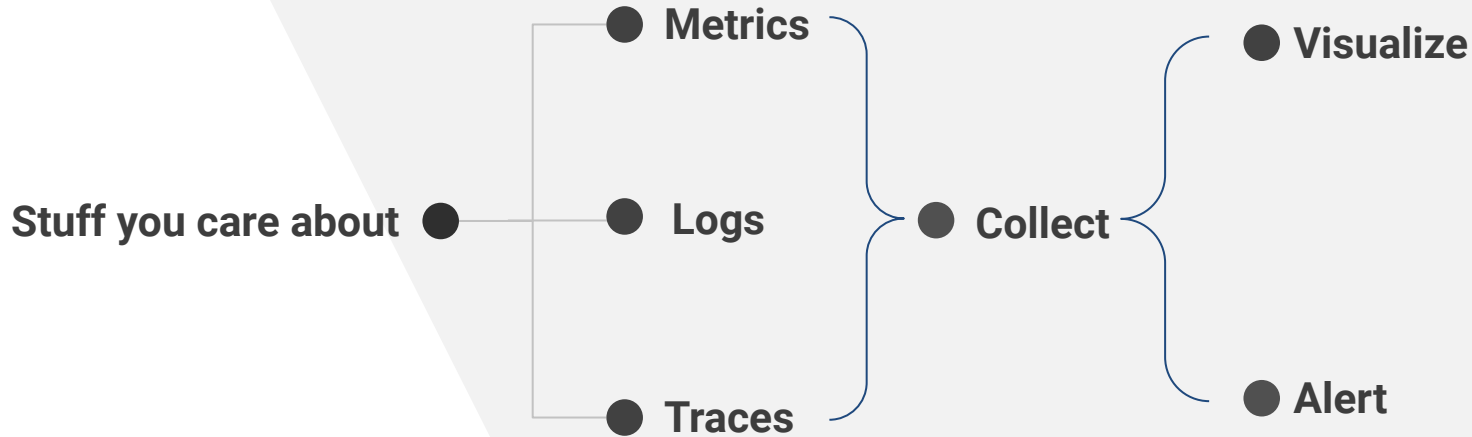
# Observability - in a nutshell



# Observability - in a nutshell



# Observability - in a nutshell



# Observability Landscape

It's crowded in here



- 1st Generation (Checks)
  - Nagios, Zabbix, PRTG, Icinga, CheckMK
- 2nd Generation (Metrics)
  - Prometheus, Graphite, InfluxDB, OpenTSDB, Cacti, Munin
- 3rd Generation (Logs)
  - ELK, Splunk
- 4th Generation (Integrated)
  - Datadog, Dynatrace, Instana, NewRelic, Grafana

# Centralized Monitoring

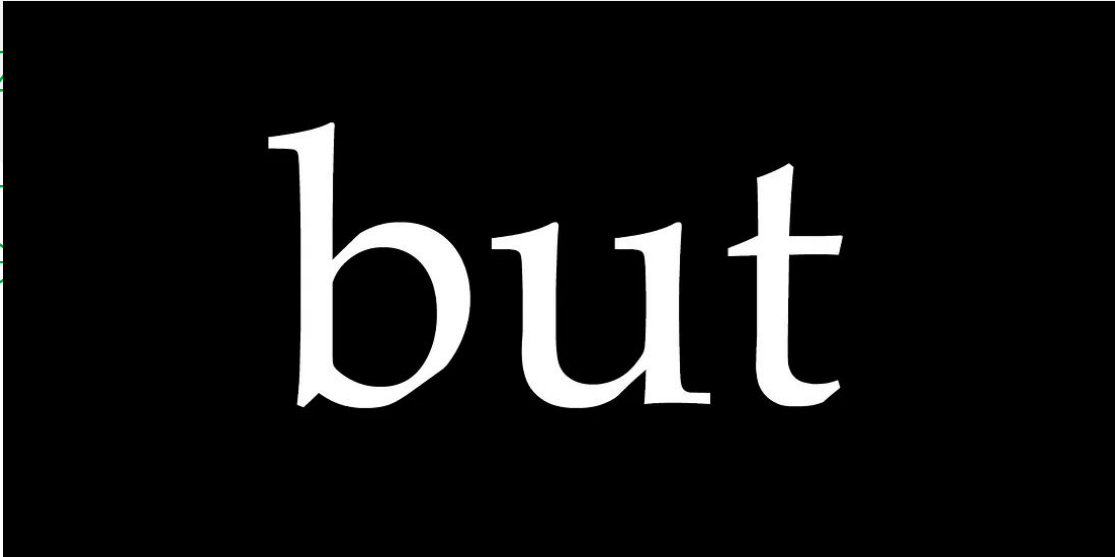
Centralized  
Observability  
is the default  
setting today

- Centralizes Metrics, Logs, Traces & Checks
- Comprehensive visibility
- Correlate trends across various data types
- Enables deep understanding of system behavior



# Centralized Monitoring

Centralized  
Observation  
is the de  
setting t



s & Checks

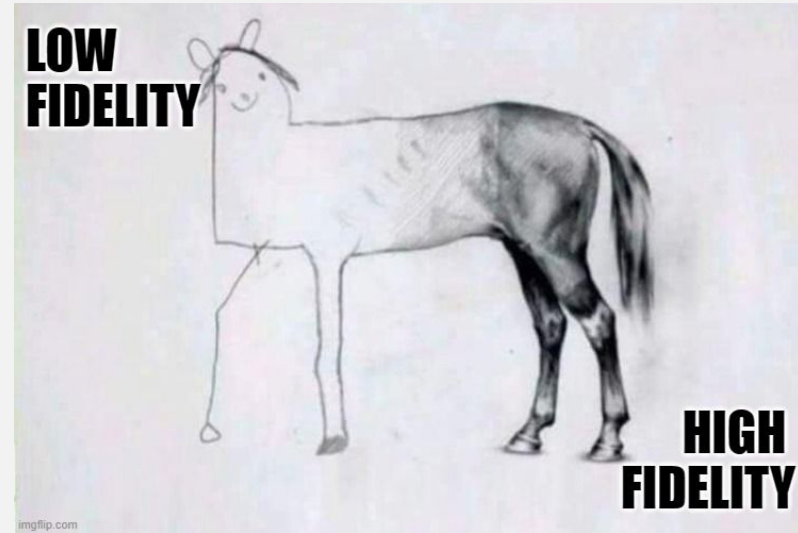
data types

system behavior

# The 7 deadly sins of Centralized Monitoring

# Fidelity

- Fidelity = Granularity + Cardinality
  - Low granularity = blurry data
  - Low cardinality = blind spots
  - Low granularity + low cardinality = abstract view lacking detail and coverage
- Centralization makes fidelity and cost, proportional to each other
  - Increasing fidelity results in higher costs
  - Reducing costs leads to a decrease in fidelity
  - Low fidelity by design



# Scalability

- Bottlenecks
- Capacity Limits
- Latency & Delays
- Complex load balancing



# Cost

- Centralized data storage
- Centralized compute
- High Data Egress
- Scaling costs grow disproportionately
- Result == teams cherry picking what to observe == bad move

## Observability is Too Damn Expensive

Why Legacy Observability Tools are So \$!&%# Expensive

## Datadog's \$65M Bill and Why Developers Should Care

## Escaping the Cost/Visibility Tradeoff in Observability Platforms

**Hacker News** new | past | comments | ask | show | jobs | submit

▲ ownagefool 10 months ago | parent | context | favorite | on: Ask HN: How do you monitor your sy  
**Alternative view point.**  
 Observability is **hella expensive**. Orgs should consider TCO when making such decisions  
 tens of millions off vendor bills.

← **r/sre** • 8 mo. ago  
 serverlessmom

**Is a \$1 million Observability bill worth it? Why are we willing to pay so much for observability?**

← **r/devops** • 1 yr. ago  
 Lost\_Understanding12

### Taming the cost of observability

My organization is currently using grafana and elastic and our observability spend is not scaling with the size of our application and infrastructure. I am guessing we are not unique when it comes to not being able to justify the ROI on observability as we scale.

# Accuracy

- Reduced Granularity
- Data Inconsistency
- Delayed Detection
- Missed alerts
- ... outages, downtime, pain



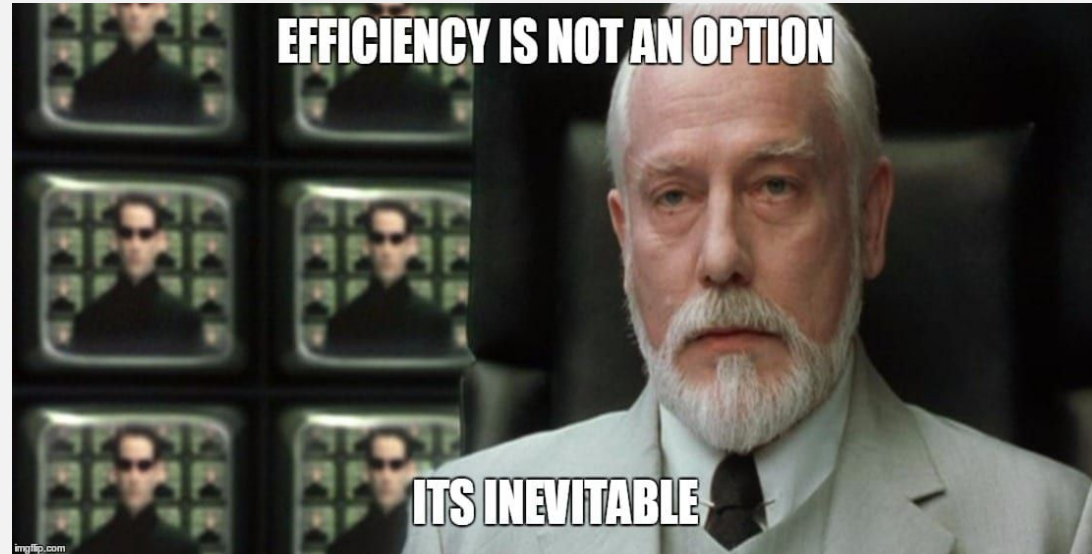
# Resilience

- Single point of failure
- Cascading failures
- Disaster Impact
- Recovery Time



# Efficiency

- Processing delays
- Inefficient data handling
- Energy consumption
- Resource overload





# Data Privacy

- Concentration of Risk
- Compliance Challenges
- Limited deployment options
- A question of trust

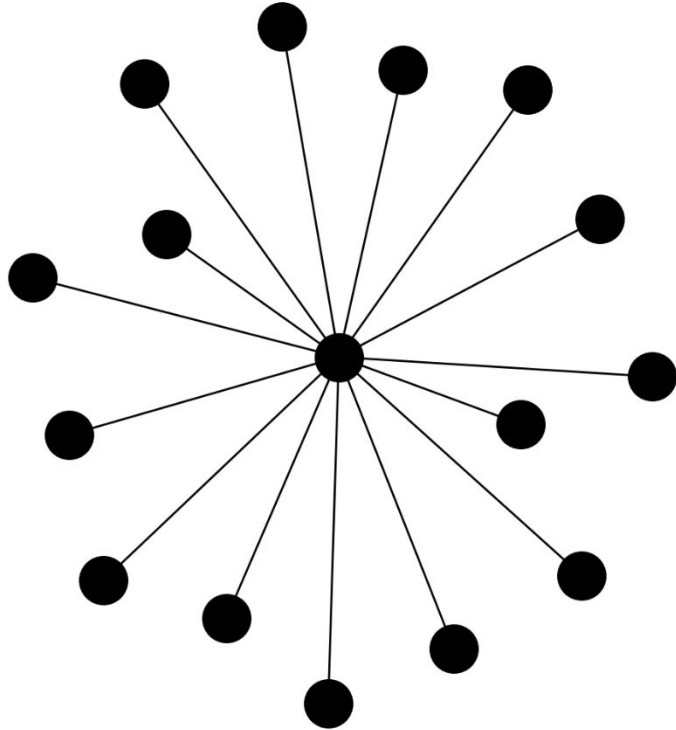


Solution = Decentralize!

# Centralized vs Decentralized

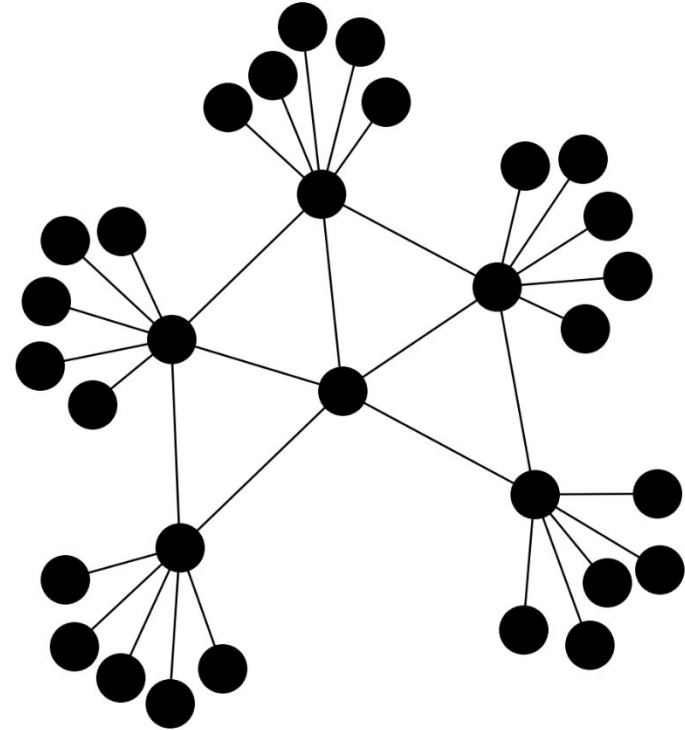
## Centralized Network

All the nodes are connected under a single authority



## Decentralized Network

No single authority server controls the nodes, they all have individual entity

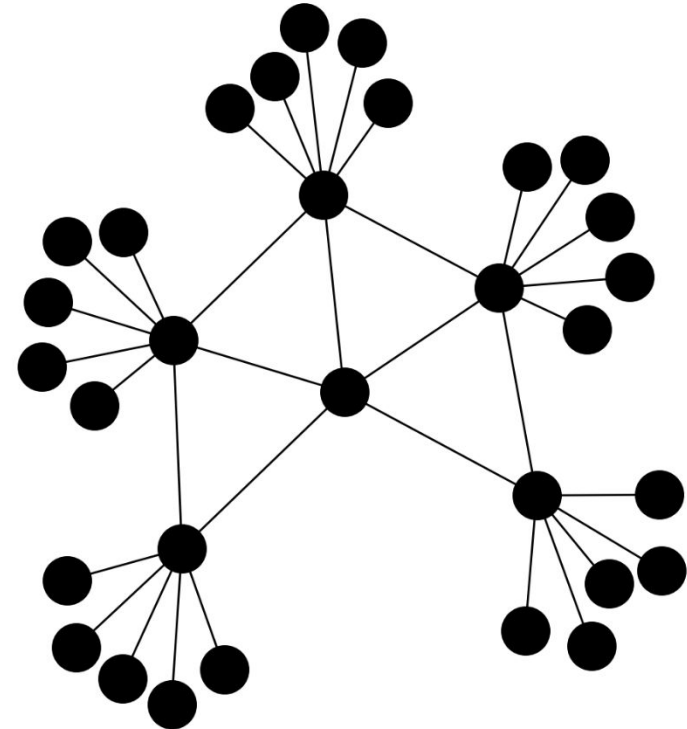


# Centralized vs Decentralized

1. FIDELITY
2. SCALABILITY
3. COST
4. ACCURACY
5. RESILIENCE
6. EFFICIENCY
7. DATA PRIVACY

## Decentralized Network

No single authority server controls the nodes, they all have individual entity

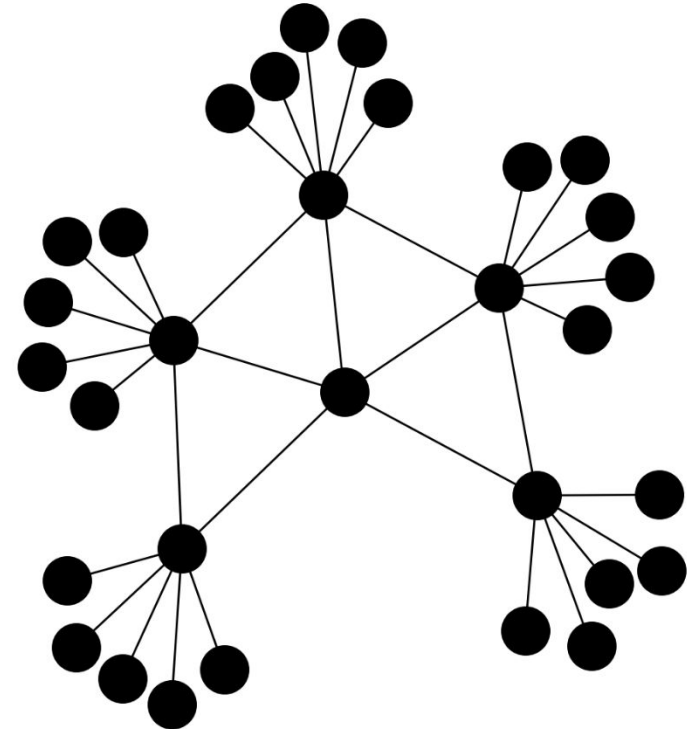


# Centralized vs Decentralized

1. FIDELITY
2. SCALABILITY
3. COST
4. ACCURACY
5. RESILIENCE
6. EFFICIENCY
7. DATA PRIVACY

## Decentralized Network

No single authority server controls the nodes, they all have individual entity



# Decentralized Design For High Fidelity

- **Keep data at the edge**

- Compute & storage resources are already available and spare
- No need for network resources
- The work to be done is small and it can be optimized, so that monitoring is a “polite citizen” to production applications

- **Make the data highly available, across the decentralized network**

- Ephemeral nodes, that may vanish at any point in time
- High availability of observability data
- Offloading “sensitive” production systems from observability work

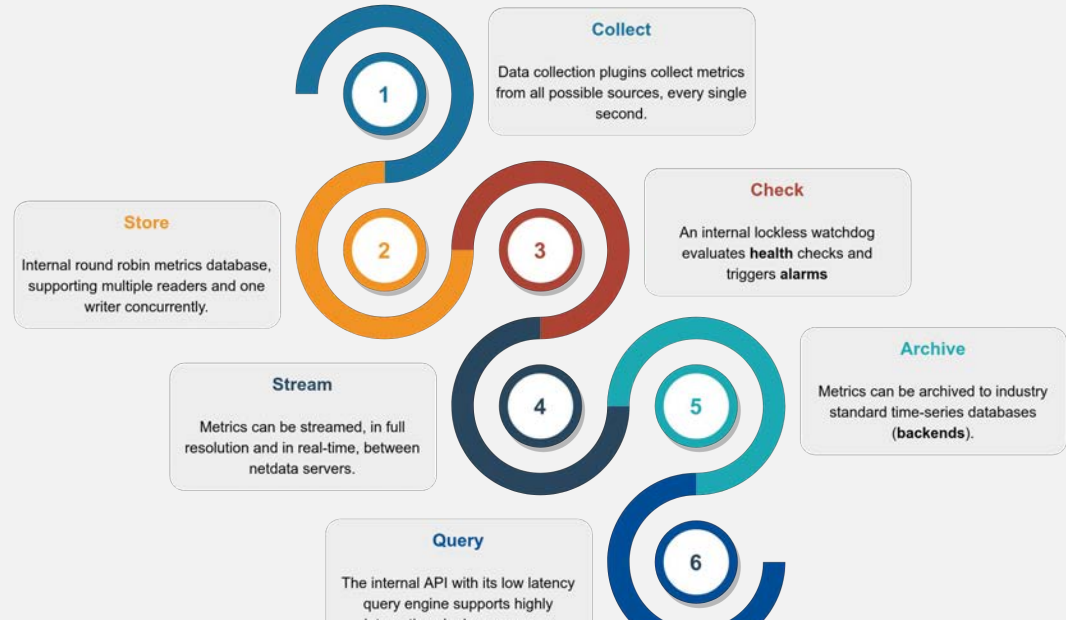
- **Unify and integrate everything at query time**

To provide unified infrastructure-wide views, query edge systems (or the mini centralization points), aggregate their responses and provide high-resolution, real-time dashboards and alerts.

# The Netdata Way

# Netdata “Agent”

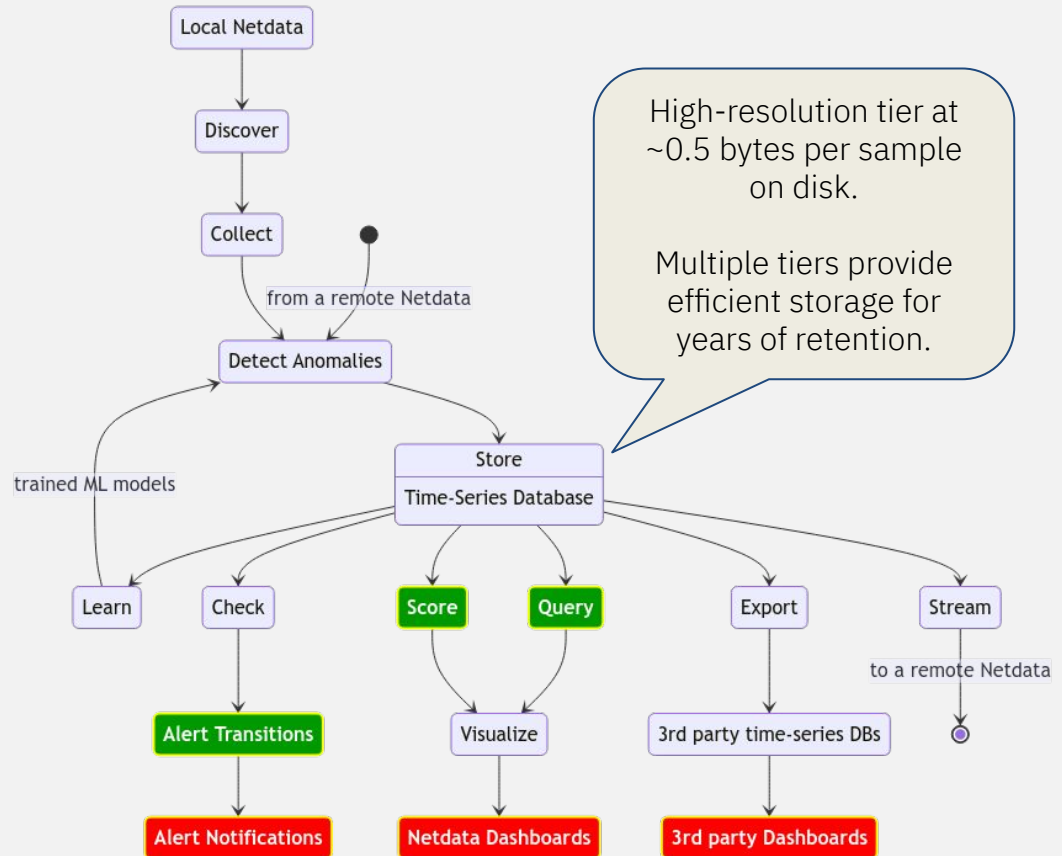
- Open Source
- Real Time Monitoring
- Discover -> Collect -> Store
- Metrics & Logs
- Alerts & notifications
- Anomaly detection & ML at the edge
- Stream data to other agents
- Query any agent from cloud





# Distributed Metrics Pipeline

The Netdata  
Metrics Pipeline  
is like lego  
building blocks



## Agents can be lightweight *and* capable

Resource	Dynatrace	Datadog	Instana	Grafana	Netdata
CPU Usage (100% = 1 core)	<b>12%</b>	<b>14%</b>	<b>6.7%</b>	<b>3.3%</b>	<b>3.6%</b>
Memory Usage	<b>1400 MB</b>	<b>972 MB</b>	<b>588 MB</b>	<b>414 MB</b>	<b>181 MB</b>
Disk Space	<b>2 GB</b>	<b>1.2 GB</b>	<b>0.2 GB</b>	-	<b>3 GB</b>
Disk Read Rate	-	<b>0.2 KB/s</b>	-	-	<b>0.3 KB/s</b>
Disk Write Rate	<b>38.6 KB/s</b>	<b>8.3 KB/s</b>	-	<b>1.6 KB/s</b>	<b>4.8 KB/s</b>
Egress Internet Bandwidth	<b>11.4 GB/mo</b>	<b>11.1 GB/mo</b>	<b>5.4 GB/mo</b>	<b>4.8 GB/mo</b>	<b>0.01 GB/mo</b>

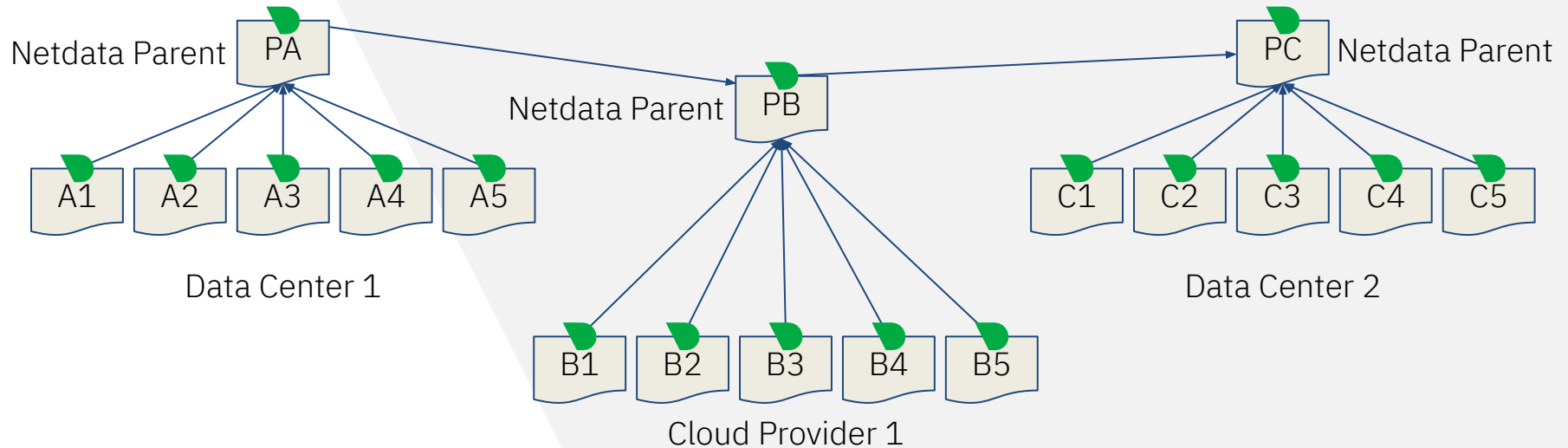
### What you get by just installing Netdata on an empty VM

[Full analysis here.](#)

- 150+ dashboard charts, 2k+ unique time-series
- 50+ unique pre-configured alerts, Anomaly detection for every metric
- 2 weeks of per-sec, 3 months of per-min, 2 years of per-hour data using just 3GiB of disk space

# Netdata “Parents”

- **Enhanced Scalability and Flexibility**
- **Resilience and Fault Tolerance**
- **Optimized Cost and Performance**
- **Always On-Prem**
- **Bottom-Up Observability**
- **Production Systems Isolation**



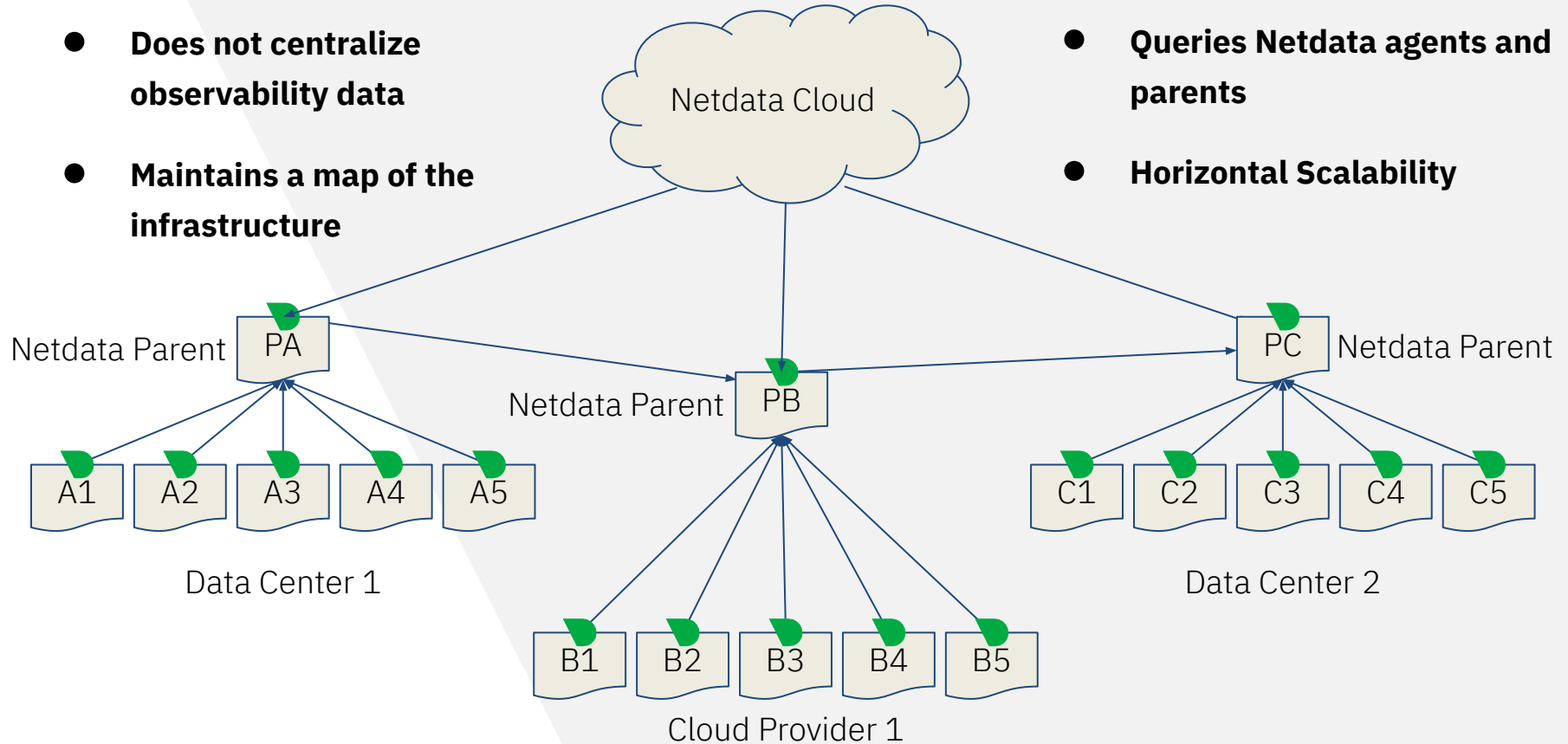
# Netdata “Cloud”

- **Does not centralize observability data**

- **Maintains a map of the infrastructure**

- **Queries Netdata agents and parents**

- **Horizontal Scalability**



# Common Concerns about Decentralized Designs

- **The agent will be heavy**

No! The Netdata agent processes thousands of metrics per second, and is one of the lightest observability agents available.

- **Querying will increase load on production systems**

No! Each agent serves only its own data. Querying such a small dataset is lightweight and does not influence operations. For very sensitive or weak production systems, a mini-centralization point next to these systems will isolate them from queries (and also offload them from ingestion, processing, storage and retention).

- **Queries will be slower**

No! They are actually faster! Distributing tiny queries in parallel to multiple systems, provides an aggregate compute power that is many times higher to what any single system can provide.

- **Will require more bandwidth**

No! Querying is selective, most of the observability data are never queried unless required for exploration or troubleshooting. And even then, just a small portion of the data is examined.

So, the overall bandwidth used is a tiny fraction compared to centralized systems.

# Time for a quick demo!



The (long and winding)  
road ahead

## What's the catch?

Where are all  
the other  
Decentralized  
Observability  
platforms?

- Developing a decentralized observability platform - is NOT easy
  - Resource consumption at the edge has to be minimal
  - Complex queries and aggregation must be handled behind the scenes
  - Keep deployment simple!
  - Learn to relinquish control





# The future is decentralized

---

Hard problems  
**CAN**  
be solved

- Do NOT compromise on fidelity
- Demand more and demand better from your observability provider
- If you operate a DIY monitoring stack, apply decentralized principles for long term benefits
- Why centralize observability in distributed, multi-cloud, auto-scaling environments?



NETDATA

Making monitoring easy  
for everyone

[www.netdata.cloud](http://www.netdata.cloud)  
[github.com/netdata/netdata](https://github.com/netdata/netdata)

[shyam@netdata.cloud](mailto:shyam@netdata.cloud)  
[linkedin.com/in/shyamvalsan](https://linkedin.com/in/shyamvalsan)