Container Observability and Monitoring

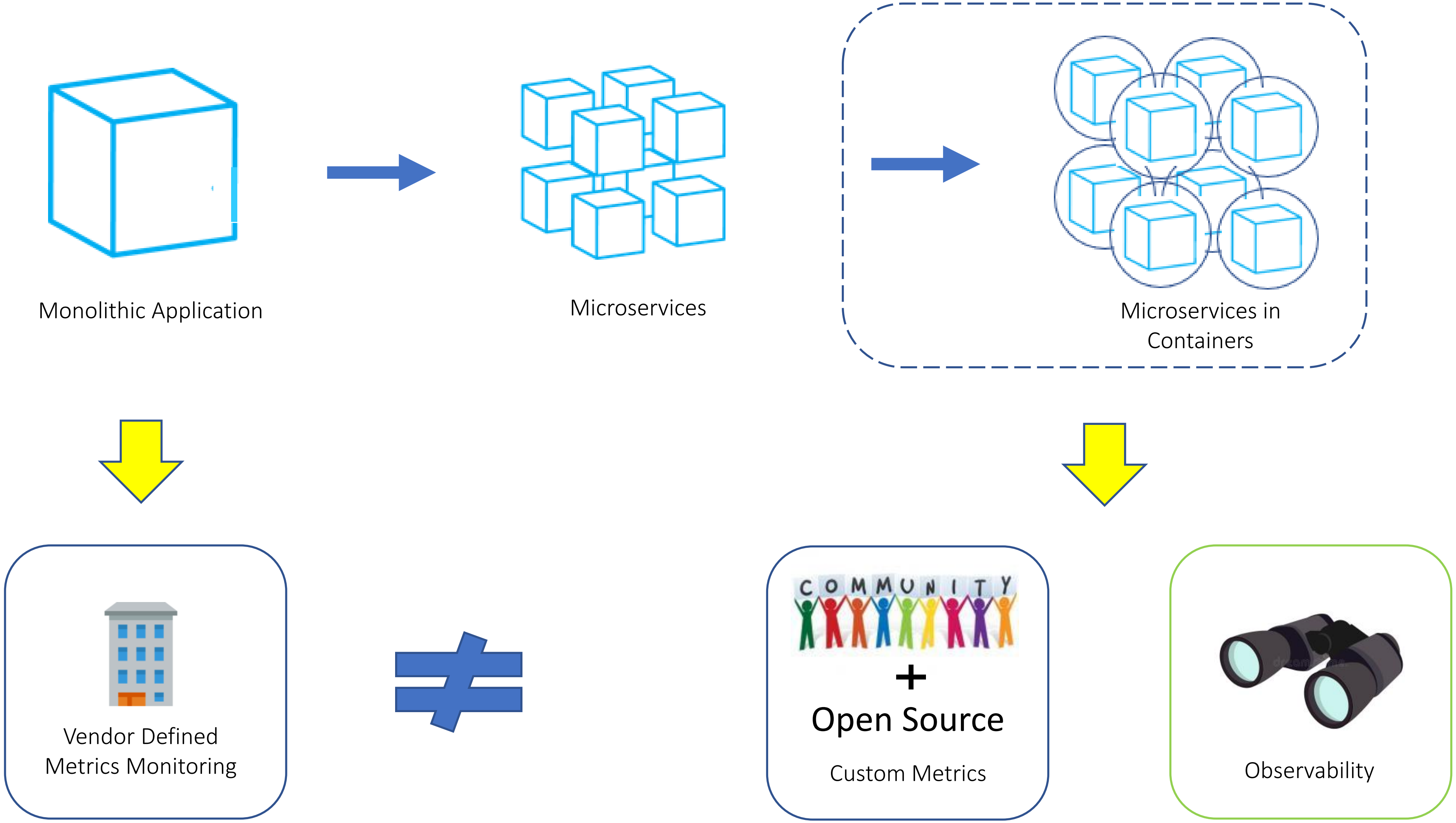
Keiko Harada
Microsoft
Senior Program Manager – Azure Compute - Application Platform

Bio:
Lehman Brothers (Japan) - IT
Goldman Sachs and Co (New York) – Senior Storage Engineer
Hitachi Data Systems (New York) – Product Strategist and Senior Product Manager
Microsoft – Senior Program Manager – Azure Container Compute

Agenda

- Shift in Monitoring with containerized microservice
- Observability and Monitoring
- How does Monitoring play in Enterprise?
- Observability/Monitoring tools

Monitoring Paradigm Shift with Microservice and Containers



Monolithic Application

Microservices

Microservices in Containers

Vendor Defined Metrics Monitoring

≠

+ Open Source

Custom Metrics

Observability

What is Observability?

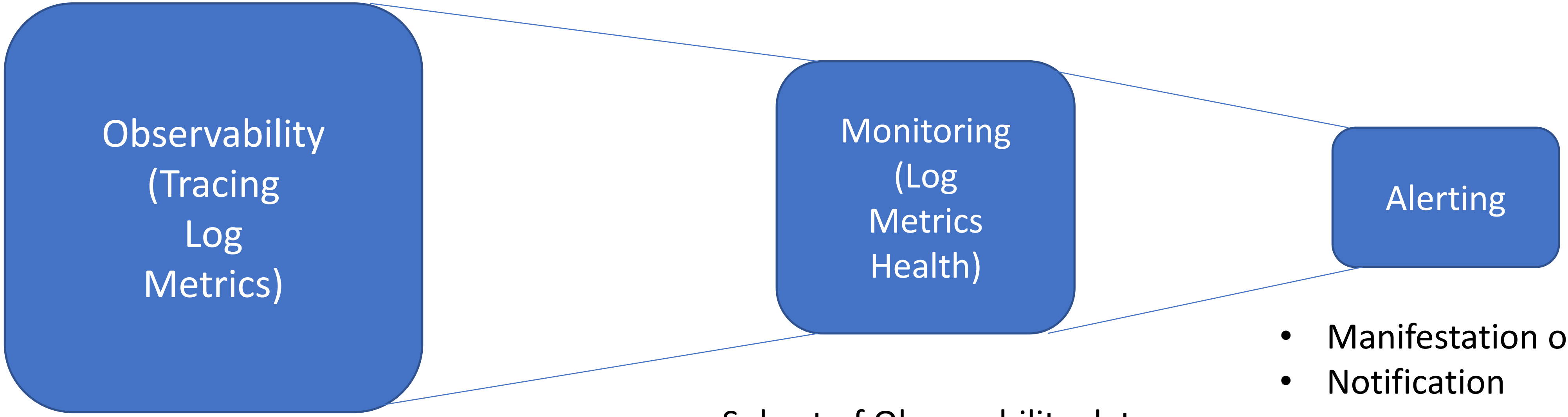
Observability:

A measure of how well internal states of a system can be inferred from knowledge of its external outputs. It is more a property of a system.

Monitoring:

Monitoring is something we perform against our applications and systems to determine their state.

Observability



Observability
(Tracing
Log
Metrics)

Monitoring
(Log
Metrics
Health)

Alerting

- Manifestation of the issue
- Notification

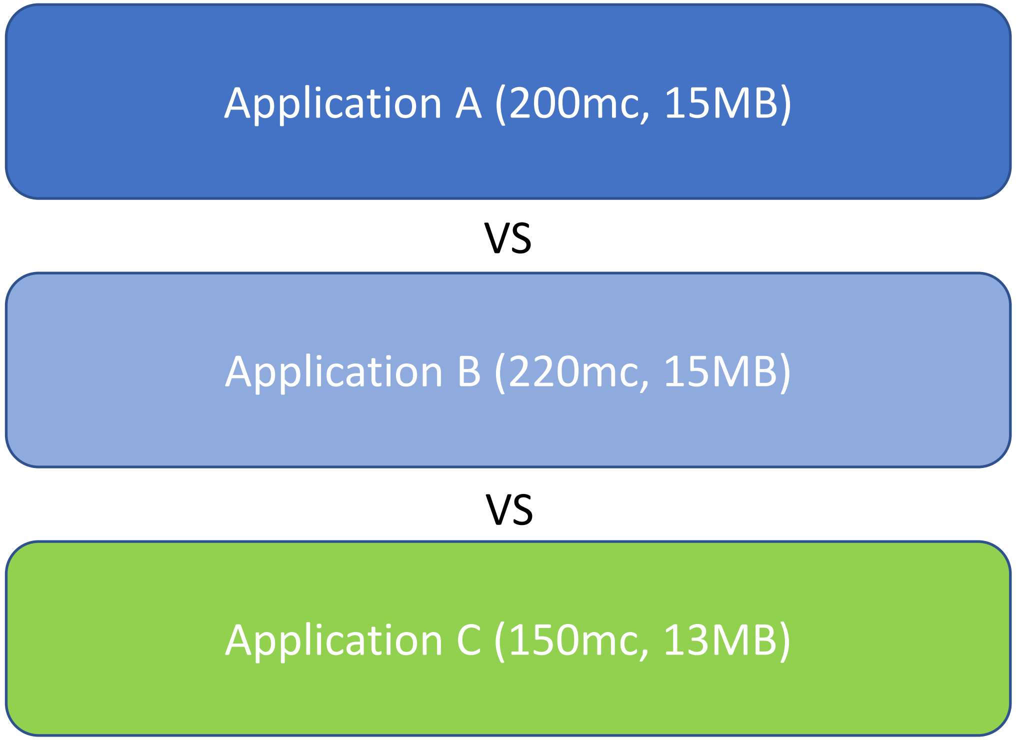
- Subset of Observability data
- Capture essence for troubleshooting
- Present what can be the issues or more insights.

- Human interaction and intuition
- There is no known issue surfaces.
- It is to just answer “any” questions regardless of issues.

Observability Example) Application Digital Footprints

Compare Digital footprint

(CPU/Memory Consumption …)



Application A (200mc, 15MB)

VS

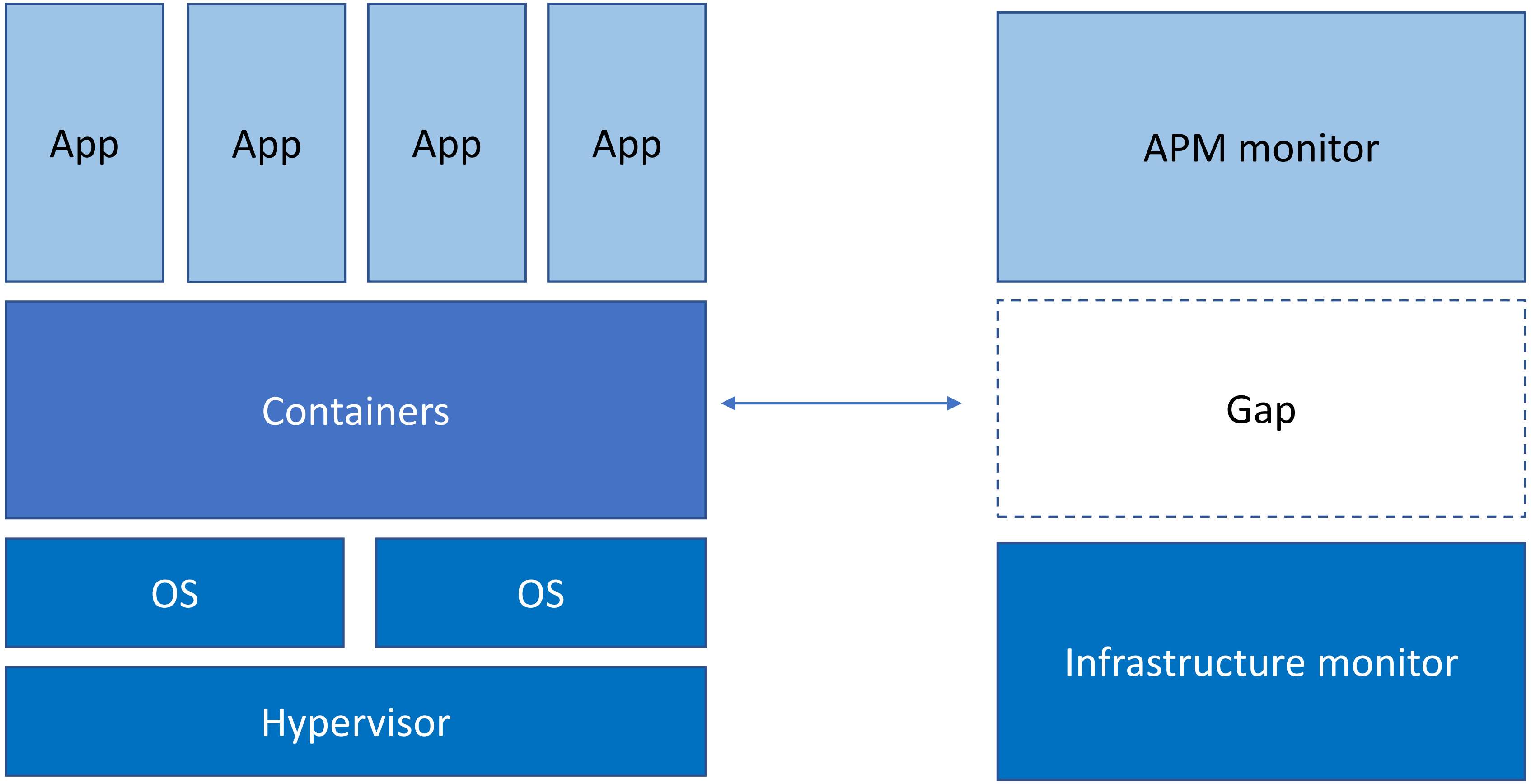
Application B (220mc, 15MB)

VS

Application C (150mc, 13MB)

How are people using “Observability and Container Monitoring” in Enterprise?

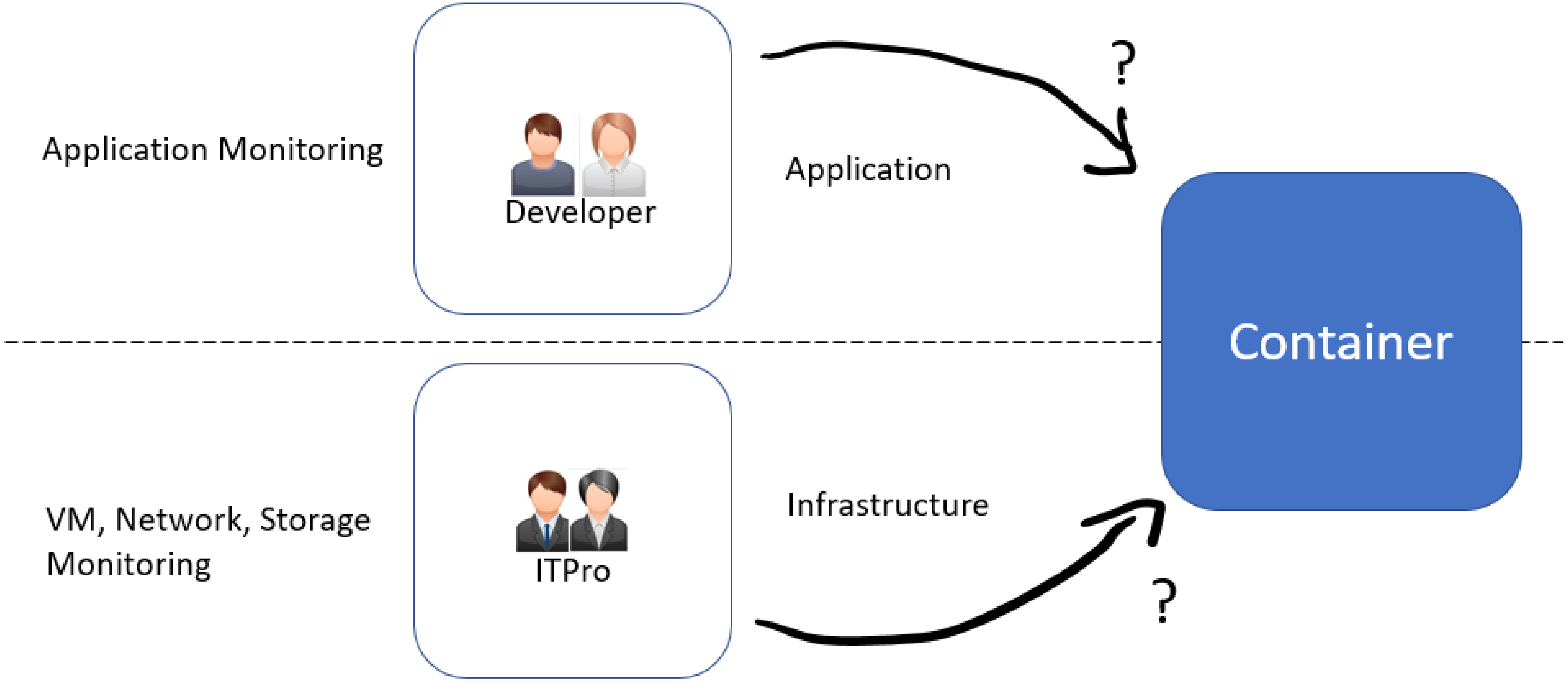
Gap in Stack



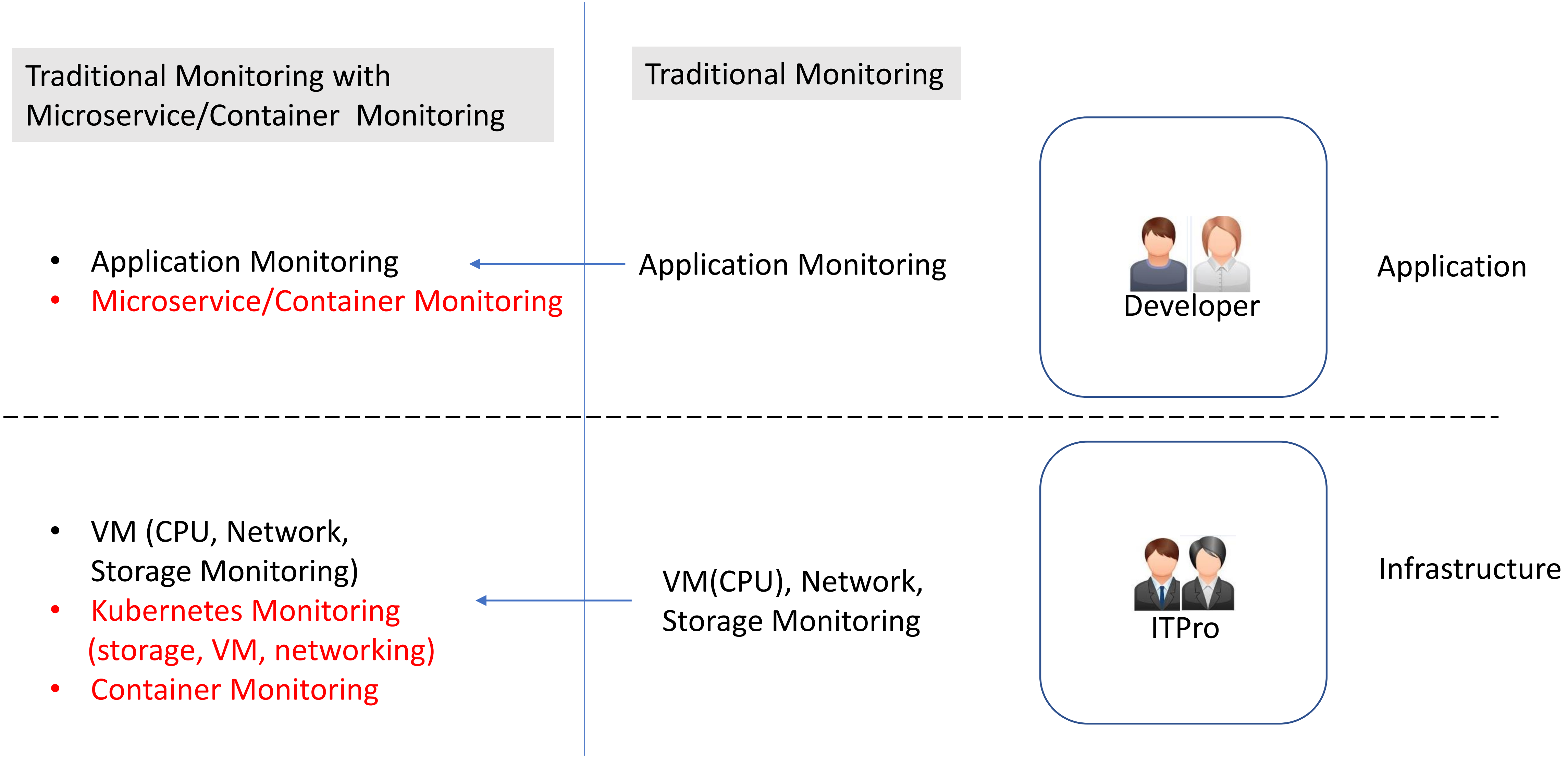
identify and pinpoint
bottlenecks or errors
thru APM

CPU, memory, network,
storage
perf and logs

Who would monitor containers?



Container bringing changes to IT Organization



Traditional Monitoring with Microservice/Container Monitoring

Traditional Monitoring

- Application Monitoring
- Microservice/Container Monitoring

Application Monitoring

Developer

Application

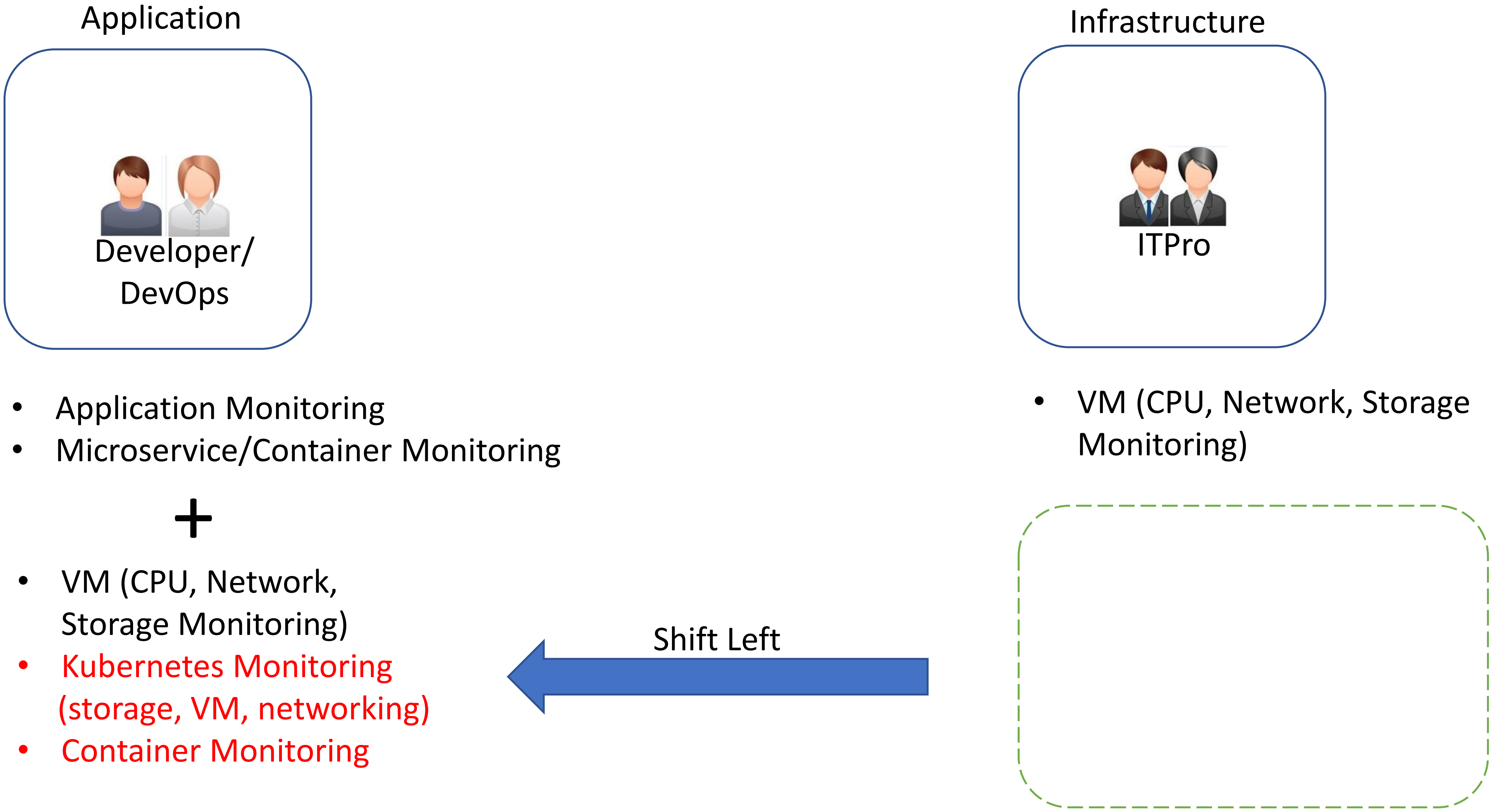
- VM (CPU, Network, Storage Monitoring)
- Kubernetes Monitoring (storage, VM, networking)
- Container Monitoring

VM(CPU), Network, Storage Monitoring

ITPro

Infrastructure

Container bringing changes to IT Organization



Application

Developer/
DevOps

- Application Monitoring
- Microservice/Container Monitoring

+

- VM (CPU, Network, Storage Monitoring)
- Kubernetes Monitoring (storage, VM, networking)
- Container Monitoring

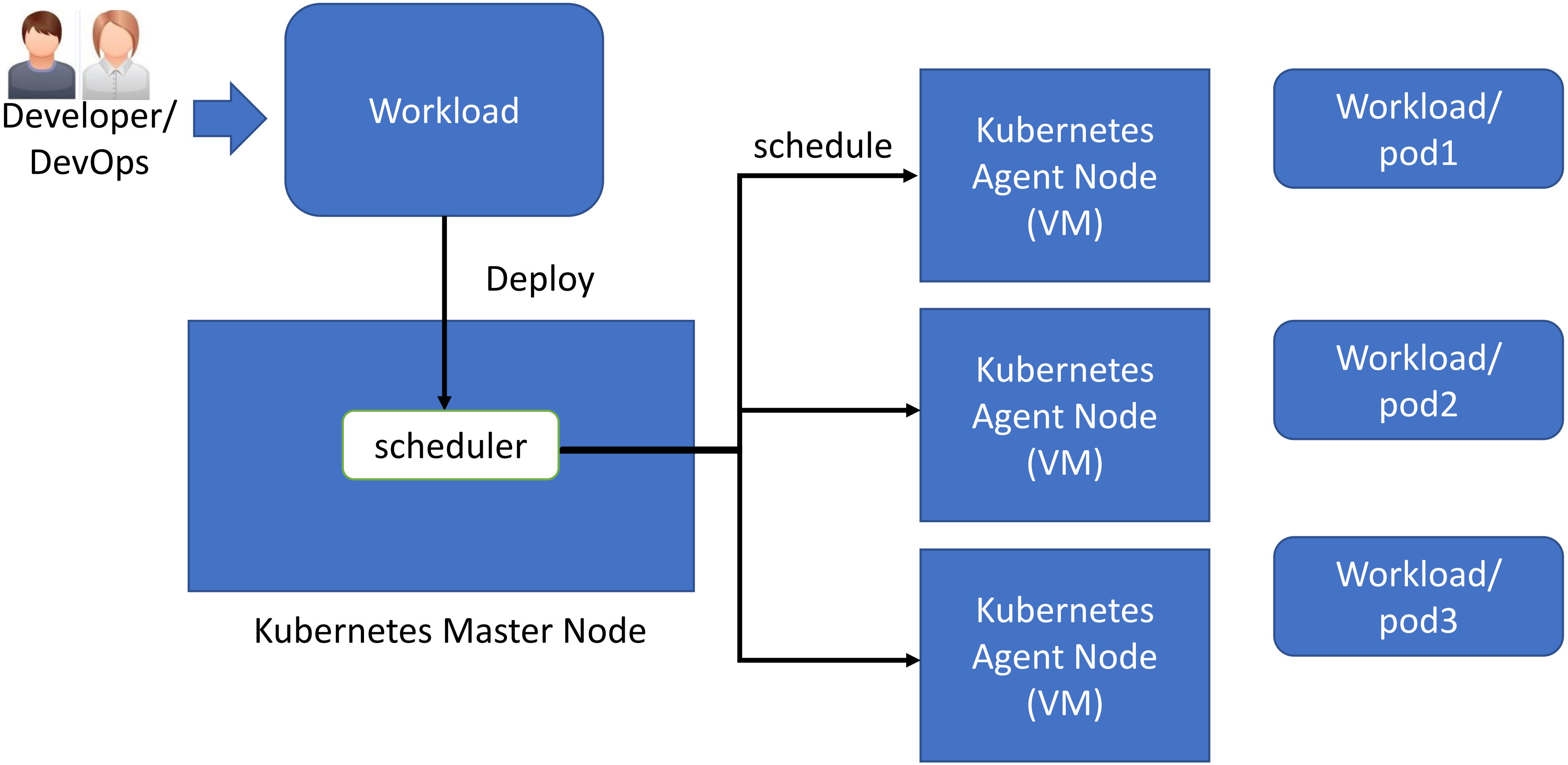
Shift Left

Infrastructure

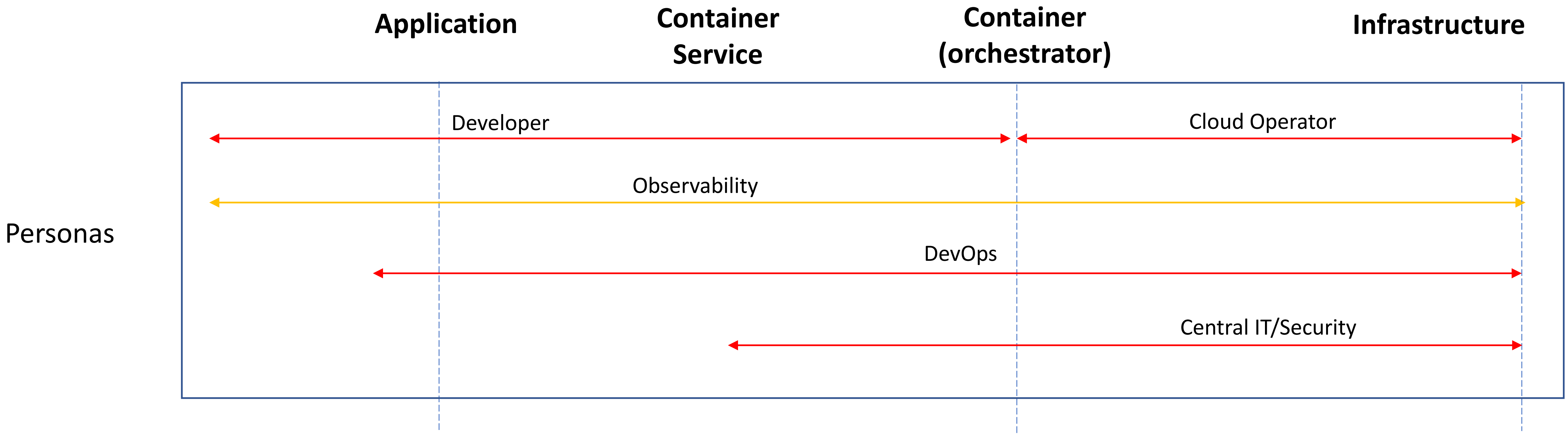
ITPro

- VM (CPU, Network, Storage Monitoring)

Troubleshooting Example



Personas and Stacks



Application

Container Service

Container (orchestrator)

Infrastructure

Personas

Developer

Cloud Operator

Observability

DevOps

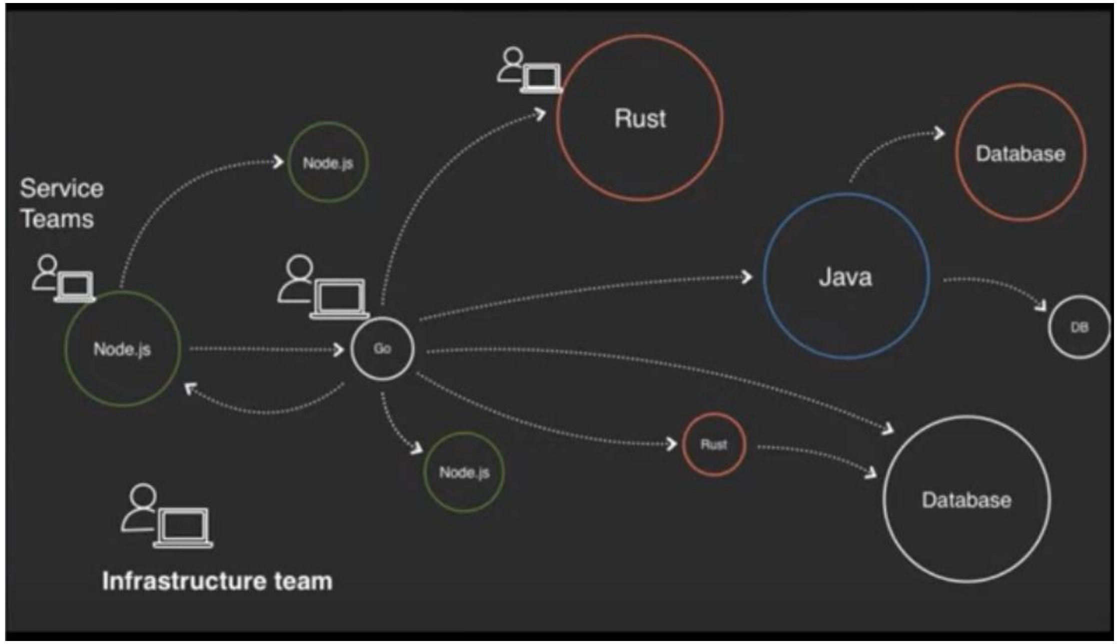
Central IT/Security

Monitoring per Kubernetes stack

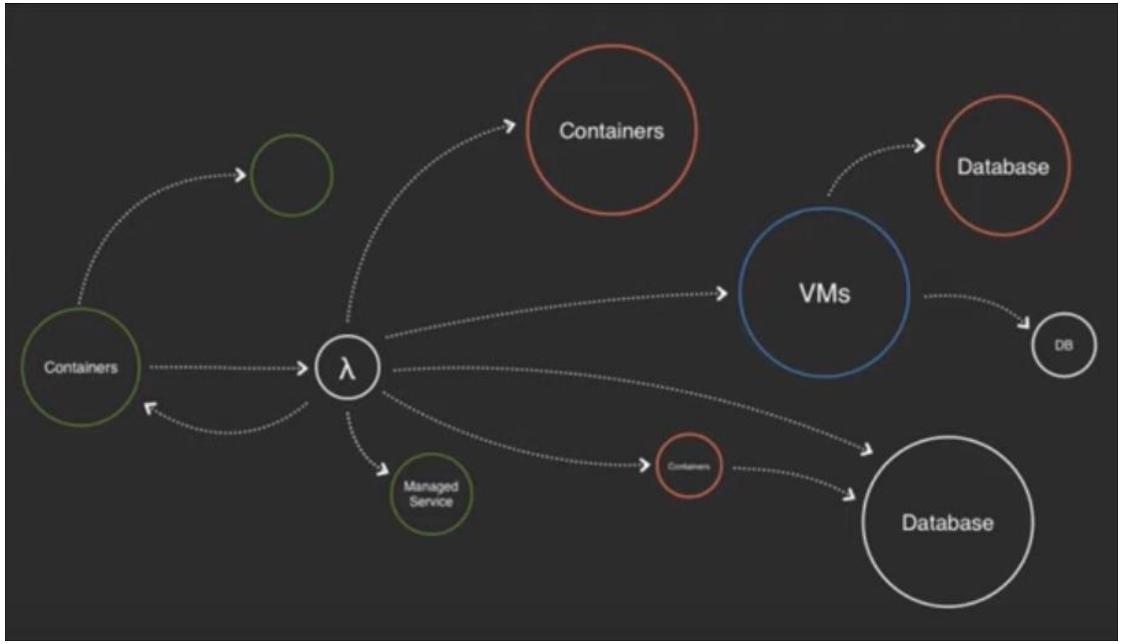
Stack	Metrics, Logs, Health, Inventory	Tools
Workload	Distributed Tracing with APM, Ingress controller (req/sec, failure rate, response time) Liveness, Readiness Probe Container logs (Stdout, stderr) Container metrics (cpu/memory/iops) Pod/container health (restart time, uptime..)	ebpf embedded tcpdump, lsof, strace sysdig fluentd
Orchestrator	Event log PV, Service Mesh/CNI/Ingress Controller, Node Conditions Pod health (pending pods), kubelet health	kubectl tcpdump
Containers	Container runtime health	docker, ps
Infrastructure (Platform)	Journalctl, systemctl, syslog Host performance/health/capacity networking latency, storage iops/bandwidth and capacity	top, sar, iostat

Development Experience

Application scope is beyond a platform



Services in multiple language



Various Service Deployments

Monitoring Vendors

APM/Tracing

- Dynatrace
- AppDynamics
- NewRelic
- Jaegar
- Zipkin
- Application Insights
- Sysdig
- Datadog
- Weavesworks

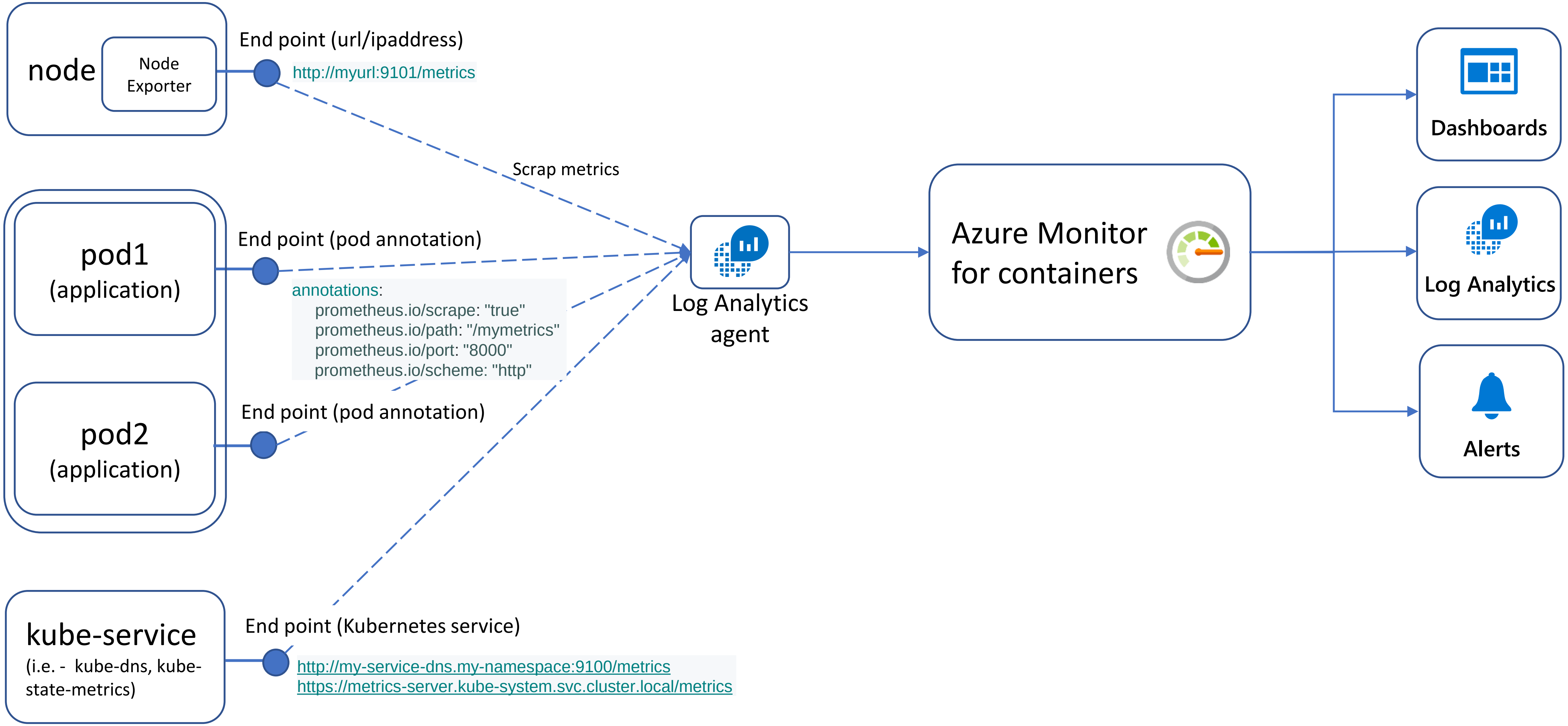
Kubernetes Infrastructure

- Dynatrace
- AppDynamics
- New Relic
- Container Insights
- Prometheus/Grafana
- InfluxDB
- Sysdig
- Datadog
- Weavesworks
- Tanzu

Observability

- Honeycomb
- NewRelic
- Prometheus/Grafana
- Container Insights
- Sysdig
- Datadog
- Tanzu

Container Insights with Prometheus Integration



node

Node Exporter

End point (url/ipaddress)

http://myurl:9101/metrics

Scrap metrics

pod1 (application)

End point (pod annotation)

```
annotations:
    prometheus.io/scrape: "true"
    prometheus.io/path: "/mymetrics"
    prometheus.io/port: "8000"
    prometheus.io/scheme: "http"
```

pod2 (application)

End point (pod annotation)

kube-service (i.e. - kube-dns, kube-state-metrics)

End point (Kubernetes service)

http://my-service-dns.my-namespace:9100/metrics
https://metrics-server.kube-system.svc.cluster.local/metrics

Log Analytics agent

Azure Monitor for containers

Dashboards

Log Analytics

Alerts

Microservice is Distributed System

- Reliability and Availability
- Performance and Capacity
- Application Model and Application Lifecycle

What to see in monitoring/observability tools?

- Application Lifecycle with platform
- Cohesiveness among platform services
- Correlation among platform to application
- Metric Cardinality
- Aggregated Insights and Interpretation (application model)
- Serverless vs Managed Kubernetes

Thank you