

Logging—a Key Element in Modern Software Systems

Peter Mechlenborg
Developer at humio.com
Email: peter@humio.com
<https://humio.com>



Why this webinar



Agenda

Observability

Architecture and tools

Making logs good

Demo

Tips & Tricks

Observability



Why observable systems

Discover
problems

Understand the user

Forecasting

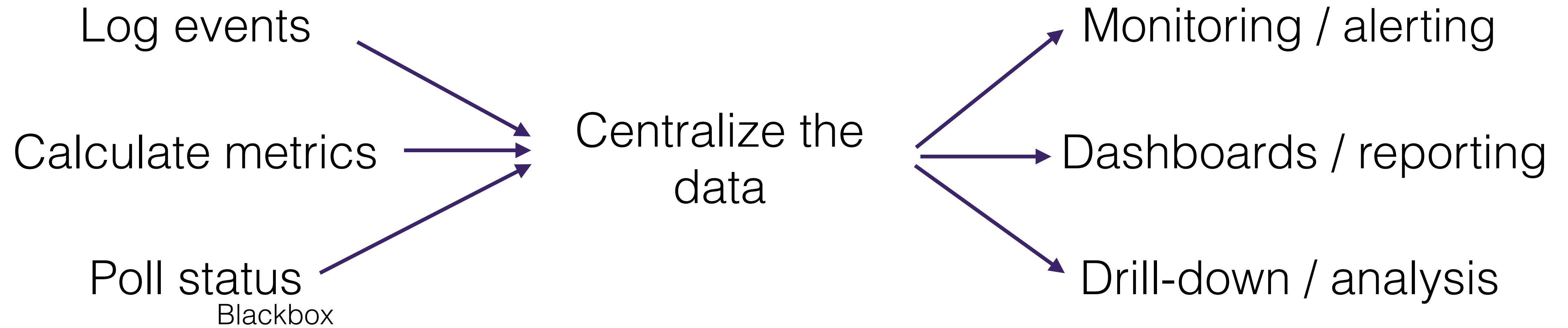
Fix problems

Understand the system

Reporting

Audit

Make systems observable



Logs

the logs-metrics spectrum

Metrics

Logs are **events** – metrics are **aggregates** of events

Logs have **high dimensionality** – metrics have **low dimensionality**

Logs tend to be **unstructured** – metrics are **structured**

Logs support **drill-down and analysis** – metrics lean towards **dashboards and alerting**

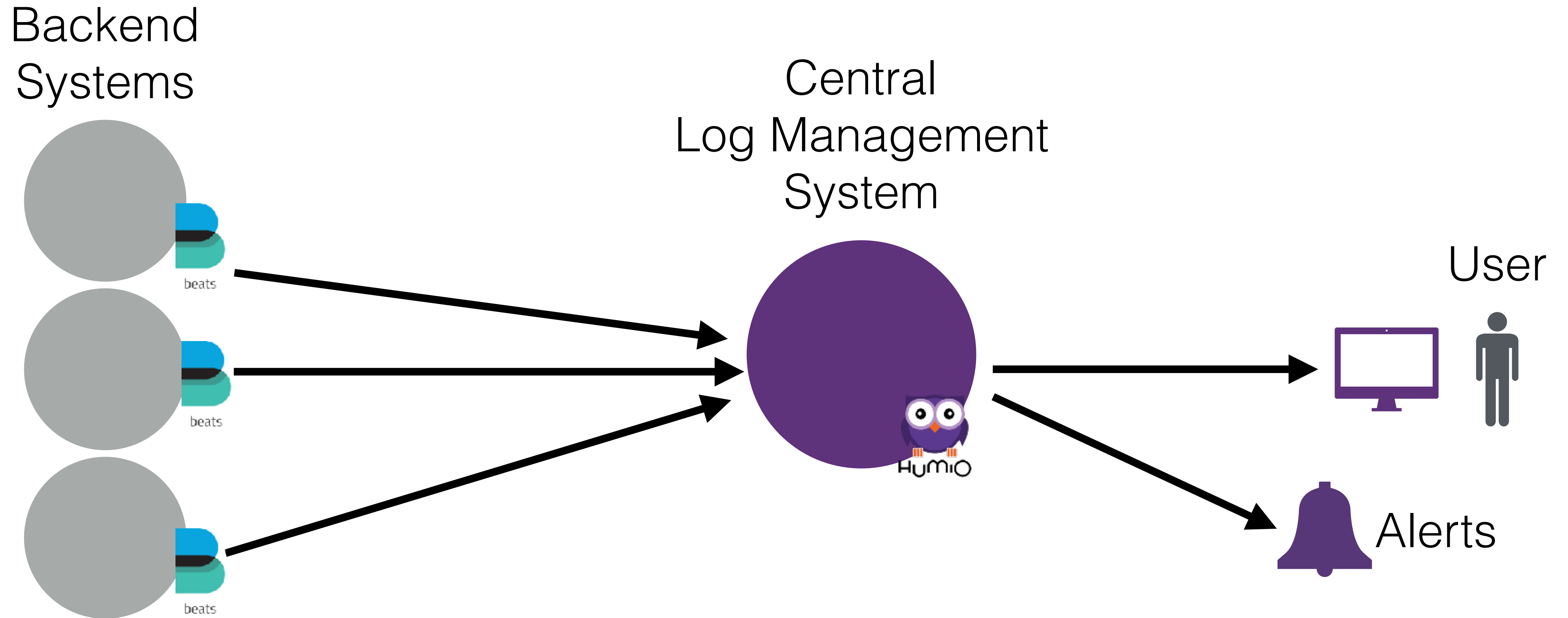
Logs will **vary** in volume – metrics have a **fixed** volume rate

Logs tend to be **high** volume – metrics tend to be **low** volume

Architecture and tools



Log Management Architecture



Log Management:
Provide access to logs across all systems

Log shippers



...and many more

Log shipping – the basics

Start simple - can be tricky to setup

Log to a file and use a log shipper to forward the logs to a log management system

Logs are available locally – debug when the pipeline is failing, better delivery guaranties

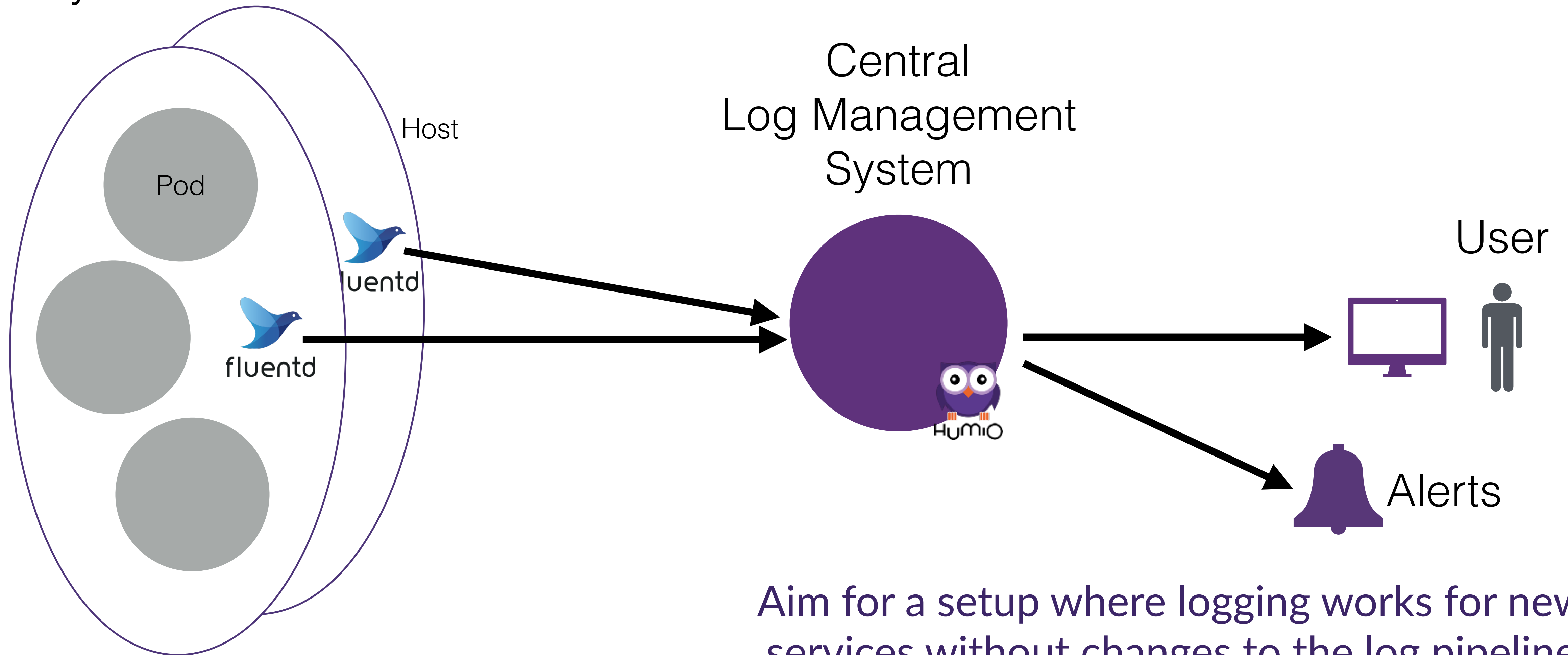
or

Configure you log library to ship logs directly to you log management system

Ease of setup, tends toward more structure – but fire and forget delivery

Orchestration

Backend
Systems



Aim for a setup where logging works for new services without changes to the log pipeline

Making logs good



Read your logs

Logs evolve

Changing code

Adjusting logs

Reading logs

Improving logs

Workflow

Logs from a workflow tell a story

- ID's - request, session, background job
- Life cycle events - start, stop, crash
- Progress
- State machine transitions
- Timings, sizes, counts - event level metrics

Context — increase dimensionality

Slice & dice your stories

- ID's
- User information
- Source file and line number
- Caller ID (Session, request, job)
- Originating ID (Session, request, job)
- Host, service, pod, task, container (name, ID)

Demo



Log Management from a User Perspective

- Filter
- Group by and time charts
- Follow the tail
- Dashboards
- Alerts

Tips & Tricks



Mixed bag of tips & tricks

- Log levels:
 - debug, info, warning, error, fatal
- Log to file vs. send directly to management system
- Log format:
 - JSON?
 - `2018-06-26T12:37:04.262+0000 [thread1] INFO
a.a.ActorSystemImpl - logged request: method=POST,
url="http://..." status=200, time=11`
- Metrics together with logs
- Git hash and config in startup log
- Follow the tail during deploy

Thank you

Download

humio.com/download

Free Trial

humio.com/pricing

Slack

community.humio.com