KubeCon | CloudNativeCon

Europe 2020

Virtual

Fluentd / fluent-bit Project Intro

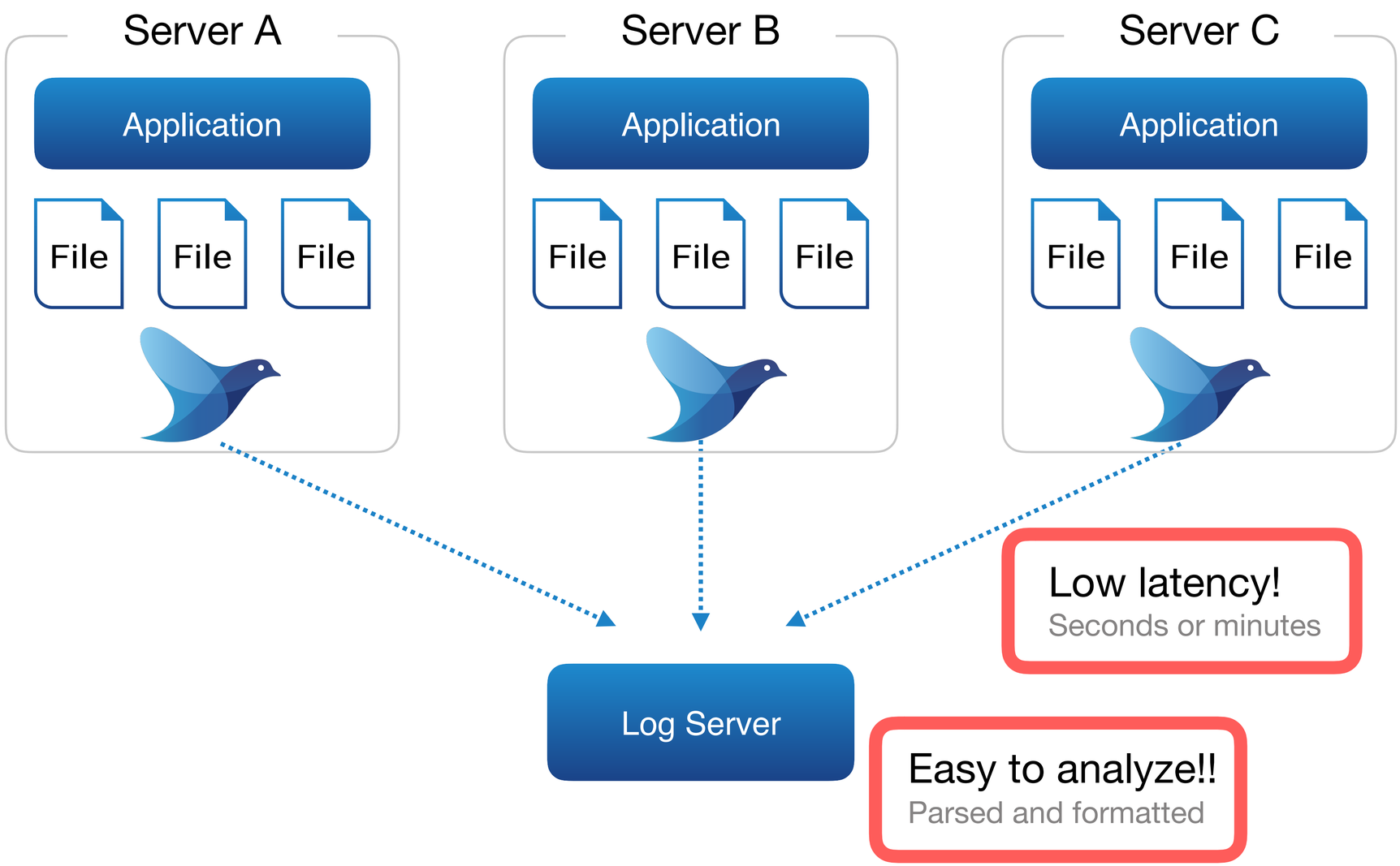
Masahiro Nakagawa / Eduardo Silva

Fluentd Overview

What’s Fluentd

- Streaming data collector for unified logging
  - Core + Plugins
- RubyGems based various plugins
  - Follow Ruby’s standard way
- Support OS packages and container images
  - https://docs.fluentd.org/installation
- Graduated Logging project in CNCF

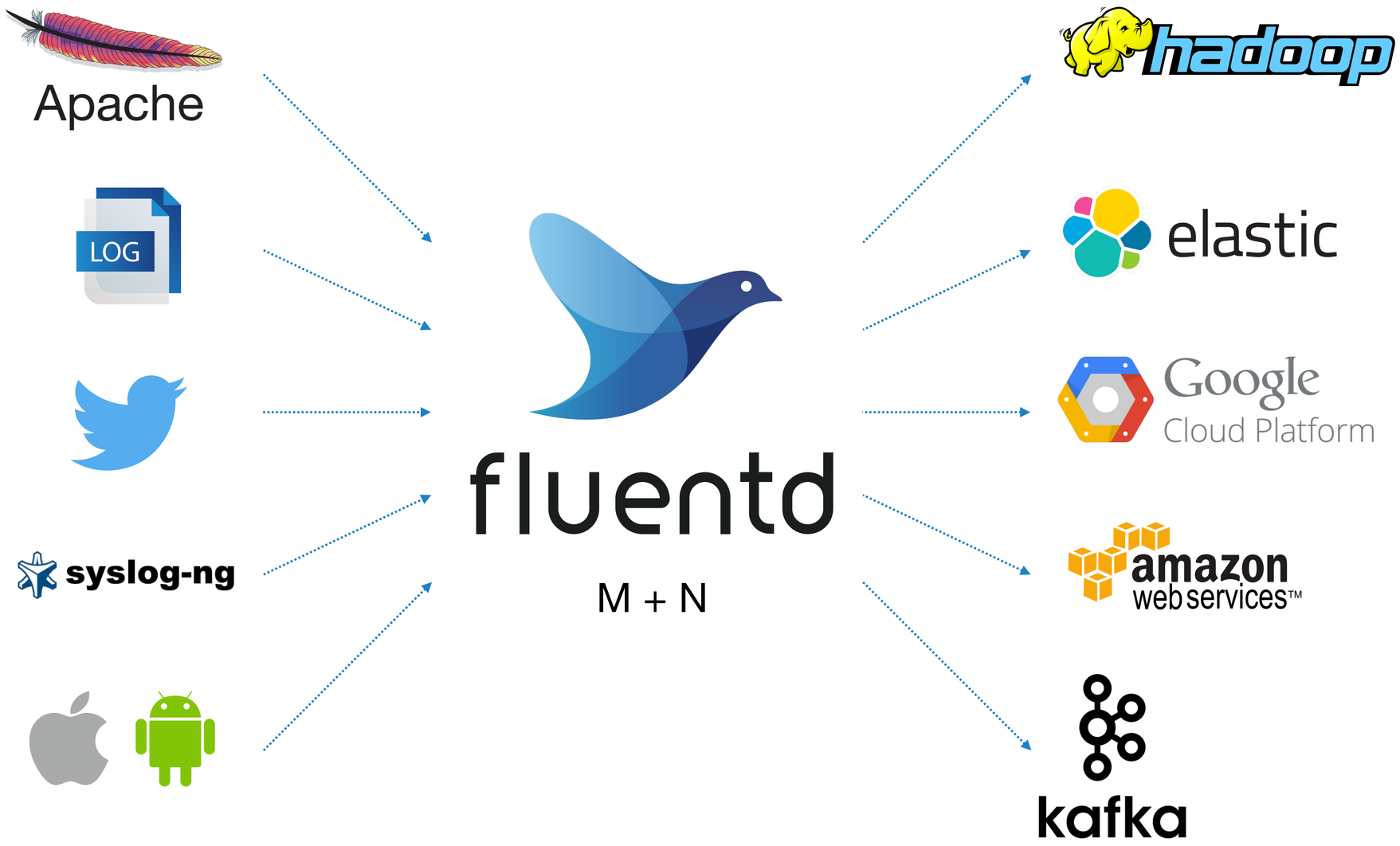
Streaming way with Fluentd



Logging on production

- Service Logs
  - software, middleware logs
  - ad, application logs
  - Transaction logs (game, ec, etc...)
- System Logs
  - syslog, systemd and other logs
  - audit logs
  - metrics (cpu, memory, etc...)

Unified logging layer



Fluentd Architecture

Design

Core

- Buffering & Retrying
- Error handling
- Event routing
- Parallelism
- Helper for plugins

Plugins

- Read / receive data
- Parse data
- Filter / enrich data
- Buffer data
- Format data
- Write / send data

Event structure

Time

- Nano-second unit
- from logs

Tag

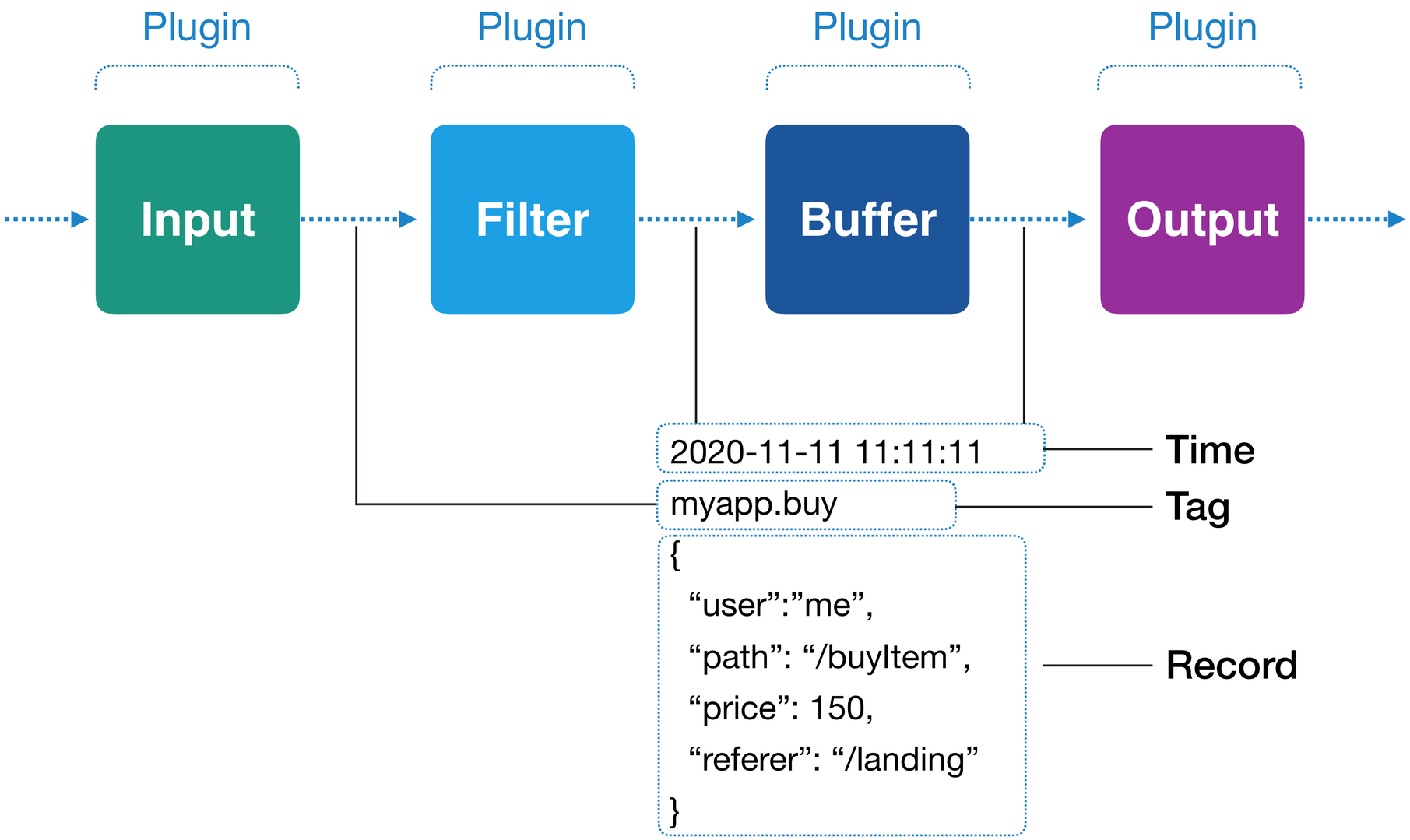
- for event routing
- Identify data source

Record

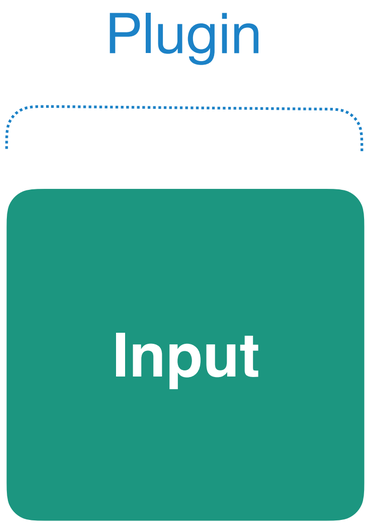
- JSON object,
  not raw string

```
{
  “str_field”:”hey”,
  “num_field”: 100,
  “bool_field”: true,
  “array_field”: [“elem1”, “elem2”]
}
```

Data pipeline (simplified)



Architecture: Input Plugins



- Receive or pull logs from data sources
- Emit logs to data pipeline
- Parse incoming logs for structured logging

HTTP+JSON (in_http)
Local files (in_tail)
Syslog (in_syslog)
…

Architecture: Filter Plugins

- Transform logs
- Filter out unnecessary logs
- Enrich logs

Modify logs (record_transformer)

Filter out logs (grep)

Parse field (parser)

…

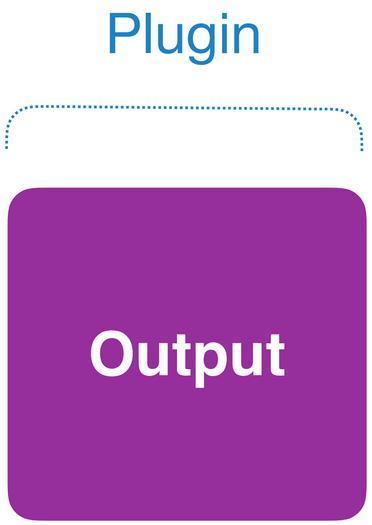
Architecture: Buffer Plugins

- Improve performance
- Provide reliability
- Provide thread-safety

Memory (buf_memory)

File (buf_file)

Architecture: Output Plugins



- Write or send event logs
- Sync or Async

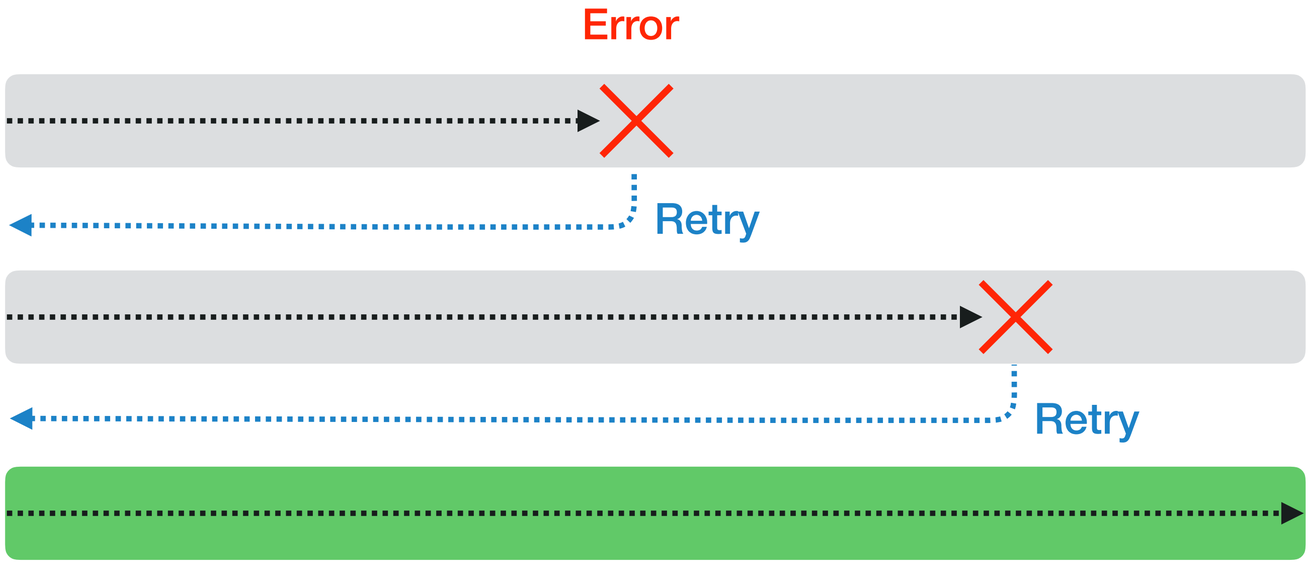
Local File (out_file)

Amazon S3 (out_s3)

Forward to other fluentd (out_forward)

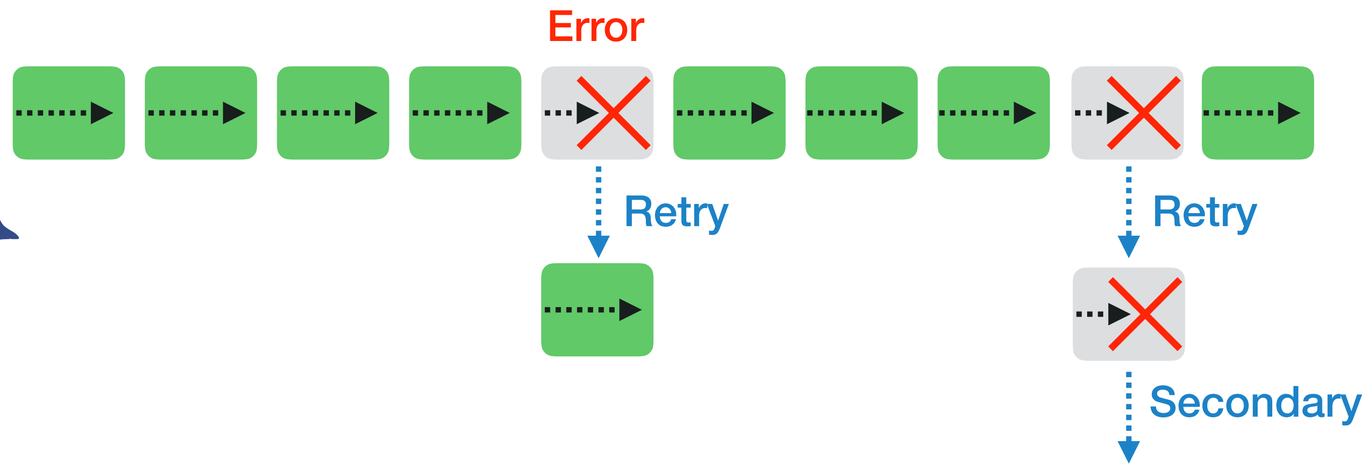
…

Divide & Conquer for retry

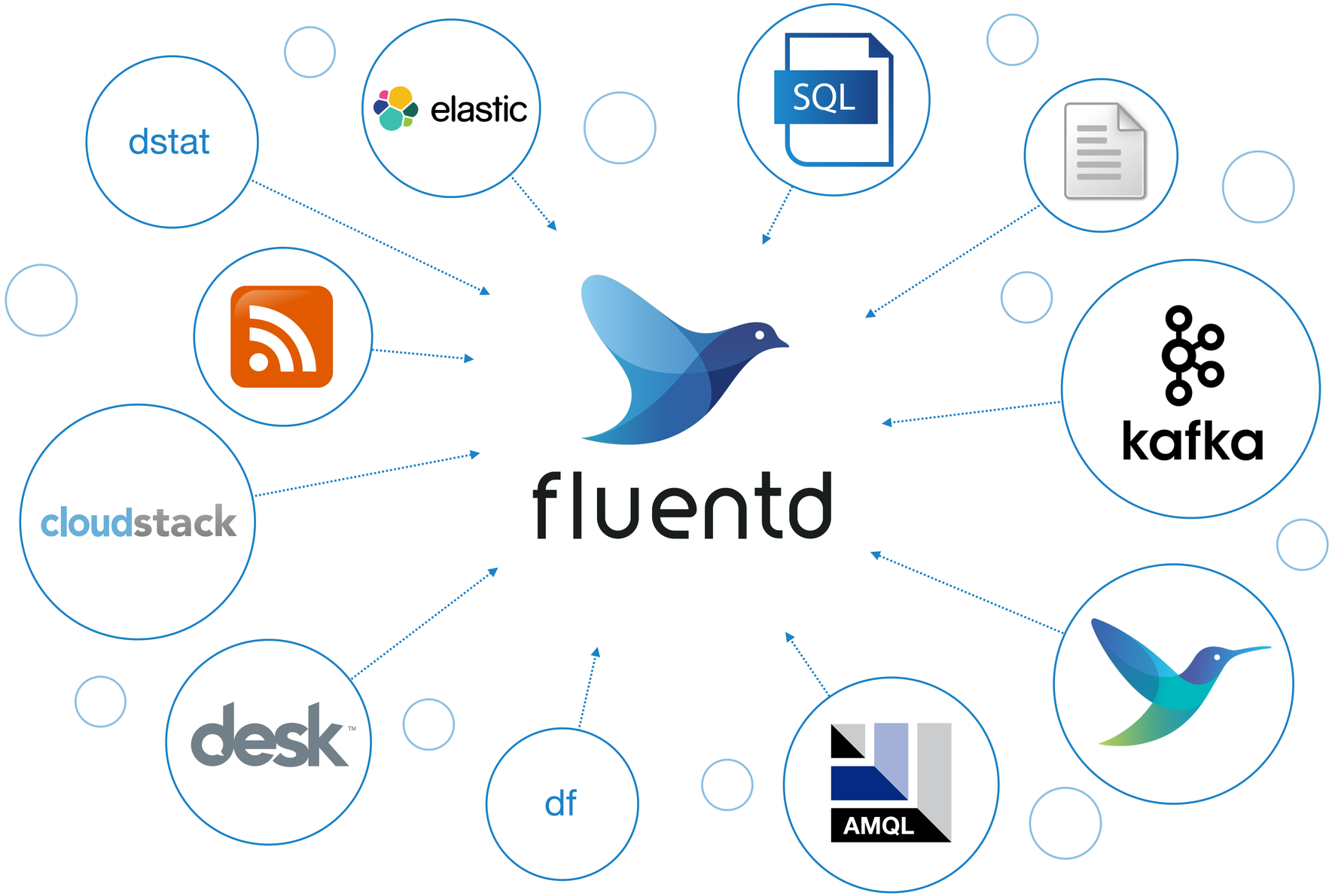


Batch

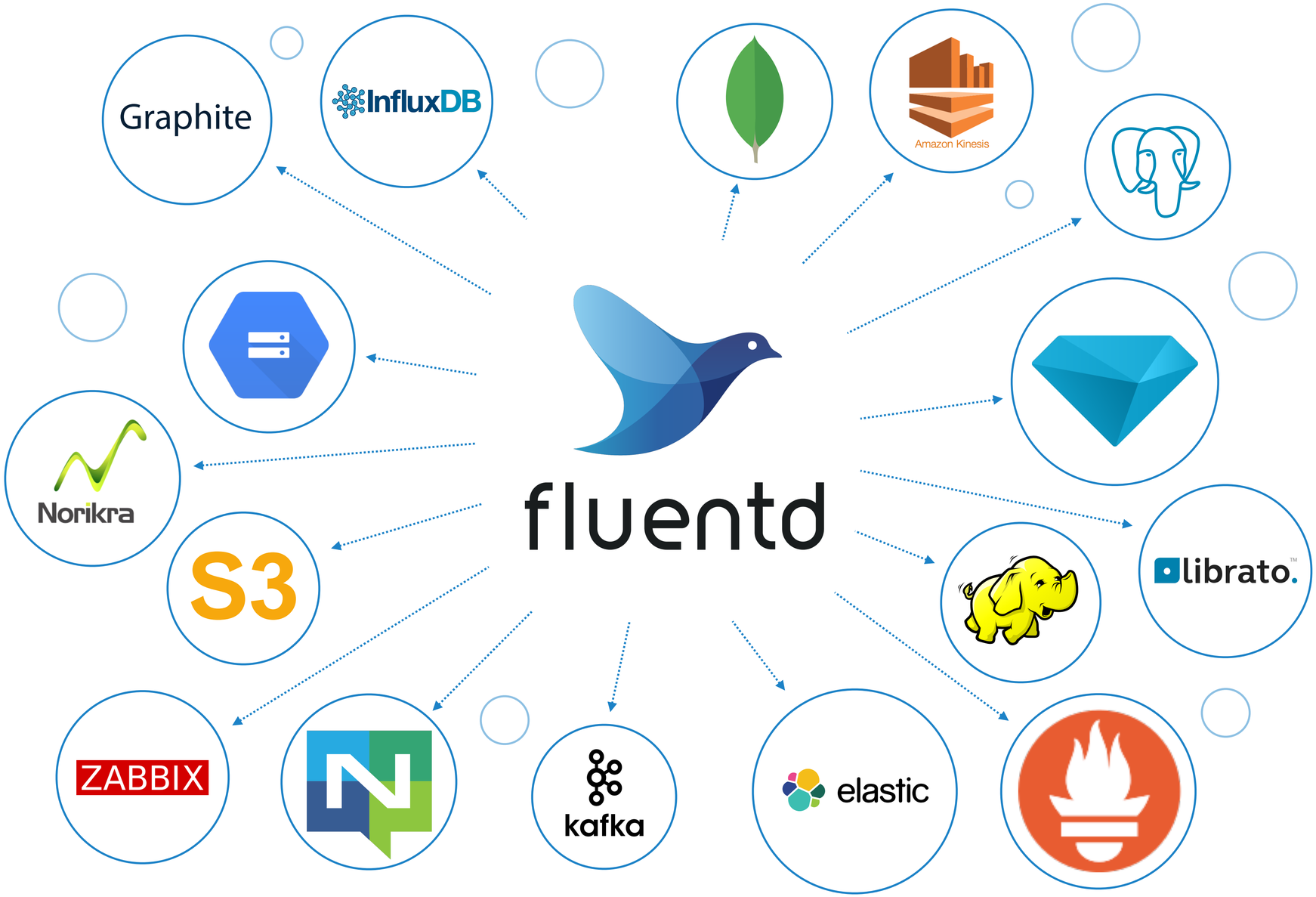
Stream



3rd party input plugins

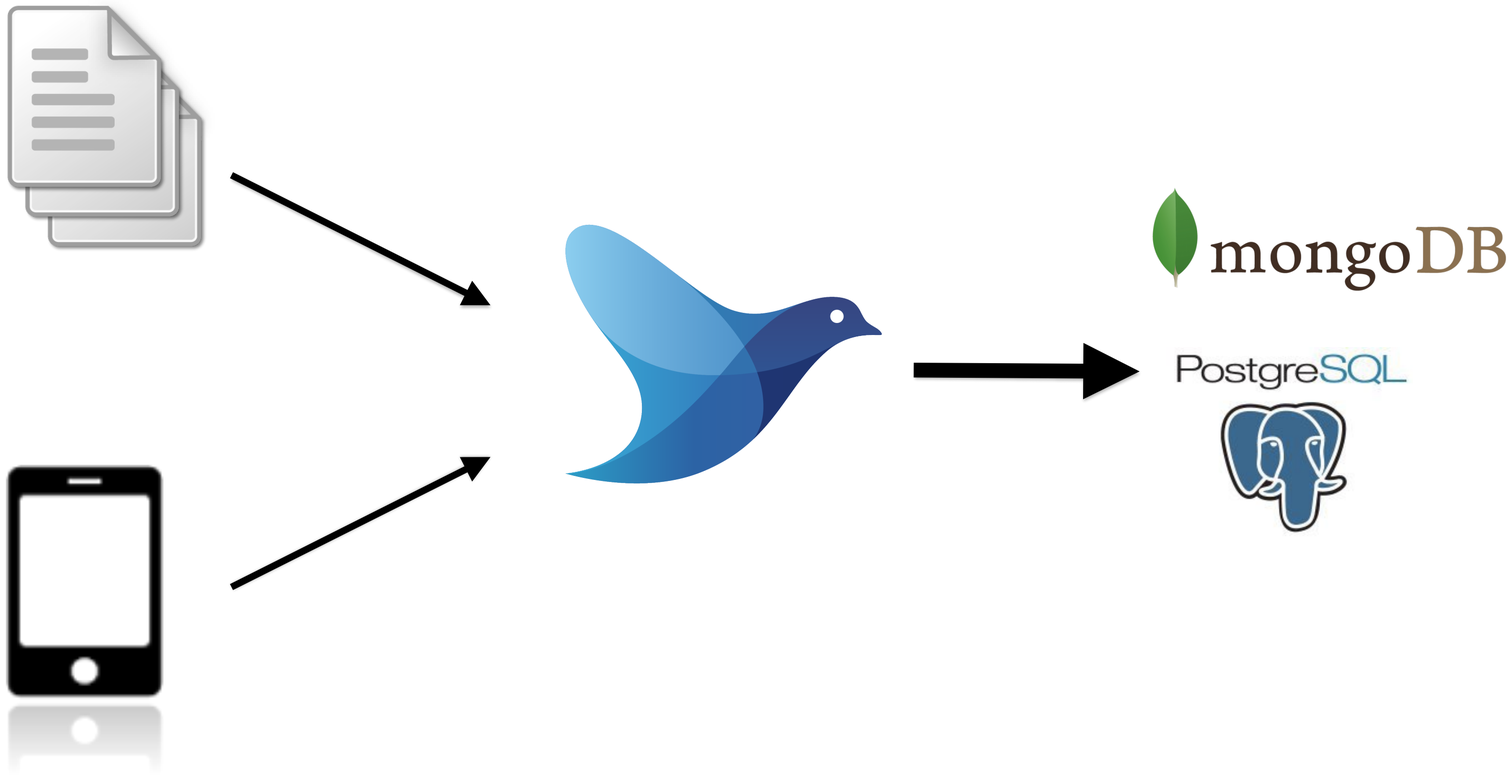


3rd party output plugins



Use-cases

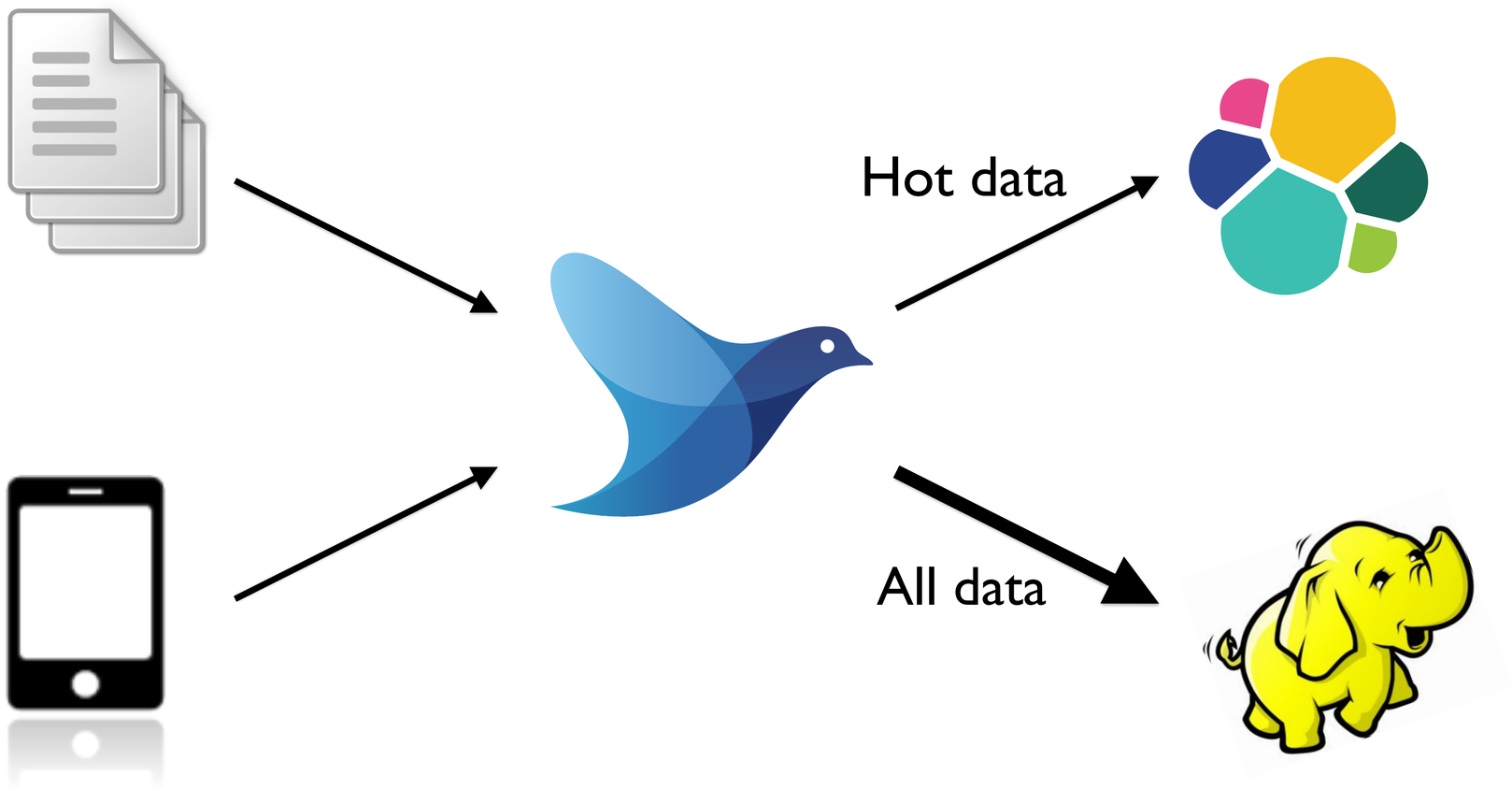
Simple forwarding



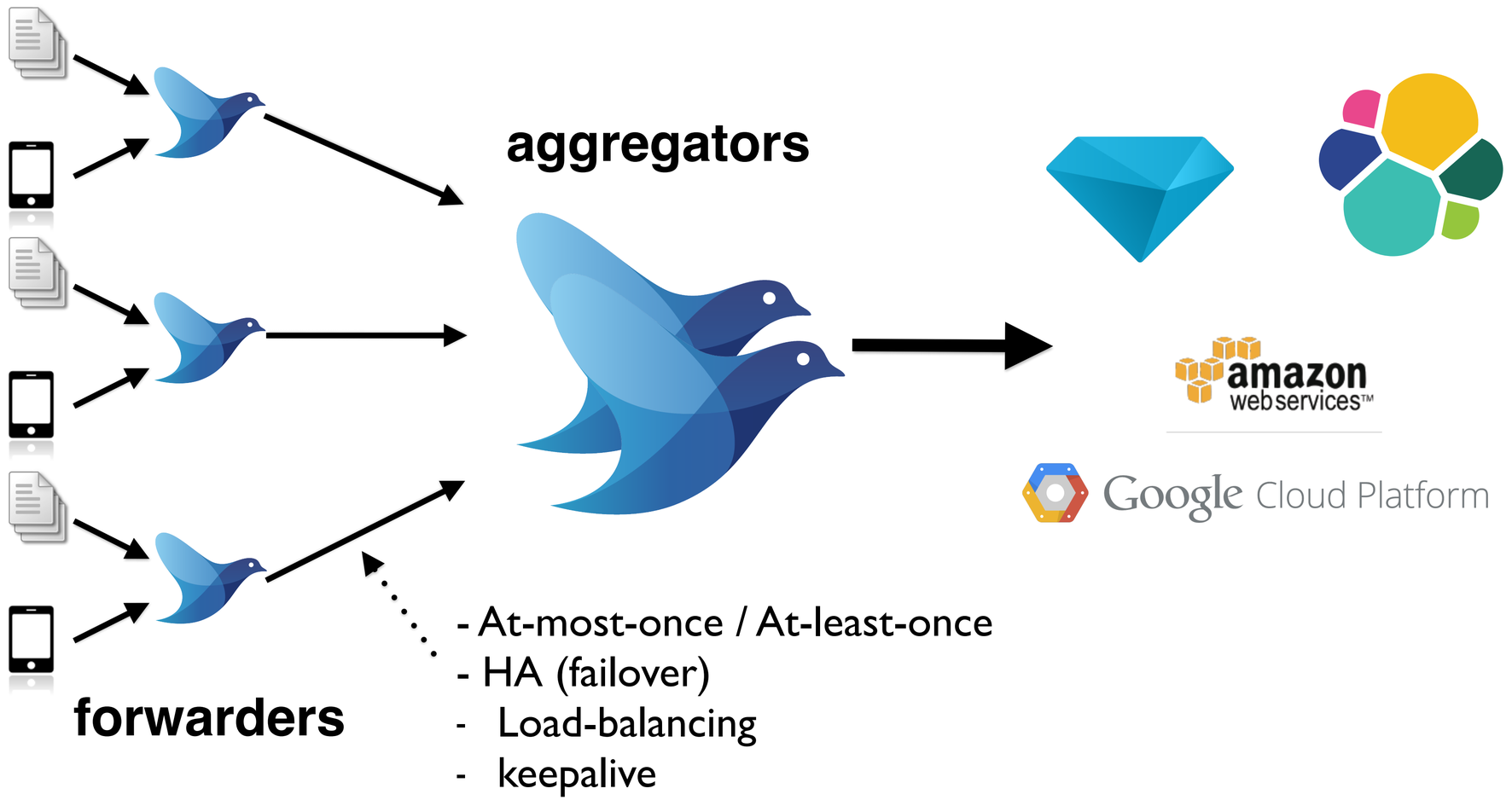
```
# logs from a file
<source>
  @type tail
  path /var/log/httpd.log
  pos_file /tmp/pos_file
  <parse>
    @type apache2
  </parse>
  tag app.apache
</source>

# logs from client libraries
<source>
  @type forward
  port 24224
</source>
```

```
# store logs to MongoDB
<match app.**>
  @type mongo
  database fluent
  collection logs
  <buffer tag>
    @type file
    path /tmp/fluentd/buffer
    flush_interval 30s
  </buffer>
</match>
```

Multiple destinations (copy)



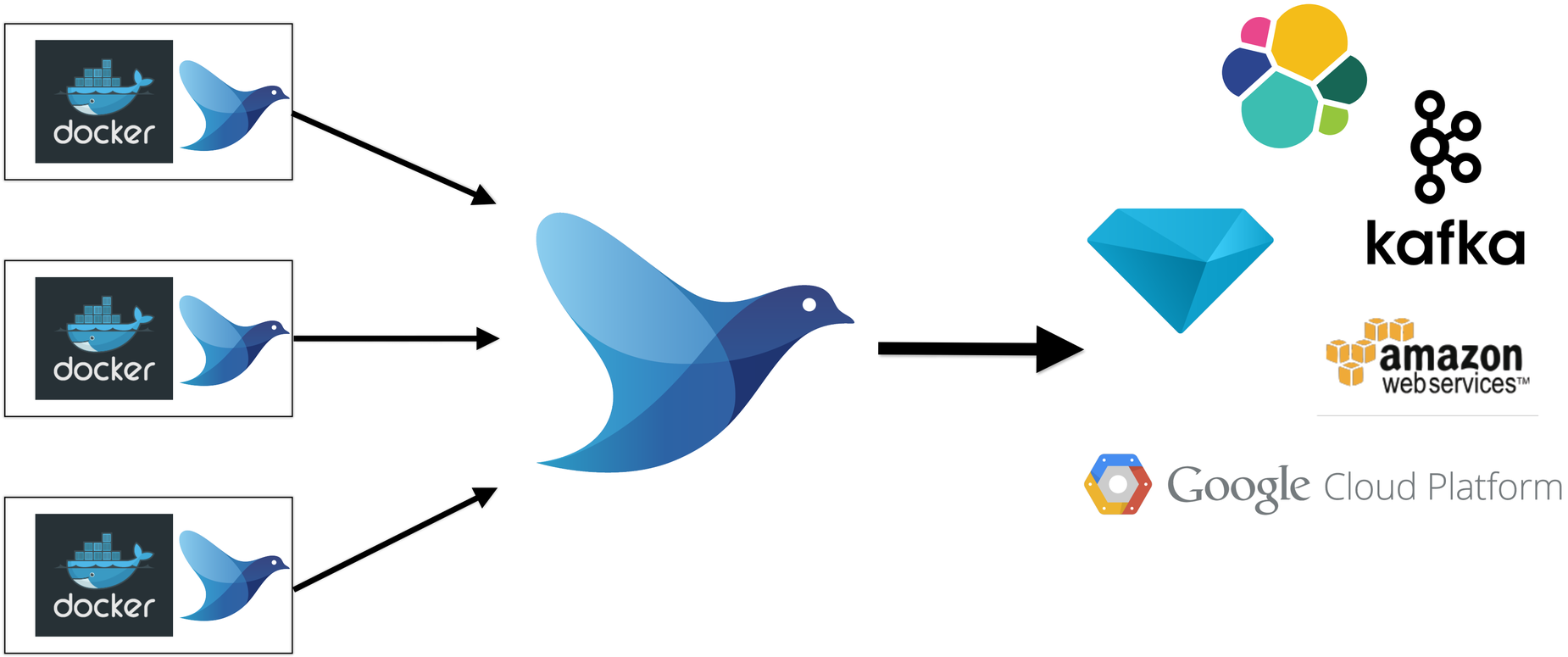
Multi-tier Forwarding



https://www.slideshare.net/repeatedly/fluentd-and-distributed-logging-at-kubecon

Container and Kubernetes

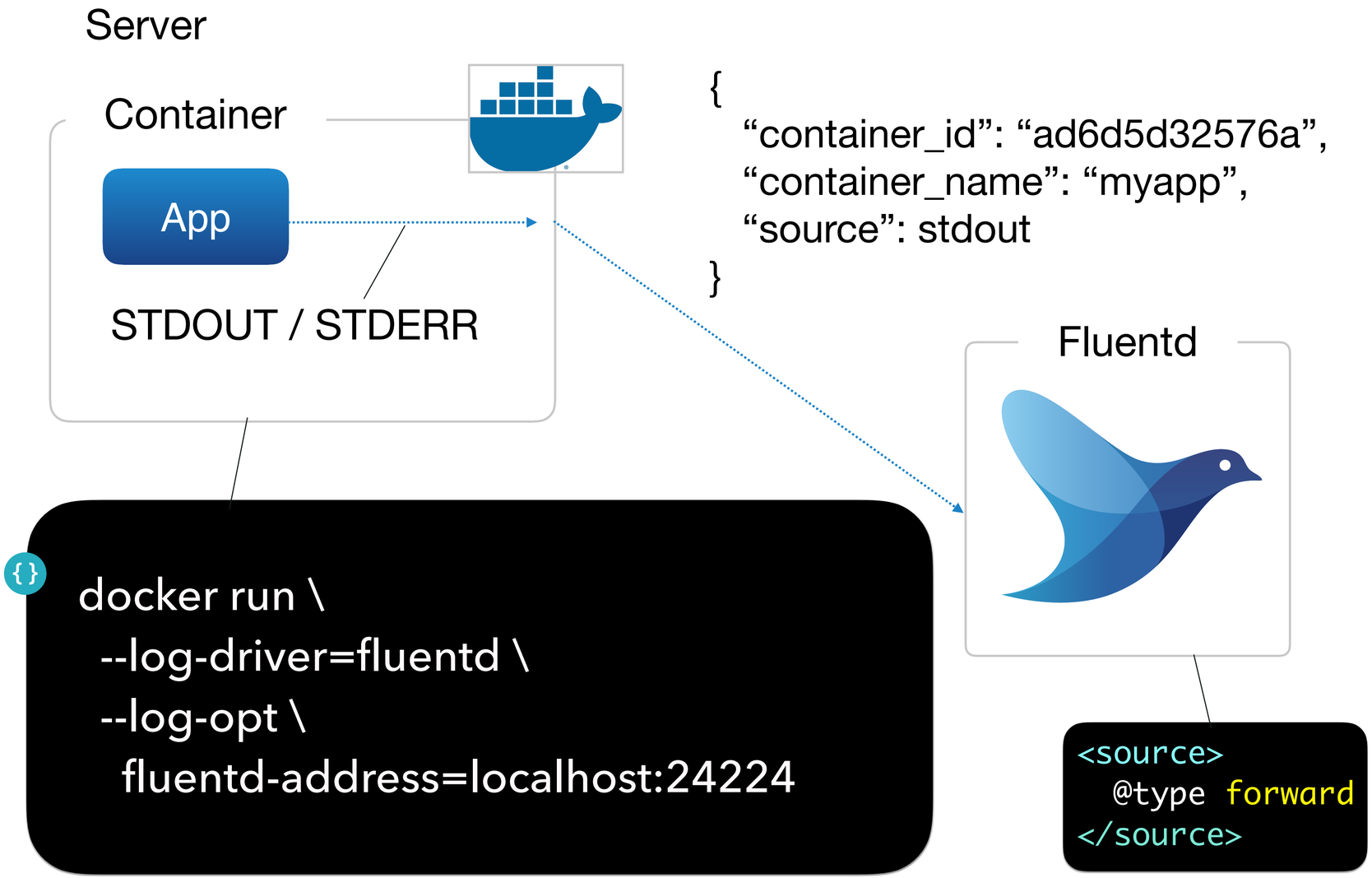
Container Logging



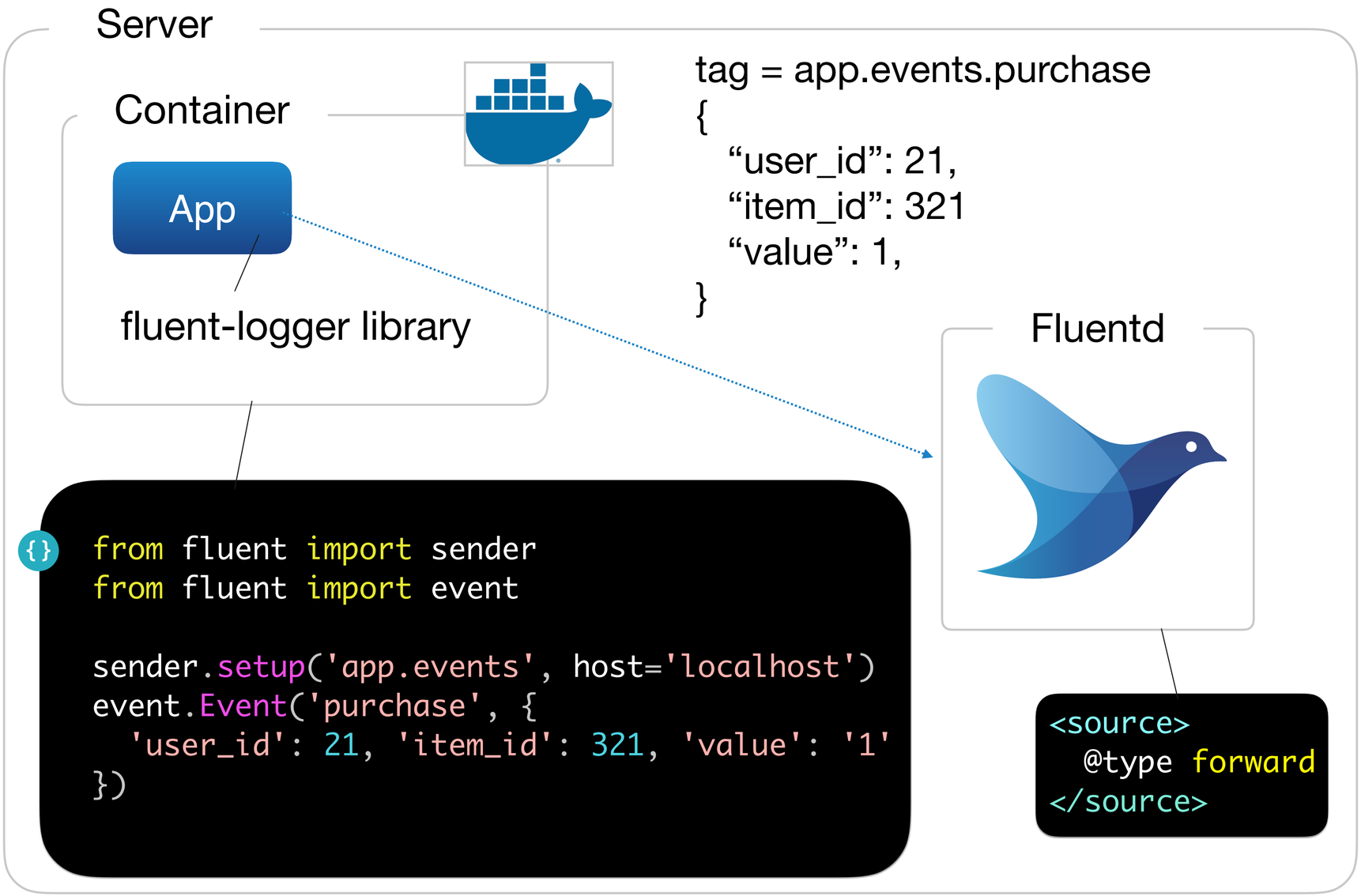
Resources

- Docker : fluentd-docker-image
  - Alpine / Debian images
  - x86, Arm, PowerPC, etc support by Docker official
- Kubernetes : fluentd-kubernetes-daemonset
  - Debian images
  - Some built-in destinations, ES, kafka, graylog, etc…
- Helm chart
  - https://github.com/helm/charts/tree/master/stable/fluentd

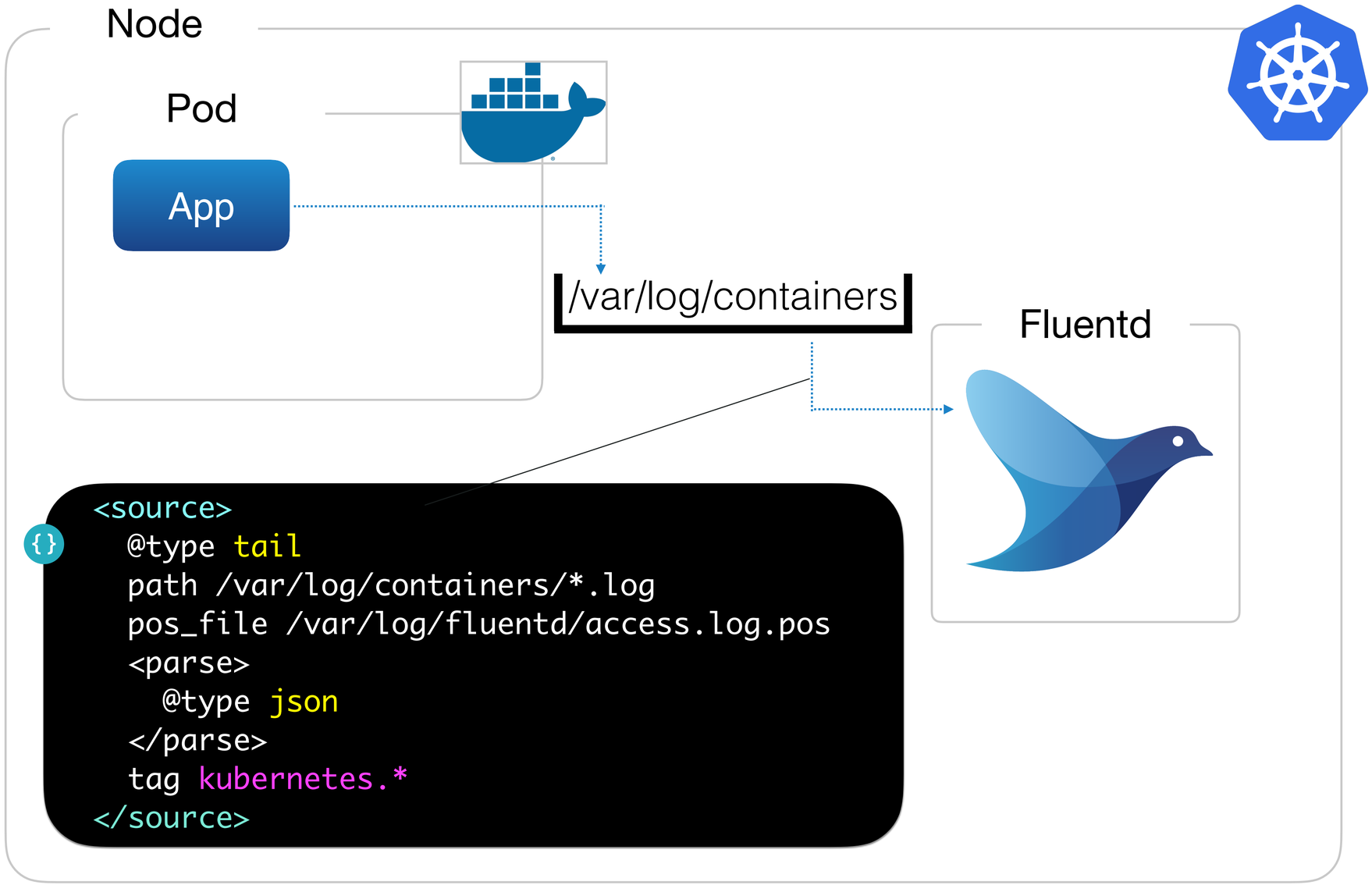
Docker logging with --log-driver=fluentd



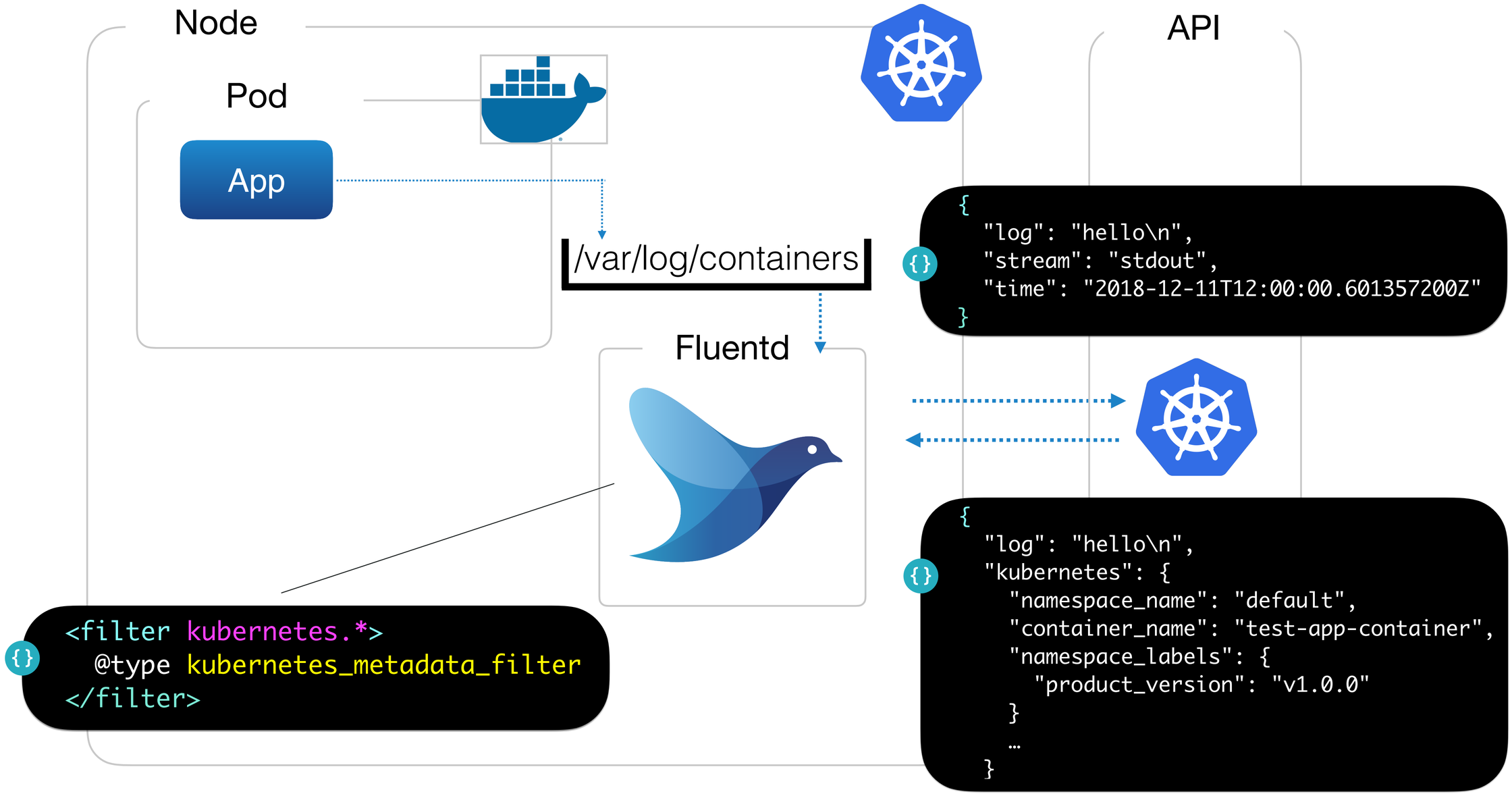
Data collection with fluent-logger



Kubernetes Daemonset



Kubernetes Daemonset & metadata



Container Logging approach summary

- Collect log messages with docker
  - --log-driver=fluentd
- Application data/metrics
  - fluent-logger
- Kubernetes Daemonset
  - Collect container logs from /var/log/containers/*
  - Add kubernetes metadata to logs
- Access logs, logs from middleware
  - Shared data volume with in_tail

Fluent-bit