

# Monitoring

Fish  
jfsc@cin.ufpe.br

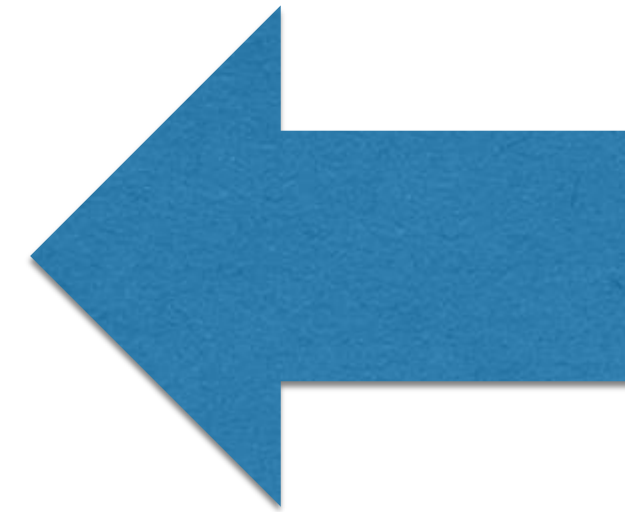
# THE ATTENDANTS WILL LEARN...

- Monitoring basics
- Traditional vs Modern monitoring
- Challenges of monitoring at scale
- What to monitor for DevOps?
- Microservices monitoring rules
- Elastic Stack fundamentals

# What is monitoring?

# Monitoring Topics

- Customer happiness
- Cost efficiency
- Safety and security
- Compliance



We'll talk about this for this class

# Customer Happiness

- Time to value (features in front of client)
- Availability
- Response time

# Cost Efficiency

- Utilization
- Optimization
- Automation

# A standard for monitoring

←

→

↺

pubs.opengroup.org/onlinepubs/009657299/c427-1/front.htm

☆

🔍

💬

🌐

🏠

🔒

🔑

Universal Measurement Architecture Guide  
Copyright © 1997 The Open Group

Frontmatter

Guide
Systems Management: UMA Specification, Part 1 - Universal Measurement Architecture
X/Open Document Number: C427
ISBN: 1-85912-117-9

©January 1997, The Open Group All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Any comments relating to the material contained in this document may be submitted to The Open Group at:

The Open Group  
Apex Plaza  
Forbury Road  
Reading  
Berkshire, RG1 1AX  
United Kingdom

or by electronic mail to:

OGSpecs@opengroup.org

Preface

This Document

This X/Open Guide reviews the problem space of performance measurement in open systems, and introduces the architecture, features and benefits provided by the three X/Open Universal Measurement Architecture (UMA) specifications:

# Traditional Monitoring

- The traditional monitoring approach tests just whether something is 'up' or 'down';
- Agents inside: Hardware, Mainframe, VMs



# Modern monitoring

- If a single component within a microservice architecture fails, there may be no **business impact**, and so **the severity of alert should match this fact**.
- Containers, events (serverless).

# Challenges: Aggregation

- Consolidating different (or without) log formats coming from a lot of containers or events with different proposal of visibility

# Distribution: Interdependence

- Maintain operational health and doing Root Cause Analysis on a large ecosystem (many containers or events) at scale becomes challenging.

# Frequency of change: ephemerality

---

- The short lifetime of containers or events has been considered a big challenge in modern monitoring.

# Unpredictable

- We don't know where the container. Has the Kubernetes moved Nginx to another zone?

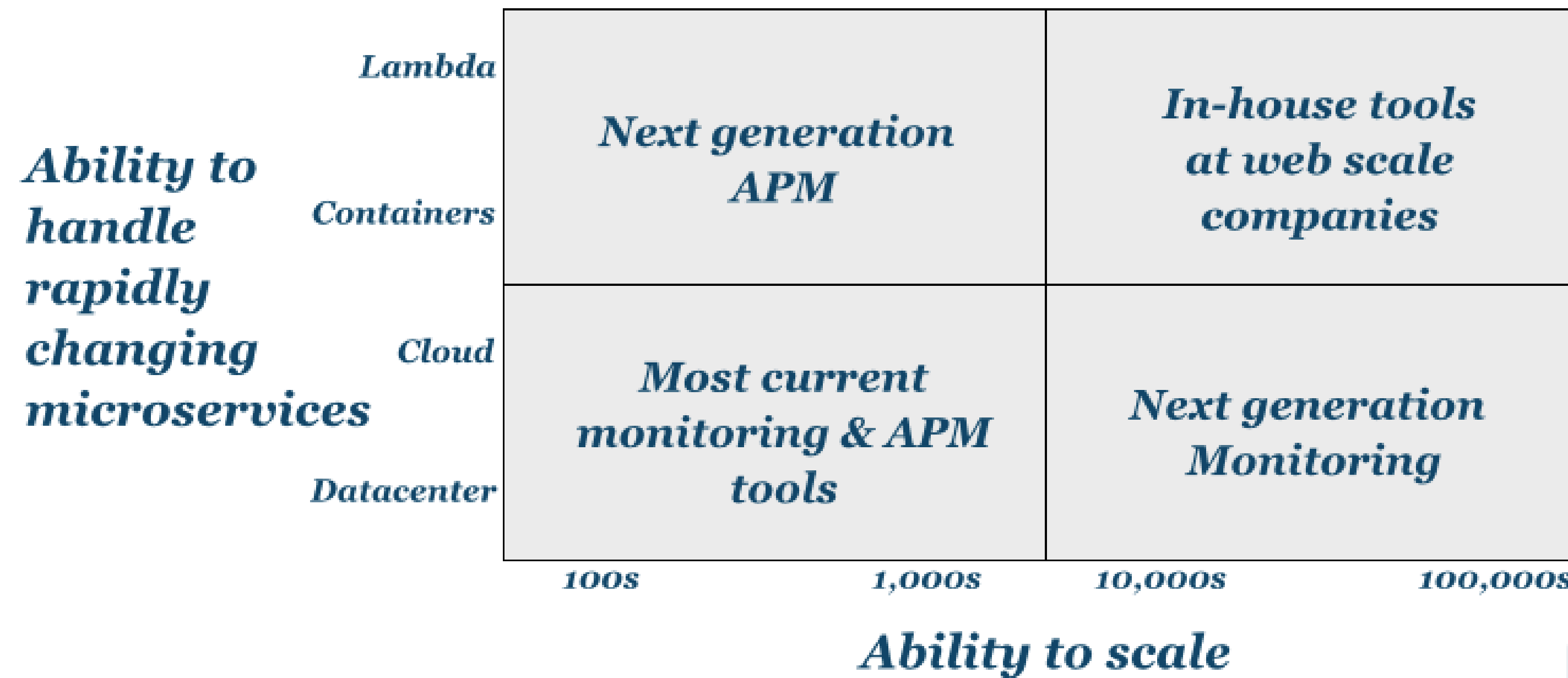


# Tips: 7 containers per host



# The Adrian's Tragic Quadrant

## A Tragic Quadrant



From: <https://www.battery.com/powered/bigpanda-analyzing-the-monitoringscape/>

# What to monitor? from orquestrators

- Swarm,
- mesos,
- kubernetes



# What to monitor? anomalies

---

- Monitoring systems have to learn normal behavior and indicate when system behavior is not normal.

# Rules for microservices

- Spend more time working on code that analyses the meaning of metrics than code that collects, moves, stores and displays metrics
- Reduce key business metric latency to less than the human attention span (~10s)
- Validate your measurement system has enough accuracy and precision. Collect histograms of response time
- Monitoring systems need to be more available and scalable than the systems (and services) being monitored
- Optimise for monitoring distributed, ephemeral, 'cloud-native', containerised microservices
- Fit metrics to models in order to understand relationships (this is a new rule)

From: <https://github.com/adrianco>

# Hands on phase

# Monitoring with Elastic Stack

# What is a log?

# A log is an event

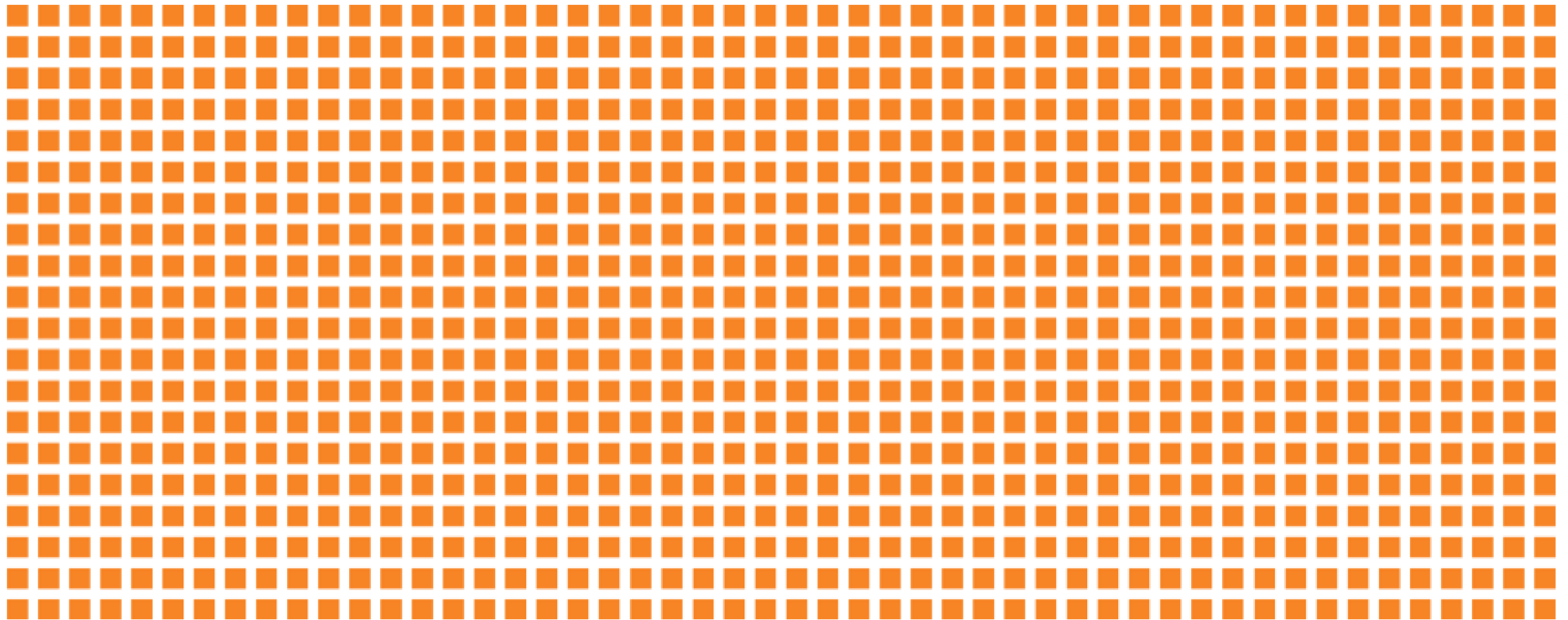
- Log lines
- Twitter feed
- Invoices
- Metrics

# Exercise 1

---

- Reading logs from containers

# At scale



1000 EC2 Instances in a Cluster



# grep

- Multiple machines
- Multiple logs
- Analysis/Discovery
- Time Period

# Different point of views

apache

```
[23/Jan/2014:17:11:55 +0000]
```

unix timestamp

```
1390994740
```

log4j

```
[2014-01-29 12:28:25,470]
```

postfix.log

```
Feb 3 20:37:35
```

ISO 8601

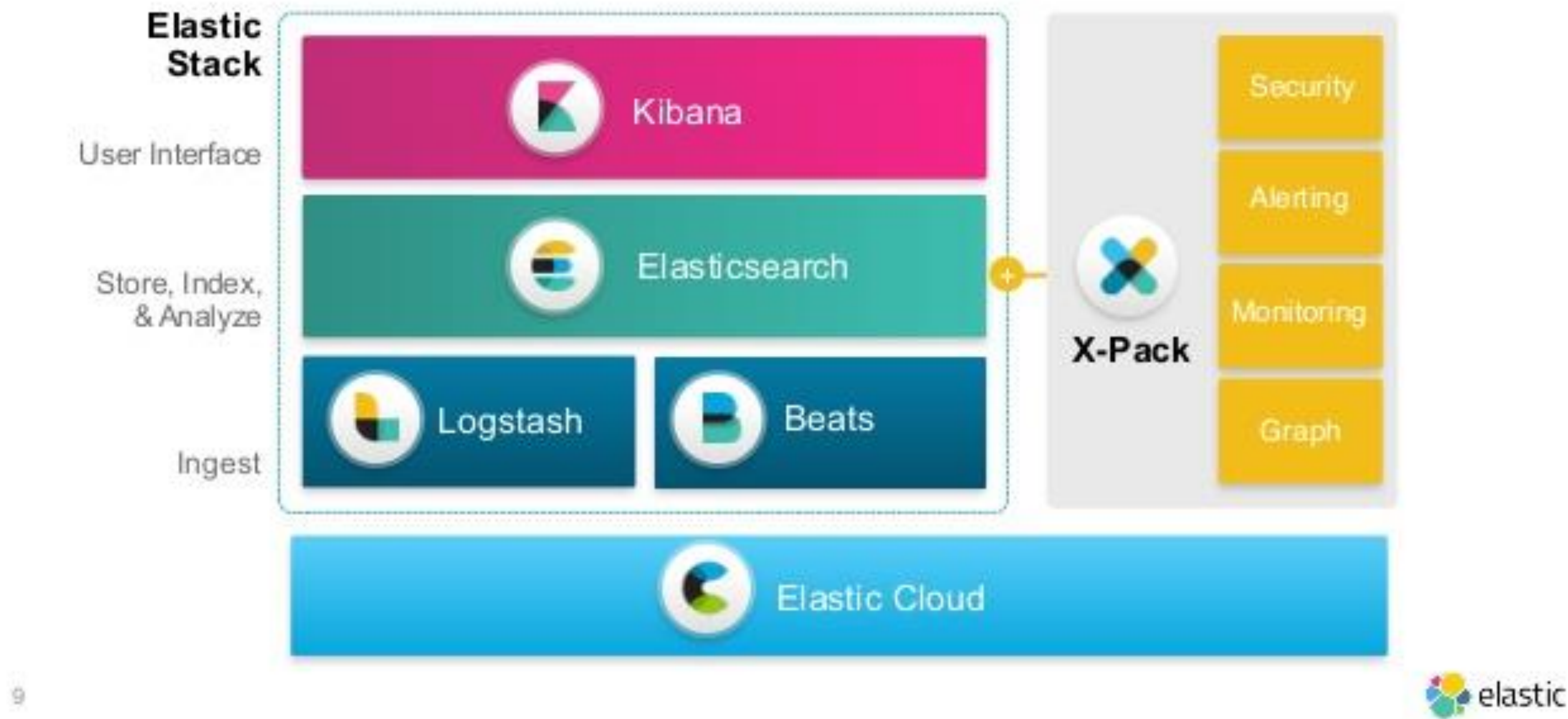
```
2009-01-01T12:00:00+01:00
```

# A Centralized logging approach can help you

- Collect data
- Parse data
- Enrich data
- Store data
- Search and aggregate
- Visualize data

# An ecosystem as response: Elastic Stack

The Elastic Stack is a set of tools developed by Elastic to provide monitoring (nowadays).



# Some Elastic Stack features

- A business centric monitoring;
- Integration-driven (flexibility to get logs from everywhere);
- Taming decentralization;
- Process (indexing) and search near real-time data;
- Use analytics to summarize data across dimensions



# Logstash : inputs

<b>Monitoring</b>	collectd, graphite, ganglia, snmptrap, zenoss
<b>Datastores</b>	elasticsearch, redis, sqlite, s3
<b>Queues</b>	kafka, rabbitmq, zeromq
<b>Logging</b>	beats, eventlog, gelf, log4j, rlp, syslog, varnish log
<b>Platforms</b>	drupal_dblog, gemfire, heroku, sqs, s3, twitter
<b>Local</b>	exec, generator, file, stdin, pipe, unix
<b>Protocol</b>	imap, irc, stomp, tcp, udp, websocket, wmi, xmpp

# Logstash : Filters

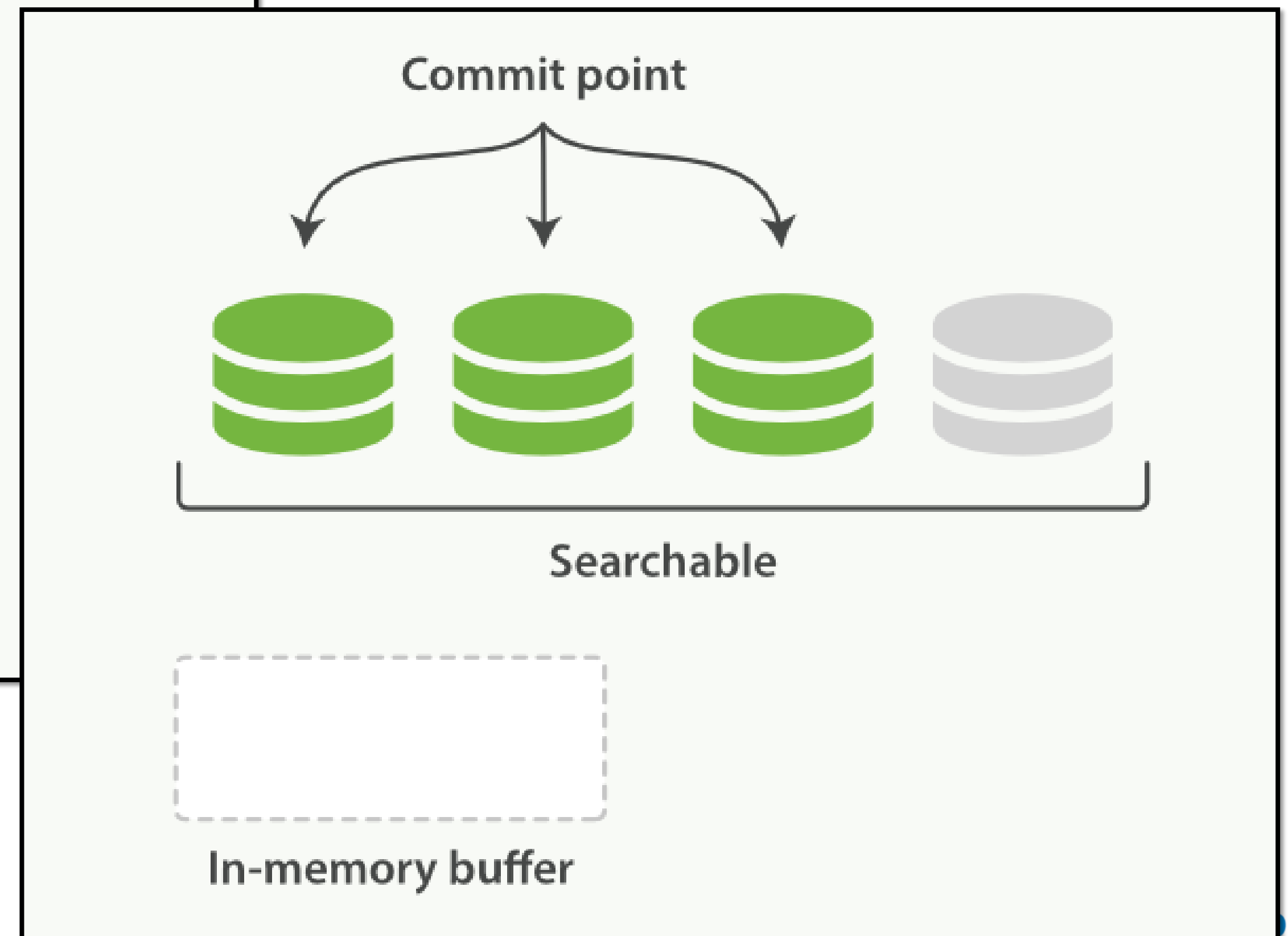
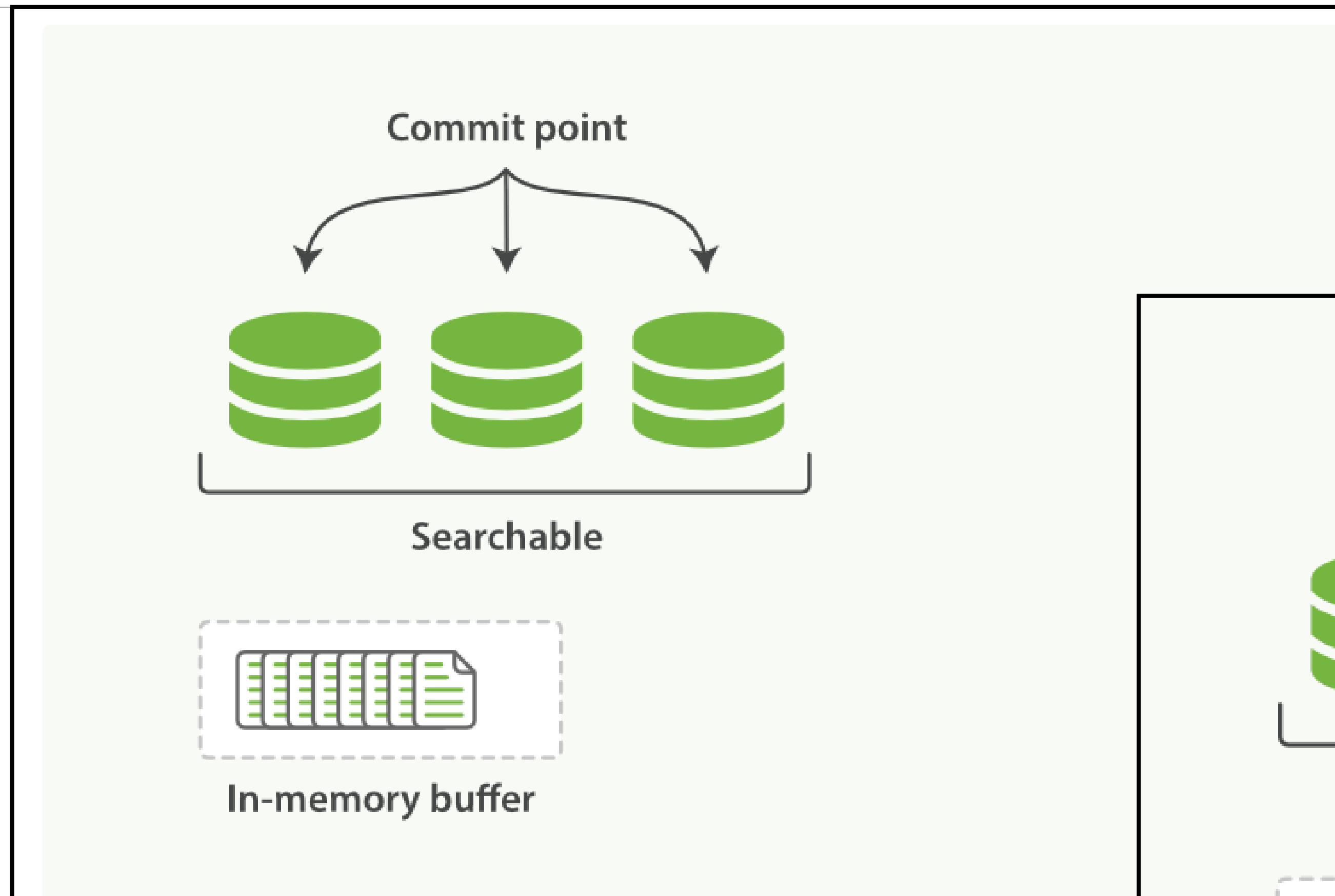
aggregate alter anonymize collate csv  
cidr clone cipher checksum date dns  
drop elasticsearch extractnumbers  
environment elapsed fingerprint geoip  
grok i18n json json\_encode kv mutate  
metrics multiline metaevent prune punct  
ruby range syslog\_pri sleep split throttle  
translate uuid urldecode useragent xml

# Logstash : Output

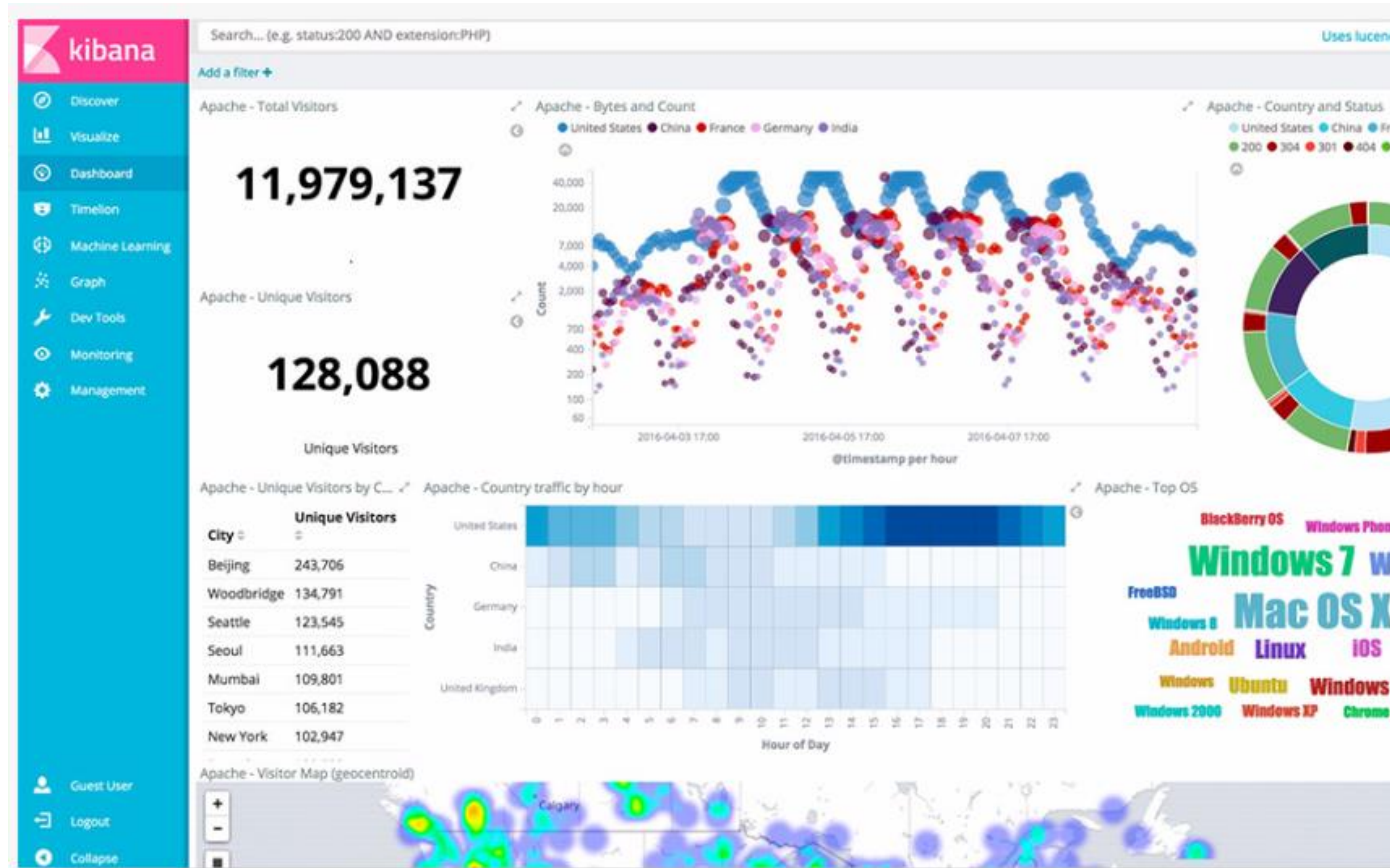
<b>Store</b>	elasticsearch, gemfire, mongodb, redis, riak, rabbitmq, solr
<b>Monitoring</b>	ganglia, graphite, graphtastic, nagios, opentsdb, statsd, zabbix
<b>Notification</b>	email, hipchat, irc, pagerduty, sns
<b>Protocol</b>	gelf, http, lumberjack, metriccatcher, stomp, tcp, udp, websocket, xmpp
<b>External service</b>	google big query, google cloud storage, jira, loggly, riemann, s3, sqs, syslog, datadog
<b>External monitoring</b>	boundary, circonus, cloudwatch, librato
<b>Local</b>	csv, dots, exec, file, pipe, stdout, null



# Elastic Search



# Kibana





# References

- <https://speakerdeck.com/elasticsearch/log-all-the-things>
- <https://www.battery.com/powered/bigpanda-analyzing-the-monitoringscape/>
- <https://www.infoq.com/articles/monitoring-containers-at-scale>
- <https://github.com/adrianco>