

# BEYOND OPENTRACING

KubeCon + CloudNativeCon — May 2-4, 2018

Allison Richardet — Asteris, LLC



ASTERIS

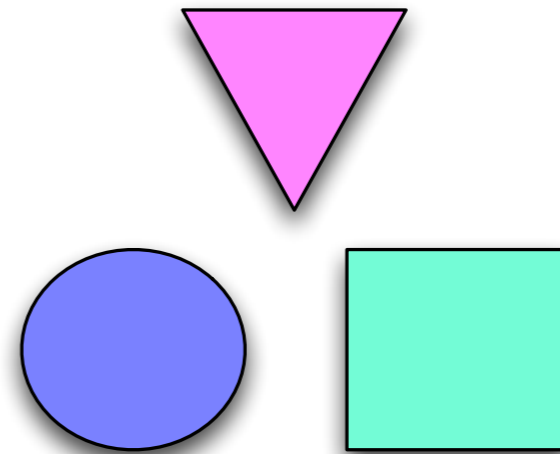


arichardet

# BACKGROUND

- Kubernetes
- Distributed tracing
- Structured logging

# MONOLITH VS MICROSERVICES



# OBSERVABILITY



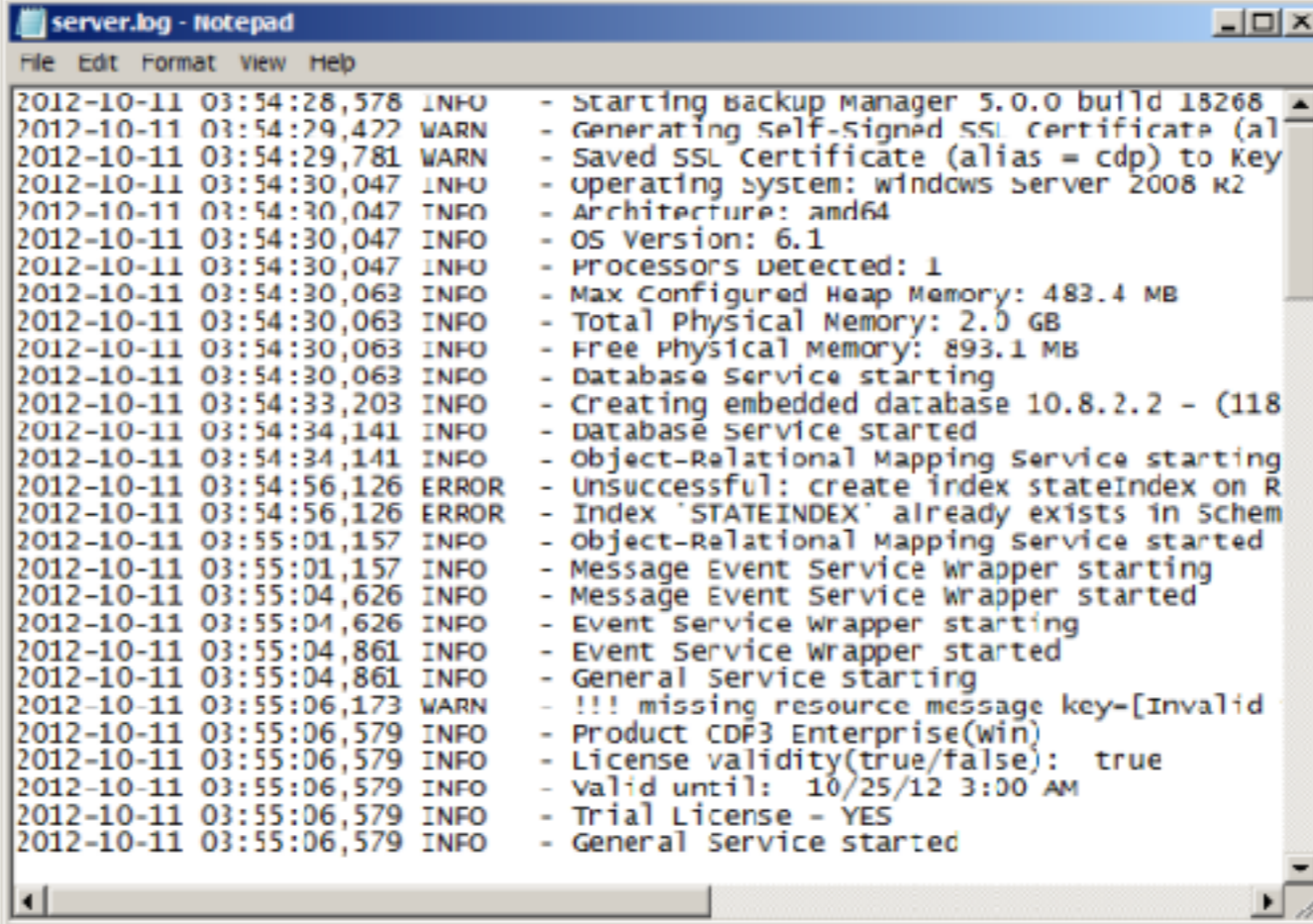
**NGINX**



# PROBLEM

- Knowledge of system state is hard
- Multiple services creating events and logging
- Need to simplify understanding the system state
- Enrich data

# MONOLITH LOGGING



```
server.log - Notepad
File Edit Format View Help
2012-10-11 03:54:28,578 INFO - Starting backup manager 5.0.0 build 18268
2012-10-11 03:54:29,422 WARN - Generating self-signed ssl certificate (al
2012-10-11 03:54:29,781 WARN - Saved SSL certificate (alias = cdp) to key
2012-10-11 03:54:30,047 INFO - operating system: windows server 2008 R2
2012-10-11 03:54:30,047 INFO - Architecture: amd64
2012-10-11 03:54:30,047 INFO - OS Version: 6.1
2012-10-11 03:54:30,047 INFO - Processors detected: 1
2012-10-11 03:54:30,063 INFO - Max Configured Heap Memory: 483.4 MB
2012-10-11 03:54:30,063 INFO - Total Physical Memory: 2.0 GB
2012-10-11 03:54:30,063 INFO - Free Physical Memory: 893.1 MB
2012-10-11 03:54:30,063 INFO - Database service starting
2012-10-11 03:54:33,203 INFO - Creating embedded database 10.8.2.2 - (118
2012-10-11 03:54:34,141 INFO - Database service started
2012-10-11 03:54:34,141 INFO - Object-Relational Mapping service starting
2012-10-11 03:54:56,126 ERROR - Unsuccessful: create index stateIndex on R
2012-10-11 03:54:56,126 ERROR - Index 'STATEINDEX' already exists in schem
2012-10-11 03:55:01,157 INFO - Object-Relational Mapping service started
2012-10-11 03:55:01,157 INFO - Message Event Service wrapper starting
2012-10-11 03:55:04,626 INFO - Message Event Service wrapper started
2012-10-11 03:55:04,626 INFO - Event Service wrapper starting
2012-10-11 03:55:04,861 INFO - Event Service wrapper started
2012-10-11 03:55:04,861 INFO - General service starting
2012-10-11 03:55:06,173 WARN - !!! missing resource message key-[invalid
2012-10-11 03:55:06,579 INFO - Product CDP3 Enterprise(win)
2012-10-11 03:55:06,579 INFO - License validity(true/false): true
2012-10-11 03:55:06,579 INFO - valid until: 10/25/12 3:00 AM
2012-10-11 03:55:06,579 INFO - Trial License - YES
2012-10-11 03:55:06,579 INFO - General Service started
```

# TOWARD SOLUTION

- Logging
- Structured logging
- Grammar structured logging



# IS THIS LOGGING?

- “store some data”
- “failed transaction”
- “open file” vs “open file failed”

# STRUCTURED LOGGING

- Impose format to logging
- Reduce cardinality - limit unique values
- Improve queries
- Improve analytics

# LOGRUS



# APPLICATION DEVELOPER

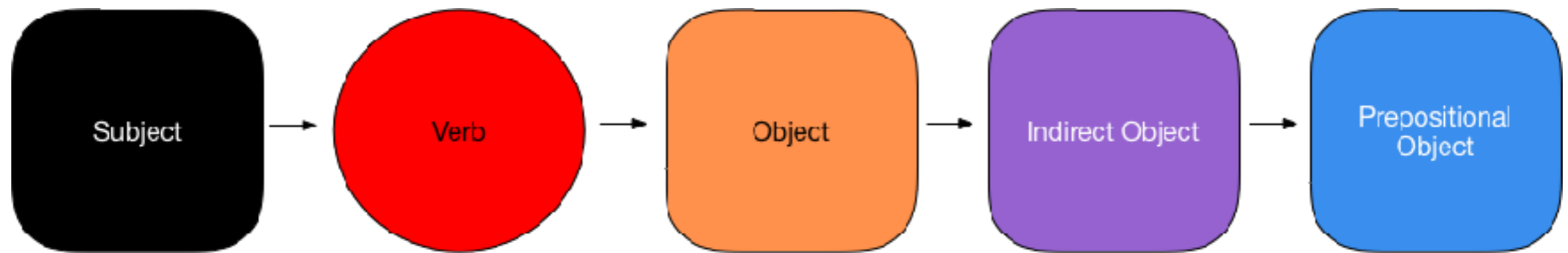
- `log.Println("did stuff")`
- `log.WithFields(log.Fields{  
"animal": "walrus",  
"size": 10,  
}).Info("A group of walrus emerges from the  
ocean")`



# EVENT GRAMMAR

- Model events with grammar - components from language
- Use for service logging, events, and tracing
- Enrich data
- Improve queries and analysis

# GRAMMAR COMPONENTS



# SUBJECT

- Person, place or thing
- Performs the action
- **Parker** rolls the ball to Kennedy on the ground.
- **CheckoutSvc** updates InventorySvc t-shirt balance in database.



# VERB

- Conveys action
- Parker **rolls** the ball to Kennedy on the ground.
- CheckoutSvc **updates** InventorySvc t-shirt balance in database.

# OBJECT

- Receiver of action
- Parker rolls the **ball** to Kennedy on the ground.
- CheckoutSvc updates **InventorySvc** t-shirt balance in database.

# INDIRECT OBJECT

- Receiver of the action
- Occurs in addition to direct object
- Parker rolls the ball to **Kennedy** on the ground.
- CheckoutSvc updates InventorySvc **t-shirt balance** in database.

# PREPOSITIONAL OBJECT

- Object of the preposition (at, for, etc.)
- Parker rolls the ball to Kennedy on the **ground**.
- CheckoutSvc updates InventorySvc t-shirt balance in **database**.

# GRAMMAR-LOG

```
logger := logger.NewLogger("commander",  
                             os.Stdout)
```

```
logger.Debug()  
    .Verb("writes")  
    .Object("event")  
    .IndirectObject("kafka")  
    .Log()
```

# GRAMMAR-LOG

```
func (e Event) Log(grams ...interface{}) {
```

```
...
```

```
}
```

```
logger.Debug().Log("writes", "event", "kafka")
```

# DISTRIBUTED TRACING

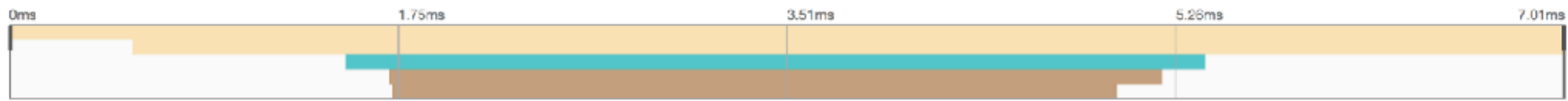


# OPENTRACING

## nginx: /commander

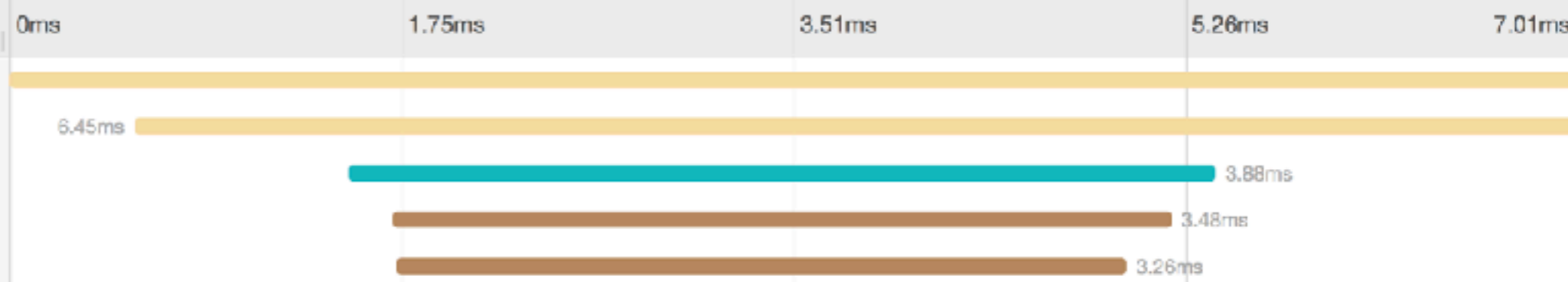


Trace Start: **May 2, 2018 11:16 PM** | Duration: **7.01ms** | Services: **3** | Depth: **5** | Total Spans: **5**



### Service & Operation

- nginx /commander
  - nginx /commander
    - commander createCommand
      - processor CREDIT\_ACCOUNT
        - processor persistTransaction





# OPENTRACING

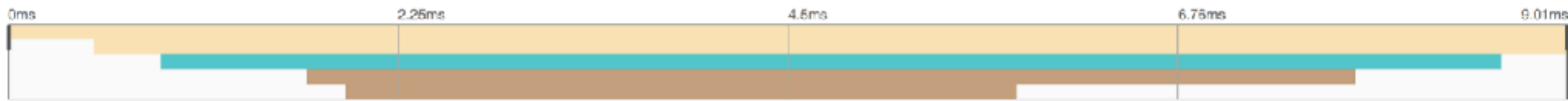
```
span := tracer.StartSpan("createCommand")
m := make(map[string]string)
m["caller"] = "createCommand"
carrier := opentracing.TextMapCarrier(m)
err := tracer.Inject(span.Context(), opentracing.TextMap, carrier)
.
.
.
grammar := actionToGrammar(cmdParam.Action)
logger.Debug().Verb(grammar.verb).Object(grammar.object).Indirect
Object(cmdParam.Data).Log()
span.SetTag(FieldTypeVerb, grammar.verb)
span.SetTag(FieldTypeObject, grammar.object)
span.SetTag(FieldTypeIndirectObject, cmdParam.Data)
```

# OPENTRACING TAGS

## nginx: /commander

🔍 Search... View Options ▾

Trace Start: May 2, 2018 11:16 PM | Duration: 9.01ms | Services: 3 | Depth: 5 | Total Spans: 5



Service & Operation	0ms	2.25ms	4.5ms	6.75ms	9.01ms
nginx /commander	[Yellow bar]				
nginx /commander	8.51ms	[Yellow bar]			
commander createCommand	7.74ms	[Teal bar]			
<b>createCommand</b>	Service: <b>commander</b>   Duration: <b>7.74ms</b>   Start Time: <b>0.88ms</b>				
> Tags:	verb = CREDIT   object = ACCOUNT   indirect_object = 32				
> Process:	hostname = 6bc8901c7682   ip = 172.22.0.6   jaeger.version = Go-2.13.0				
	SpanID: 717d0ff67f2e510f				
processor CREDIT_ACCOUNT	6.06ms	[Brown bar]			
<b>CREDIT_ACCOUNT</b>	Service: <b>processor</b>   Duration: <b>6.06ms</b>   Start Time: <b>1.72ms</b>				
> Tags:	verb = CREDIT   object = ACCOUNT   indirect_object = 32				
> Process:	hostname = 8ed82d8f88c1   ip = 172.22.0.8   jaeger.version = Go-2.13.0				
	SpanID: 519db20880bb333e				
processor persistTransaction	[Brown bar] 3.89ms				

# BEYOND

- Get context through OpenTracing
- Take advantage of OpenTracing tags
- Event Grammar bridges gap between OpenTracing and service logging and events

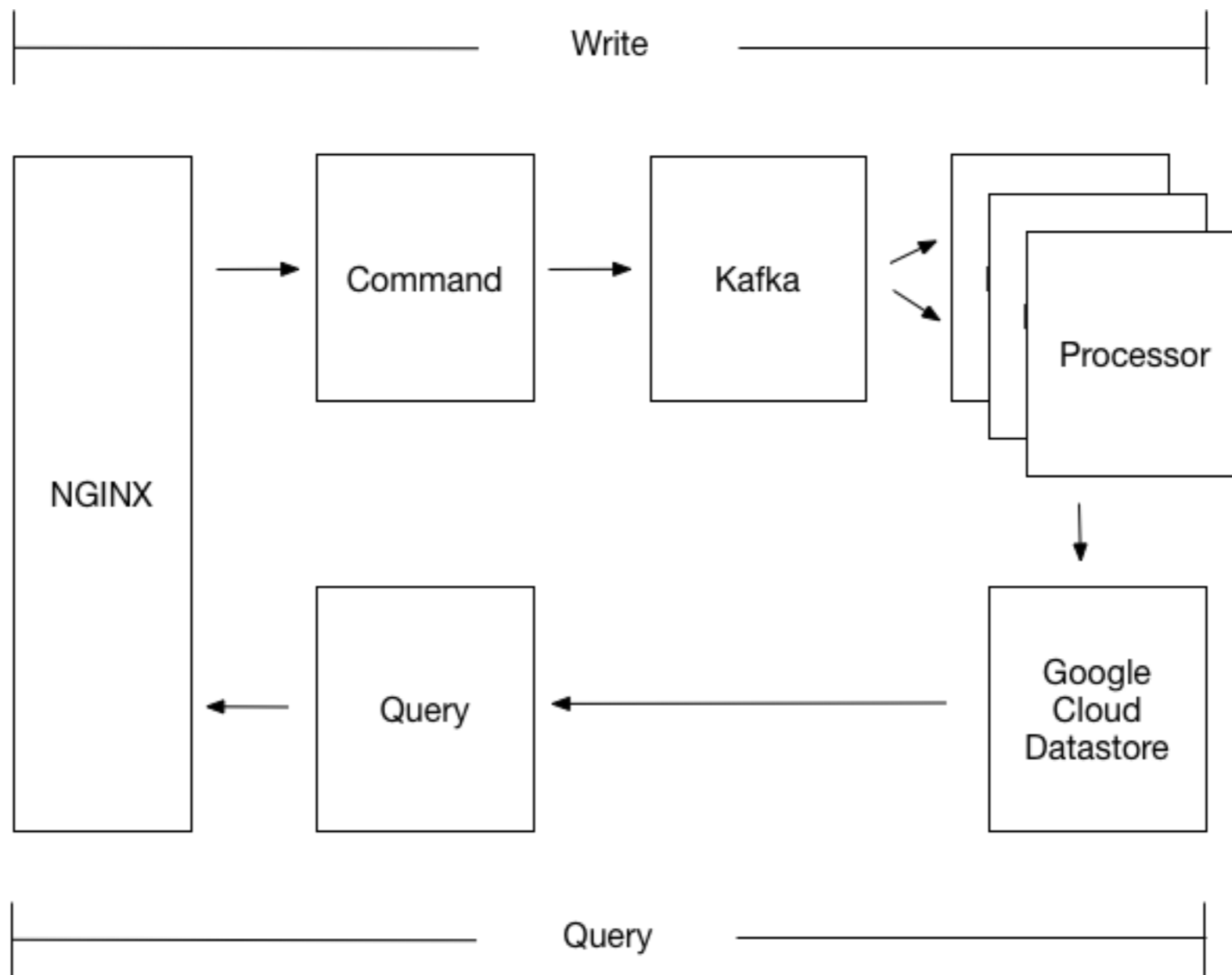
# BEYOND

- Semantics for application developers & operations
- Data structured to be analytic friendly
- Reduce cardinality

# EVENT SOURCING

- Services publish/subscribe to state change events
- Audit log & overall application state
- Ability to replay events

# CQRS / EVENT SOURCING



DEMO

# SUMMARY

- Structured logging with a defined event grammar
- Reduce cardinality
- Improve analysis
- Enhance ML



# SUMMARY

- Enrich data
- Bridge gap between application developers and operations
- Tie OpenTracing to events and logging

THANK YOU

# REFERENCES

- Towards Universal Event Analytics - Building an Event Grammar, Snowplow, <https://snowplowanalytics.com/blog/2013/08/12/towards-universal-event-analytics-building-an-event-grammar/>
- Logrus, <https://github.com/sirupsen/logrus>.
- OK Log: Distributed and Coordination-Free Logging, Peter Bourgon. <https://www.youtube.com/watch?v=gVWK2eyZ-sc>
- OpenTracing, [opentracing.io](http://opentracing.io).
- Jaeger Tracing, [jaegertracing.io](http://jaegertracing.io).
- <https://github.com/capitalone/cqrs-manager-for-distributed-reactive-services/blob/master/doc/rationale.md>