One Log

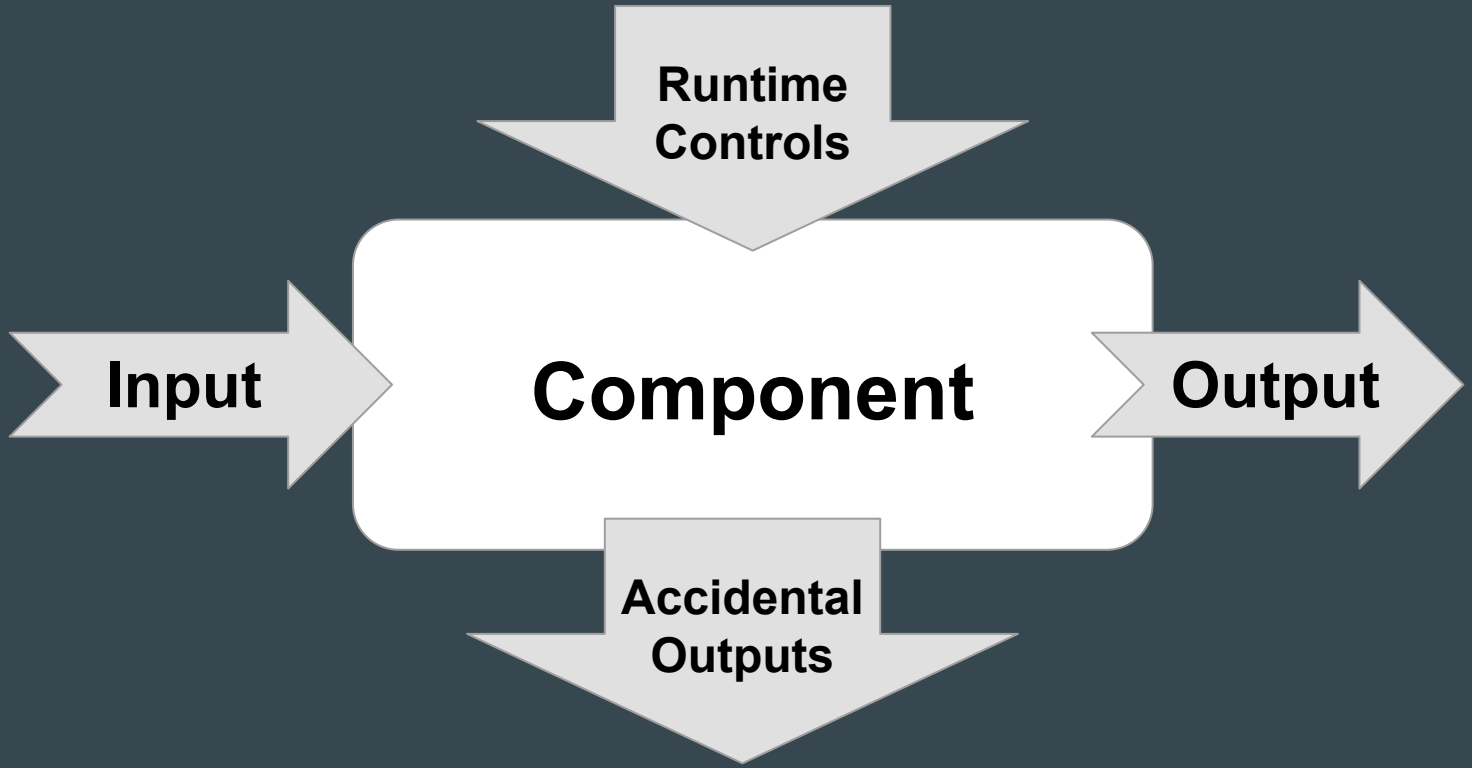
...

Arnaud Bailly - @dr_c0d3
Yann Schwartz - @abolibibelot

Agenda

- Log as the System's State
- Shaping the Log
- Log as a Language

Functional versus Operational Plane



What We Believe

Input

\+ Output

\+ **Accidental Output**

make up **the true model of a running system**

Shaping the Log

...

Demo time

```
[Warning] [14:42:58.133 21 Sep 2018 UTC] [Main.app#27] [ThreadId 7] Starting application...
[Debug]   [14:42:59.135 21 Sep 2018 UTC] [Main.example#22] [ThreadId 7] app: First message...
[Info]    [14:42:59.137 21 Sep 2018 UTC] [Main.example#23] [ThreadId 7] app: Second message...
[Error]   [14:42:59.138 21 Sep 2018 UTC] [Main.exceptionL#47] [ThreadId 7] app: ExampleException
[Info]    [14:42:59.139 21 Sep 2018 UTC] [Main.app#32] [ThreadId 7] app: Application finished...
[Warning] [14:42:59.141 21 Sep 2018 UTC] [Main.app#27] Starting application...
[Debug]   [14:43:00.143 21 Sep 2018 UTC] [Main.example#22] app: First message...
[Info]    [14:43:00.144 21 Sep 2018 UTC] [Main.example#23] app: Second message...
[Error]   [14:43:00.145 21 Sep 2018 UTC] [Main.exceptionL#47] app: ExampleException
[Info]    [14:43:00.145 21 Sep 2018 UTC] [Main.app#32] app: Application finished...
```

The way to One Log

- Log typed messages, not strings
  - Structured logs
- Exactly Once Logging
  - Vanilla Logs, metrics, traces, system info all treated as log messages
- Think system, not application
  - Aggregate all the logs
- Don’t sample first
- Transports and Materialized Views

Event Sourcing

"As above, so Below"

Event Sourcing is a "natural" fit to One Log

Conceptually: It structures the whole system around the concept of a stream of events

Physically: Append-Only log storage is the core stream of the aggregated view of One Log

Log as a Language

...

- Words
- Grammars
- Language

The Words of Logs

...

Log events as Words of our Language

- Counting
- Grouping
- Monitoring and dashboards

322	docker	yyyy/MM/dd HH:mm:ss [ERR] consul: "Catalog.Register" RPC failed to server 10.XXX.XXX.XXX:8300: rpc error making call: rpc error making call: failed inserting node: Error while renaming Node ID: "*": Node name * is reserved by node * with name *
290	docker	yyyy/MM/dd HH:mm:ss [WARN] agent: Syncing service "vault:10.XXX.XXX.XXX:8250" failed. rpc error making call: rpc error making call: failed inserting node: Error while renaming Node ID: "*": Node name * is reserved by node * with name *
256	docker	yyyy/MM/dd HH:mm:ss [ERR] agent: failed to sync changes: rpc error making call: rpc error making call: failed inserting node: Error while renaming Node ID: "*": Node name * is reserved by node * with name *
228	docker	yyyy/MM/dd HH:mm:ss [INFO] agent: Synced service "vault:XXX.XXX.XXX.XXX:8250"

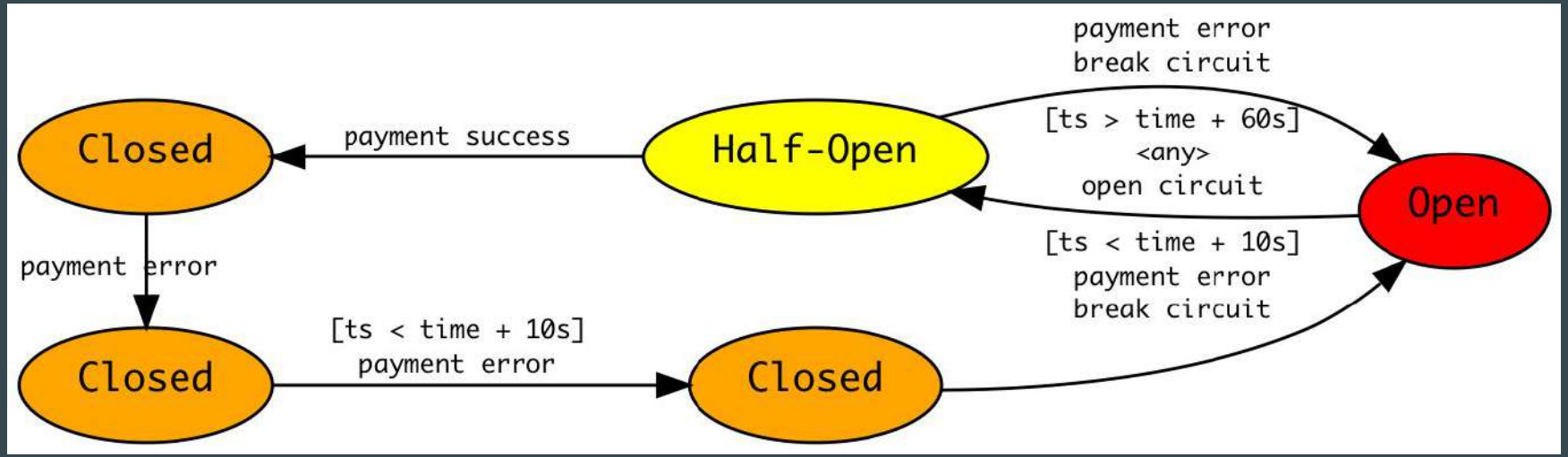
The Grammar of Logs

...

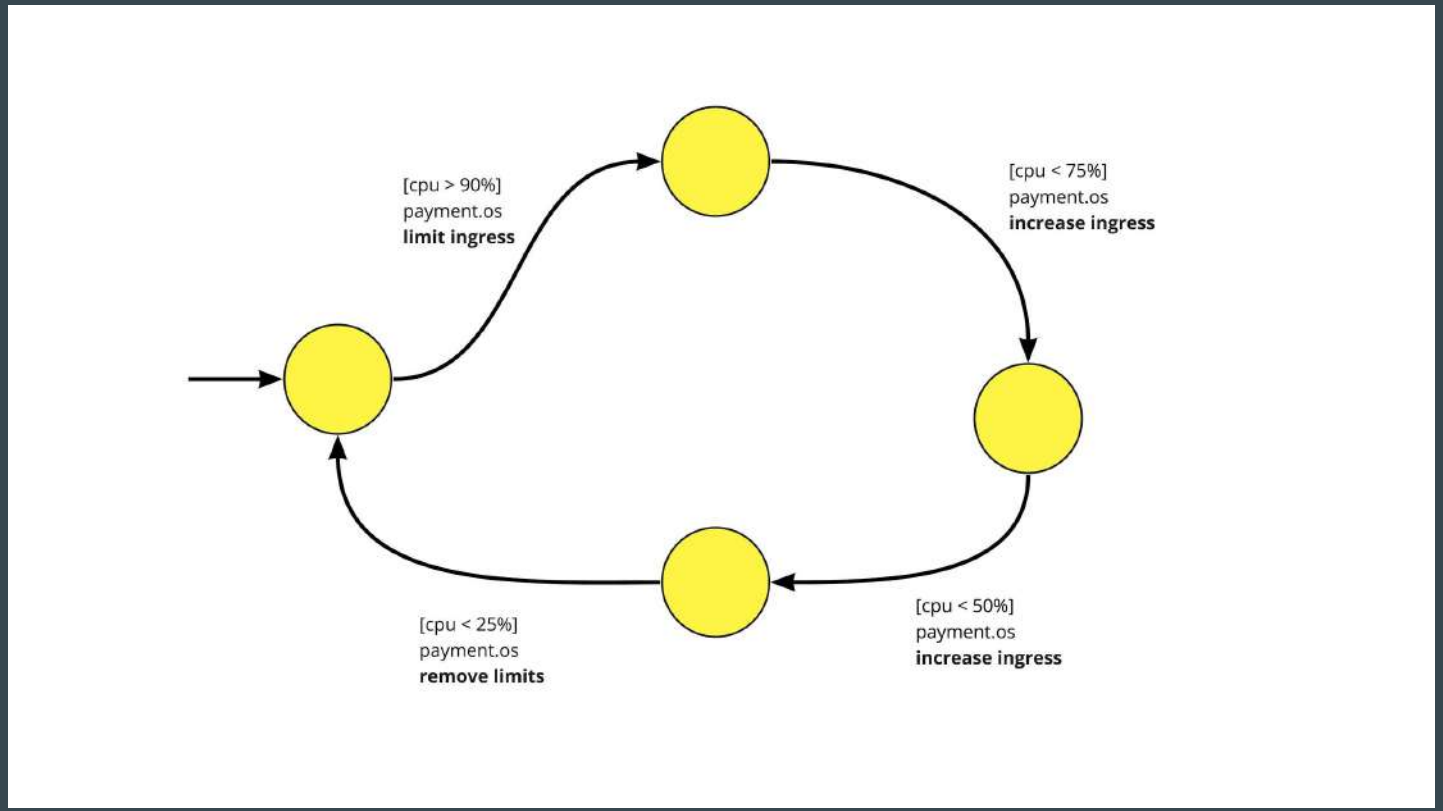
Relation between Words

{"log":{"message":{"tag":"Error","reason":"InvalidPayment"},"timestamp":"2018-11-07T20:30:57.930068Z"},"node":"pet-store-server","timestamp":"2018-11-07T20:30:57.930164Z"}
{"log":{"message":{"tag":"Add","pet":{"petType":"Rabbit","petPrice":10,"petName":"Bella"}},"timestamp":"2018-11-07T20:33:42.151195Z"},"node":"pet-store-server","timestamp":"2018-11-07T20:33:42.151343Z"}
{"log":{"message":{"tag":"PetAdded","pet":{"petType":"Rabbit","petPrice":10,"petName":"Bella"}},"timestamp":"2018-11-07T20:33:42.152093Z"},"node":"pet-store-server","timestamp":"2018-11-07T20:33:42.152216Z"}
{"log":{"message":{"tag":"Error","reason":"InvalidPayment"},"timestamp":"2018-11-07T20:34:57.930068Z"},"node":"pet-store-server","timestamp":"2018-11-07T20:34:57.930164Z"}
{"log":{"message":{"tag":"Error","reason":"InvalidPayment"},"timestamp":"2018-11-07T20:34:59.930068Z"},"node":"pet-store-server","timestamp":"2018-11-07T20:34:58.930164Z"}
{"log":{"message":{"id":2,"paymentOk":false},"timestamp":"2018-11-07T20:33:42.071468Z"},"node":"pet-store-server","timestamp":"2018-11-07T20:33:42.072911Z"}
{"log":{"tag":"Broken","breakTime":"2018-11-07T20:33:42.072912Z"},"node":"circuit-breaker","timestamp":"2018-11-07T20:33:42.072912Z"}

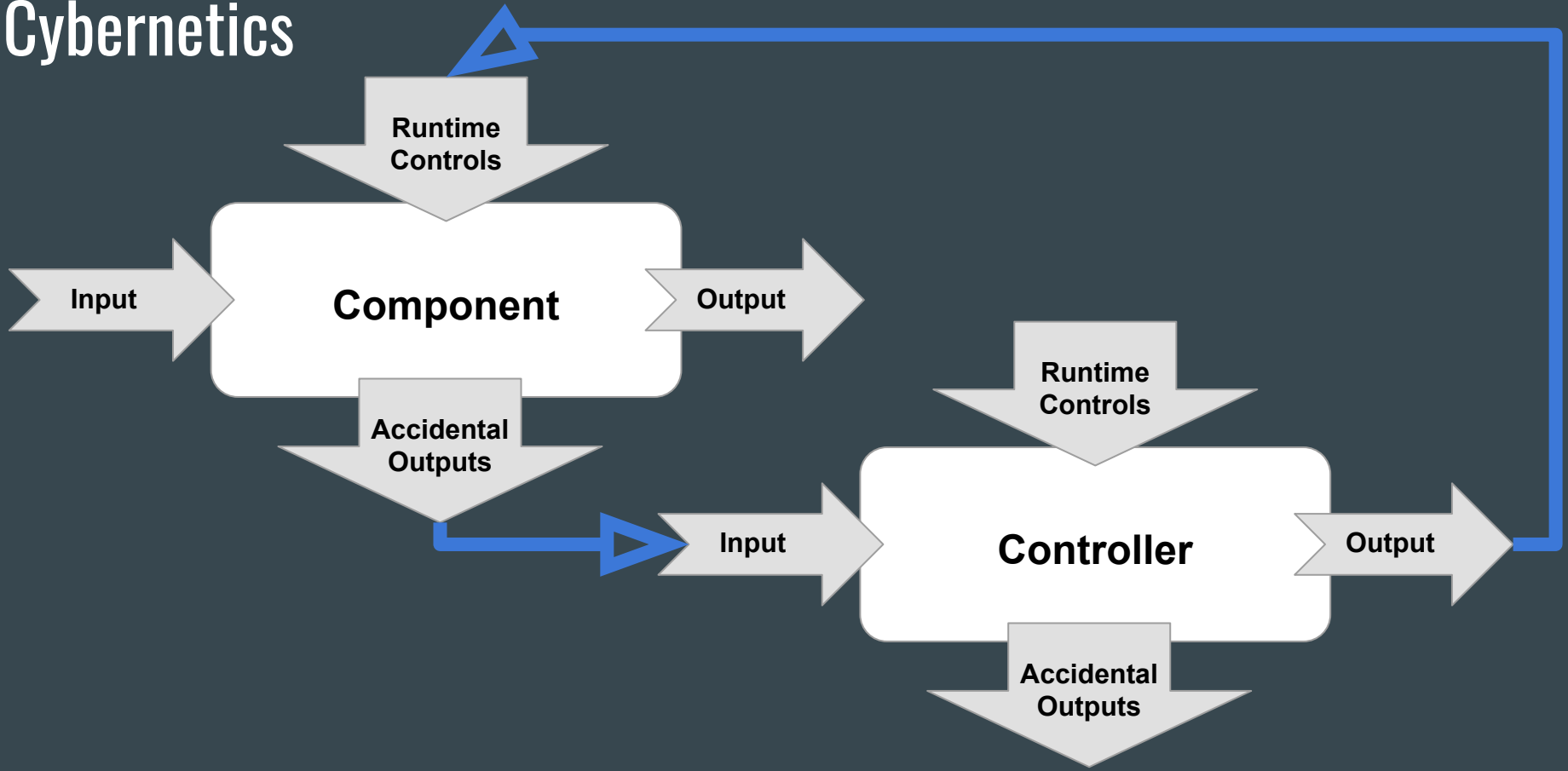
Circuit Breaker



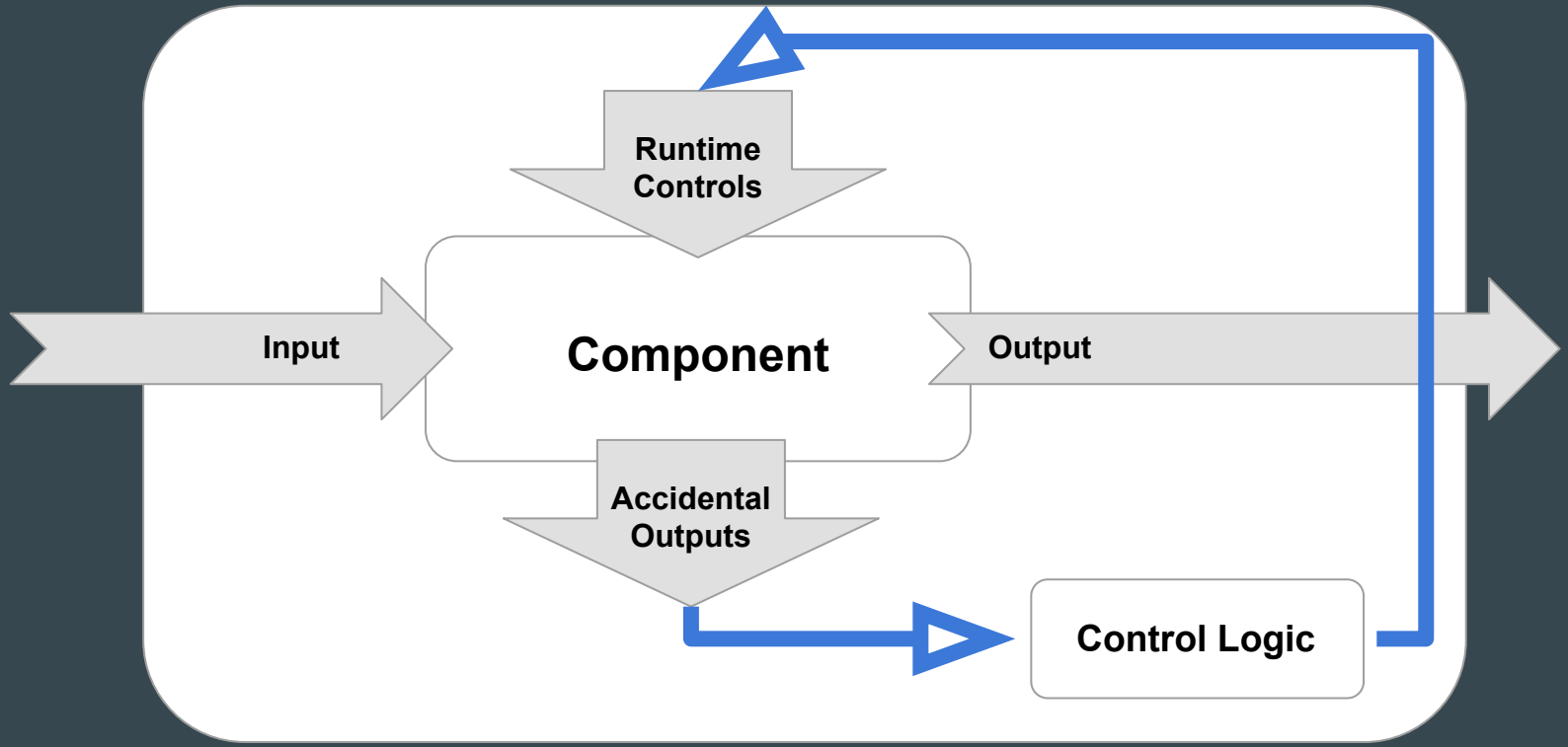
Back-Pressure



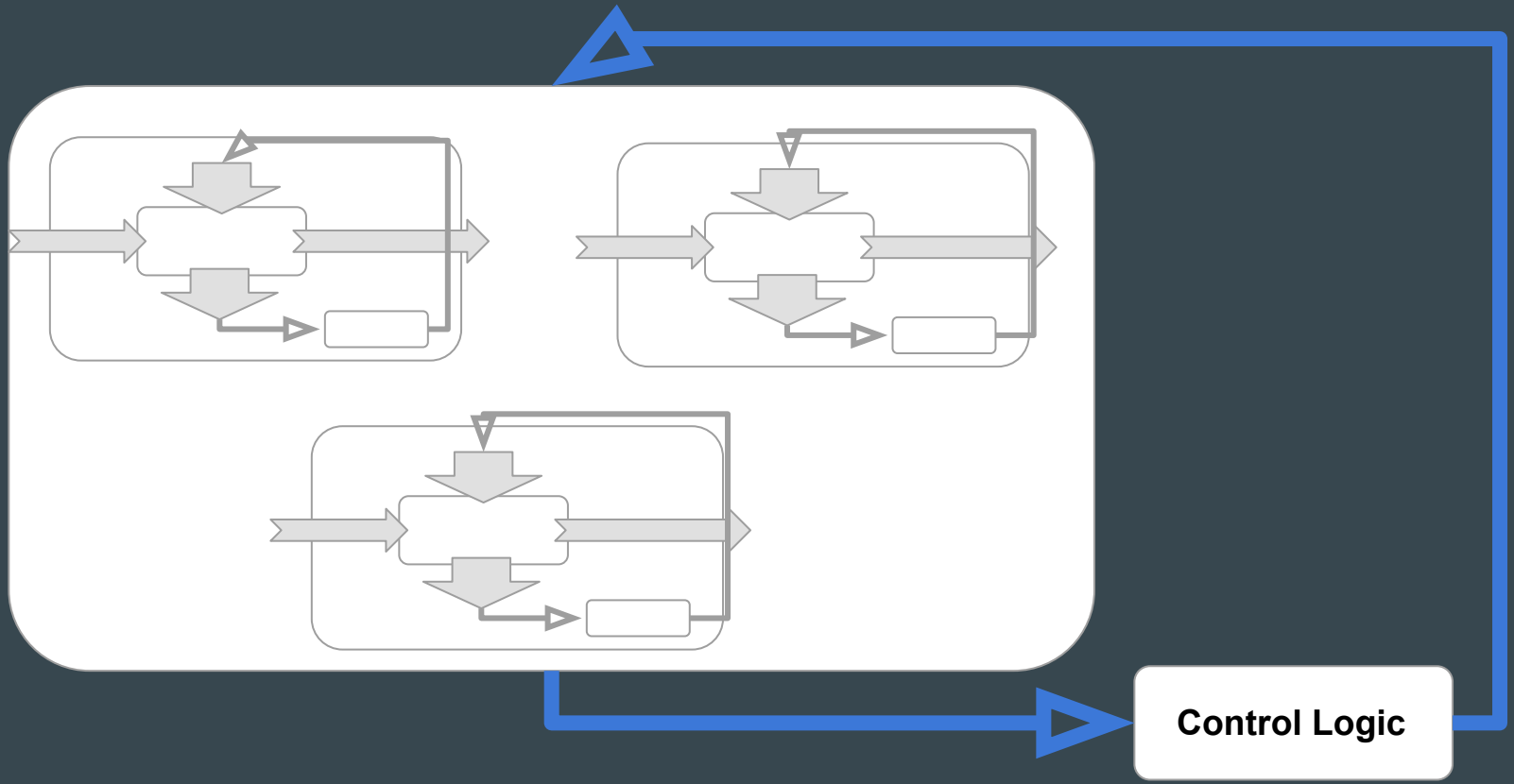
Cybernetics



Cybernetics



Cybernetics - Composition



The Language of Logs

...

Log streams as sentences

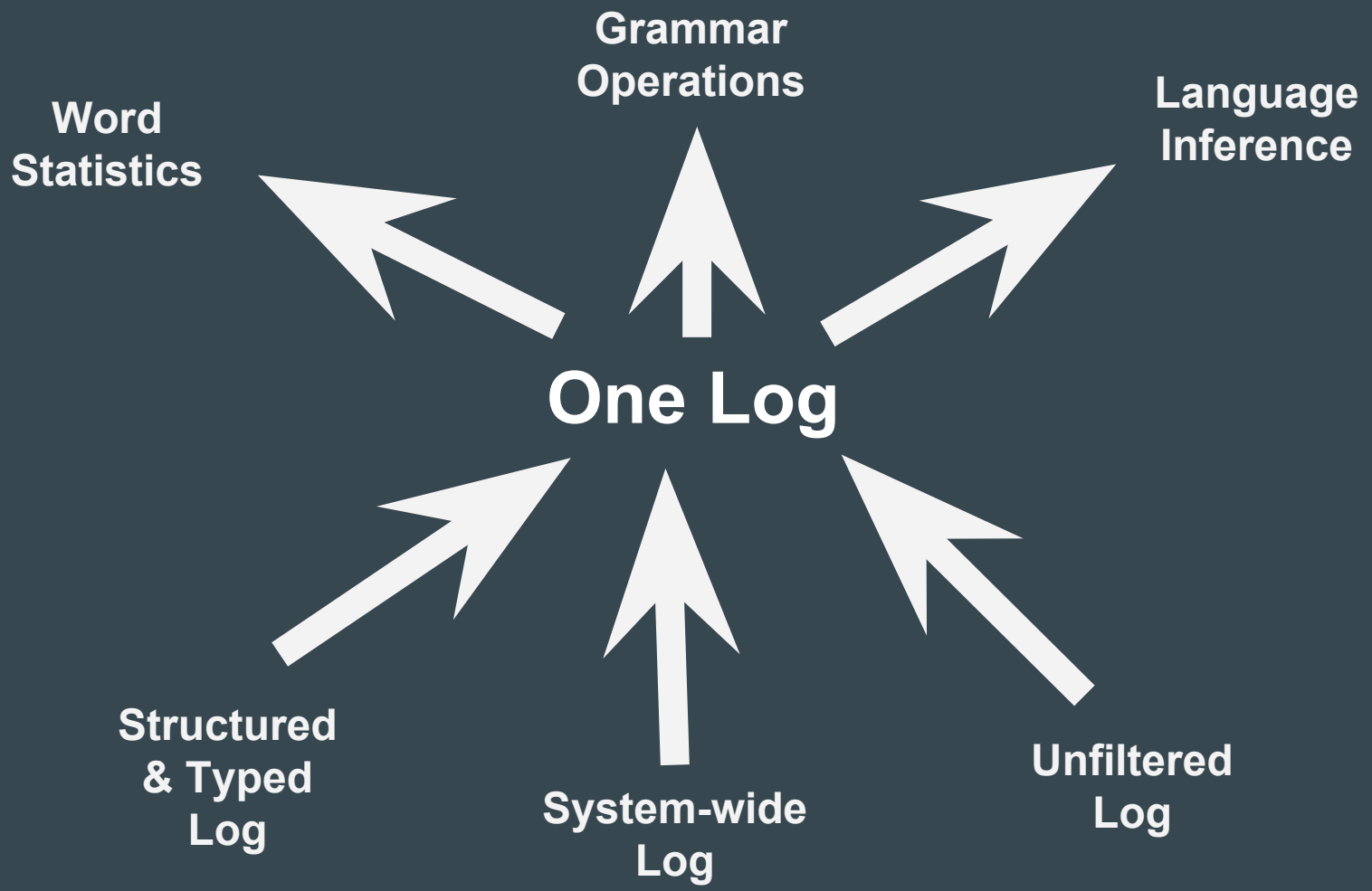
Log productions as NLP problems

- Log2Vec:
  - Similitudes in a distributed system

- RNN: a model of likely productions
  - Low perplexity : compress production
  - High perplexity: detect outliers

Parting Words

...


Grammar
Operations
Word
Statistics
Language
Inference
One Log
Structured
& Typed
Log
System-wide
Log
Unfiltered
Log

Mechanical Antipathy?

- Don't use JSON
- Structured Log is compression
- Aggregate from call site
- Append only, materialized views
- Standardize on structure, vary your storage
- Standardize on kafka for transport / intermediate storage
- Push versus pull on monitoring?
  - Maintain dual systems
  - Generate logs from monitoring events

Related and inspirational work

Riemann

Effective Log (Osterhout)

Cindy Sridharan(Health Checks and Graceful Degradation in Distributed Systems)

Grammatical Inference (Colin de la Higuera)

Code

https://github.com/aleryo/one-log

Contains a simplified system and ways to exploit logs

Work-in-progress

Future Work

Expressing controllers as *Tree Transducers/Term Rewriting*

→ A DSL to define controllers "easily"

Logs as a NLP problem

Replay & Time-travel

With an Event Sourced systems, the log is the journal of the changes to the state of the system

We can reconstruct the state from the logs

Replay is a powerful time-travelling and troubleshooting feature afforded by One Log