Data to Decisions in a Globally Distributed Computing Environment

Larry Underhill, Senior Engineering Manager
Custom Government Division
Akamai Technologies, Inc

Agenda

- Akamai - The Elevator Pitch
- Data to Decision Enablement
  - Culture
  - Architectures
  - Patterns
- Q&A

Akamai – The Elevator Pitch

Q: What's Akamai

A: If you've ever shopped online, downloaded music, watched a web video, or connected to work remotely, you've probably used Akamai's Cloud Platform

Akamai optimizes Web experiences, helping businesses securely and reliably connect users to any experience, on any device, anywhere. And faster than ever.

Akamai Facts and Figures

- 127,000 servers
  - in 81 countries
  - within 1,150 networks
- Delivers between 15-30% of all Web traffic
- Daily peak traffic reaching more than 10 Terabits per second
- Delivers over 2 trillion daily Internet interactions

CHMNI Topic Excerpt

In the twenty-first century, critical decisionmaking will increasingly rely on collaboration among humans, machines and information …

Moreover, the explosion in the means for collecting, categorizing and analyzing data is transforming the decisionmaking process …

… include findings on (1) key research goals in several enabling subfields as they relate to the abilities of humans, machines and networks to share the cognitive load to make decisions …

(4) systems-integration challenges to improving data-to-decision capabilities

Topic Restatement and Refinement

- In an environment of ever-increasing complexity, how do we build and operate systems that naturally enable good decision making?
  - Environment = Humans, Machines, & Networks
    - Pretty much a given, no?
  - Build + Operate = Service
    - I don’t write software. I deliver a service
- In this light, Akamai’s platform delivers multiple services worthy of analysis.
  - But today we will further narrow our focus.

Big Data ~ Big Decisions :: Real Time ~ Small Teams

My Bias

- Services are what matters
  - A physician is not a health care service
- “Operators” deliver services
  - Extending rich, expressive, extensible controls into the operations realm is *the essential element* for creating better organizational decisions
- Data to decision enablement is all about culture and architectures

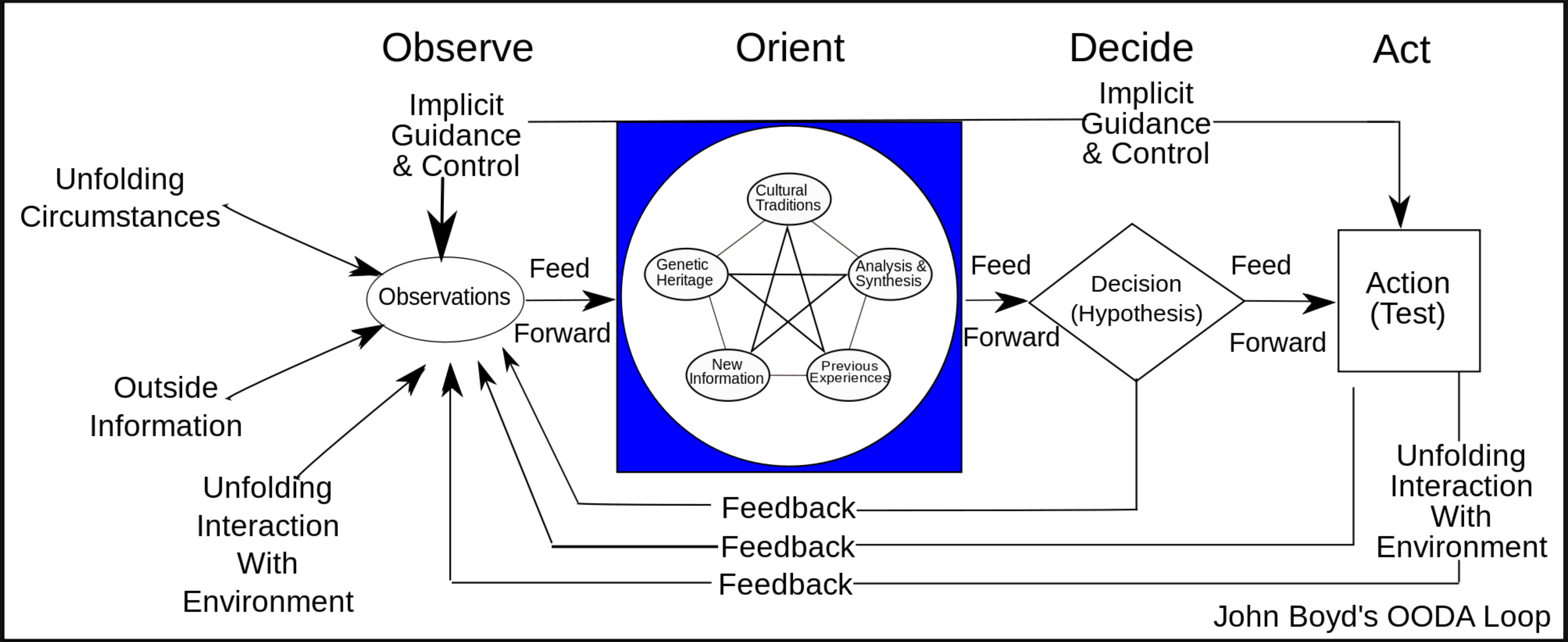
Culture

- Complexity
  - Embrace it; do not contain it
  - Humans are ill equipped to handle “today’s complexity?”
    - Take a walk in the woods
- Ambiguity
  - Perfect prior knowledge does not exist when designing and implementing systems
  - *Over time*, seek to reduce ambiguity
- Distributed Control & Execution
  - Generates Superior Domain Knowledge

Akamai Architecture Principles

- Ensure significant redundancy in all systems to facilitate failover
- Use software logic to provide message reliability
- Use distributed control for coordination
- Fail cleanly and restart
- Zoning
- The network should notice and quarantine faults

OODA Loop

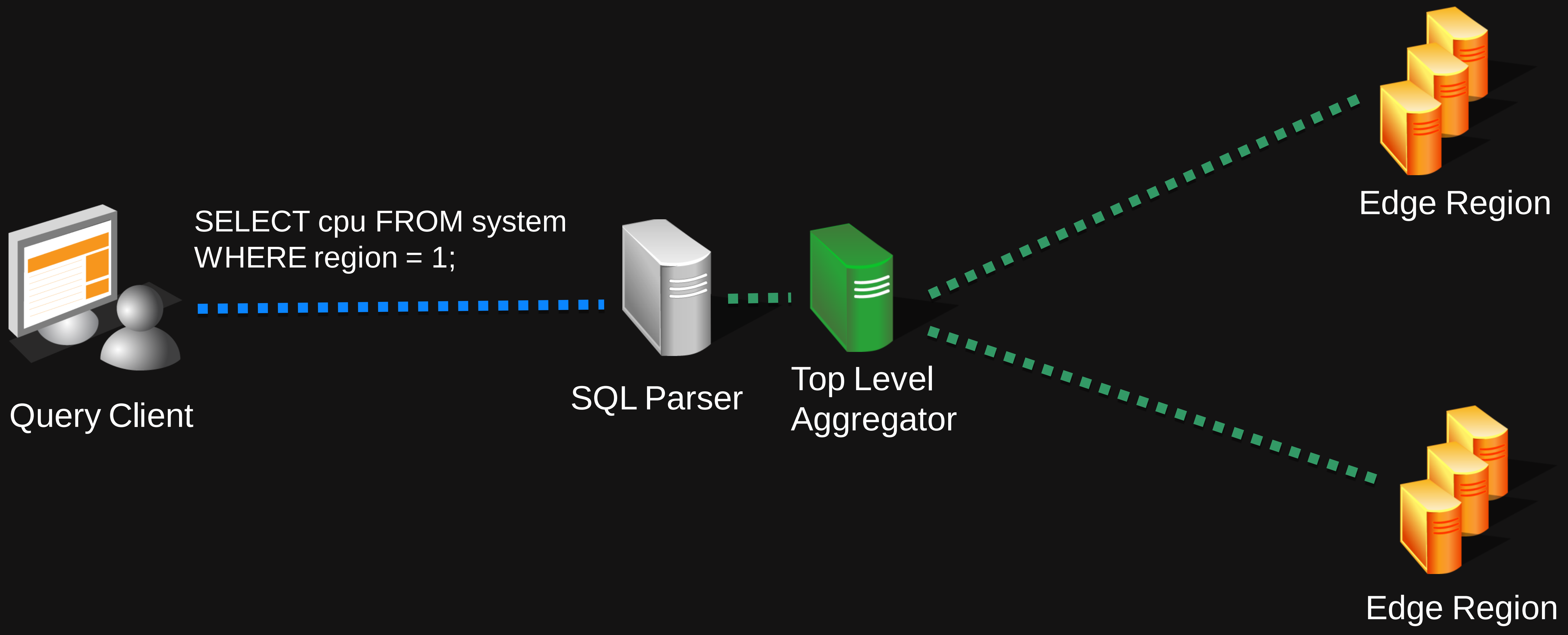


John Boyd's OODA Loop

OODA Architecture Implications

- To execute an OODA loop you need messaging
  - Monitoring = “Observation”
  - Command & Control = “Act”
- One Size Fits All messaging architecture does not work.
  - Employ multiple systems tailored to the message
    - Time Sensitivity, Reliability, Persistence, etc.
  - This talk will highlight specific features of Akamai’s Query monitoring system.

Query Diagram



SELECT cpu FROM system
WHERE region = 1;

Attractive Query Properties

- Each service running on a node can expose voluminous metrics without centralized control/coordination.
  - Developers can arbitrarily modify the global schema?!
    - Do it. Embrace complexity.
- Ad hoc Socratic Dialogues

Decision Enablement Patterns

- Partitions
- Primitives
- Abstractions

Partitions

The rest? Our focus is on real-time decisions. Can we partition by time to make an offline analysis?

Also. Sometimes problems are just hard.

Focus on Aberrations

Partitions

- Don’t make a decision!
  - Autonomic
    - Self managing
    - Adaptable to unpredictable changes
    - Hides intrinsic complexity to operators and end users
  - Akamai’s global load balancing is an example.
    - OODA Loop on “Auto Pilot”
      - Observe: load reporting
      - Orient: algorithmic + operator defined tuning
      - Decide: load balancing decision published
      - Act: request-by-request load balanced

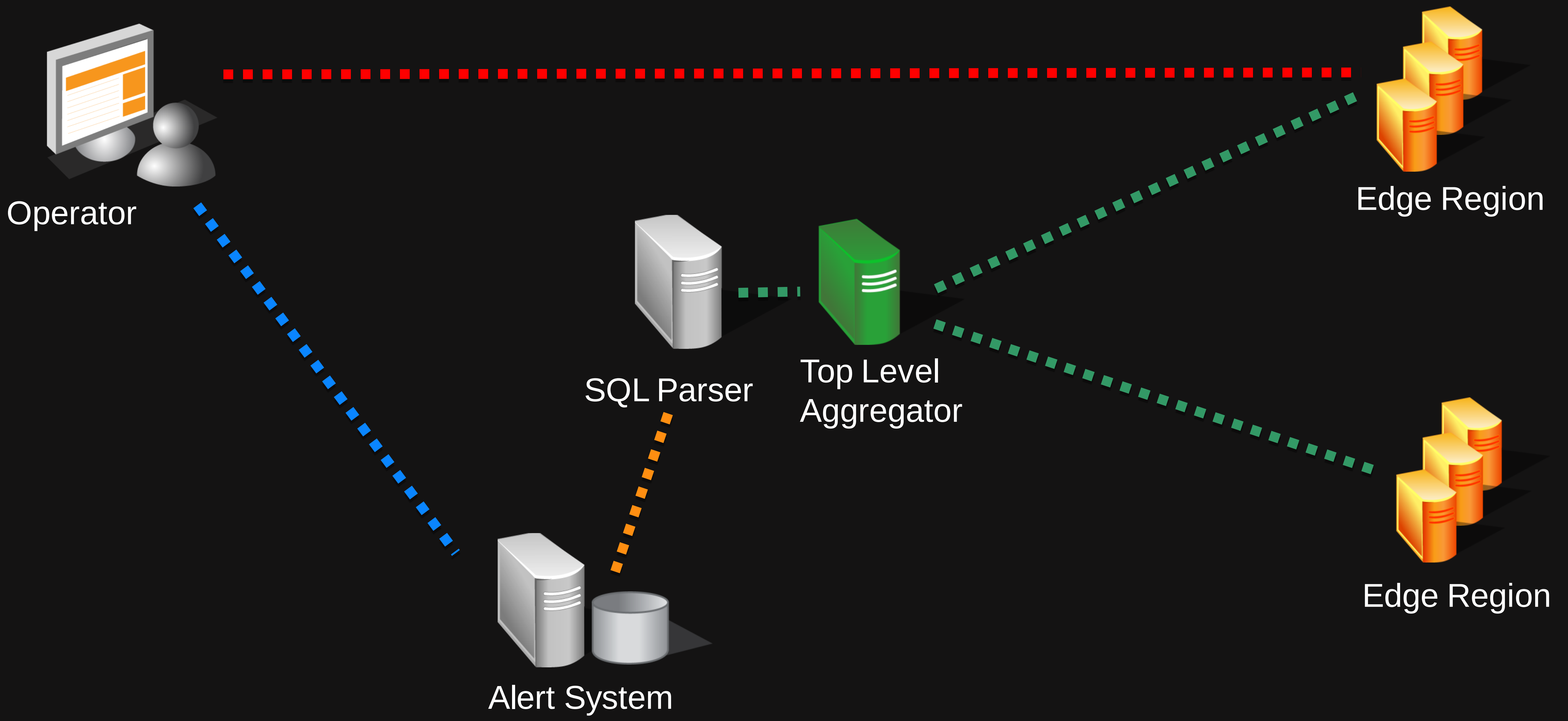
Partitions

- If we have to make a decision, can we just make a ~~dumb~~, ~~simple~~, reflexive one?
  - Recovery Oriented Computing
    - One of our key architectural principles, "Fail cleanly and restart"
    - System behaviors are often emergent, not deterministic
      - Heisenbugs
      - Windshield Wipers
    - System reliability can, in fact, be increased with *preemptive* component restarts.
      - Ghost rolls and memlimits

Partitions

- Focus on pathology
  - Assuming sufficient telemetry…
  - Craft a statement that returns “true” for a pathological condition.
  - Consistently evaluate the current telemetry with this statement
  - Raise an alert if the statement returns true

Alert Management System



Alert SQL Example

```
SELECT
  machineip ip,
  mountp mnt ,
  bavail*bsize free_space,
  (100*bavail)/blocks pct
FROM
  filesystem a
WHERE
  blocks > 0 and
  (100*bavail)/blocks < 3;

         ip  mnt  free_space pct
------------ ---- ----------- ---
10.123.123.1 /var 150,179,840   2
10.123.123.7 /var  72,216,576   1
```

Alert System Properties

- Content Management System for Alert Definitions and Alert Procedures
- EASY TO USE
- Enables further partitioning
  - Time: Alerts are handled asynchronously
  - Specialization: Engineers and Operators own the alerts for their components
  - Delegation: Procedures allow a non-expert to resolve issues
- Fully Expresses an OODA Loop cycle
- Integrated with other systems to enable further decision making
  - Examples: Issue tracking and Alert History

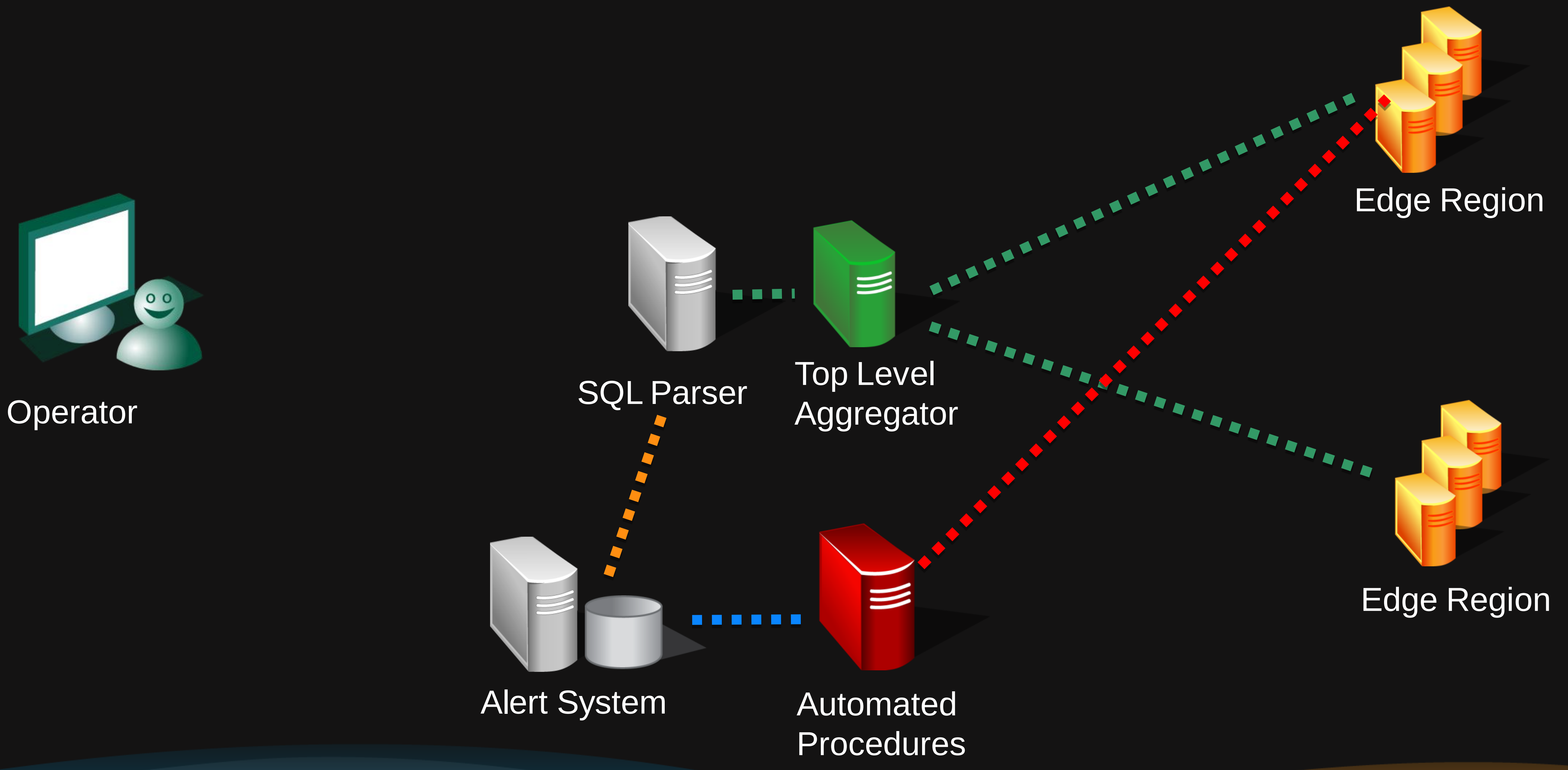
Alert System Culture

- Who is responsible for authoring and tuning alerts?
  - Engineers and Operators
  - Engineers start with an ambiguous model of how their components will work in production…
  - And then reality intrudes

Towards autonomics

- Can my alert management be autonomic? Should it?
  - To a degree…
  - You can enable High RPM OODA Loops with automation
    - Cultural side note: embrace automation

Alert Management System Automated Procedures



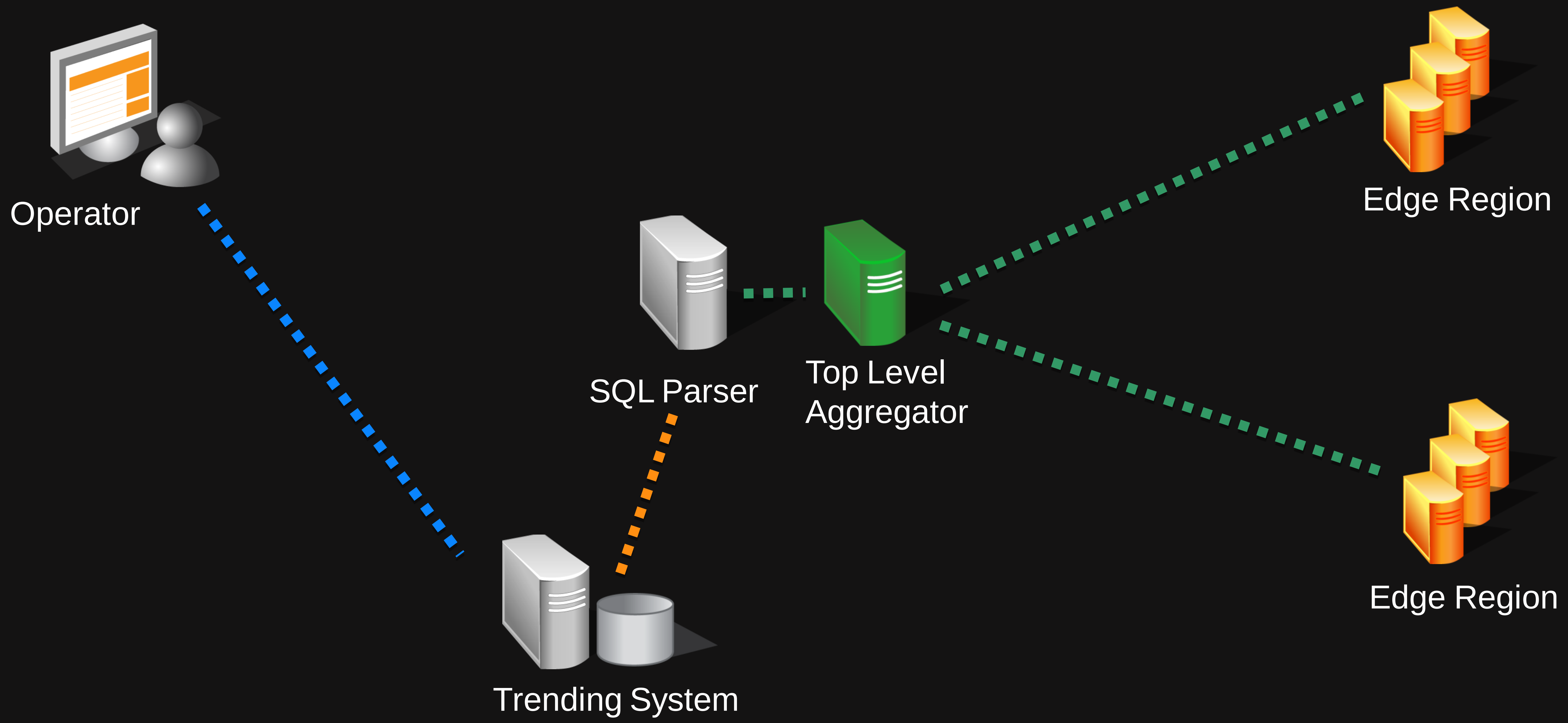
AMS as Business Continuity

- Assertion: the resulting corpus of alert definitions and procedures is a literal distillation of your Institutional Knowledge
  - The Alert Management System turns the OODA Loop into a Flywheel

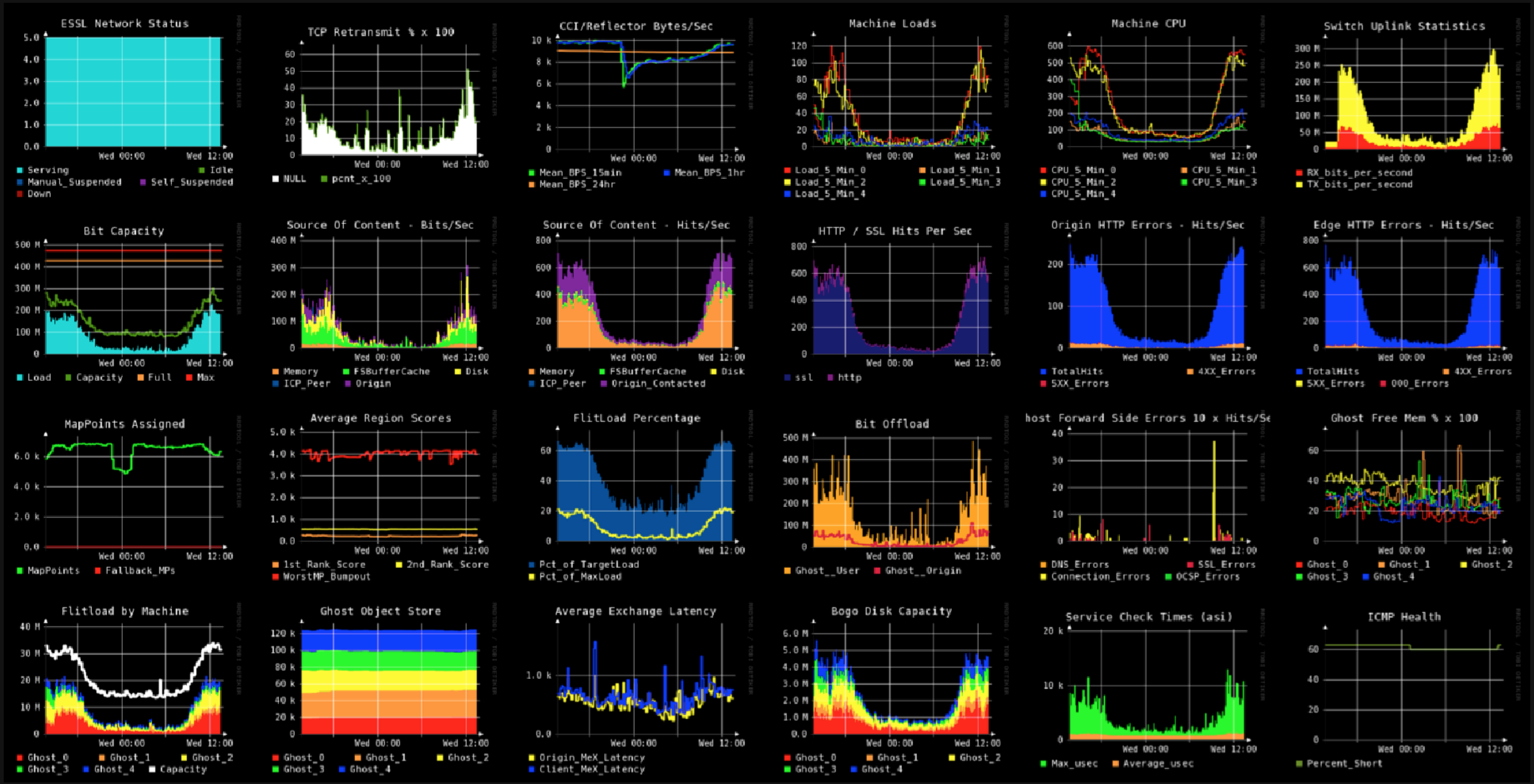
Partitions

- Further partition the decision making process by focusing on aberrations.
- Aberrant behavior is contextual. Our context is time.
- We leverage our Query system to build a collection and visualization system for time series data
- Humans have excellent pattern recognition software
  - If you build a tool that makes it easy to collect and visualize complex information…
    - People will use it!

Trending System



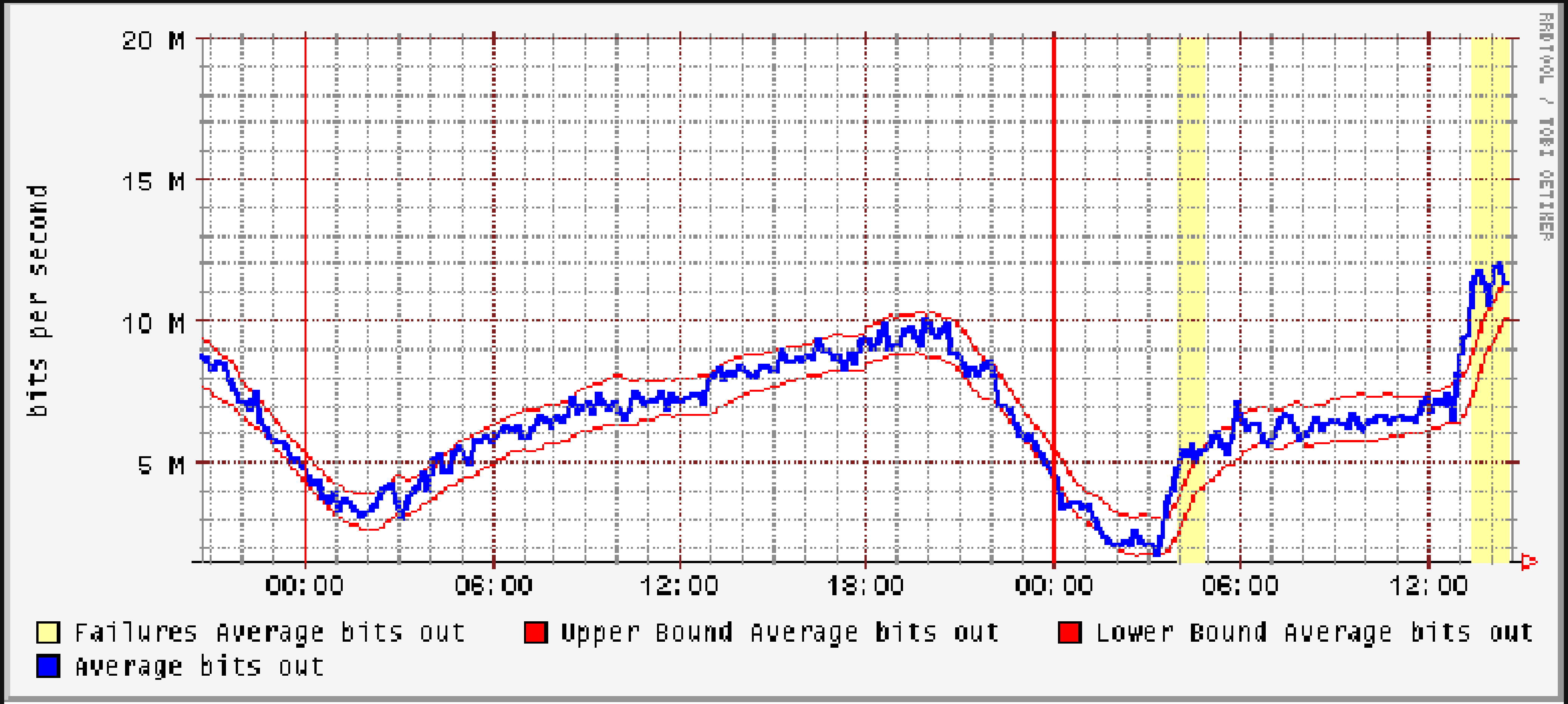
Trending Dashboards



Trending Criticisms

- Humans must participate in the OODA loop.
  - Orient -> Decide -> Act
    - Trending is not *actionable*
- True. But contextual value from trending is still useful
  - Can help surface previously undiscovered pathological conditions
- One more technique up our sleeve…

Holt-Winters Forecasting



Trending and Alerting Integration

- Complete the circle.
  - Create a trending definition and collect data
  - At each data collection iteration, compute confidence bands based on seasonality.
  - Publish data points and confidence bands for each trending definition into Query.
  - Create alert to catch values falling outside confidence bands.

Tooling Lifecycle

- Want incredible data-to-decision tools? We have noticed the following pattern in tool creation:
  - Scratch an itch
  - Share
  - Get excited
  - Adoption
  - Polishing
  - Recognition
  - "Properly" Resourced
- Develop architectures and cultures that enable and encourage these behaviors throughout the organization.

Wrap Up

- Build and operate services
- Embrace complexity and ambiguity
- Have a bias for strong operations
- OODA
- Architect services to include autonomic and reflexive responses
- Address pathology and aberrations

Thanks!

Q&A