
AWS
re:Invent

BLD205

Monitoring production services at Amazon

David Yanacek

Sr. Principal Engineer, AWS Lambda

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

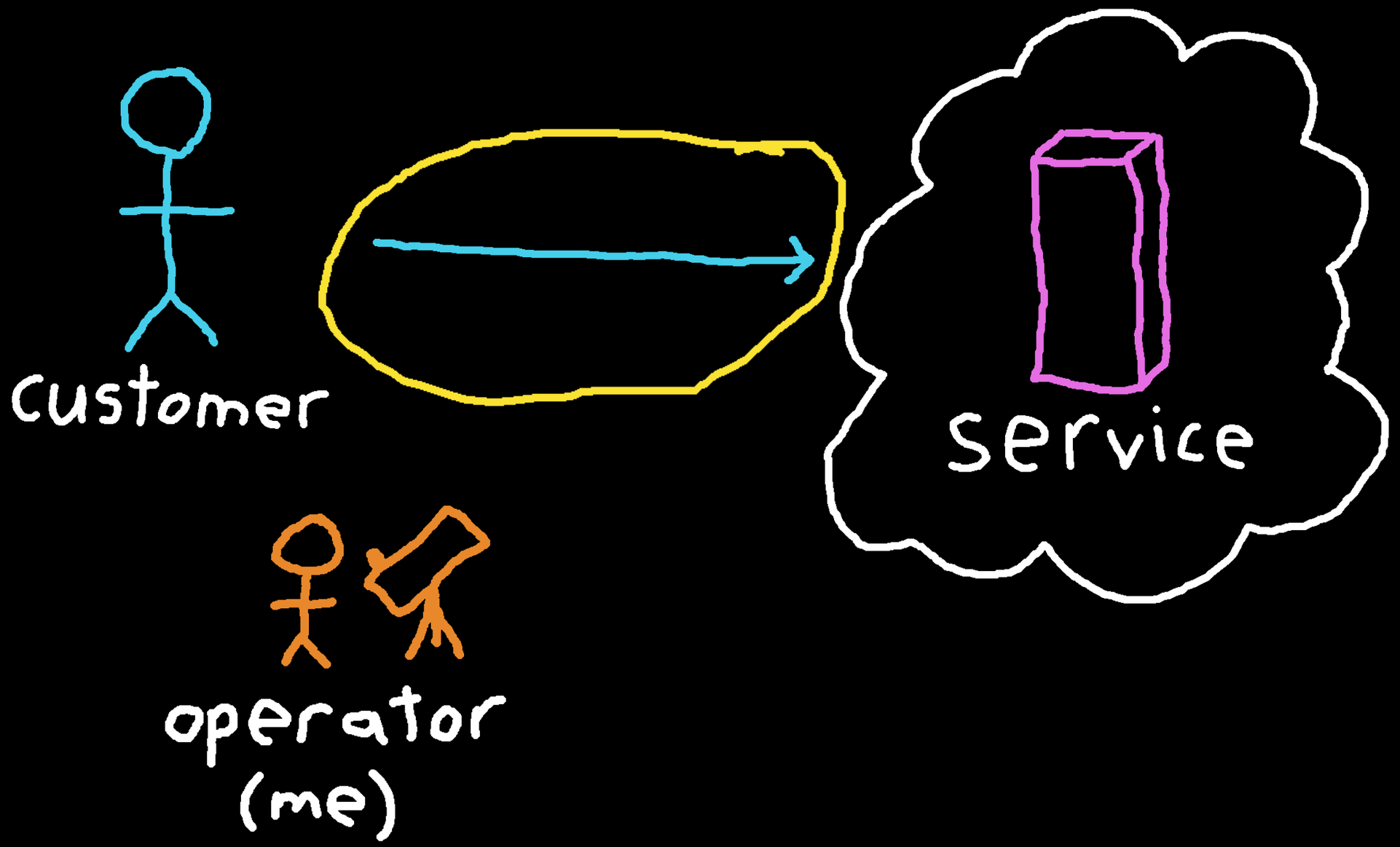
Monitoring philosophy
Noise
When metrics lie

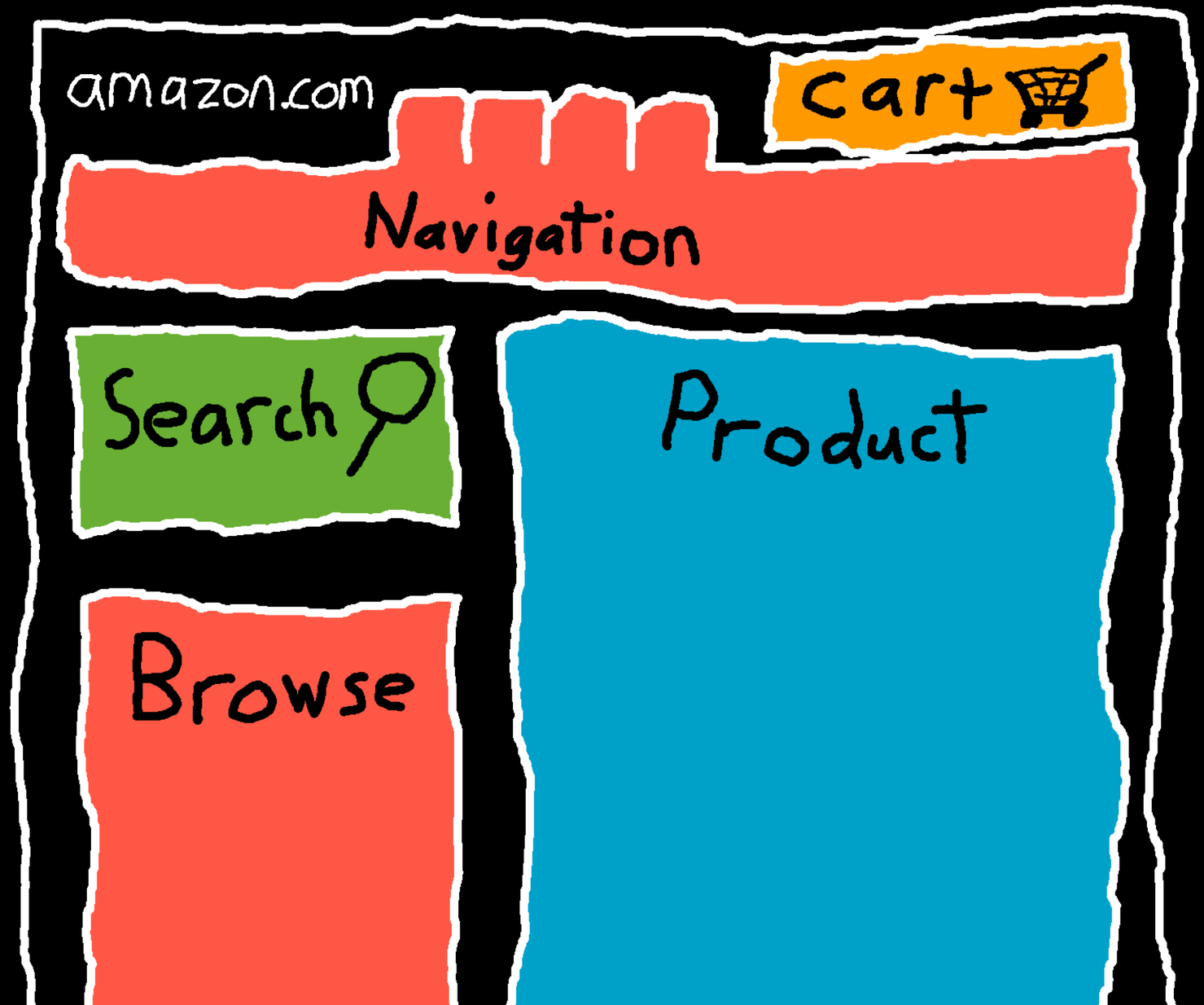


AWS re:Invent

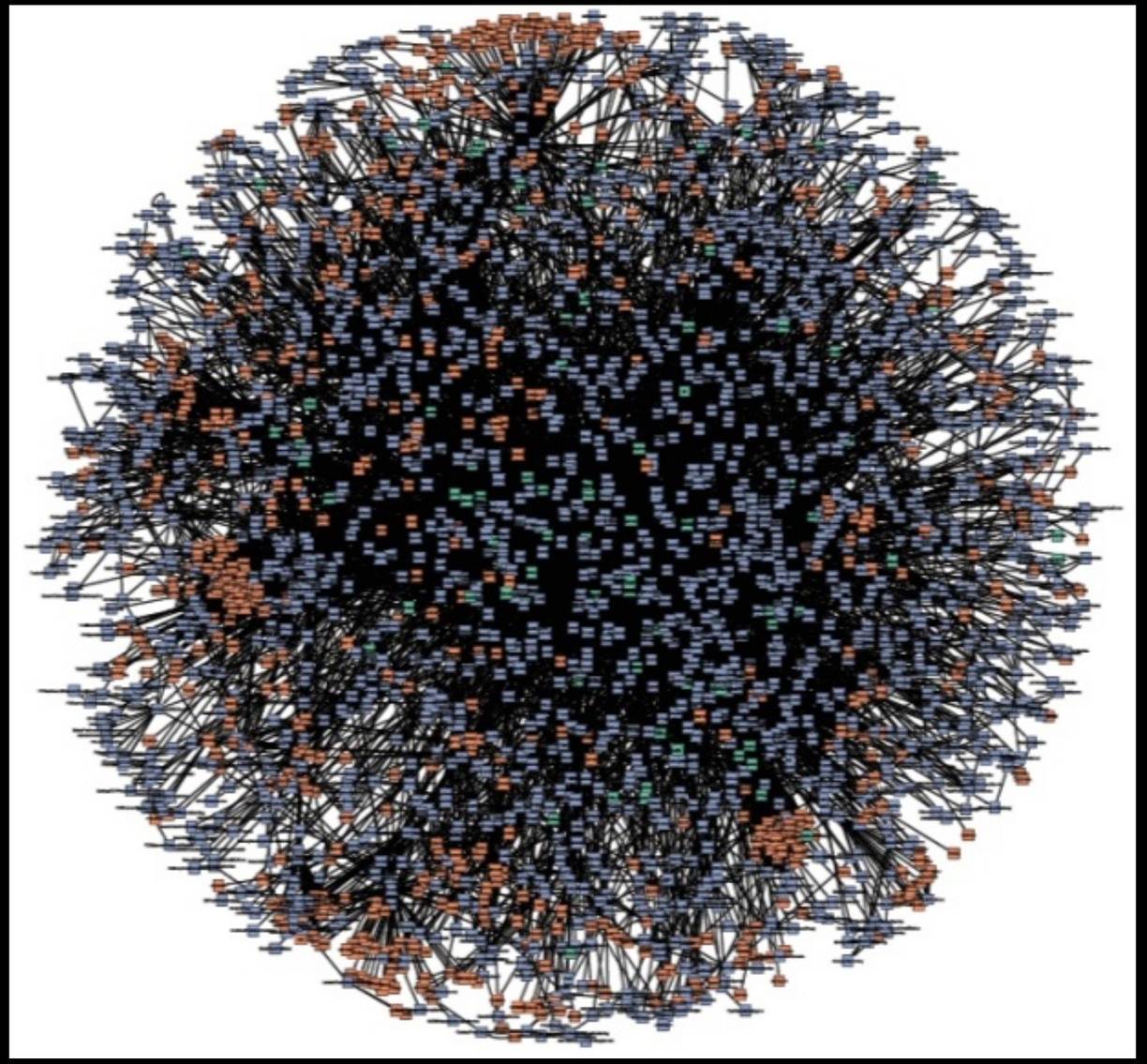
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Measuring the customer experience




amazon.com
Cart
Navigation
Search
Product
Browse

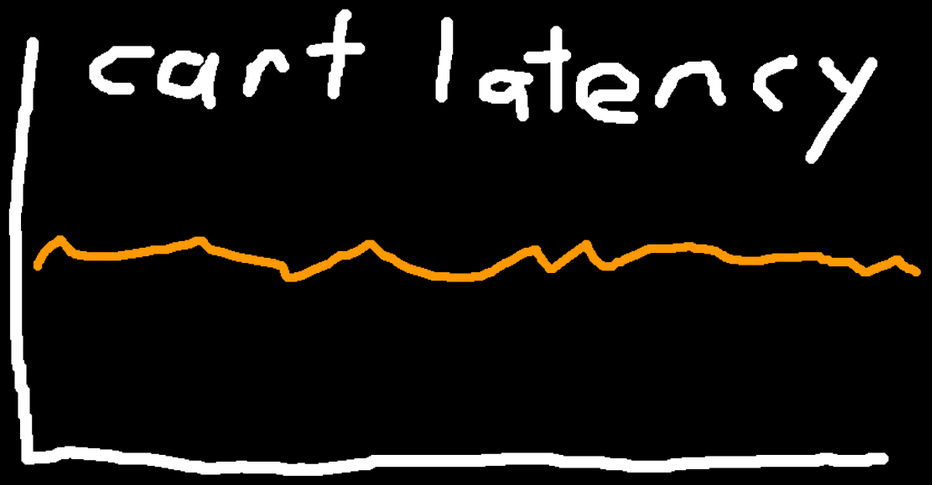
Service-oriented architecture



Per-page performance over time



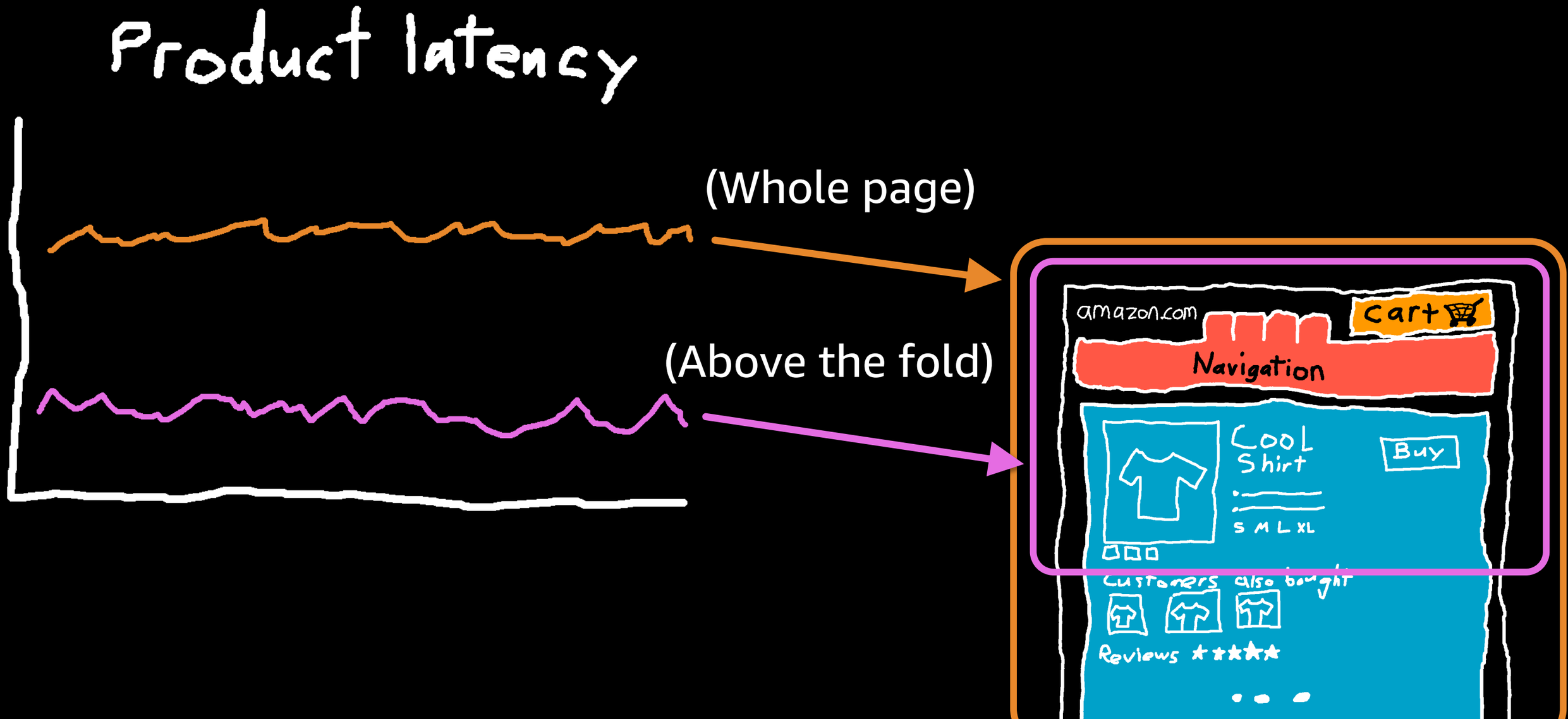





amazon.com
Cart
Navigation
Cool Shirt
Buy
S M L XL
Customers also bought
Reviews ★★★★★


http://
amazon.com
Cart
Navigation
Cool Shirt
Buy
S M L XL
Customers also bought
Reviews ★★★★★

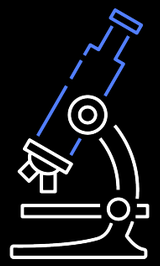
Measuring “above the fold” latency



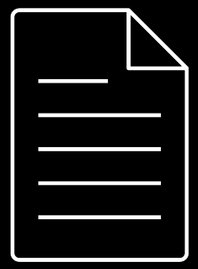
(Some of) Amazon’s monitoring philosophy

- Measure things that customers care about
- Measure from multiple perspectives in the system
- Continuously introspect metrics
- Continuously question the metrics themselves

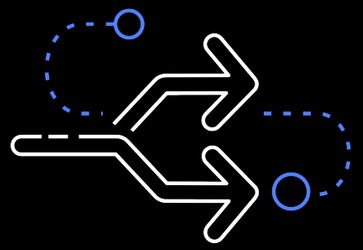
Monitoring: How is it made?



Instrumentation



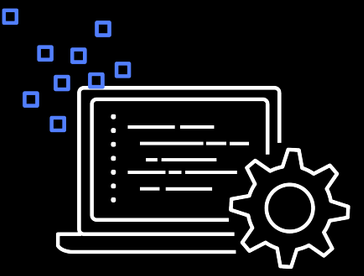
Logs



Traces



Log archives



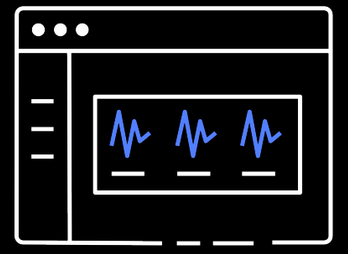
Log analysis



Metrics

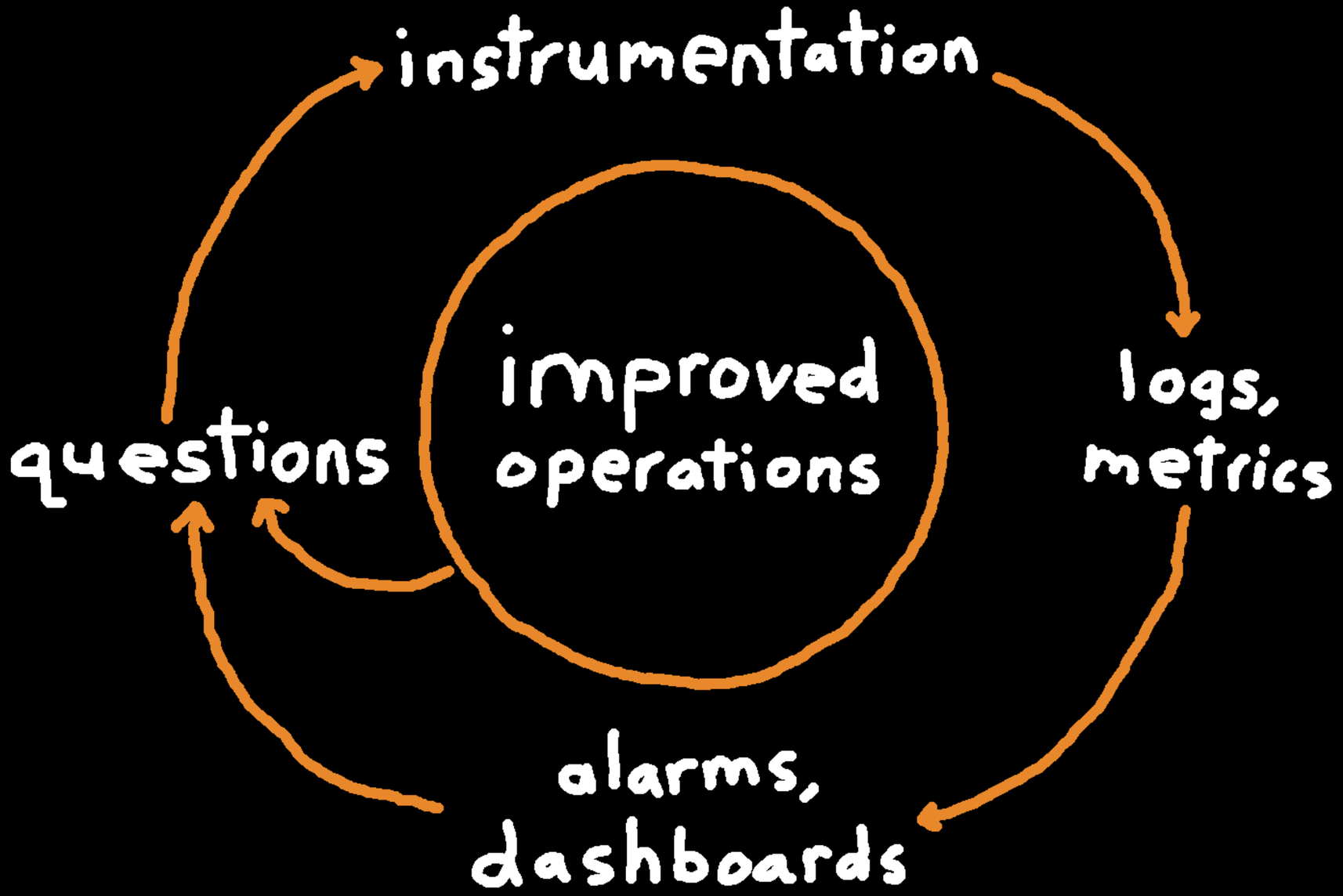


Alarms



Dashboards

The cycle of monitoring



Monitoring philosophy

Noise

- Noisy latency
- Unpredictable traffic
- Distributed noise

When metrics lie

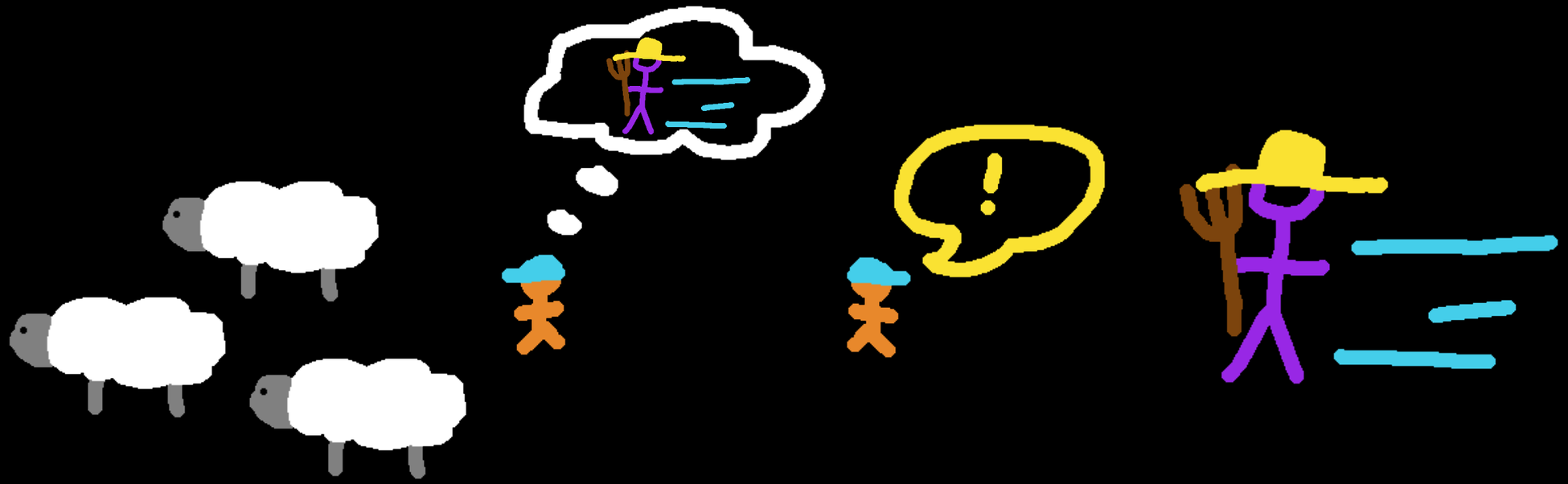


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

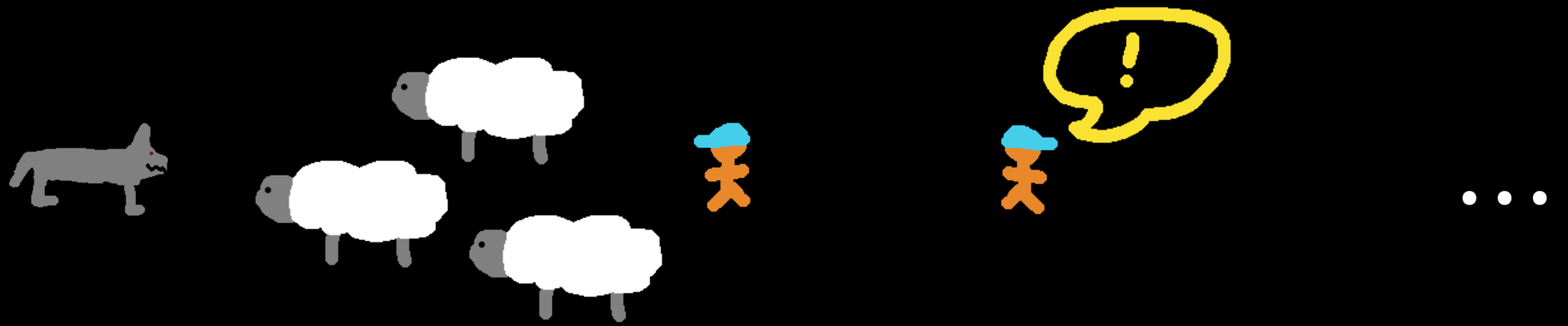
The alarm who cried wolf



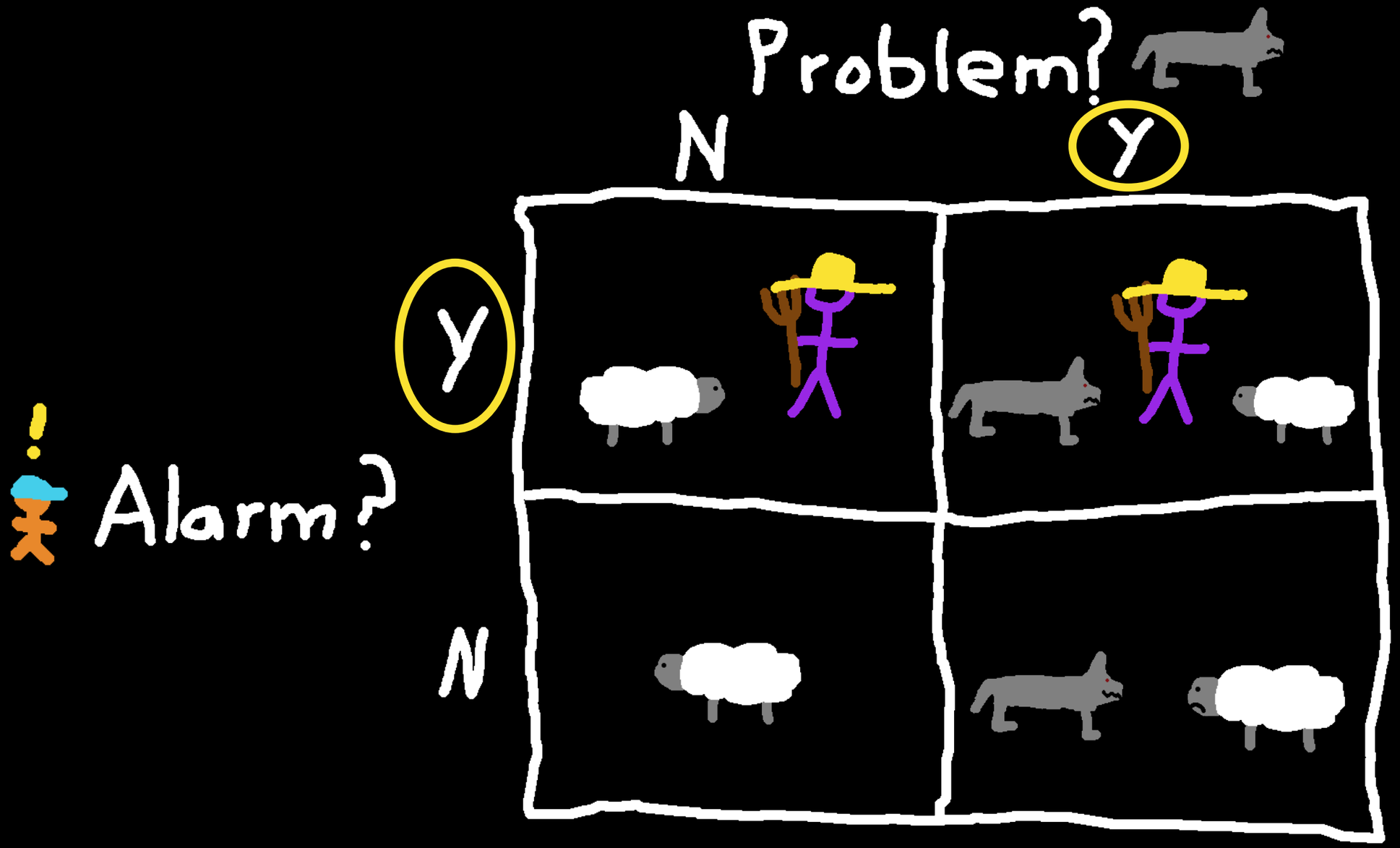
The alarm who cried wolf



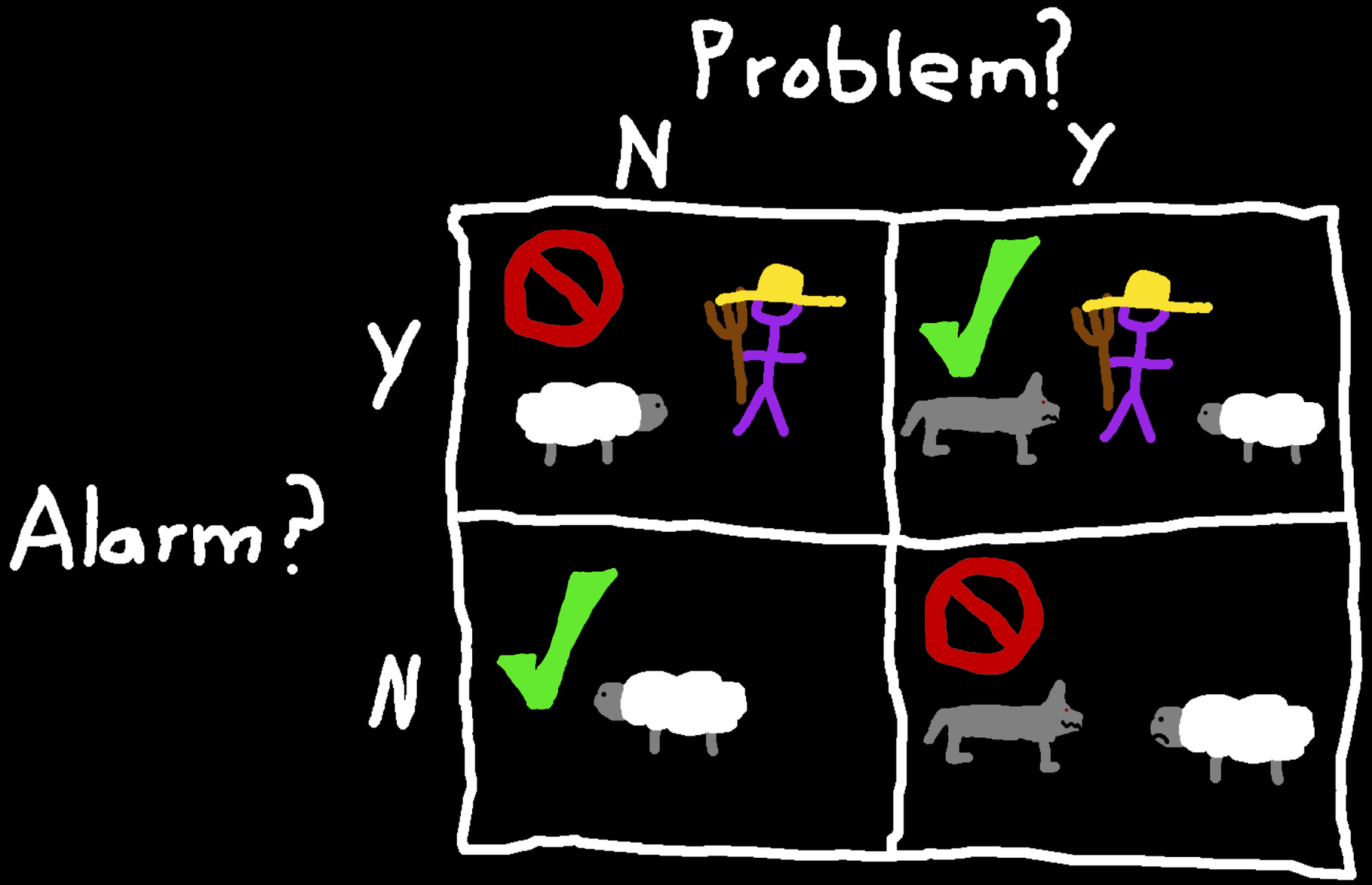
The alarm who cried wolf



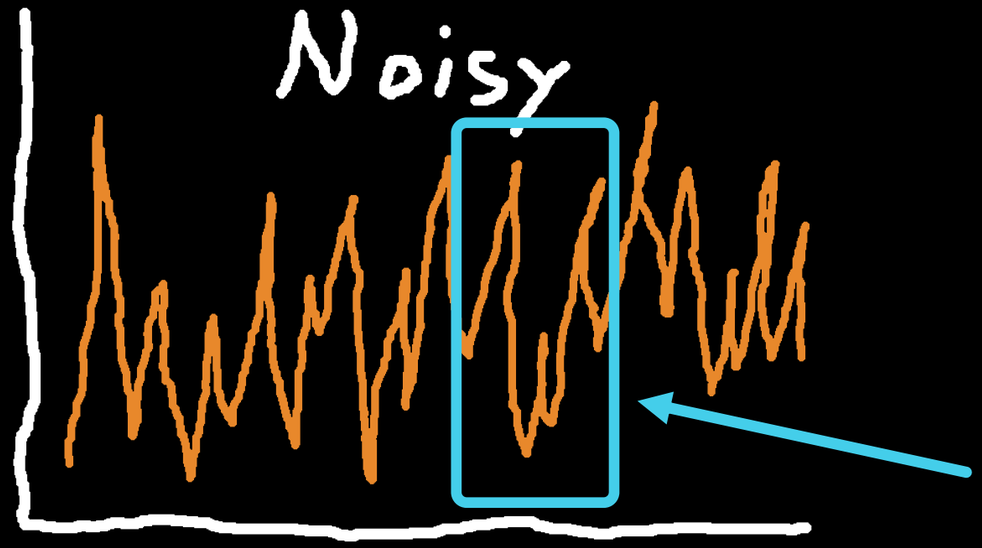
Separating signal from noise



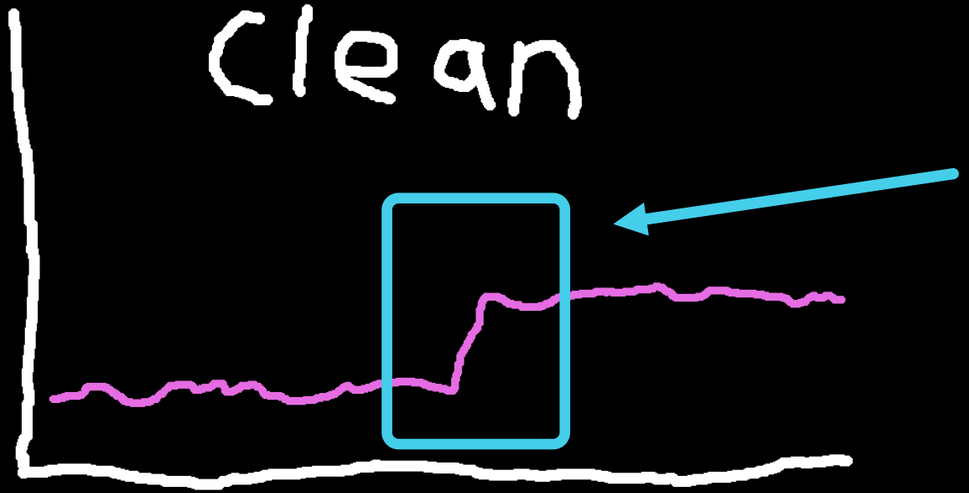
Separating signal from noise



Signal-to-noise ratio



(Did things get worse around here?)

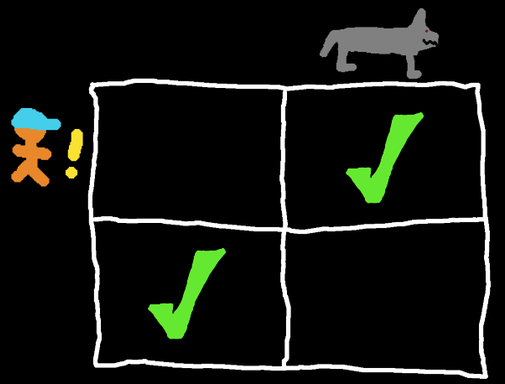


(Obvious inflection point)

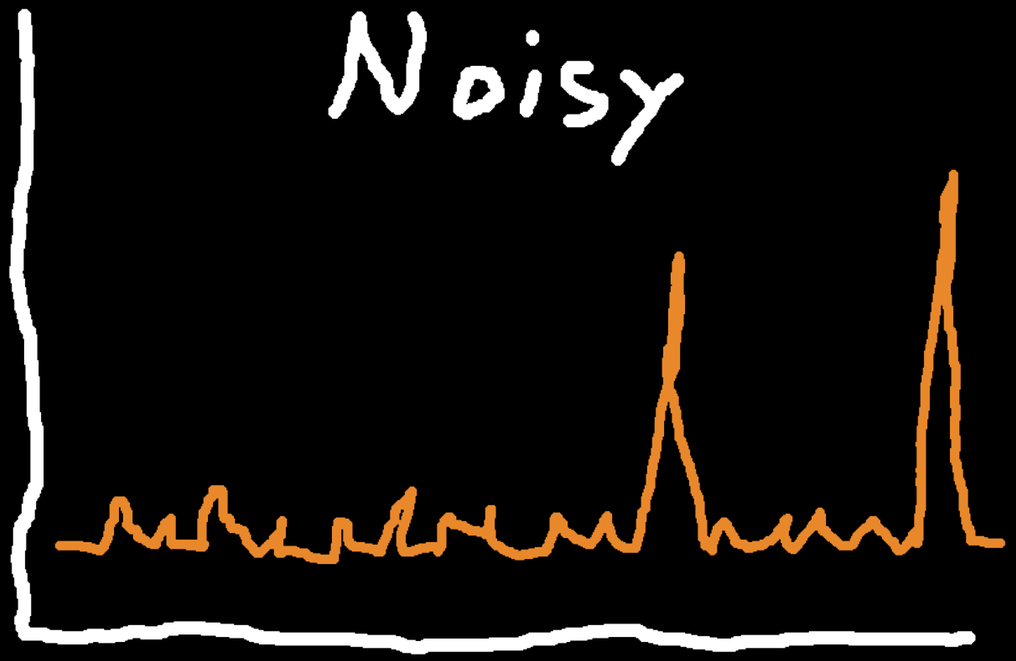
Signal-to-noise ratio







The noise that cried wolf

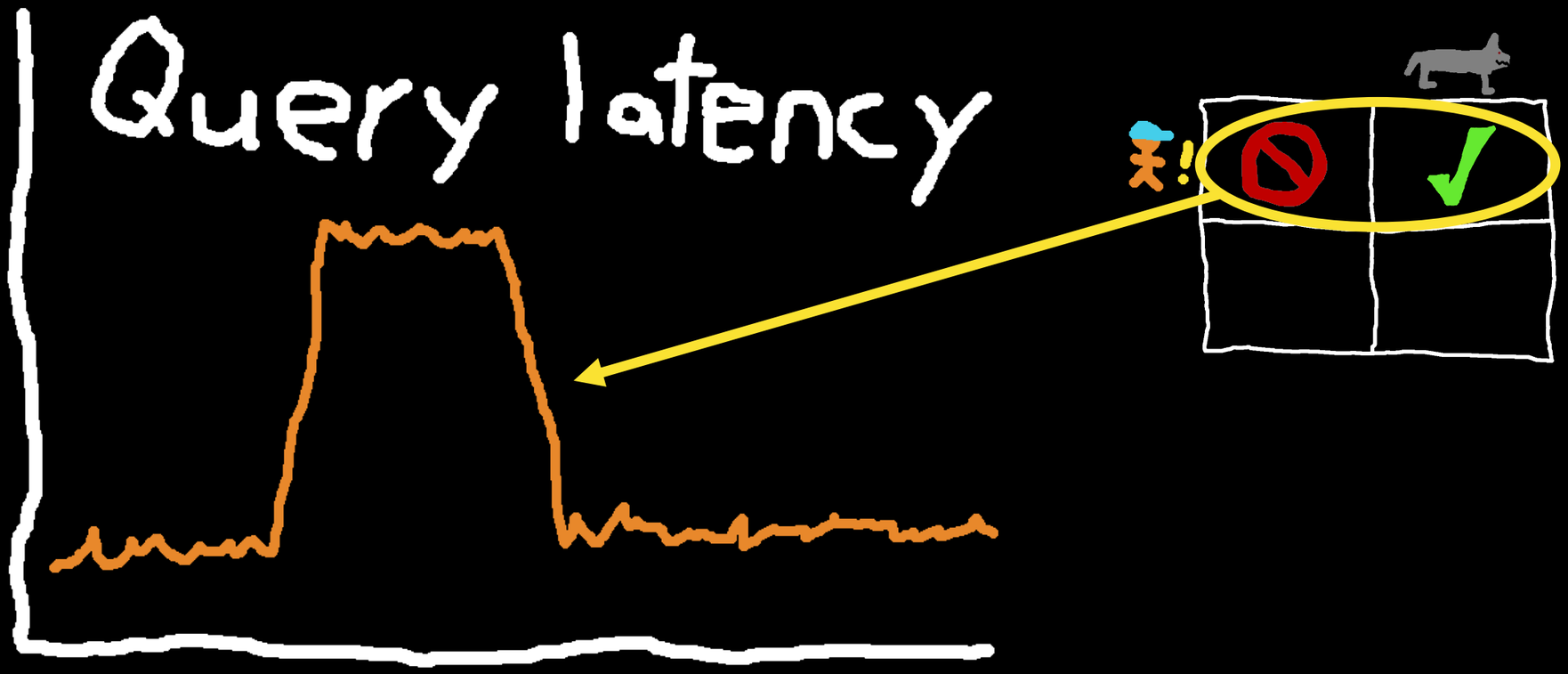


Example: Query latency tiering



Amazon
DynamoDB

Noisy latency



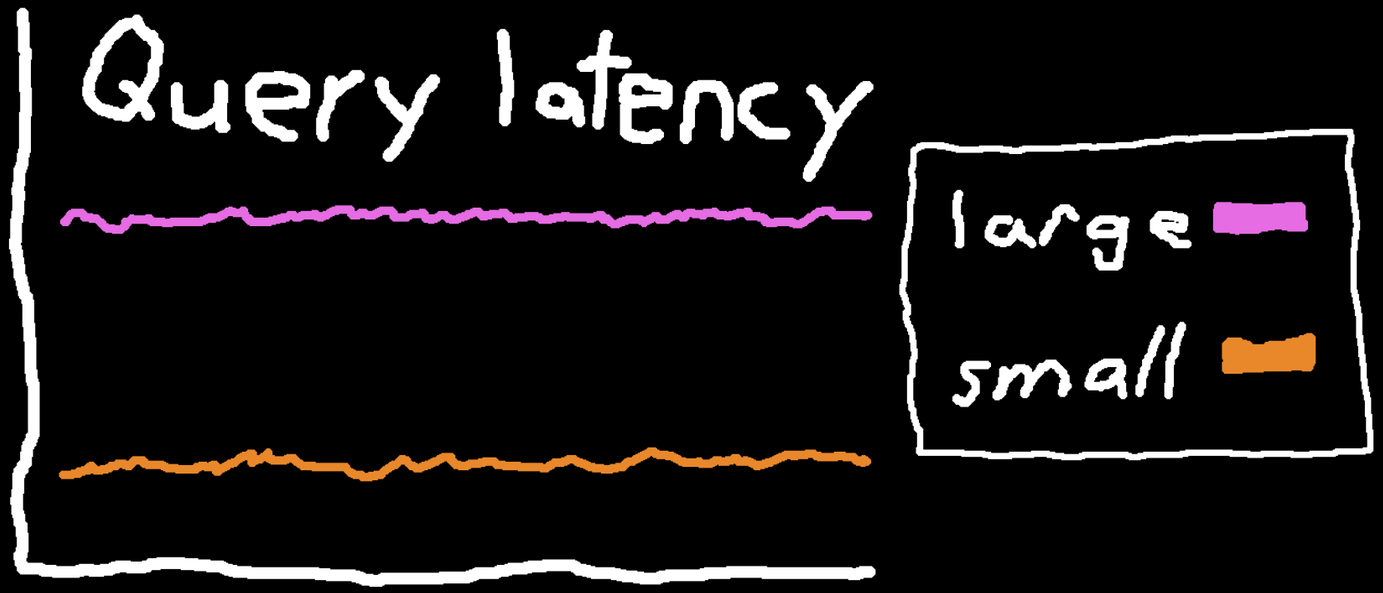
Separating signal from noise

```
metrics.addTime("QueryLatency", duration);
```

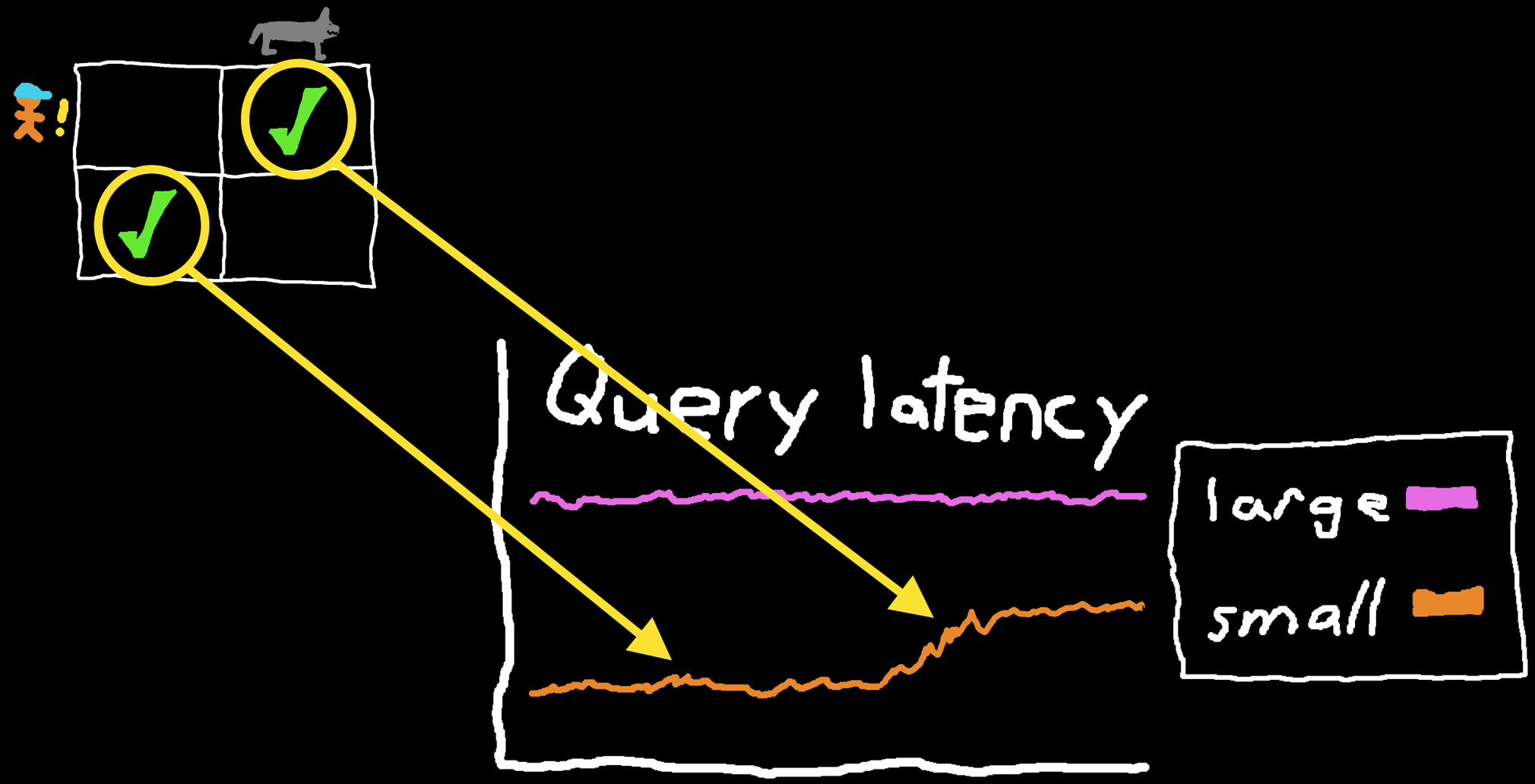


Separating signal from noise

```
if( queryCost < 10 )
  metrics.addTime("QuerySmall", duration);
else
  metrics.addTime("QueryLarge", duration);
```



Separating signal from noise



Monitoring philosophy

Noise

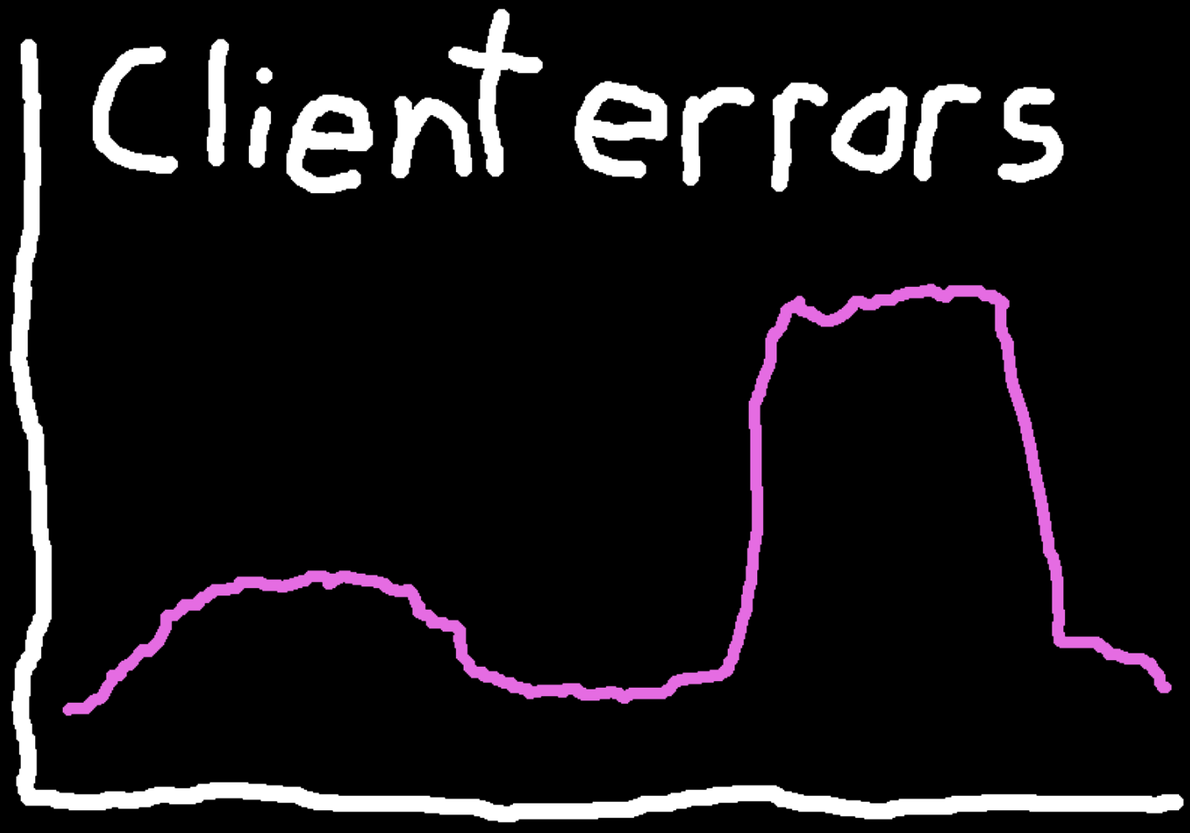
- Noisy latency
- Unpredictable traffic
- Distributed noise

When metrics lie



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Alarming on client behavior



Client vs. server fault

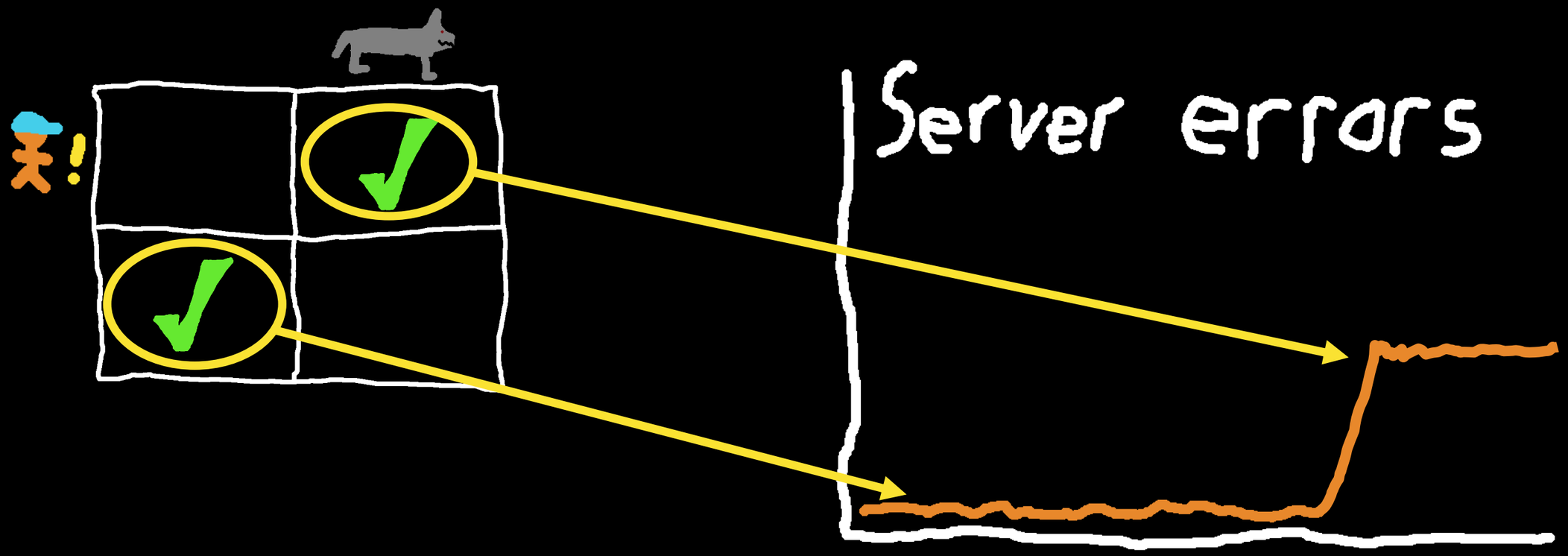
Server fault (5XX)

500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported

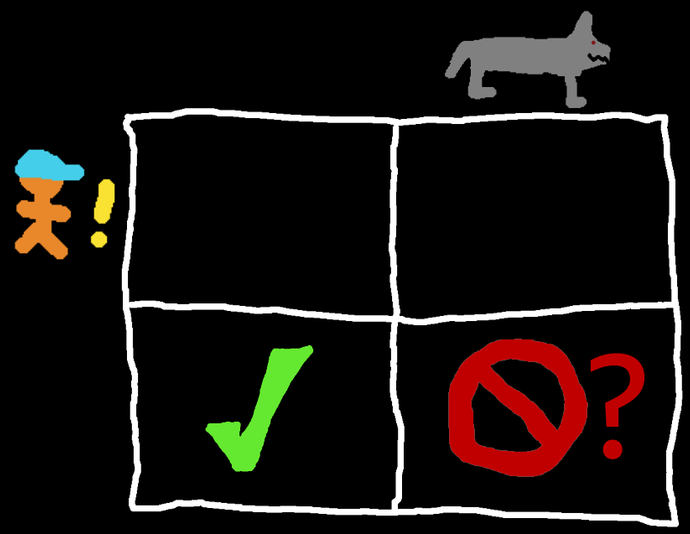
Client fault (4XX)

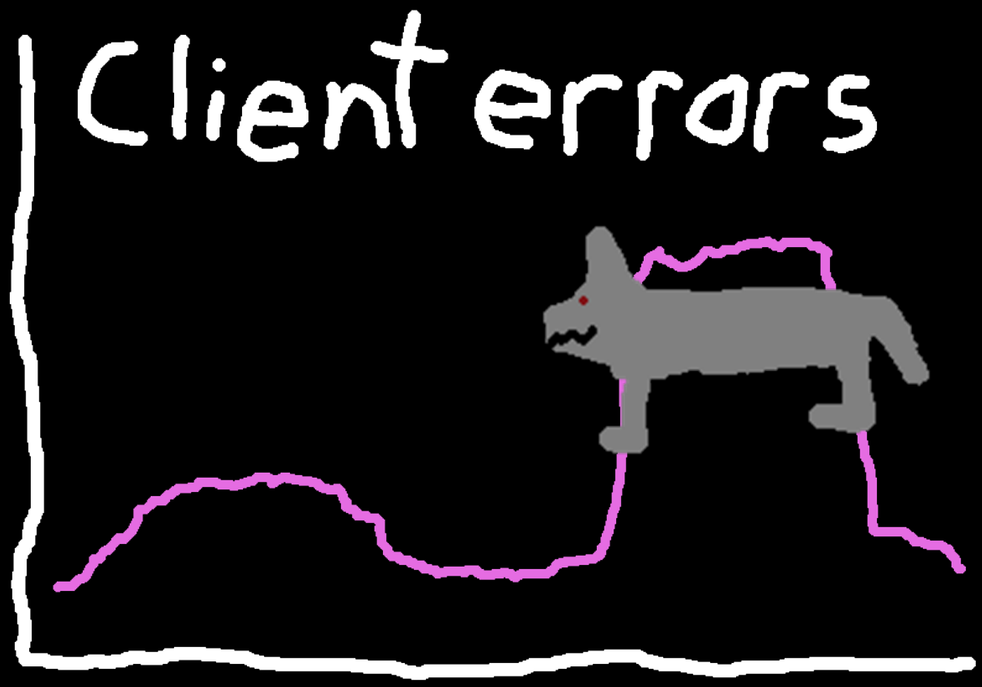
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Payload Too Large
414	URI Too Long
415	Unsupported Media Type
416	Range Not Satisfiable
417	Expectation Failed
426	Upgrade Required

Server faults: Easy



Client faults: Noisy





AWS Lambda CreateFunction

FunctionName

The name of the Lambda function.

Name formats

- **Function name** - `my-function`.
- **Function ARN** - `arn:aws:lambda:us-west-2:123456789012:function:my-function`.
- **Partial ARN** - `123456789012:function:my-function`.

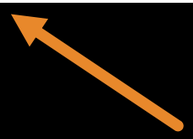
The length constraint applies only to the full ARN. If you specify only the function name, it is limited to 64 characters in length.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 140.

Pattern: `(arn:(aws[a-zA-Z-]*)?:lambda:)?([a-z]{2}(-gov)?-[a-z]+-\d{1}:)?(\d{12}:)?(function:)?([a-zA-Z0-9-_]+)(:(\$LATEST|[a-zA-Z0-9-_]+))?`

Required: Yes

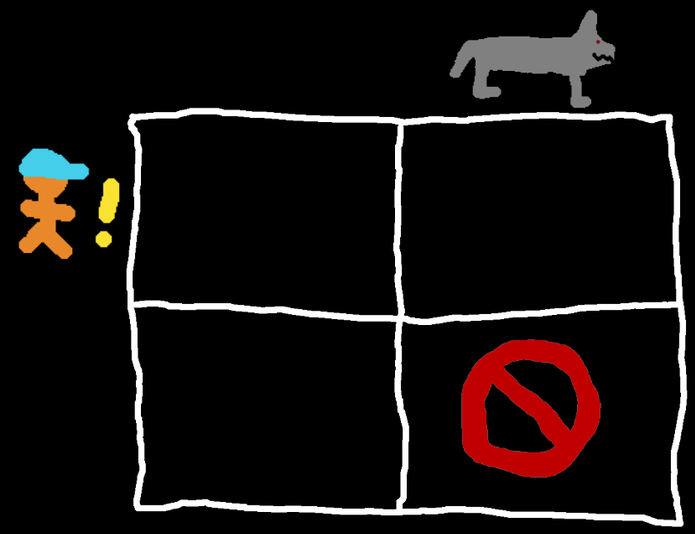


(Constraints on length, required, type, pattern)

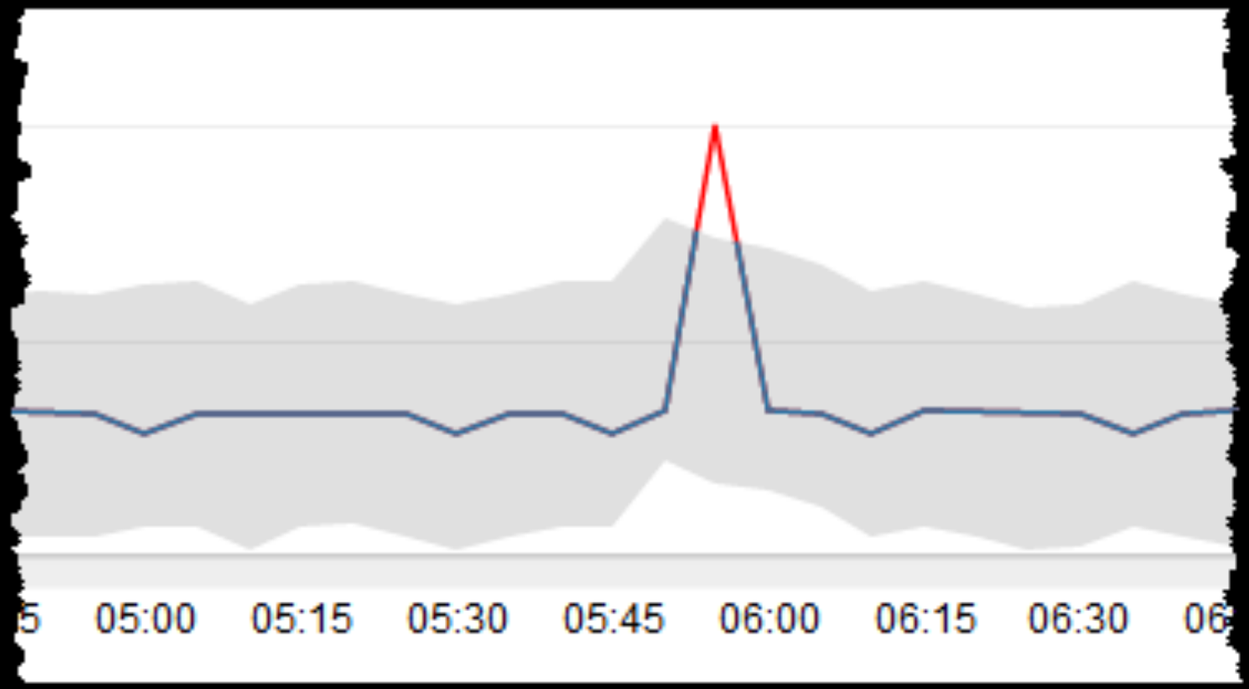
Input validation bugs

(Bug: Making input validation more strict)

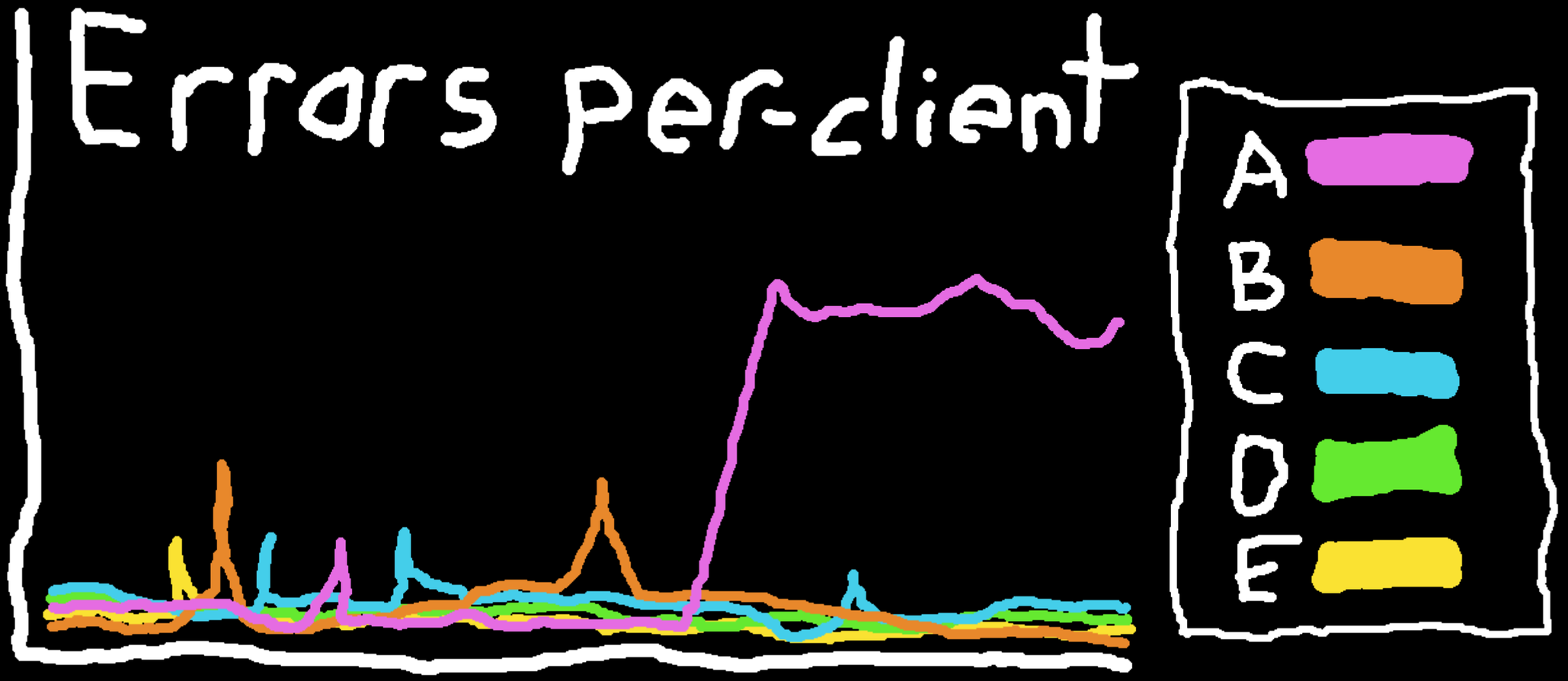
```
    public void validate() {
-     if(input.length() > 140) {
+     if(input.length() > 100) {
        throw new ClientError();
      }
    }
```



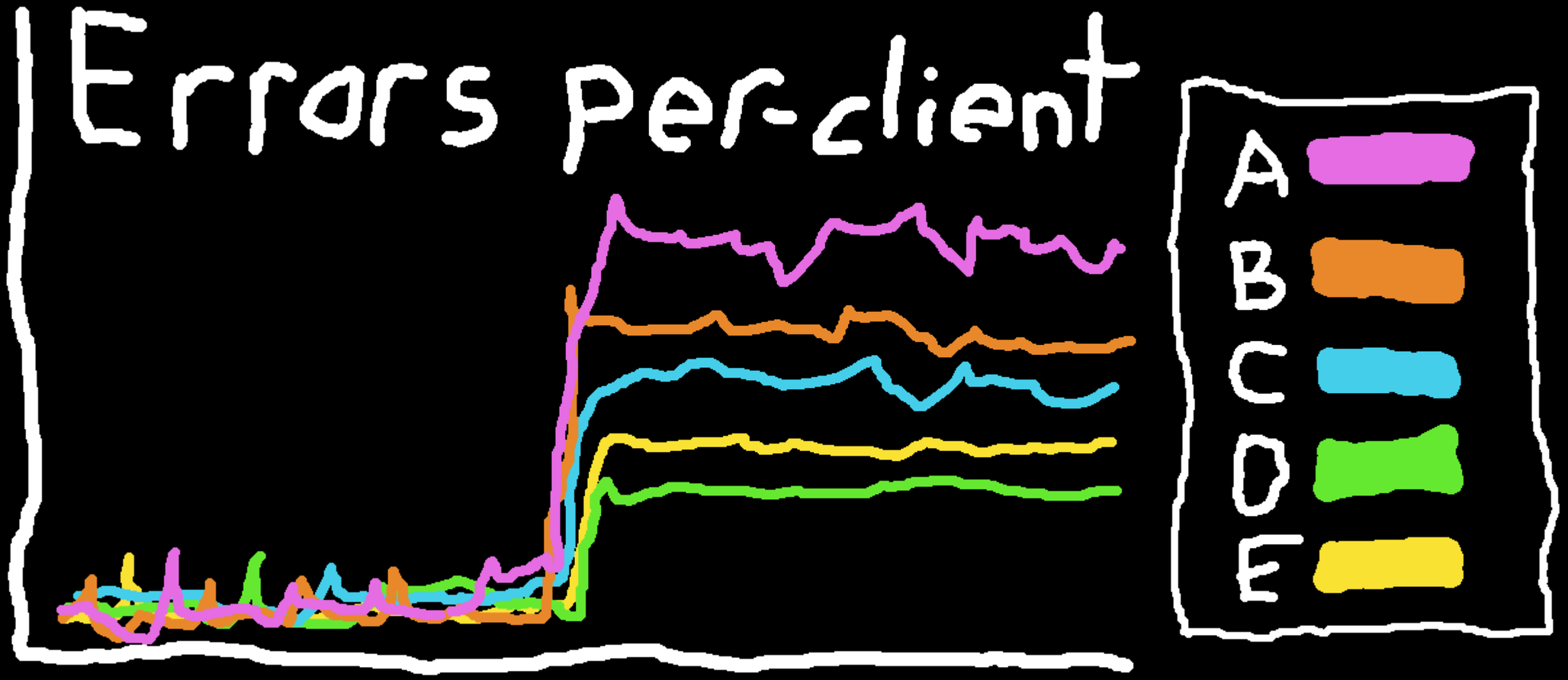
Anomaly detection



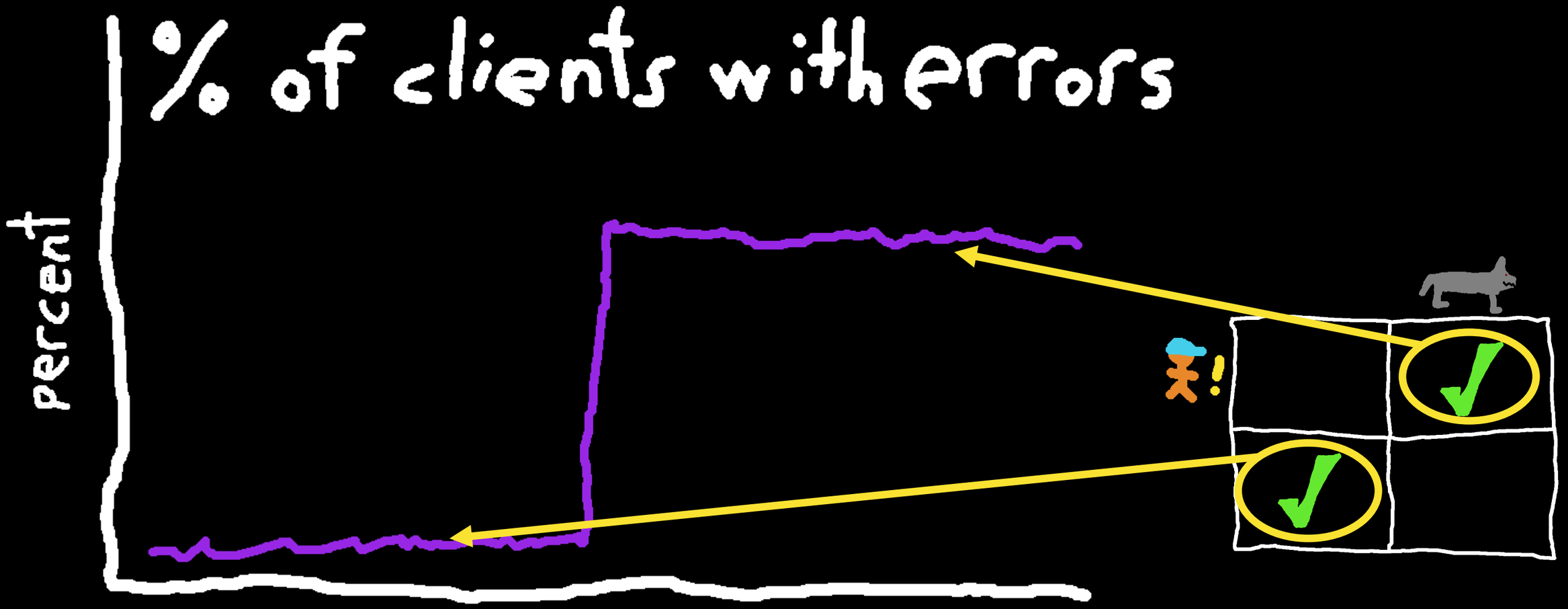
Graphing top contributors



Graphing top contributors



Client faults: Deployment time



Monitoring philosophy

Noise

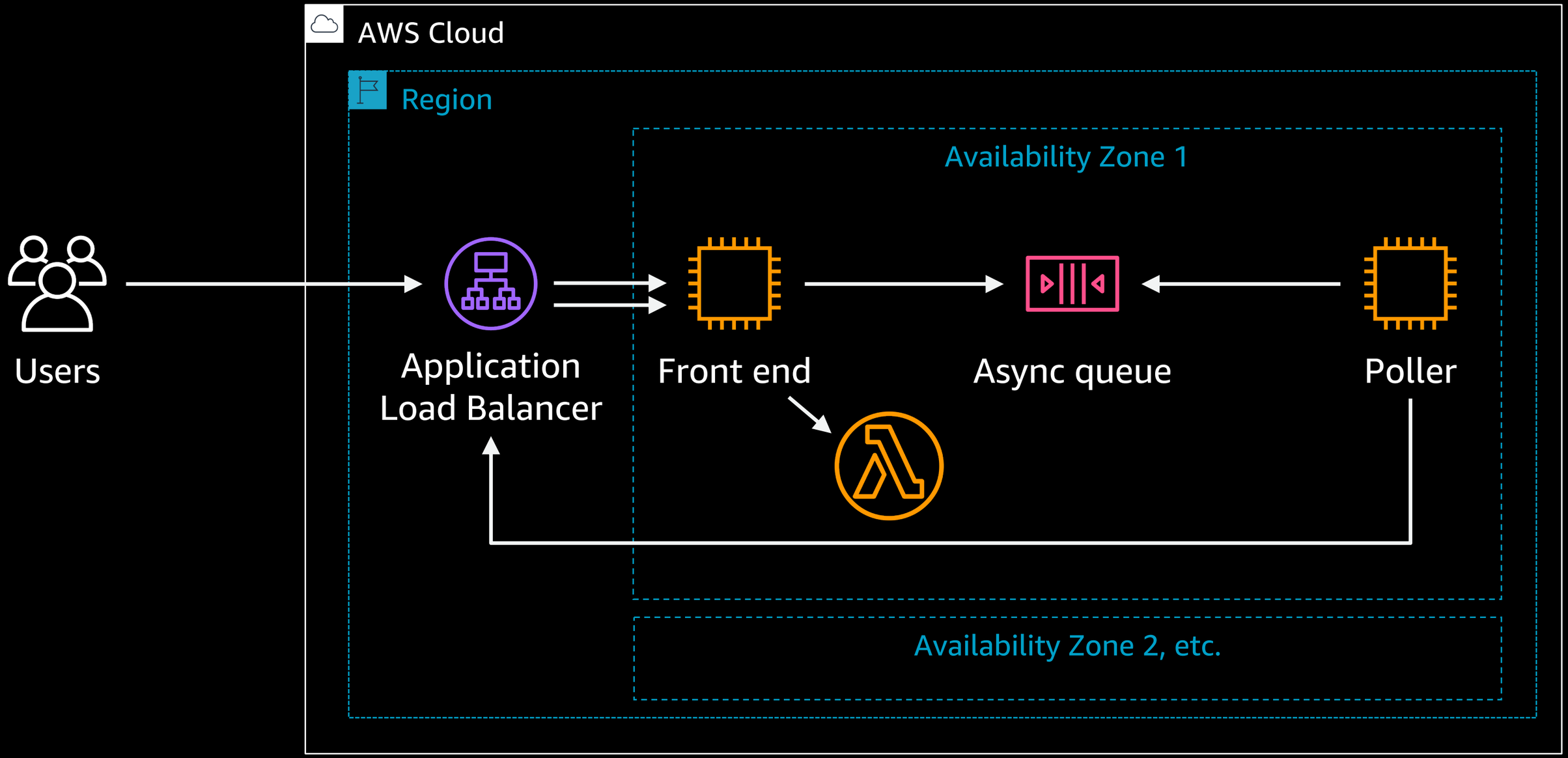
- Noisy latency
- Unpredictable traffic
- Distributed noise

When metrics lie

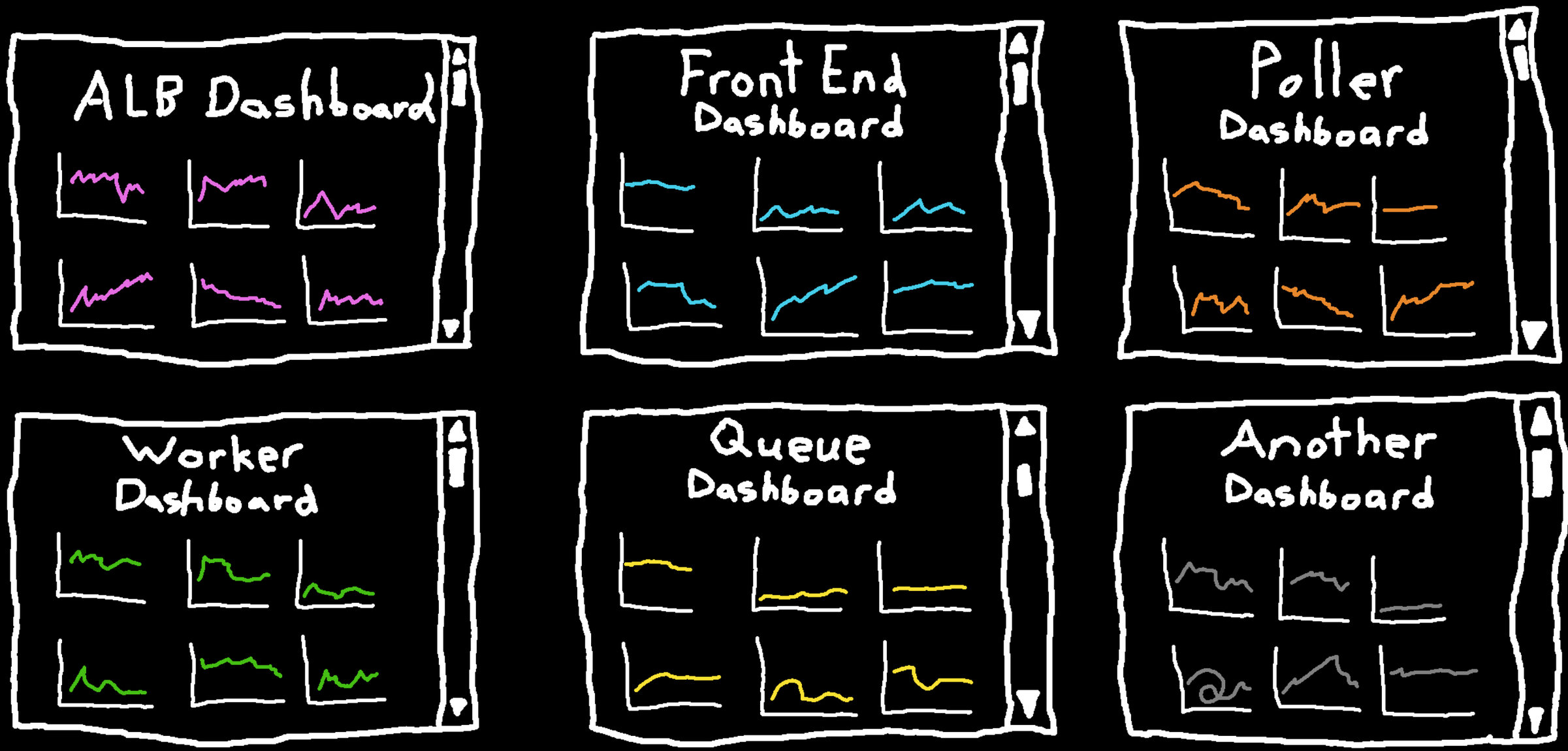


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

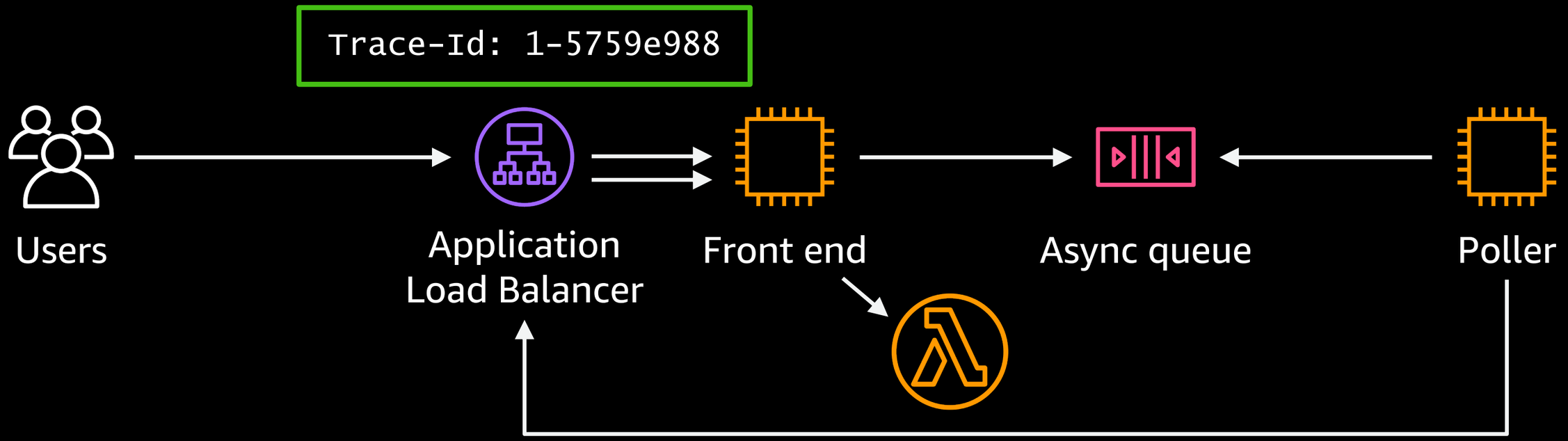
Reasoning about distributed systems



Visibility into distributed systems

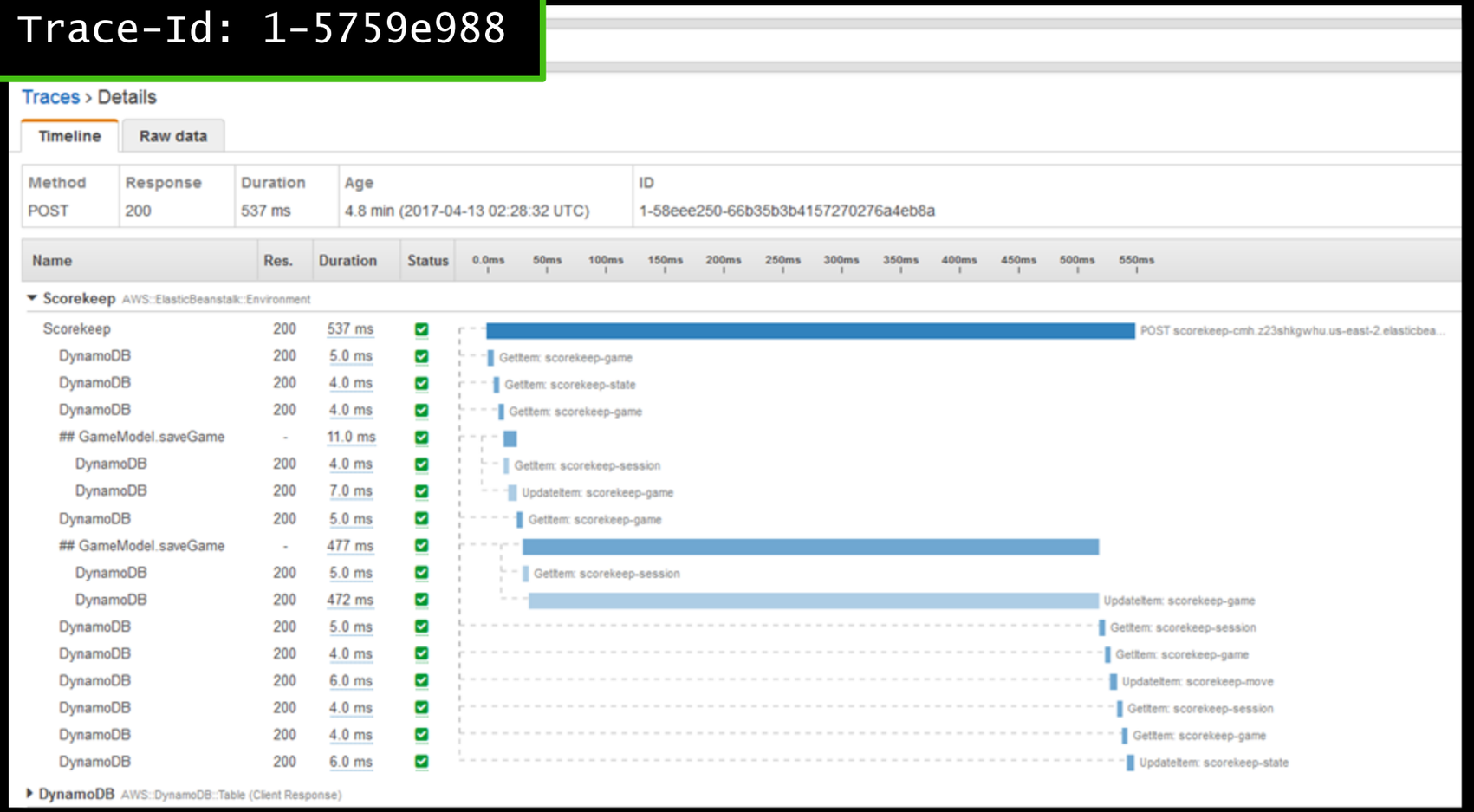


Distributed tracing

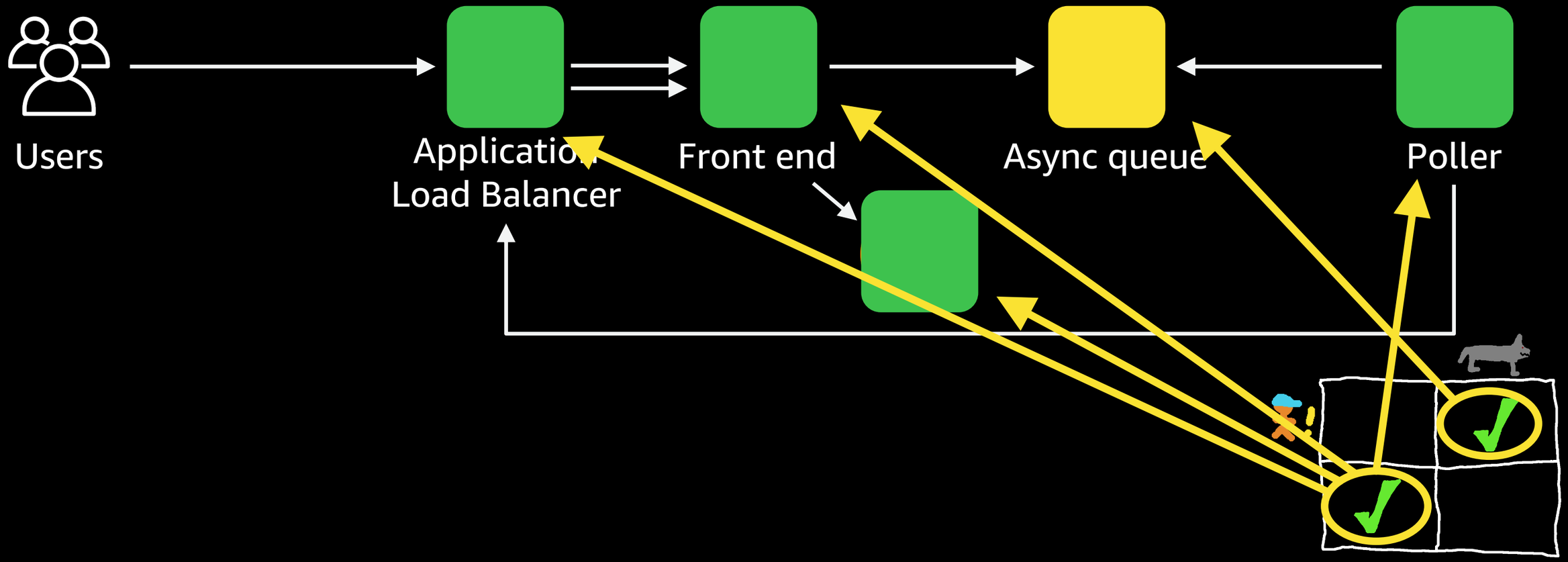


Per-request visibility

```
Trace-Id: 1-5759e988
```



Distributed system-wide visibility



Monitoring philosophy
Noise
When metrics lie

Summary statistics

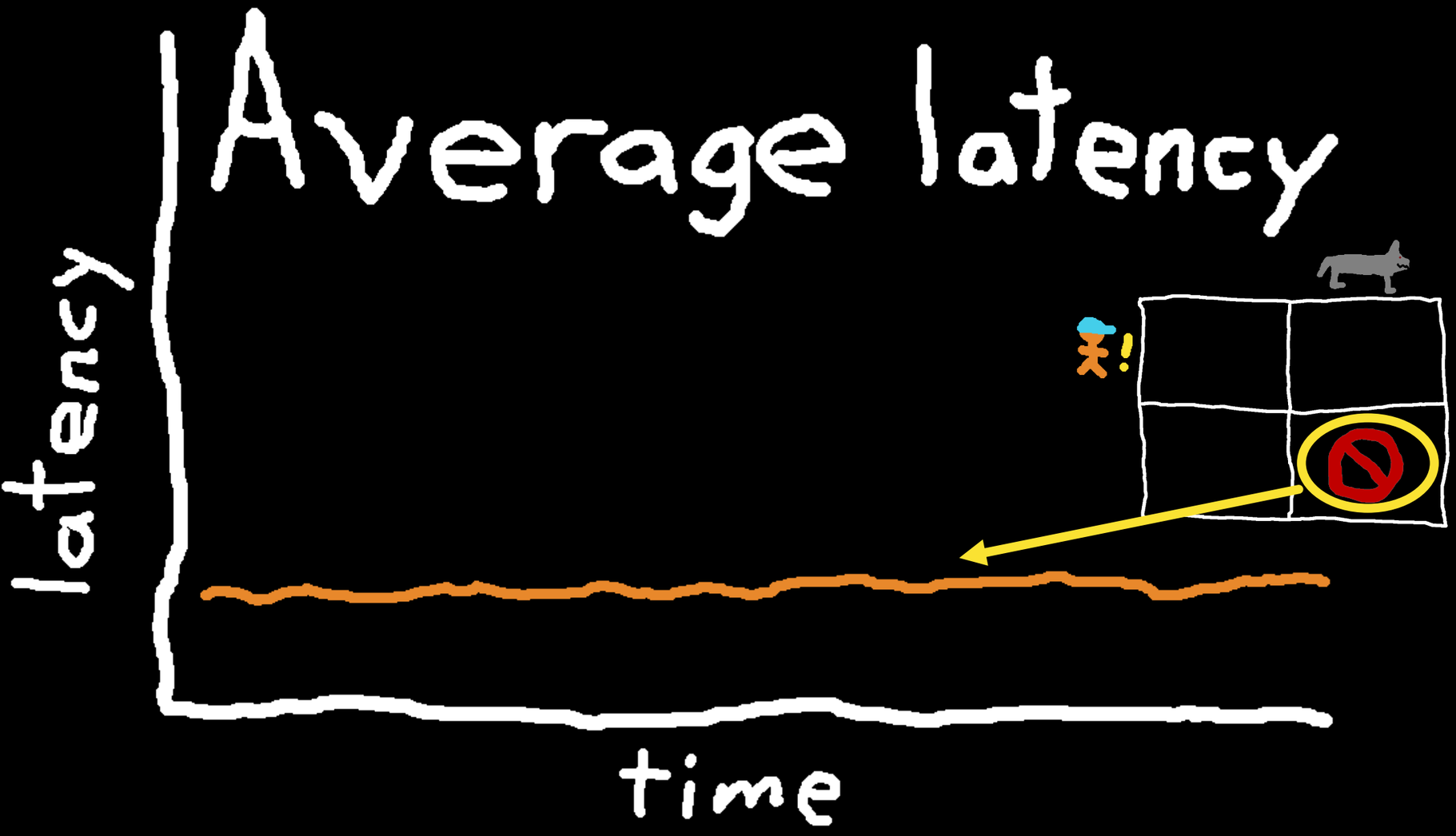
Misleading measurement

Coarse-grained

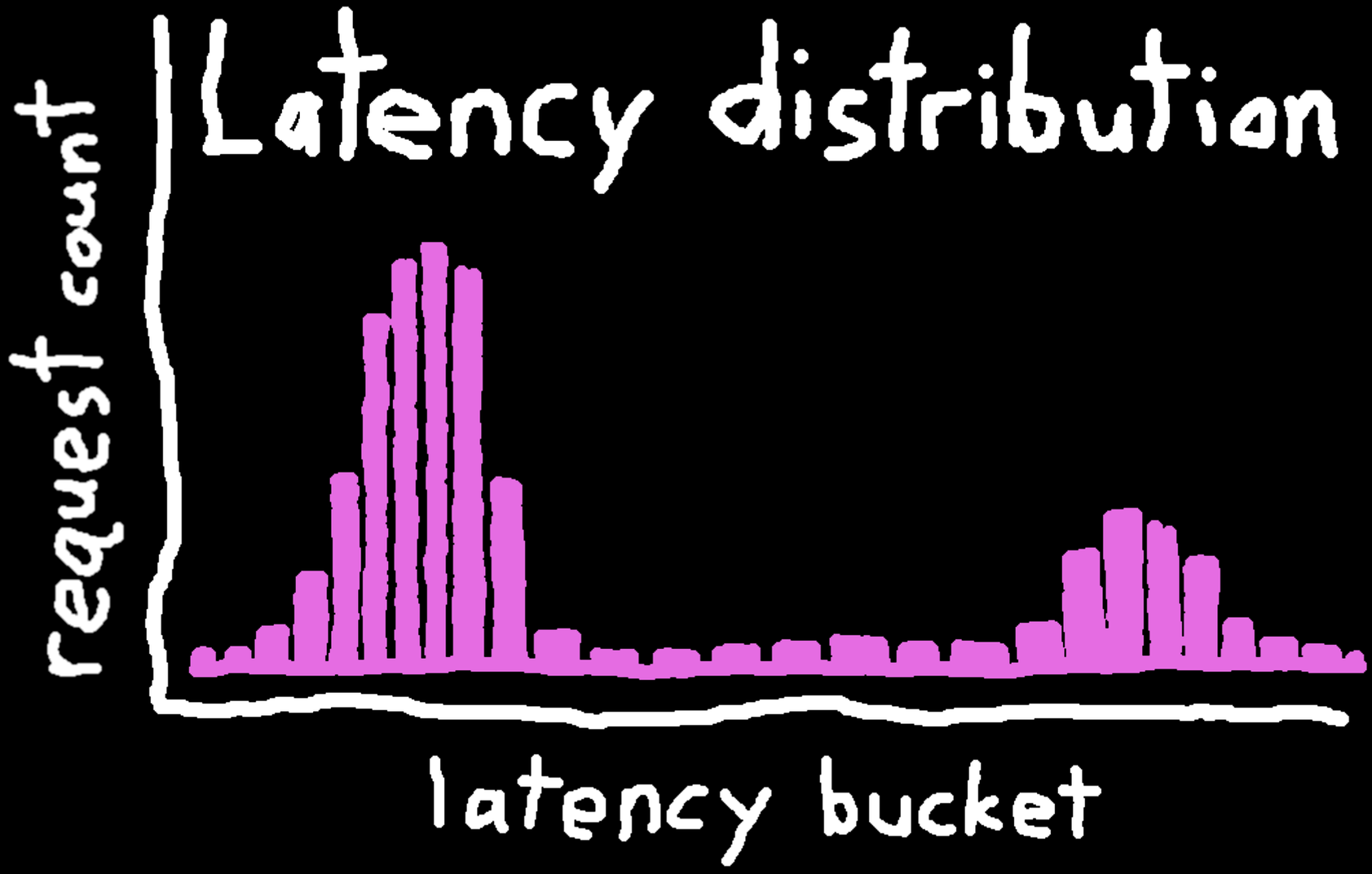


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

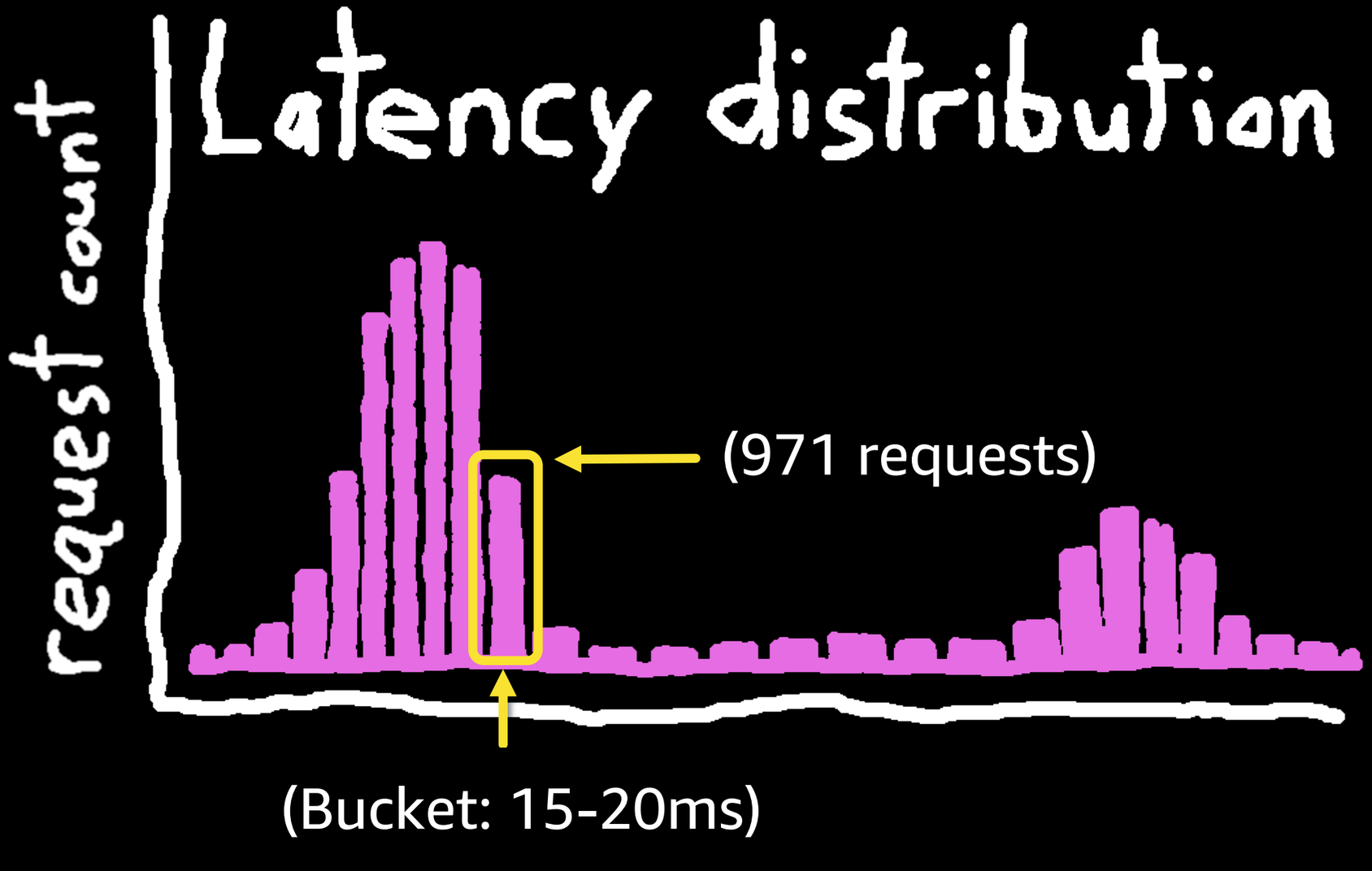
When summary statistics lie



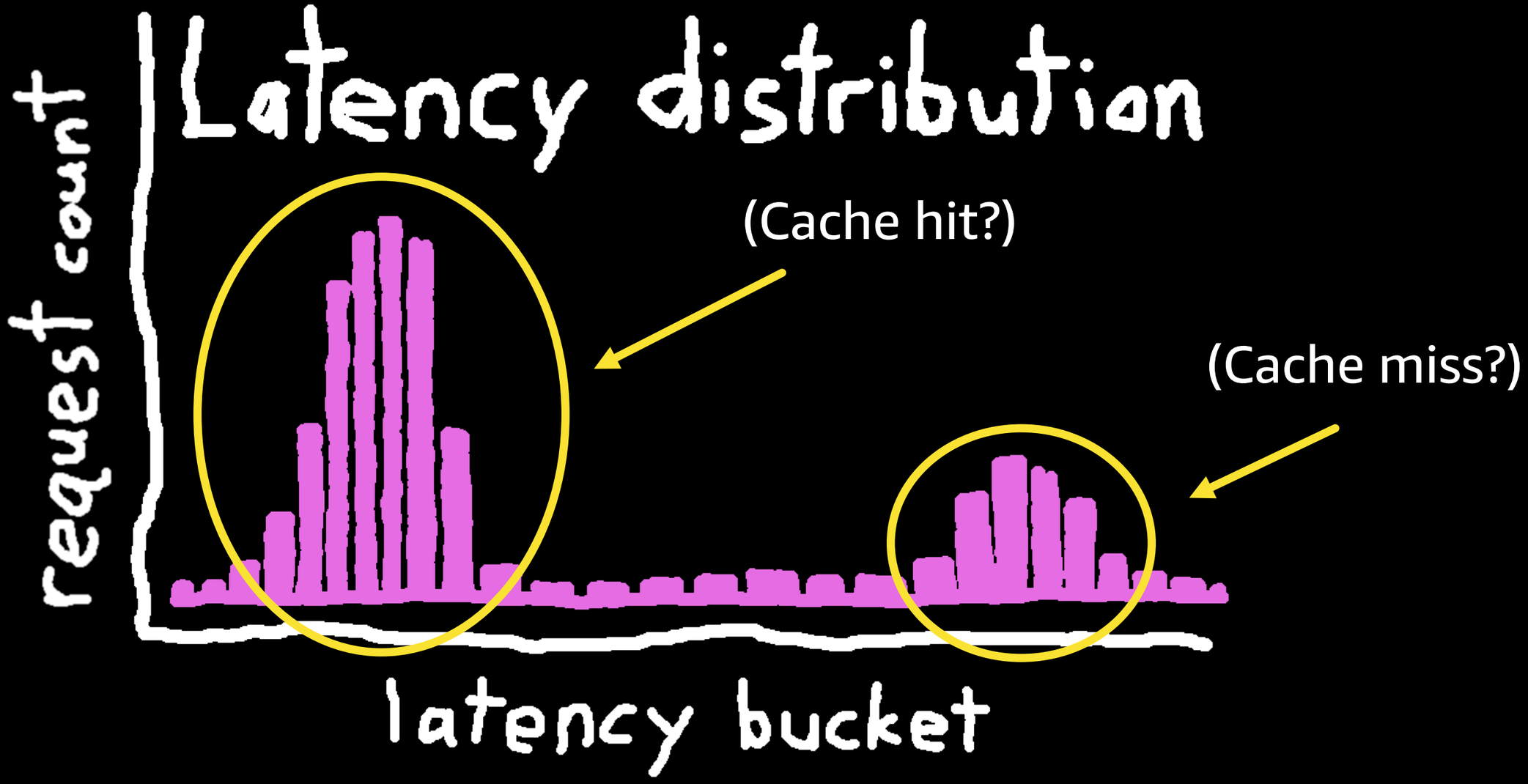
Histograms



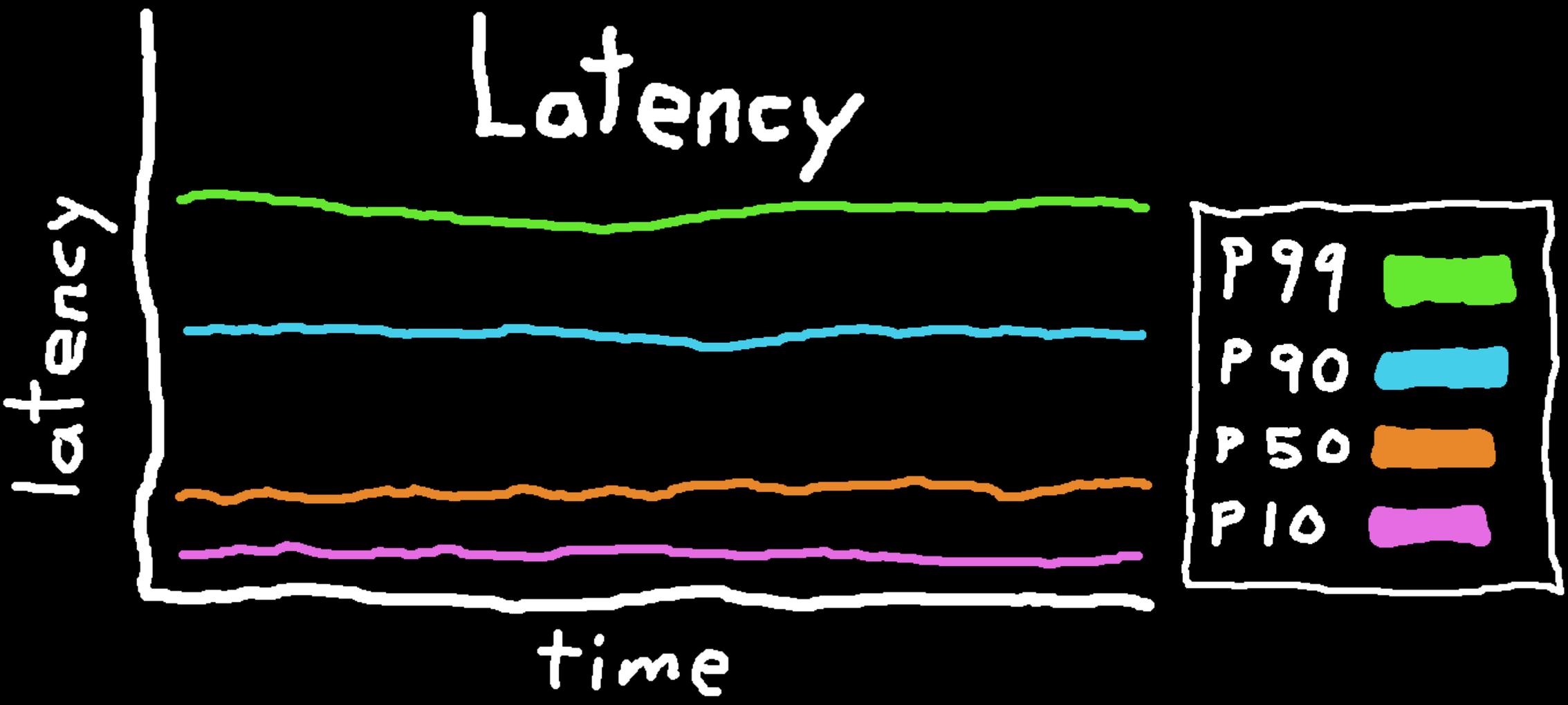
Histograms


Latency distribution
request count
(971 requests)
(Bucket: 15-20ms)

Averages lie



Percentiles



Monitoring philosophy
Noise
When metrics lie

Summary statistics

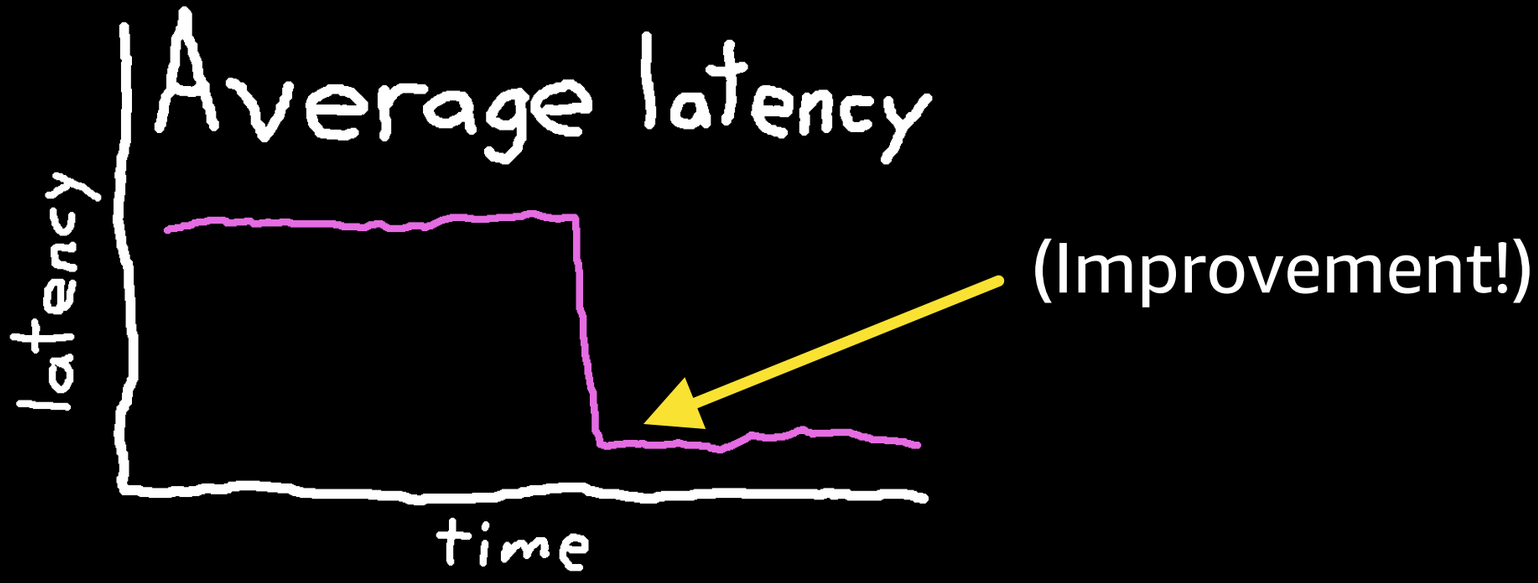
Misleading measurement

Coarse-grained

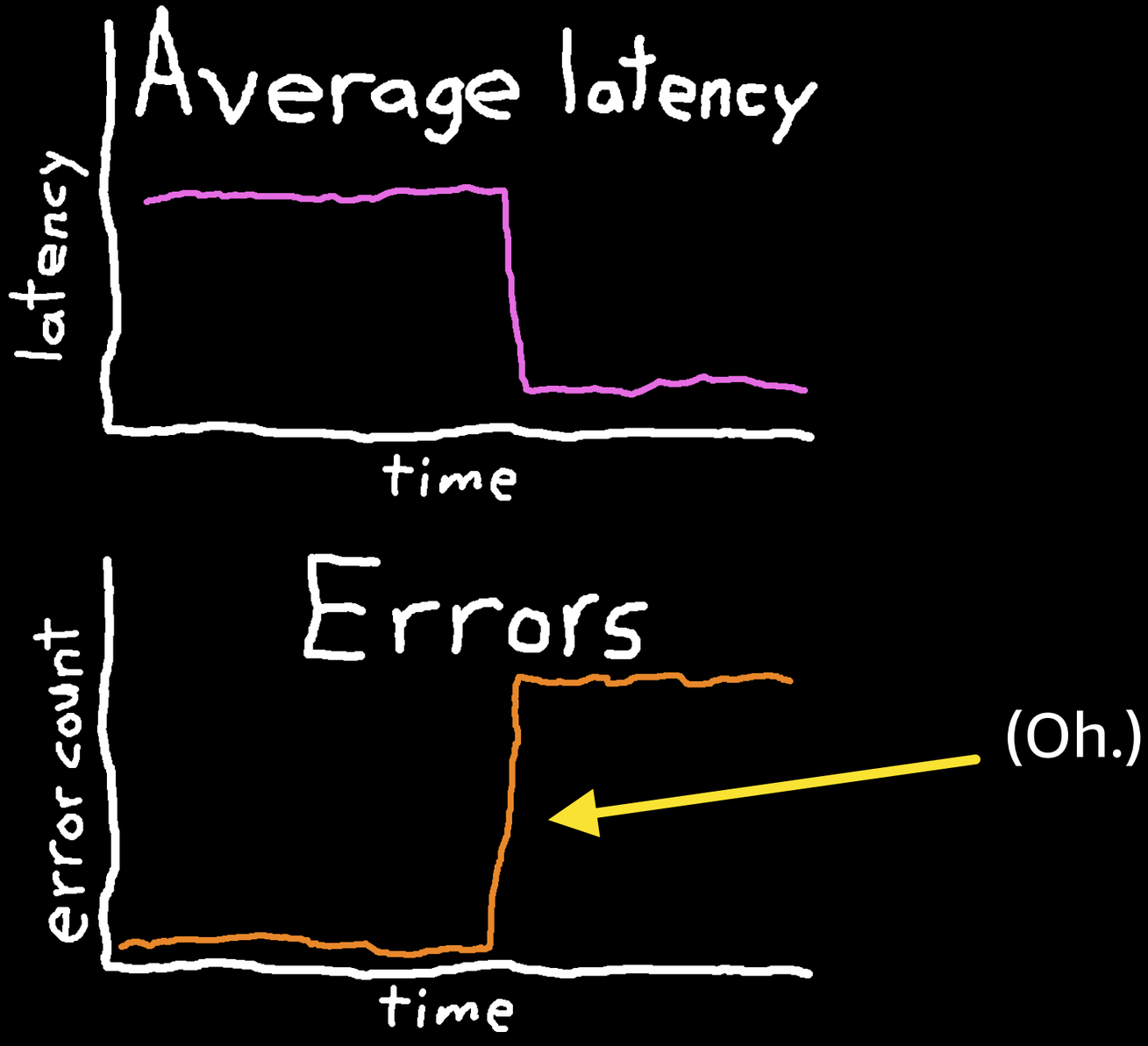


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

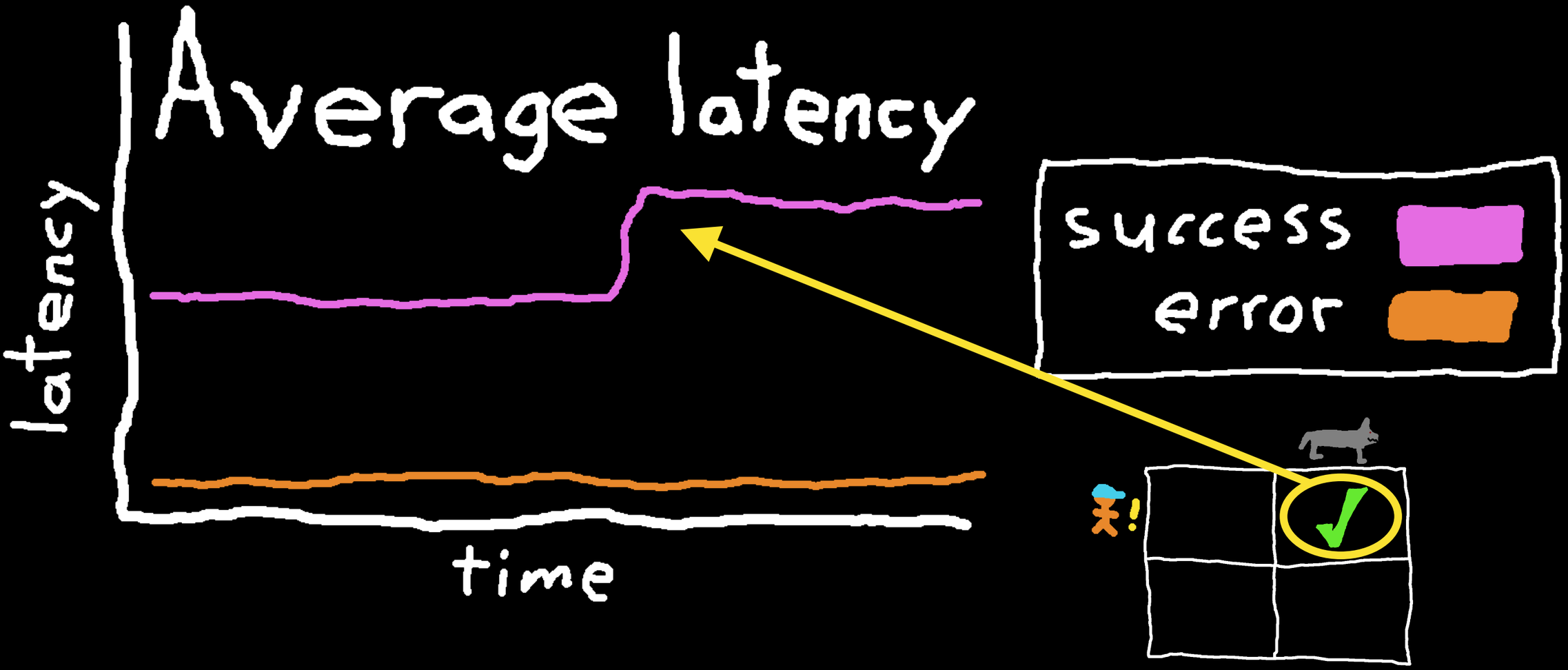
When latency metrics lie



Latency vs. errors



Success latency vs. error latency



Monitoring philosophy
Noise
When metrics lie

Summary statistics

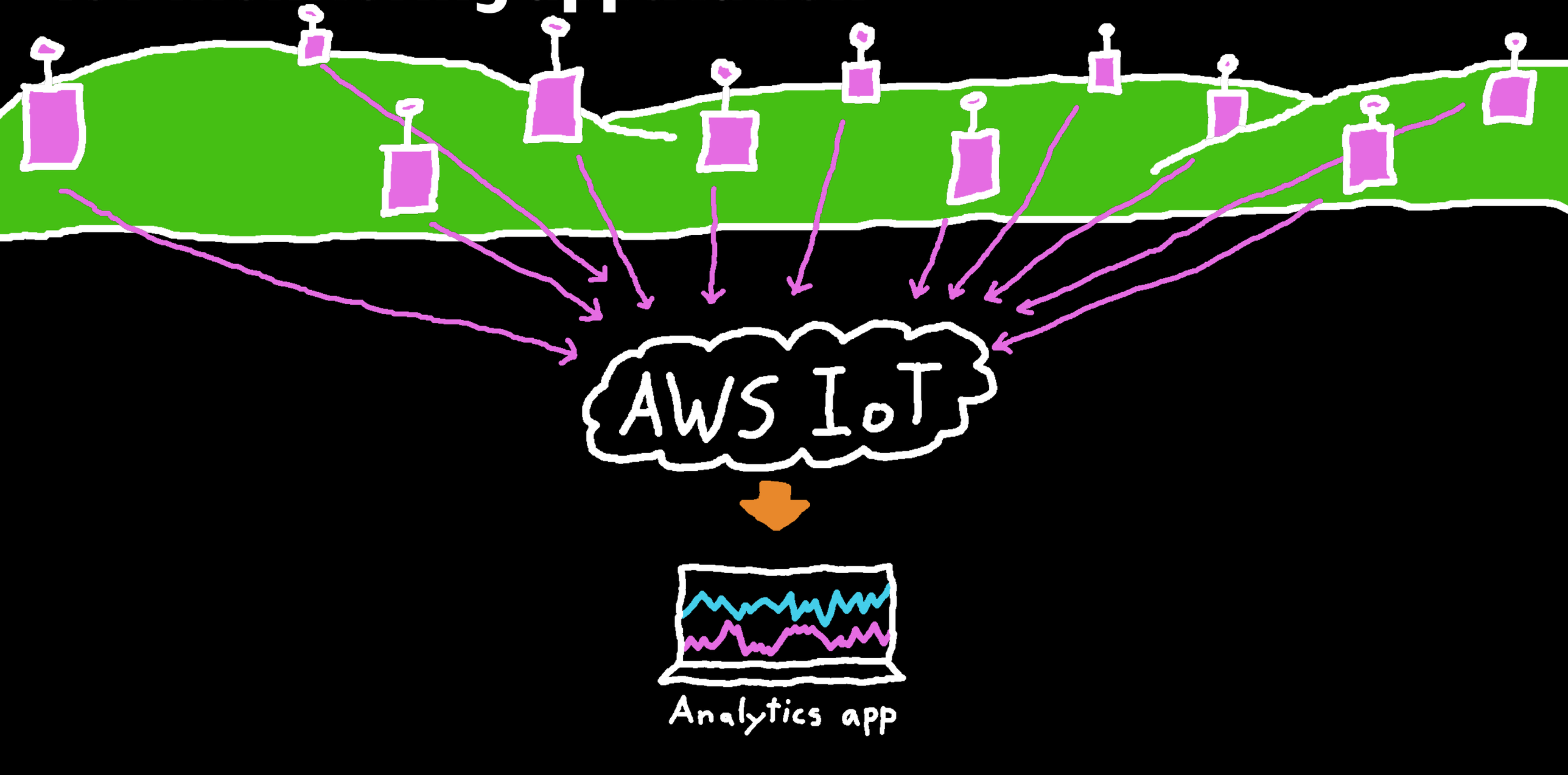
Misleading measurement

Coarse-grained

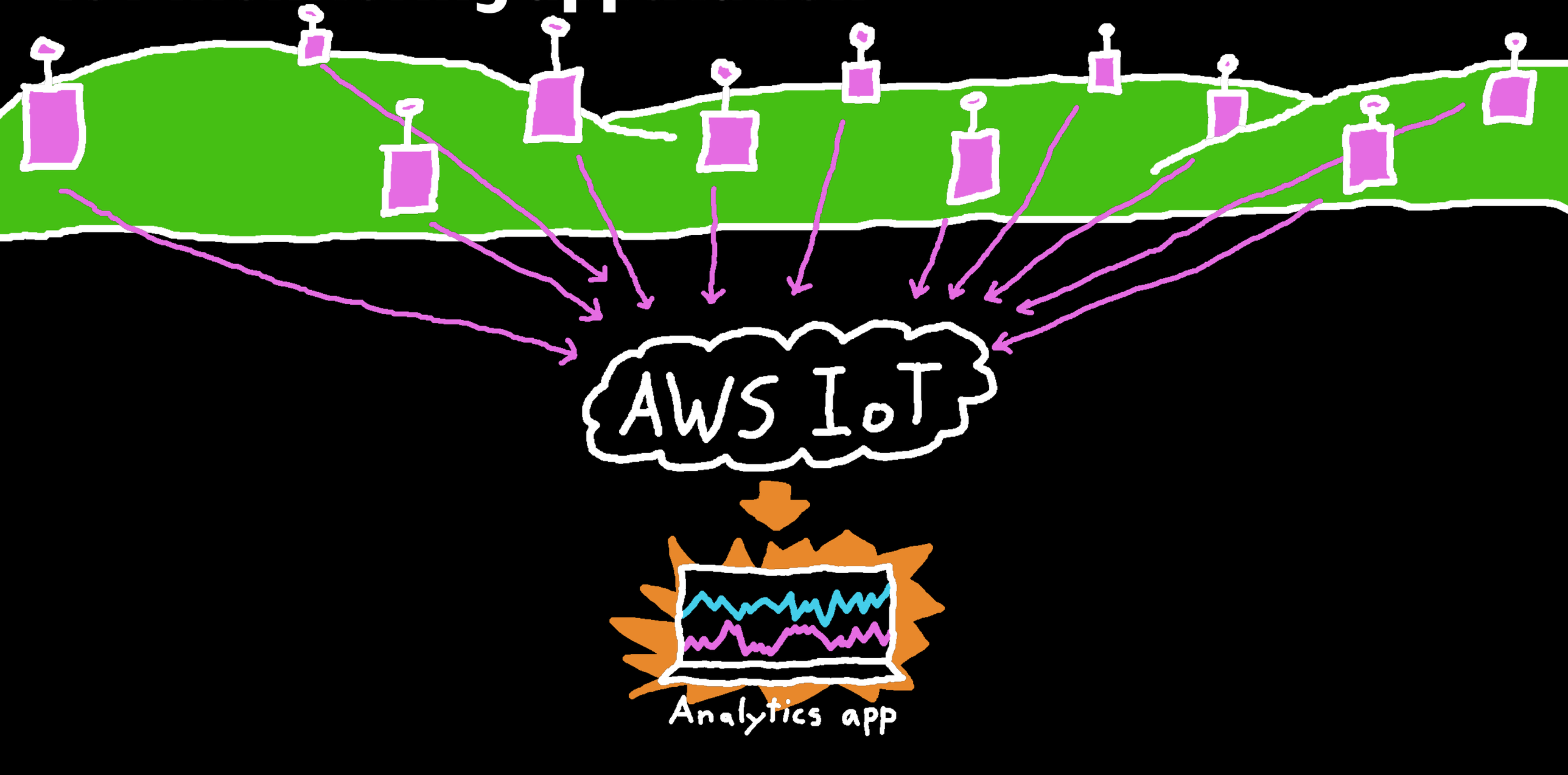


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

IoT monitoring application



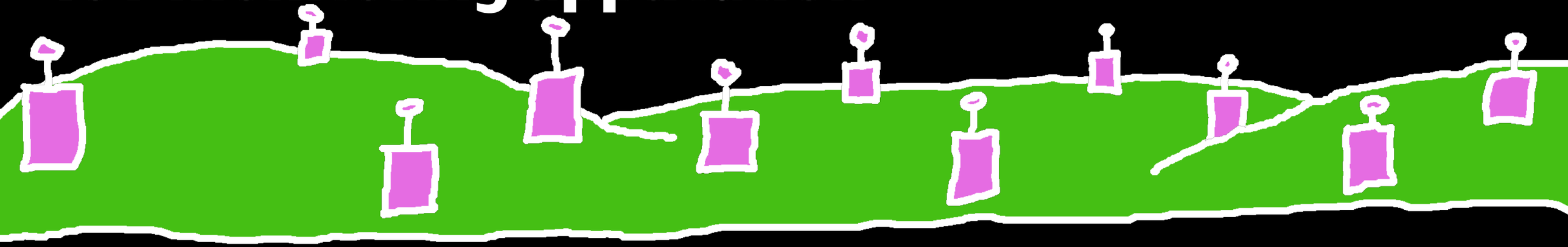
IoT monitoring application



When request rates are a lie

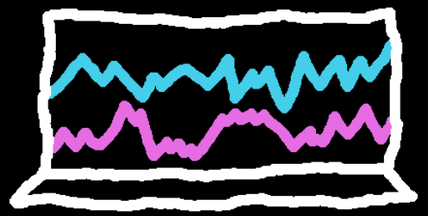


IoT monitoring application



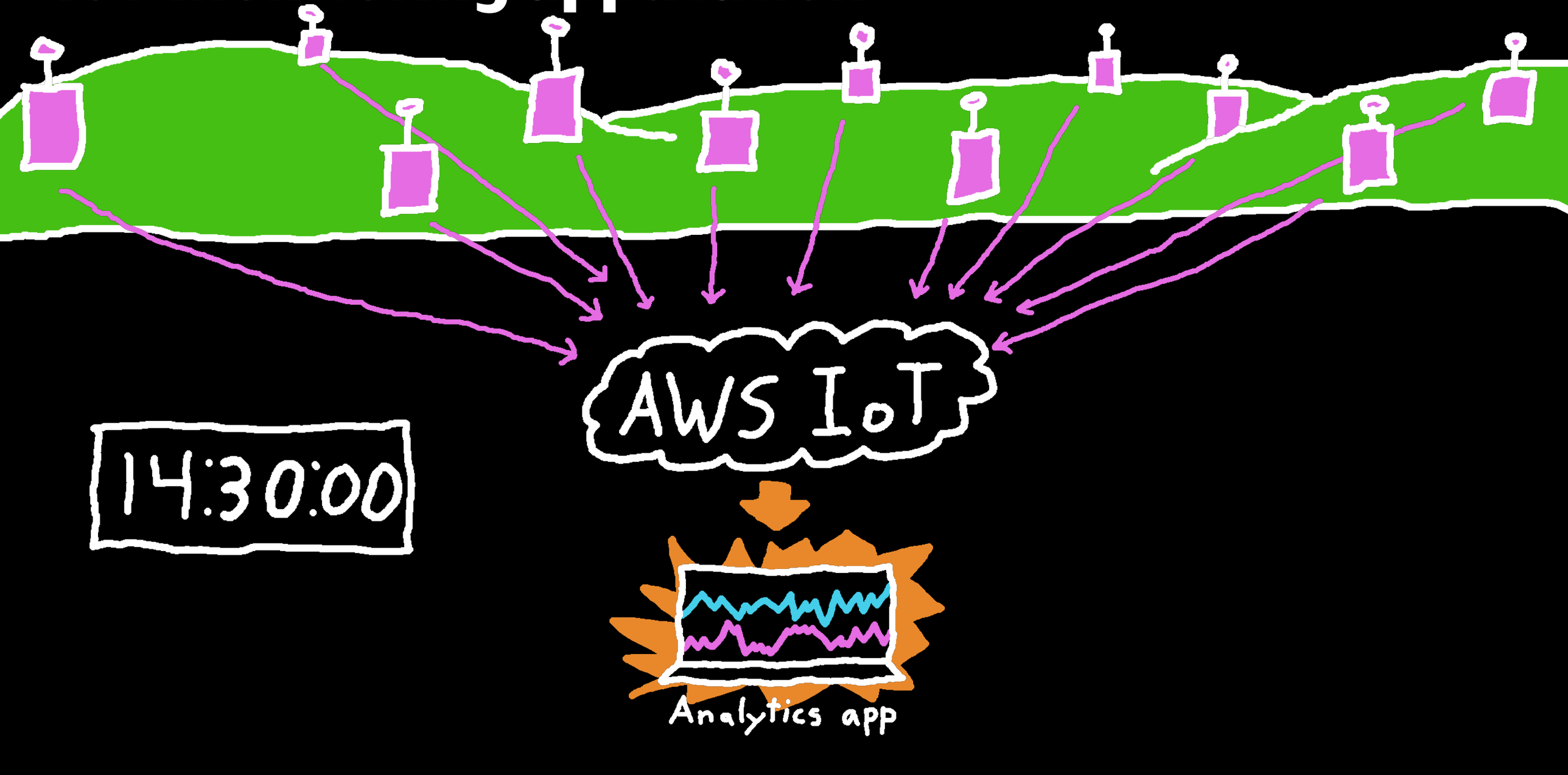
AWS IoT

14:29:59

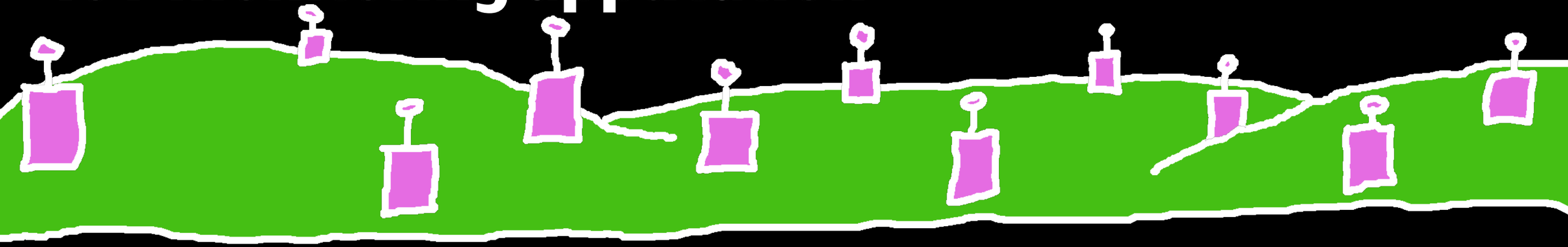


Analytics app

IoT monitoring application

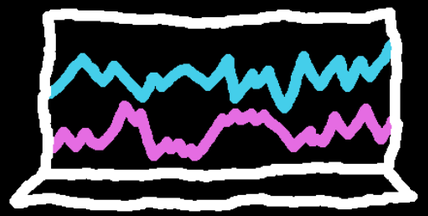


IoT monitoring application



AWS IoT

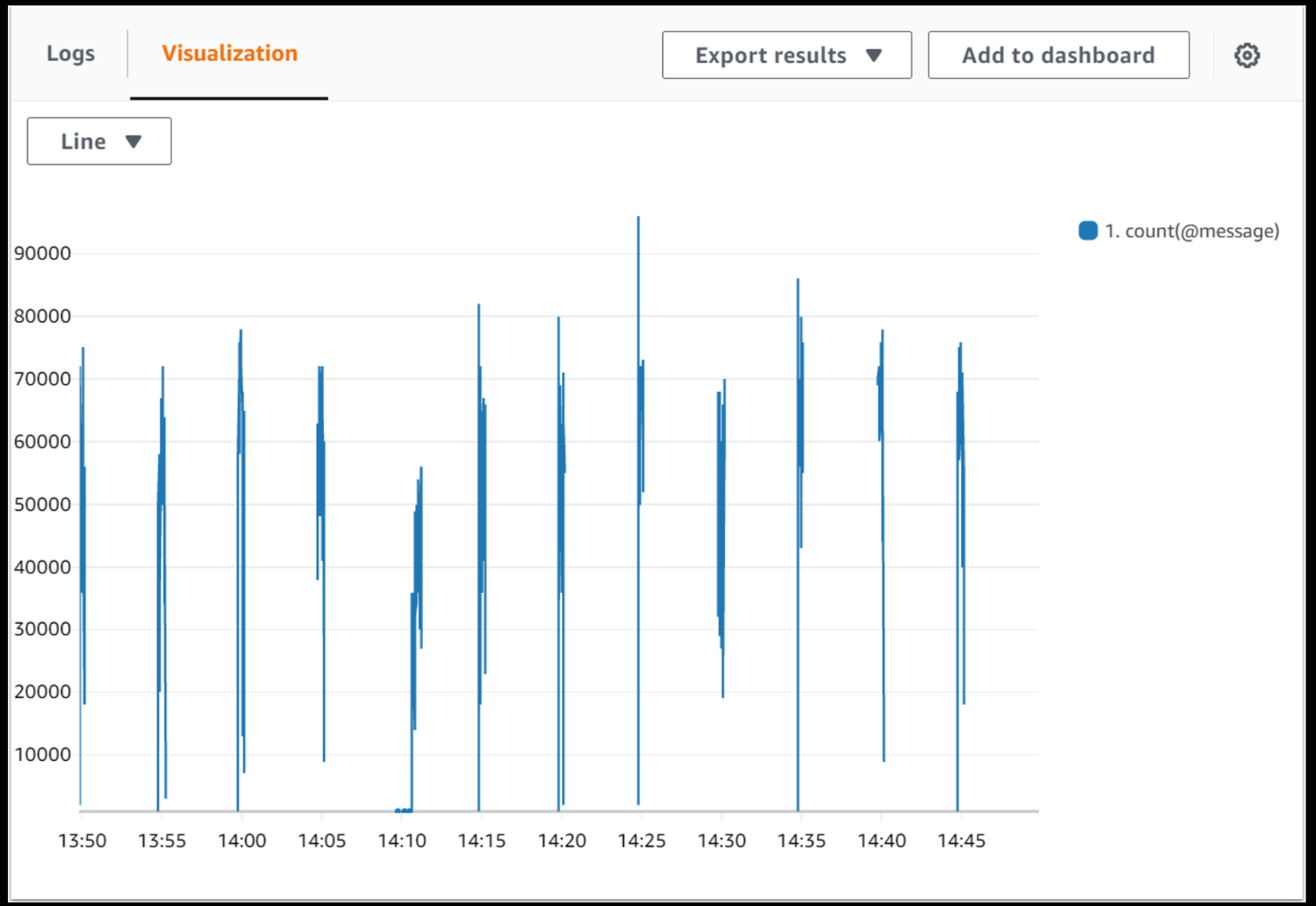
14:30:01



Analytics app

When request rates are a lie

```
1  fields @timestamp, @message
2  | sort @timestamp
3  | stats count(@message) by bin(1s)
4
```



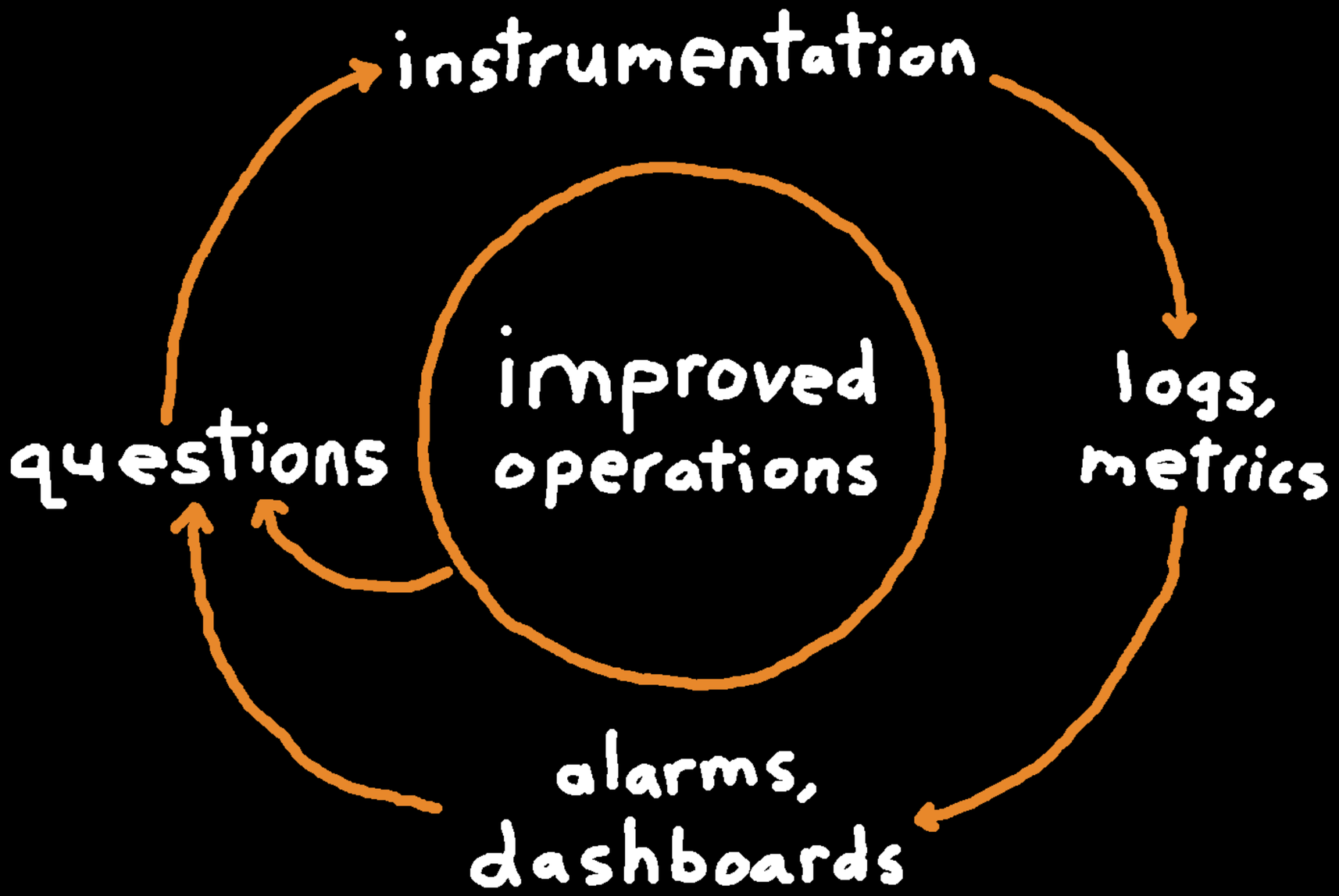
Monitoring philosophy
Noise
When metrics lie
Takeaways



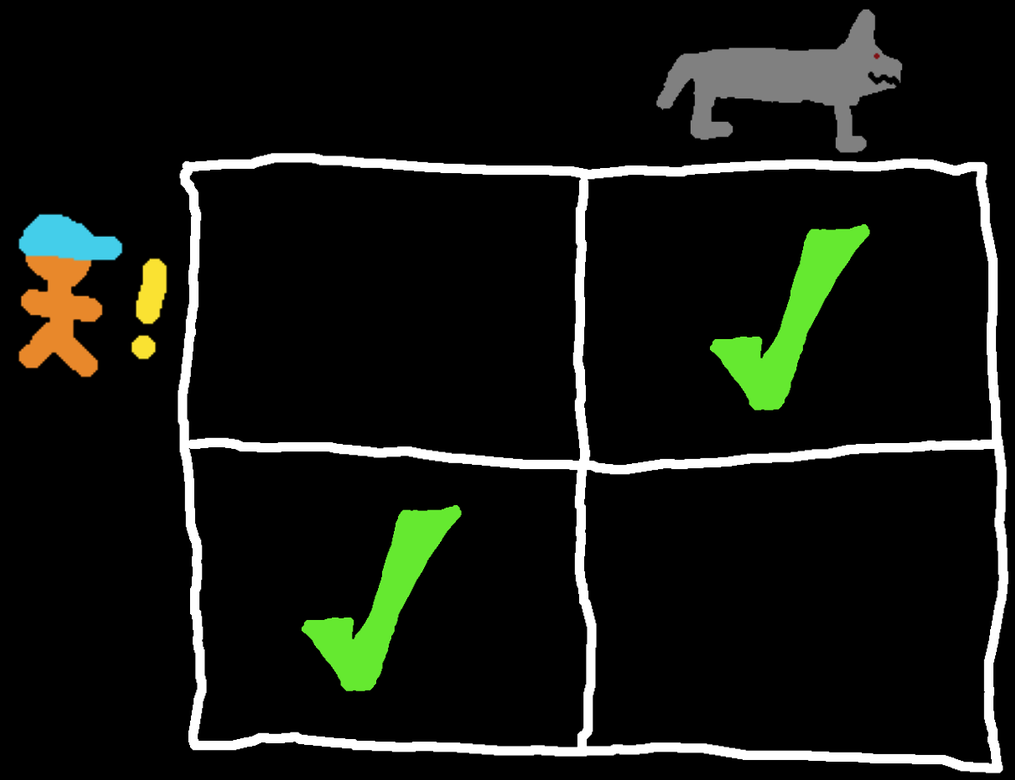
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Takeaways: Constantly improve



Takeaways: Avoid noisy or misleading metrics



Obligatory Amazon Builders' Library plug

SOFTWARE DELIVERY AND OPERATIONS NEW

LEVEL 300



Building dashboards for operational visibility

Author: John O'Shea

Building dashboards to monitor, dive deep, audit, and review distributed services and automated systems.

SOFTWARE DELIVERY AND OPERATIONS NEW

LEVEL 300



Automating safe, hands-off deployments

Author: Clare Liguori

Strategies for continuously deploying to production while balancing safety and speed.

SOFTWARE DELIVERY AND OPERATIONS

LEVEL 400



Instrumenting distributed systems for operational…

Author: David Yanacek

Gaining operational visibility into production systems, and troubleshoot failures with software instrumentation.

Thank you!

David Yanacek

Sr. Principal Engineer, AWS Lambda
@dyanacek

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.