Scout:
Improving the diagnosis process through domain-customized incident routing

Jiaqi Gao, Nofel Yaseen, Robert MacDavid, Felipe Vieira Frujeri, Vincent Liu, Ricardo Bianchini, Ramaswamy Aditya, Xiaohang Wang, Henry Lee, David Maltz, Minlan Yu, Behnaz Arzani

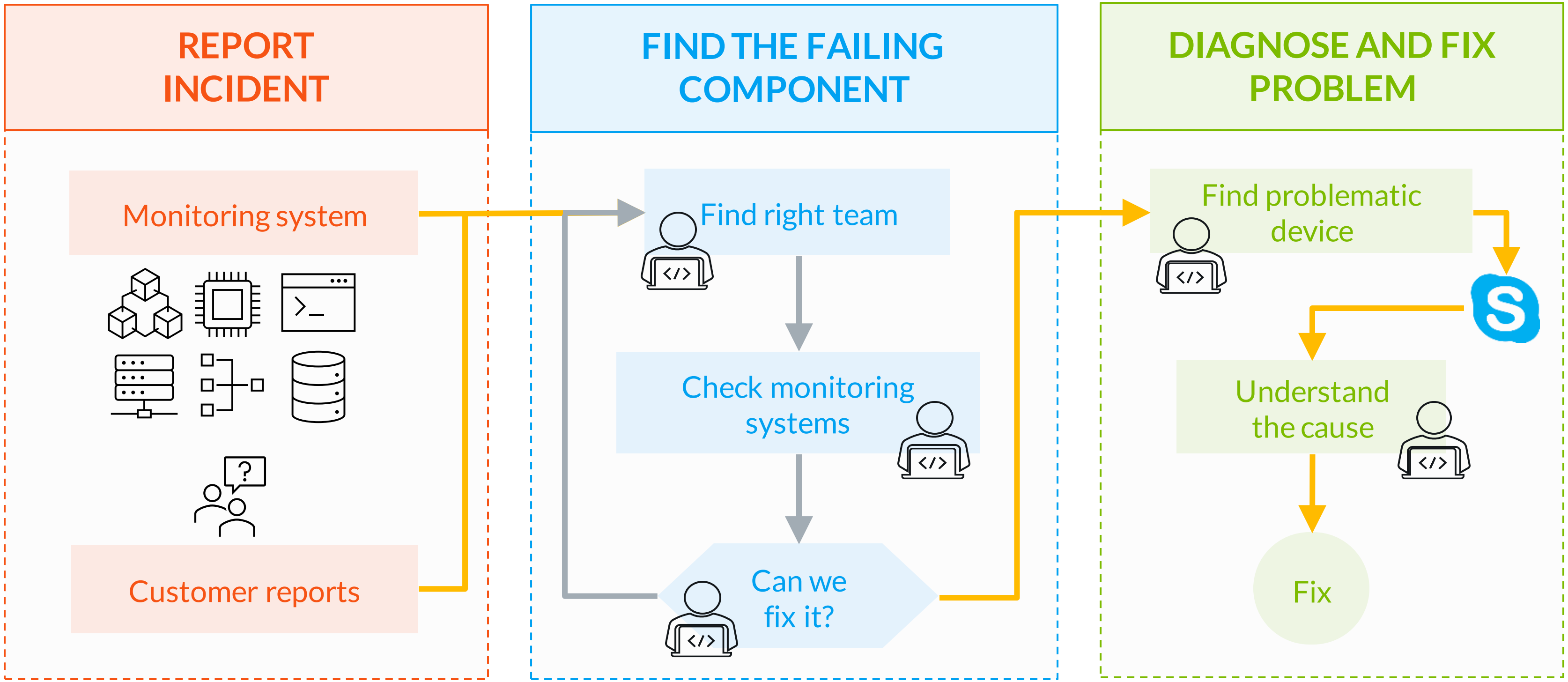
HARVARD UNIVERSITY Penn UNIVERSITY of PENNSYLVANIA Microsoft Research Microsoft

Availability and maintaining service level objectives (SLOs) are the biggest challenges facing cloud operators today

Incidents can and do happen

Number of public incidents between February to July 2020	69	21
Maximum resolution time	14 h 12 m	19 h 49 m
Average resolution time	4 h 40 m	5 h 28 min

Life cycle of an incident



REPORT INCIDENT

Monitoring system

Customer reports

FIND THE FAILING COMPONENT

Find right team

Check monitoring systems

Can we fix it?

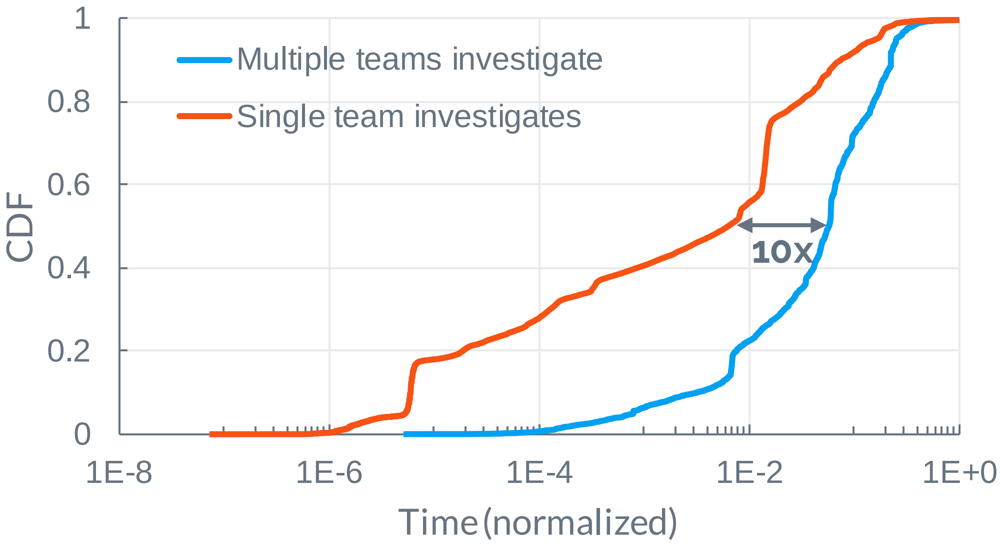
DIAGNOSE AND FIX PROBLEM

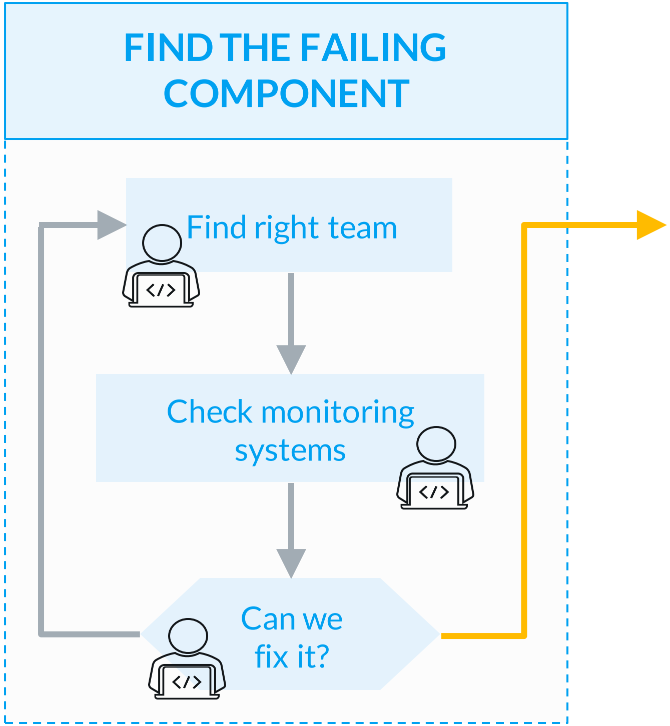
Find problematic device

Understand the cause

Fix

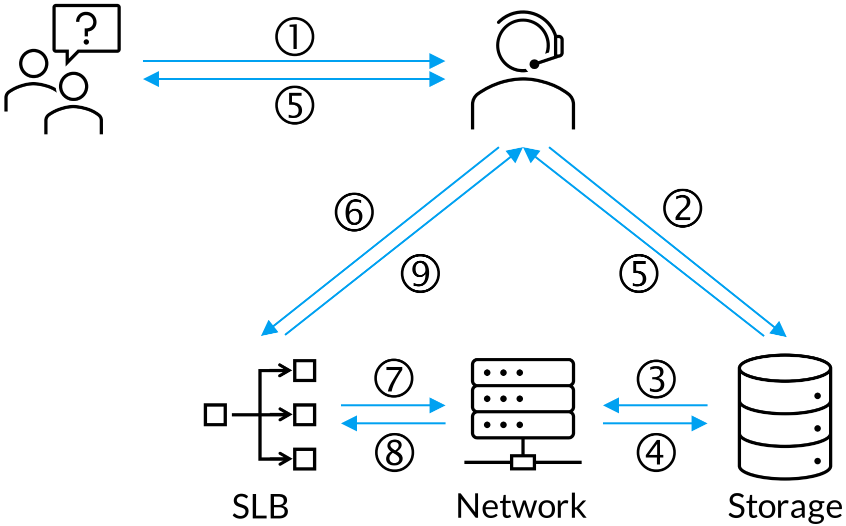
Finding the right team is time consuming





Example incident: storage problem

1. Can’t write to storage!
2. Must be storage issue
3. Storage is good, network must be slow
4. No congested links
5. Need more information from customer
6. Connection fail to init, SLB failing
7. SLB is good, network must be dropping
8. Packet is reaching to SLB
9. Customer opens too many connections and exhaust SNAT pool, behavior is expected

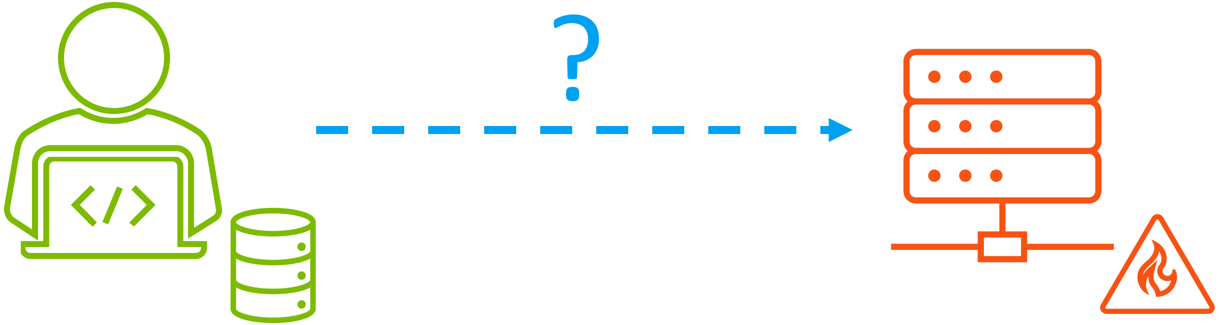


Why multiple teams get involved?

Studied 200 misrouted incidents in Azure

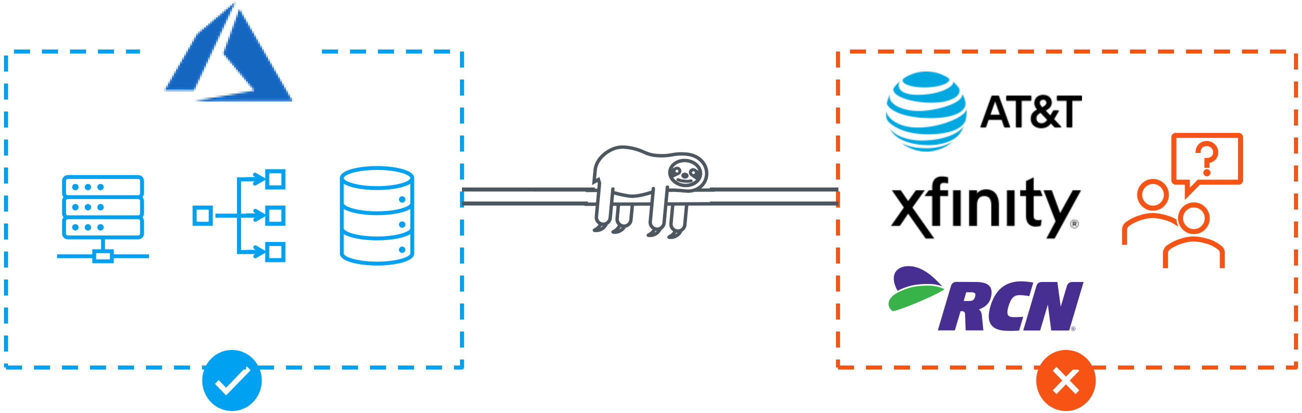
1. Lack of domain knowledge

- Storage team doesn’t know network is functioning or not



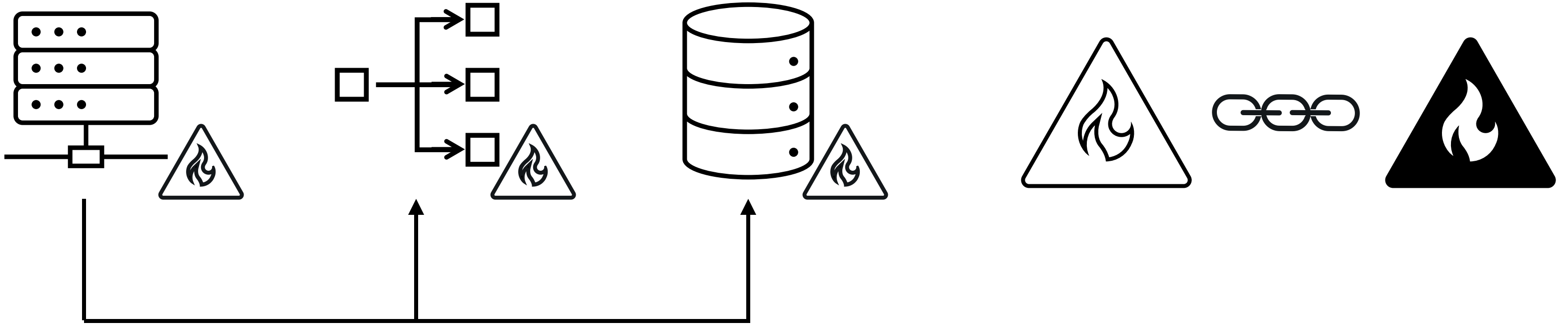
2. No cloud teams are responsible, more misrouting

- ISP or customer outside the cloud is experiencing issues



3. Concurrent incidents

▷ One failure causes multiple incidents in multiple teams



How to reduce misrouting?

Existing solutions

Application specific diagnosis system

NetPoirot
[SIGCOMM-16]

DeepView
[NSDI-18]

Sherlock
[SIGCOMM-07]

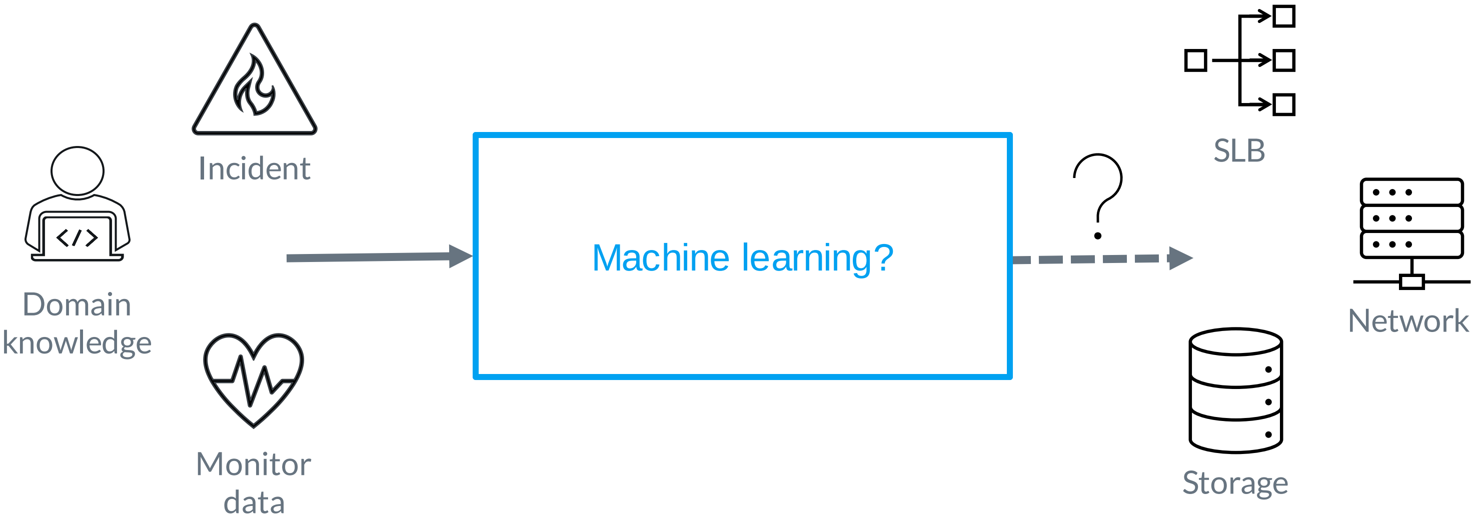
Too many applications in the data center

Natural language processing

NetSieve
[NSDI-13]

Ignores essential domain knowledge

Incident routing problem revisit



Solve the whole problem at once?

- Hard to build a single, monolithic incident routing system

Curse of dimensionality

Huge feature vector with no enough training examples

Uneven instrumentation

A subset of teams will always have gaps in monitoring

Constantly changing

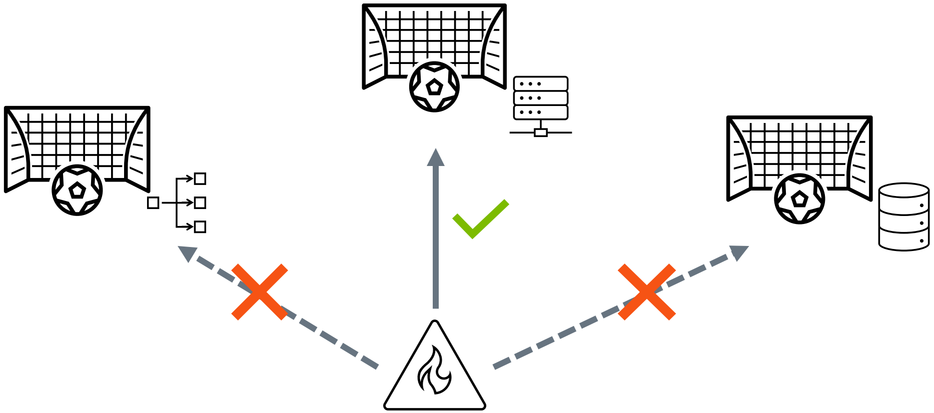
Stale components and monitors

Limited visibility

Hard to understand appropriate feature set for each team

Scout: team-specialized ML-assisted gatekeeper

- “Is my team responsible for the incident?”

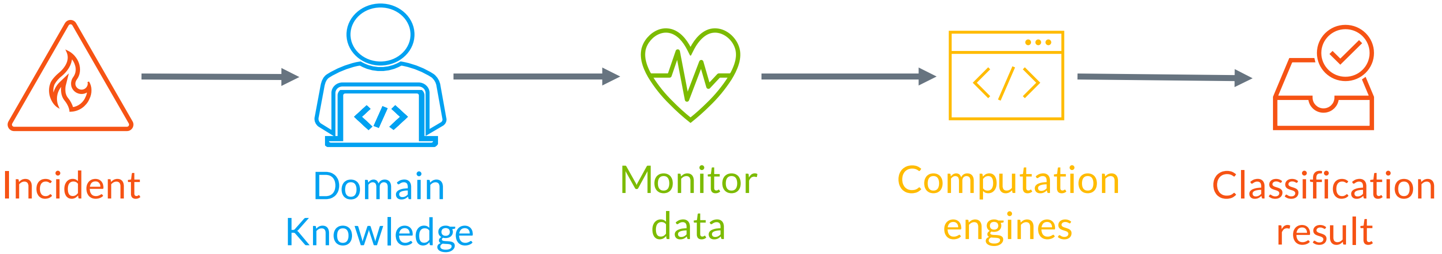


One team, one scout

Leverage domain knowledge

Evolve independently

Scout design



Physical networking team

Scope

Every switch & router in DC

Monitors

PingMesh, Everflow, NetBouncer, etc.

Statistics

58% incidents investigated by PhyNet went through multiple teams

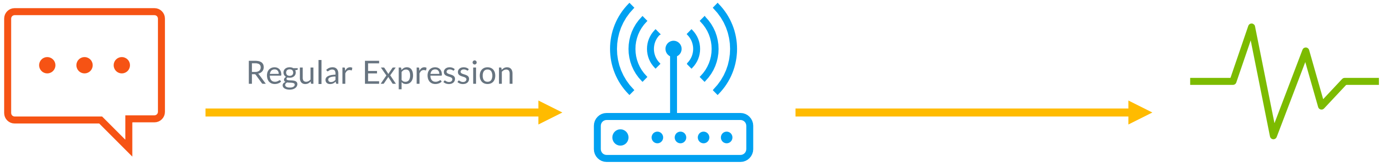
97.6 hours per day wasted on unnecessary investigations

CHALLENGE 1

▷ *How to process huge amount of monitoring data?*

Millions of devices in the Cloud

Incident guided investigation



Regular Expression

Incident Description

Devices

Monitor data

“Server X.c10.dc3 is experiencing problem connecting to storage cluster c4.dc1”

Server: X.c10.dc3
Cluster: c4.dc1

CPU Usage
Link loss rate
Ping latency

CHALLENGE 2

- *How to create a feature vector out of the monitoring data?*

Variable number of devices

Mixed types of monitoring data

How to build a fixed width feature vector?

- Per-component feature
  - Event: count number of events
  - Time-series: normalize and calculate statistics (percentiles, average, etc.)
- Multiple components
  - Compute statistics across multiple components (percentiles, average, etc.)

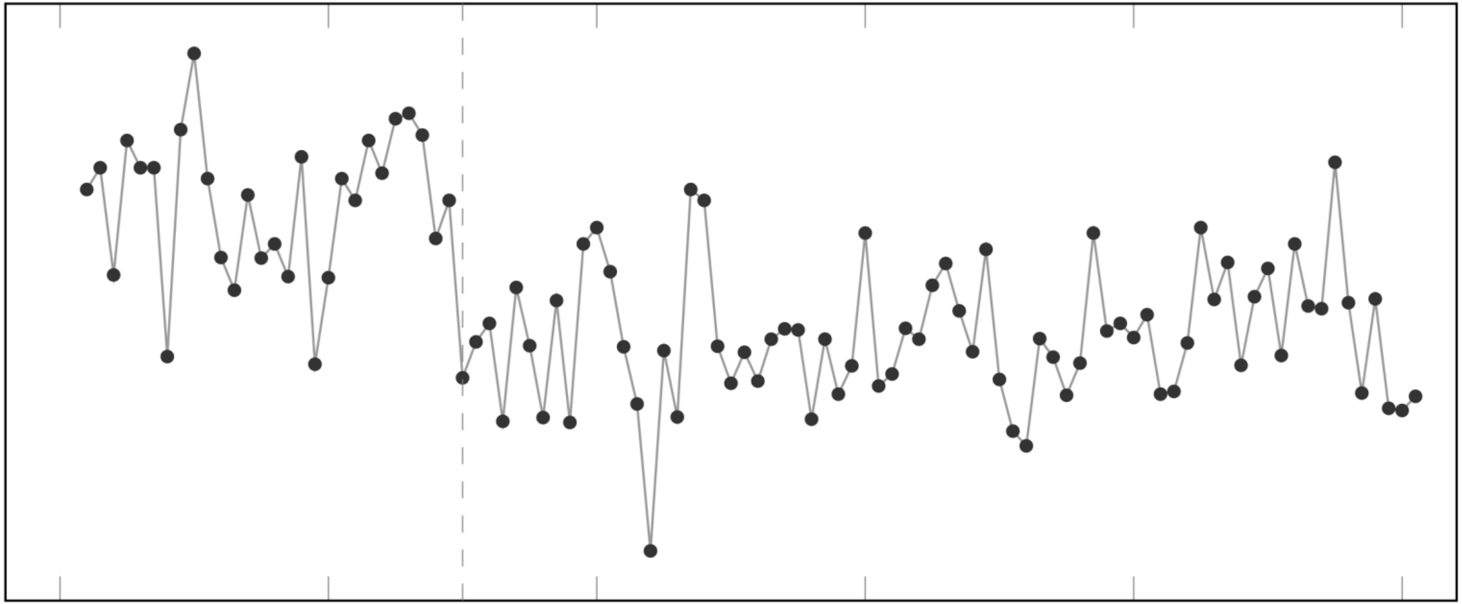
CHALLENGE 3

- ▷ *Which computation engine?*

Supervised learning: random forest

- 🙂 Learns based on history incidents, high accuracy
- 😣 Low accuracy on new incidents
- 🙂 Interpretable, able to provide more insights

Change point detection for new incidents



Change point detection for new incidents

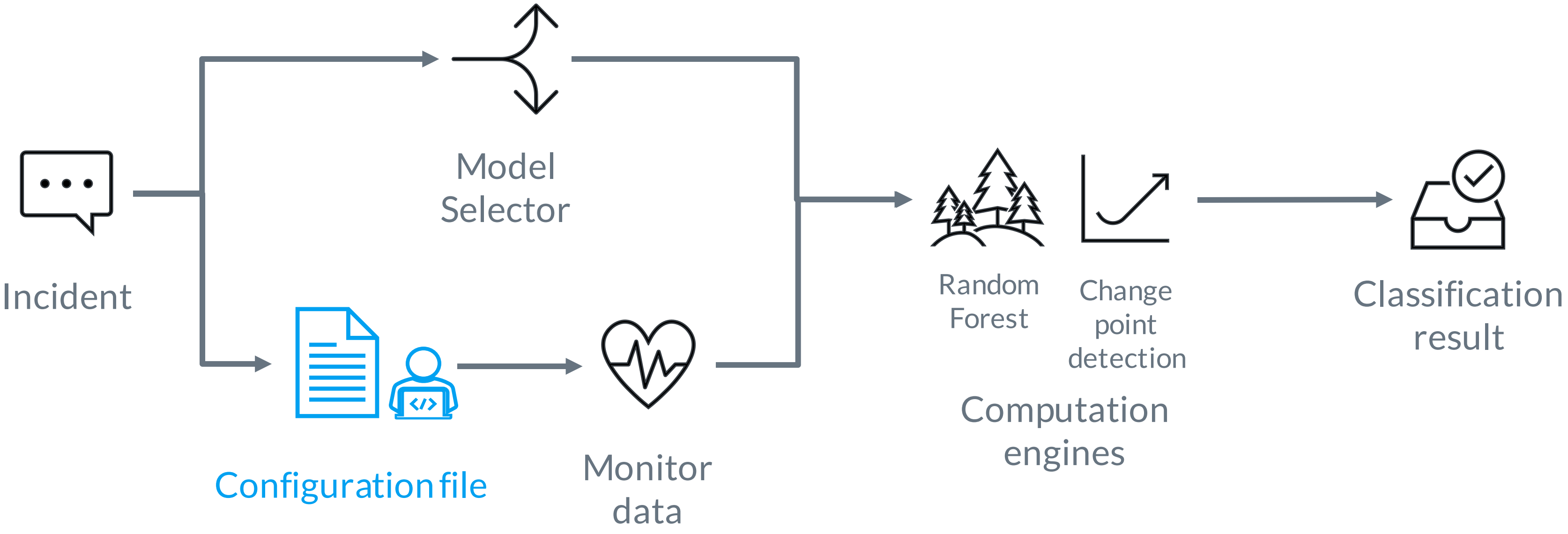
- 🙂 Easy to compute
- 🙂 Higher accuracy on new incidents
- 😣 Low accuracy on old incidents

Model selector

☺ Incident itself tells whether it is new or not

☺ Use meta-learning to identify new incidents

The anatomy of a Scout



Evaluation

Evaluation setup

DATASET

9 months of incidents in Azure

Randomly split into training and testing set

LABEL

Whether incident is resolved by PhyNet

BASELINE

Current incident routing system without Scout

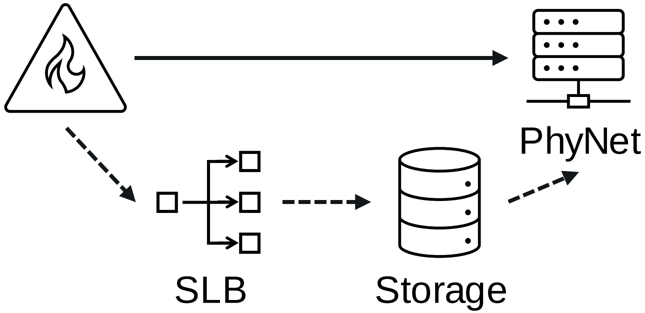
Runbooks, past-experience, NLP based routing system

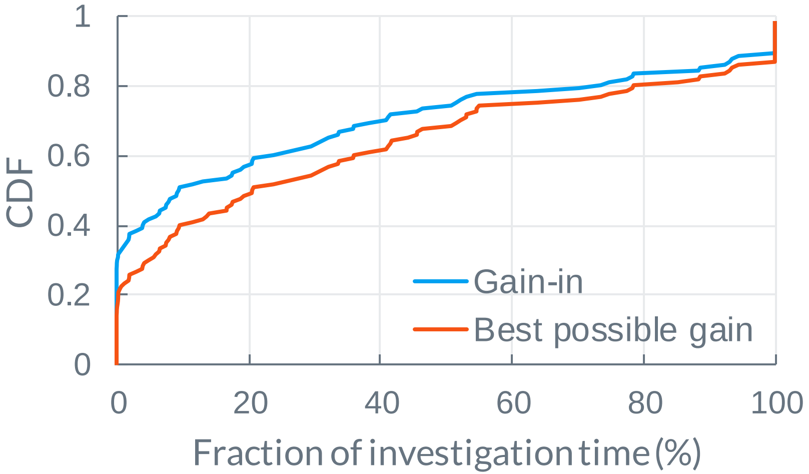
Overall performance

	Precision	Recall	F1-Score
Baseline	87.2%	91.9%	0.89
PhyNet Scout	97.5%	97.7%	0.98
Delta	10.3%	5.8%	0.09

10% improvement means ?

Gain in of the PhyNet Scout

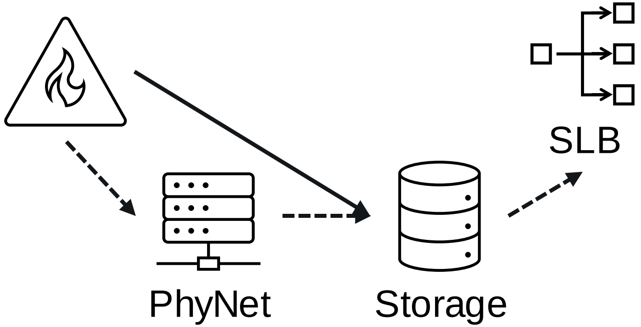


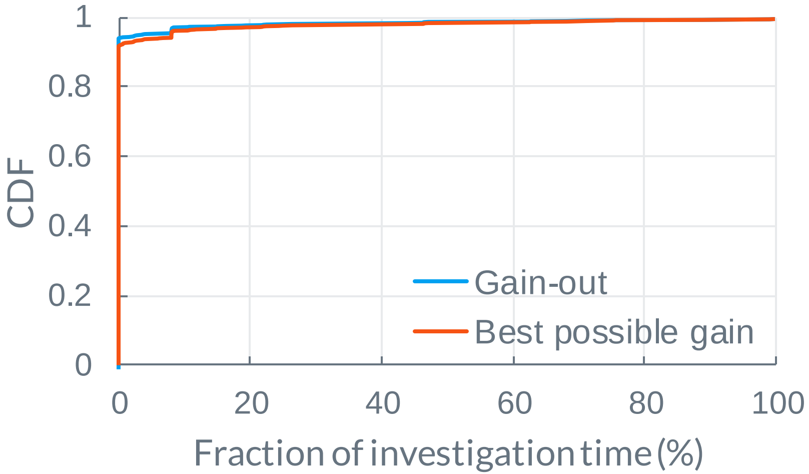


Gain in
Send incident to PhyNet directly

Save more than 20% of the total investigation time in 40% of incidents

Gain out of the PhyNet Scout





Gain out
Reject incident to PhyNet

Close to the best possible gain

Things we did not talk about

- The design and evaluation of Scout Master
- Extended evaluation
  - System performance over time
  - Sensitivity analysis
  - Other supervised learning algorithms
- Lessons we learnt from deployment

Please check our paper for more detail

Conclusion

- Incident misrouting is the main challenge for maintaining service level objectives in the cloud
- Scout: a distributed team-specialized gate-keeper can reduce investigation time.

Thanks!