Monitoring AI Pipelines as Product

Hila Fox

AUGURY

Hila Fox

Squad Lead &
Backend Developer

Agenda

- Augury - Company and Product
- Machine Health AI - How we do it at Augury
- The Detection Management layer
  - Why it’s important
- Monitoring
  - What do we want to achieve
  - The Hybrid approach
  - Being Proactive
- Conclusions

THE WORLD RUNS ON MACHINES

WE'RE ON A MISSION TO MAKE THEM RELIABLE

AUGURY

MACHINES TALK, WE LISTEN

www.augury.com

info@augury.com

How it works

Connect

Monitor 24/7 via IoT devices. Vibration, temperature and magnetic field sensors.

Diagnose

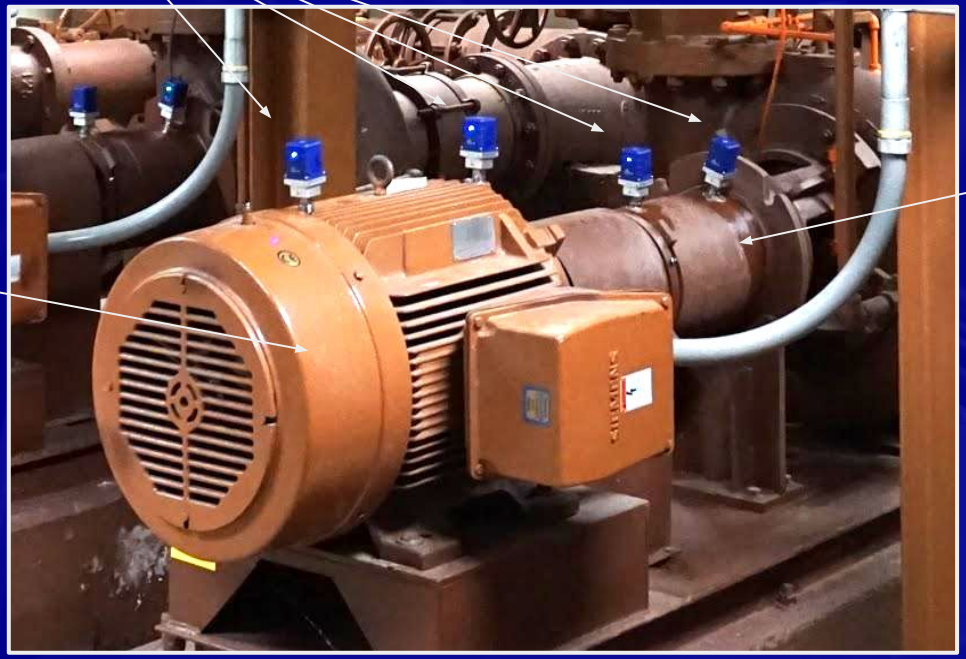
Algorithms diagnose malfunctions and provide alerts and recommendations

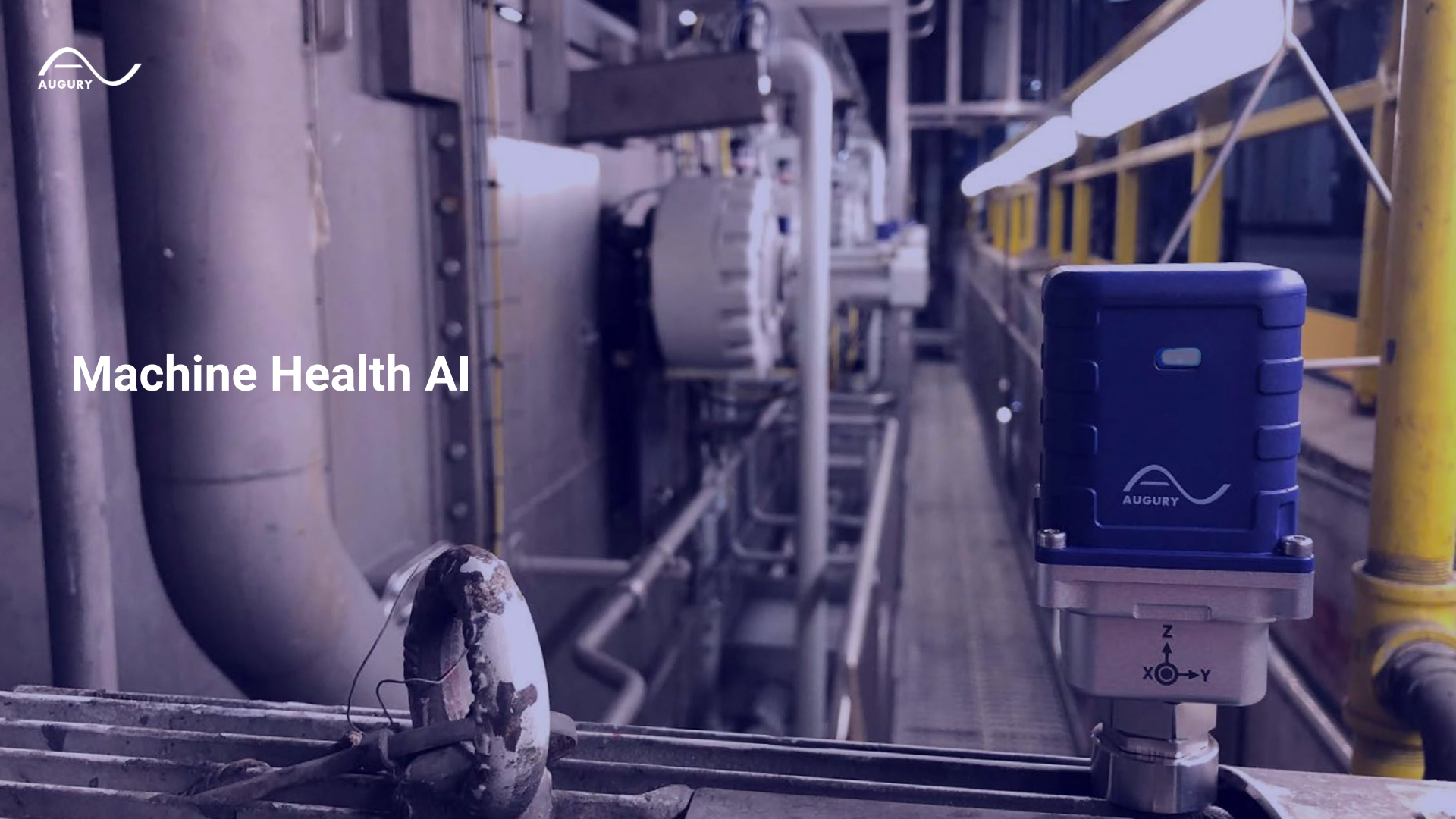
Visualize

Visualization and communication includes a web, mobile, emails, sms and more.

How does it look

How does it look





Machine Health AI

AUGURY

Some Important Numbers

50M+ Hours of Machines Monitored

80K+ Machines Diagnosed

Multiple Customers with Global Expansion

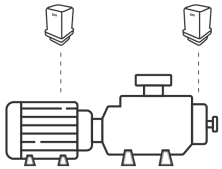
Dozens of Thousands of Machines

Dozens of ML Algorithms

3 Product Squads

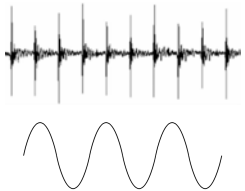
3 Algo Squads + Data Engineering Team

Augury Machine Health AI Flow



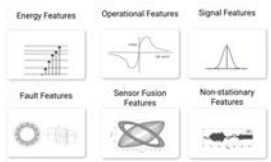
Sensors data
Operational data

1.3M points/hr/machine



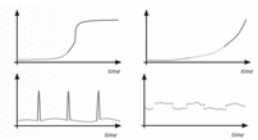
Validity and DSP pipeline

Time, FFT, Hilbert, Cepstrum



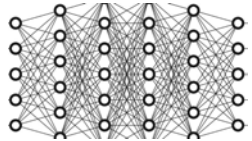
Feature extraction pipeline

~ 1000 features/hr/machine



Time series algorithms pipeline

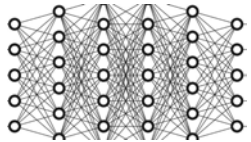
Time dependent, ops idx, stats



Algo detectors
Based on **50M** machines hrs

Machine learning detectors

Augury Machine Health AI Flow (cont.)



Algo detectors
Based on **50M**
machines hrs

Machine learning detectors

User

AI detection

Augury analyst

High Confidence, Critical Alerts

Low Confidence

0.13%

Detection Management

99.87%

Confidence Scores

Anomaly detection

Fault detection

Augury Machine Health AI Flow (cont.)

User

AI detection

Augury analyst

High Confidence, Critical Alerts

0.13%

Low Confidence

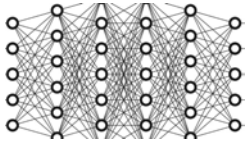
Detection Management

99.87%

Confidence Scores

Anomaly detection

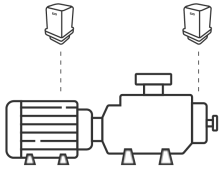
Fault detection



Algo detectors
Based on **50M** machines hrs

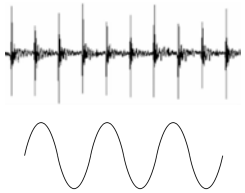
Machine learning detectors

Augury Machine Health AI Flow (cont.)



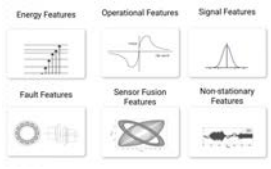
Sensors data
Operational data

1.3M points/hr/machine



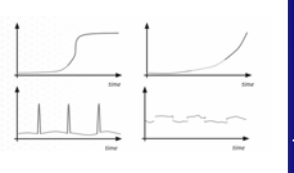
Validity and DSP pipeline

Time, FFT, Hilbert, Cepstrum



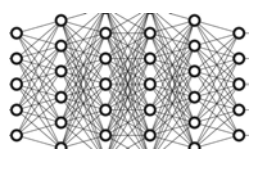
Feature extraction pipeline

~ 1000 features/hr/machine



Time series algorithms pipeline

Time dependent, ops idx, stats



Algo detectors
Based on **50M** machines hrs

Machine learning detectors

User

AI detection

Augury analyst

High Confidence, Critical Alerts

Low Confidence

0.13%

Detection Management

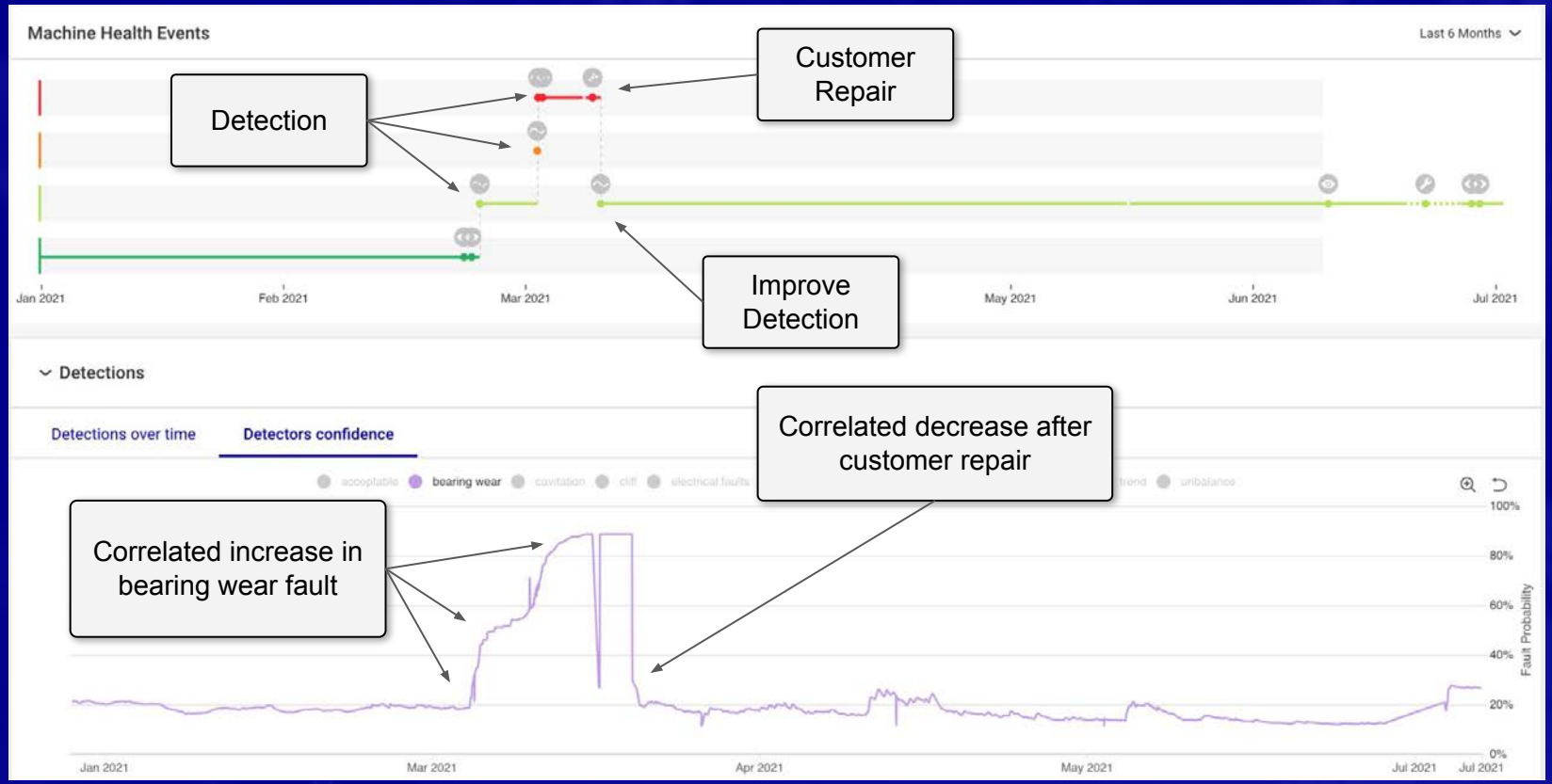
99.87%

Confidence Scores

Anomaly detection

Fault detection

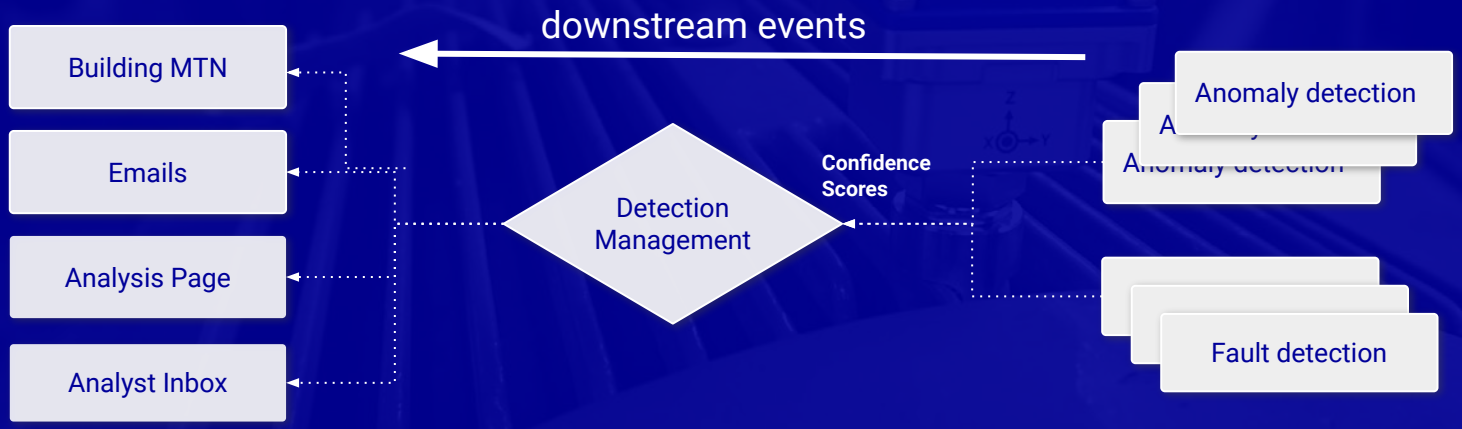
How We Use Detections In The Product



The Detection Management Layer

Why Is It So Important

- Connects the AI engine to customer facing products
- Contains logic which effects product decisions



AUGURY

Being Confident in the Changes We Make

Expected changes (new features) :

- **AI Engine** - Test in staging environment (was already happening)
- **Detections Management Layer** - Test in dry run in production environment

Unexpected changes (bugs) :

- **AI Engine** - Add metrics and alerts on CPU, Queue lag and more
- **Detection Management Layer** - Add metrics on the detection flow

AUGURY

Our motivation - Avoid Production Issues

Simple Bugs

Bad Deployments

Change in Interfaces Between Squads

Negative Effects From Configuration Changes

Changes in Logic Based on Wrong Assumptions

* Due to the nature of downstream flows, an error in the top of the funnel can cause major issues to several consumers


All detection
AUGURY
Detection reached state
Detection finish state
Monitoring!!!
Detections breakdown
Filtered vs non filtered with breakdown
pass 68
didnt pass filter_demo 13
didnt pass filter_other_detectors_confidence 13
didnt pass filter_already_in_danger 3
Filtering filters
filter_demo 13
filter_other_detectors_confidence 13
filter_already_in_danger 3
filter_machine_is_not_halo_in_baseline 3
Filtered detectors
acceptable 16
feature_thresholds 13
bearing_wear 4
cliff 3
Total per filter
filter_raw_detector 106
filter_demo 106
filter_acceptable_machines 93
filter_halox 93

What Do We Want To Achieve

Good Service/Support

- It’s the core of our product
- Catch issues before customers
- Find issues as fast as possible

What Do We Want To Achieve

Good Service/Support

- It’s the core of our product
- Catch issues before customers
- Find issues as fast as possible

Have Consistent AI Insights

- Quality of our insights
- Find Machine Health Issues
- Minimize False Alerts

What Do We Want To Achieve

Good Service/Support

- It’s the core of our product
- Catch issues before customers
- Find issues as fast as possible

Have Consistent AI Insights

- Quality of our insights
- Find Machine Health Issues
- Minimize False Alerts

Better Collaboration between our teams

- Growing from 8 people to 7 squads
- Improve communication
- Enable Fast response

What Do We Want To Achieve

Good Service/Support

- It's the core of our product
- Catch issues before customers
- Find issues as fast as possible

Have Consistent AI Insights

- Quality of our insights
- Find Machine Health Issues
- Minimize False Alerts

Top Goal

Retain Trust From our Customers

Better Collaboration between our teams

- Growing from 8 people to 7 squads
- Improve communication
- Enable Fast response

AUGURY

Types of Monitoring

- **White box monitoring** — Monitoring based on metrics exposed by the internals of the system, including logs, interfaces like the Java Virtual Machine Profiling Interface, or an HTTP handler that emits internal statistics (CPU, memory usage and more).
- **Black box monitoring** — Testing externally visible behavior as a user would see it.

* According to google SRE book

The Hybrid Approach
Monitoring Pipelines as a Product

Blackbox Monitoring

+

Whitebox Monitoring

=

The Hybrid Approach
Monitoring Pipelines as a Product

Blackbox Monitoring

+

Whitebox Monitoring

=

The Detection Management Layer:

- **Customer/Consumer** to the AI Engine.
- Executes **product logic**
- Decides on the **detections states**

Conclusion: even though it's an **internal product process**, this layer is the **decision flow** for what the external customers get.

Patterns We Commit To

Consistent # of Propagations

- Considering detection confidence
- Considering machine state

Consistent Detections Generation for each detector

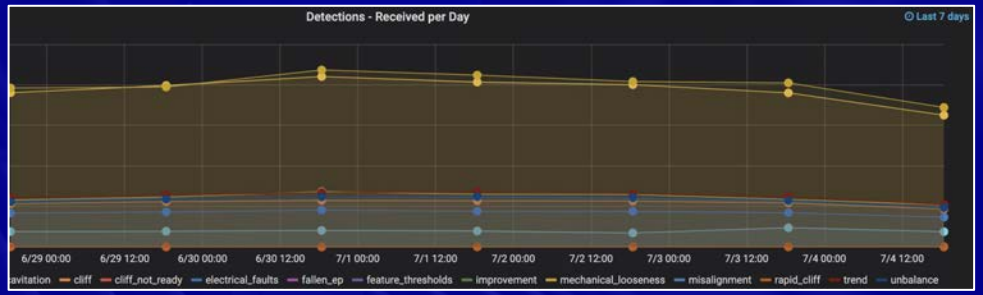
Detection lifecycle

- Init (reached detection management layer)
- Filtered by detection confidence
- Filtered by machine state
- Propagated to customers

Detection lifecycle

- Init (reached detection management layer)
- Filtered by confidence
- Filtered by logic
- Propagated to customers

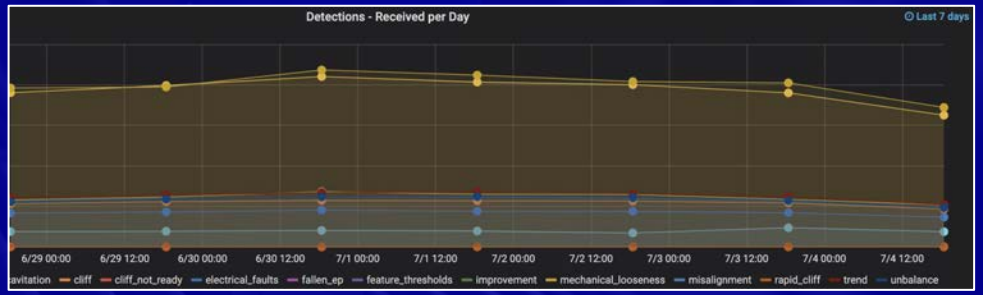
Using Graphite & Grafana



Detection lifecycle

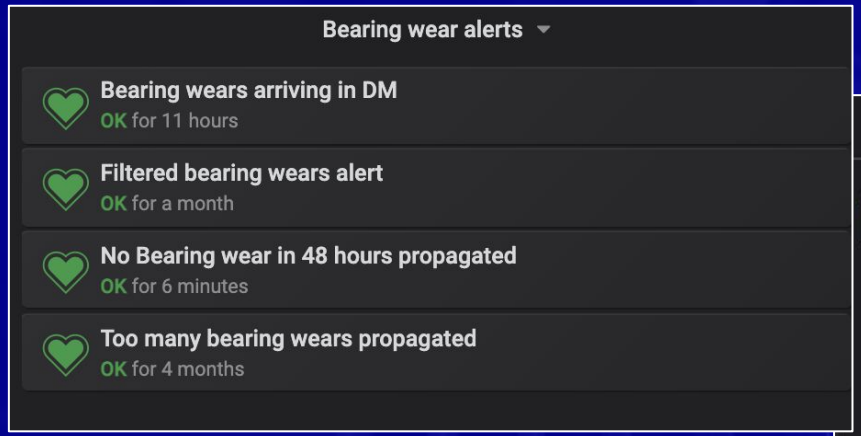
- Init (reached detection management layer)
- Filtered by confidence
- Filtered by logic
- Propagated to customers

Using Graphite & Grafana



Let's Get Proactive

- Setting up alerts per detector on breaking patterns
- Creating a collaborative slack channel for monitoring alerts



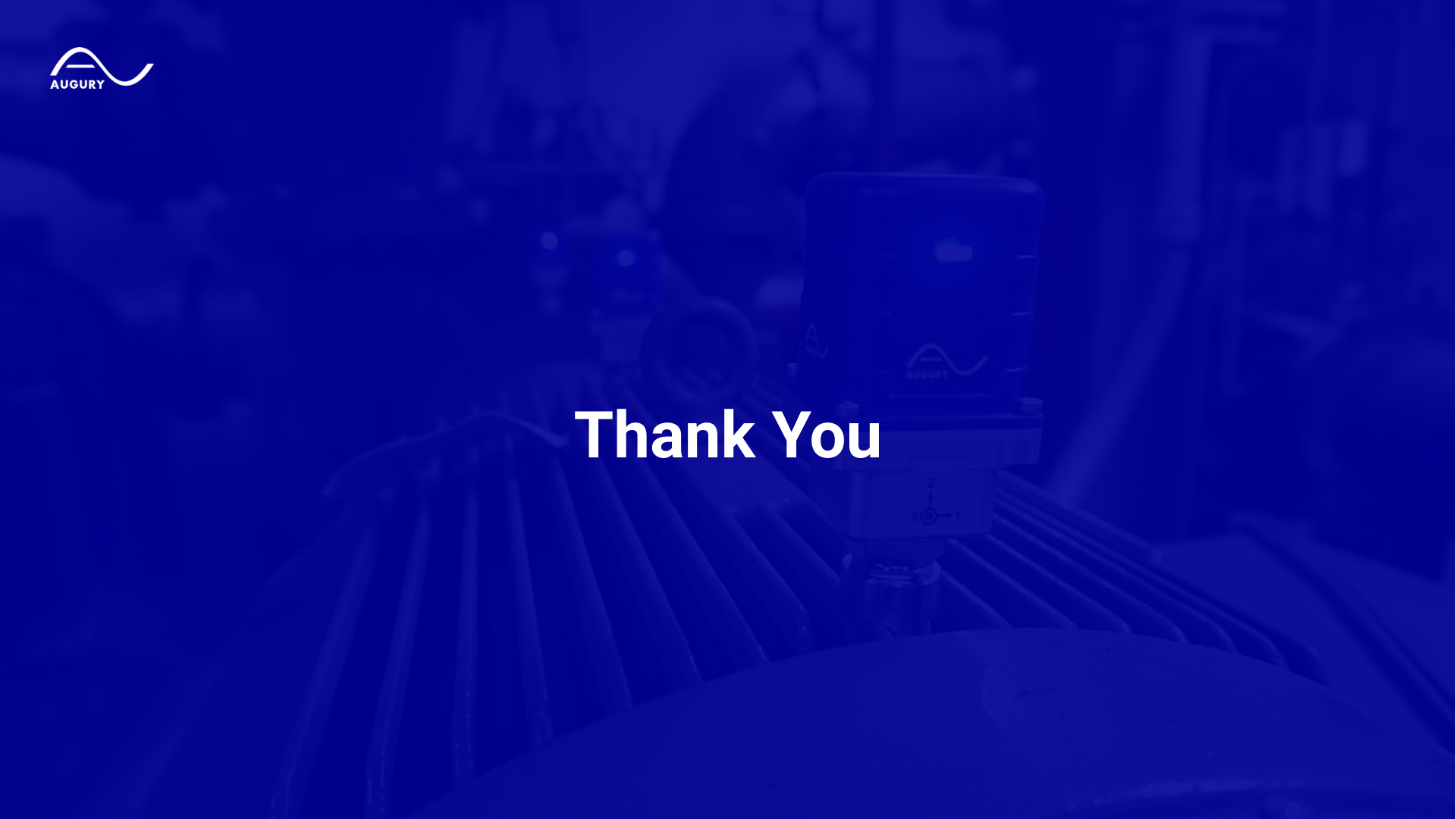
Example: Consistent Detections Generation Graph

- Lines indicating thresholds for alerts
- Purple line - extra information, deployment tags
  - enables fast understanding of what changed

AUGURY

Conclusion

- Keep the customers in the center (internal and external)
- Internal teams can consume products from each other
- It’s not about having a zero bugs product, it’s about fast response
- To move fast, we need high confidence in our process
- Having an easy way to communicate across teams is crucial


AUGURY
Thank You