Dynatrace

Innovation & Strategy

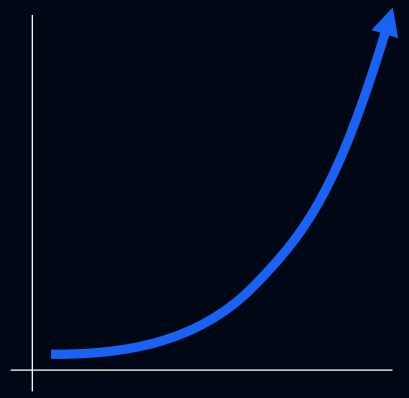
PRESENTER

Roman Spitzbart

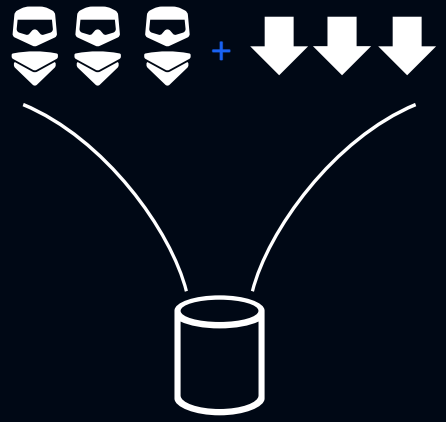
VP Solutions Engineering EMEA

MACRO TRENDS SHOW TOOLS ARE BREAKING

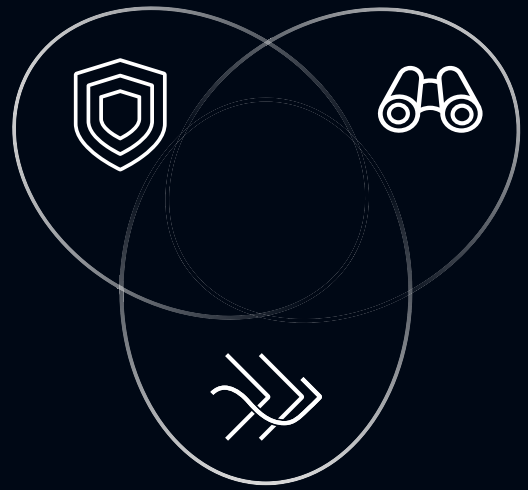
Data collection explosion



Data ingest grows significantly



Observability + security, automation convergence



Enterprise Saasification continues to rise

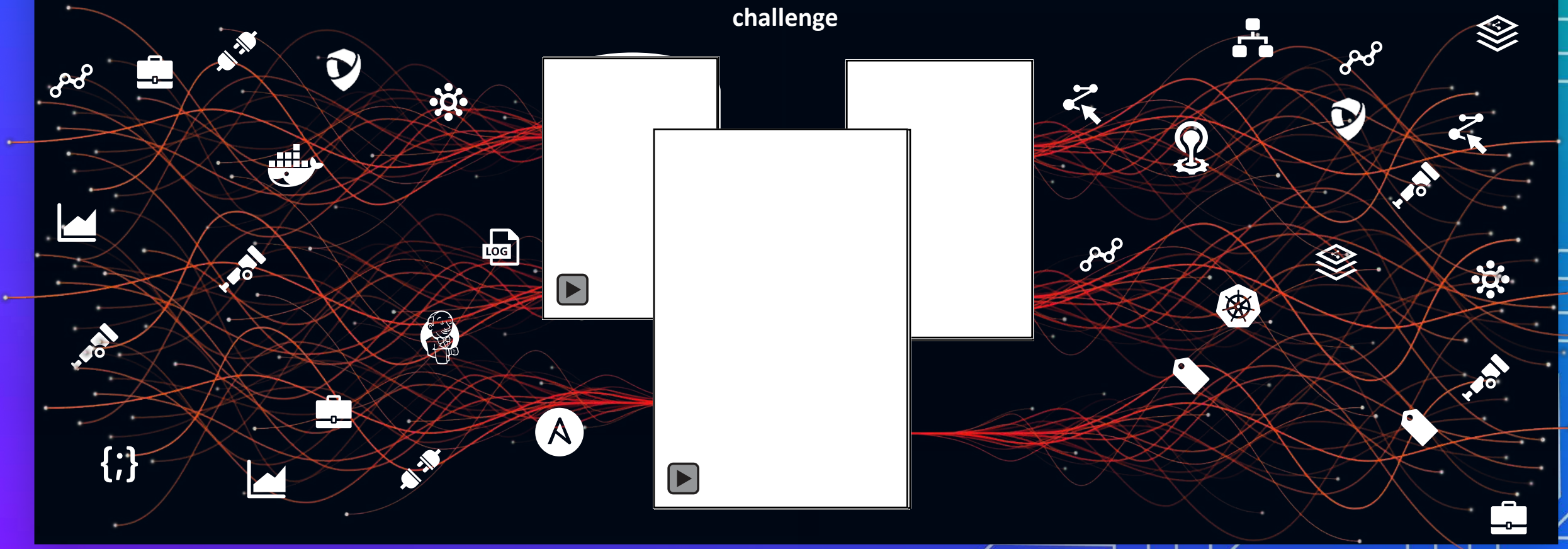
51% → **86%**

2018 2023

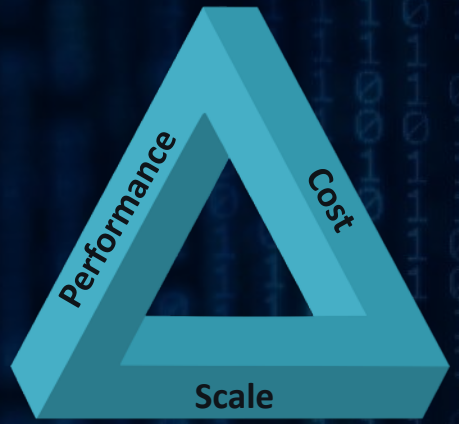
DATA COMPLEXITY EXPLODES

Multiple data types, continuously created, across different parts of the toolchain, in an evermore complex environment

Storing and extracting value from data is a massive challenge



Cost of existing data platforms are outpacing the value provided



Today's data stores lack full context, require expensive infrastructure and/or cloud resources, and consume the time of expensive staff in reactive mode

scale challenges

expensive

slow

cumbersome

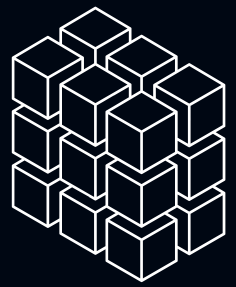
rigid

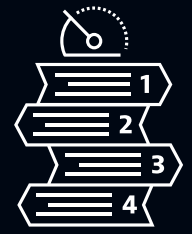
cross purpose

WE'VE REACHED A TIPPING POINT IN TODAY'S DATA PLATFORMS

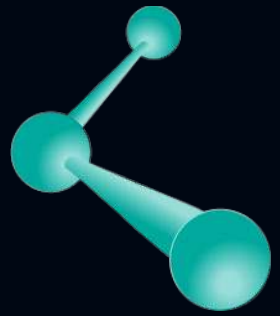
WE NEED INNOVATION

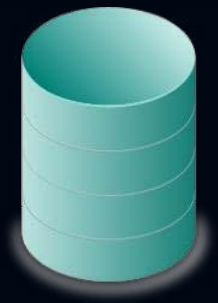
FROM DISPERSED DATABASES to a CAUSATIONAL MPP DATA LAKEHOUSE

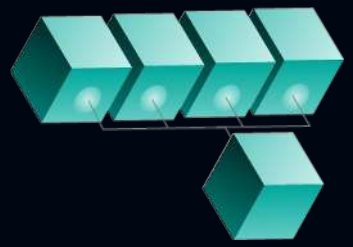


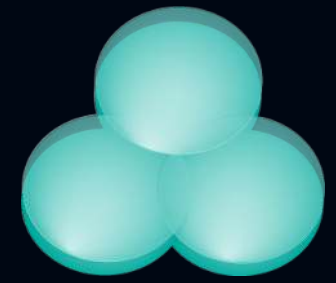


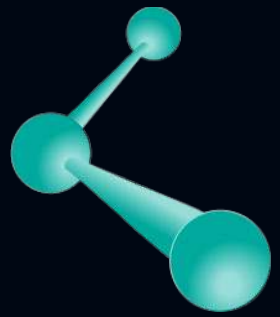
Siloed, incompatible	Data lake	Data warehouse	Indexed wait, admins, expensive	Generic
Causational Context	1000 petabytes Cost effective	Instant Schema-on read	Query at scale Massively Parallel Processing	Observability and Security analytics

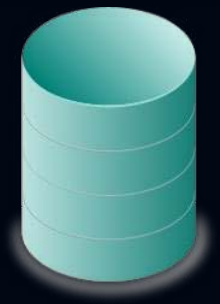
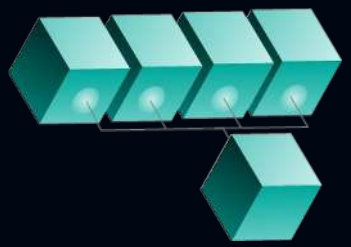
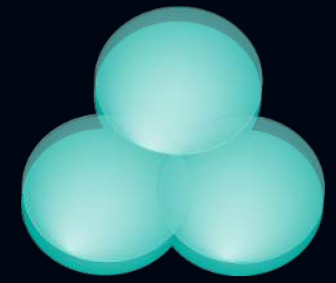


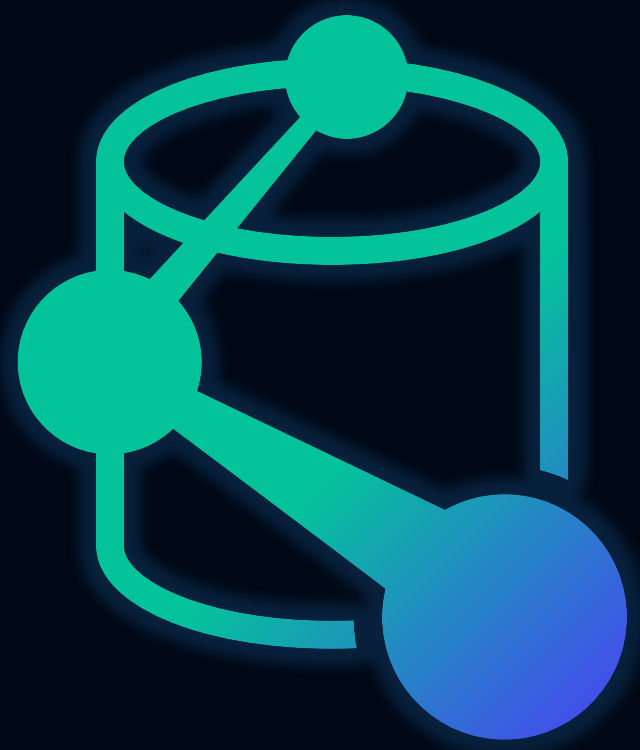










Grail™

Grail is a causational data lakehouse with a massively parallel processing (MPP) analytics engine. It leverages the new Dynatrace Query Language (DQL) for context-rich analytics

WITH GAME CHANGING VALUE ATTRIBUTES

Grail™

BILLIONS

of dependencies can be analyzed in context

CONTEXT

1000 petabytes/day

all retained for more than 12 months

INGEST

5-100x

faster to query and parse

executed in massive parallel on 1000s of nodes

SPEED

0 secs

instant query,

no index,
no rehydration,
no schema creation, no extra parsing step

SCALE

1st

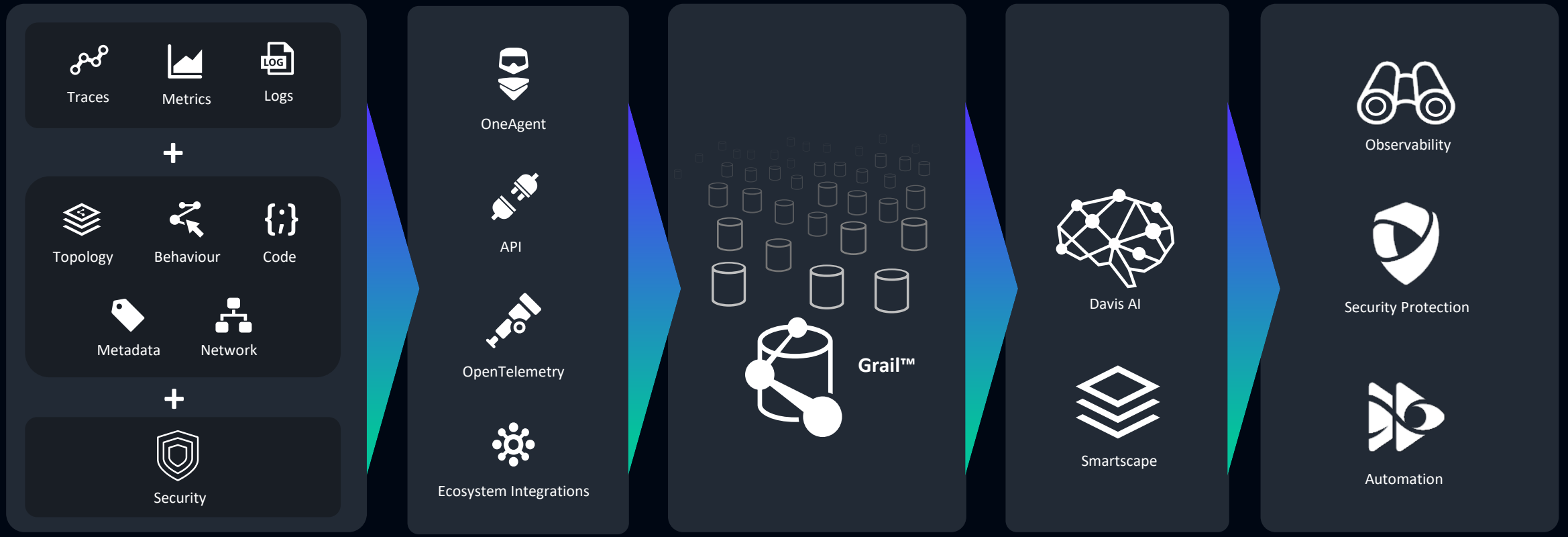
All-in-one platform that brings observability, security and business together

PLATFORM

* Grail™ architecture is enabling to scale to that dimension for a multi-tenant cluster, we reserve right to deliver based on future market needs

** potential combined performance gain from massive-parallel processing query engine, and faster parsing compared to regular expressions, for many advanced query use-cases on large datasets

WOVEN INTO THE ARCHITECTURE TO DELIVER
ANSWERS AND INTELLIGENT AUTOMATION FROM DATA



Deep context rich, full stack beyond observability sources

Automatically captured in context & pre-processed

NOW
Stored contextually with massive processing and retrieval capabilities

Accessed by our causal AI for analysis & answers

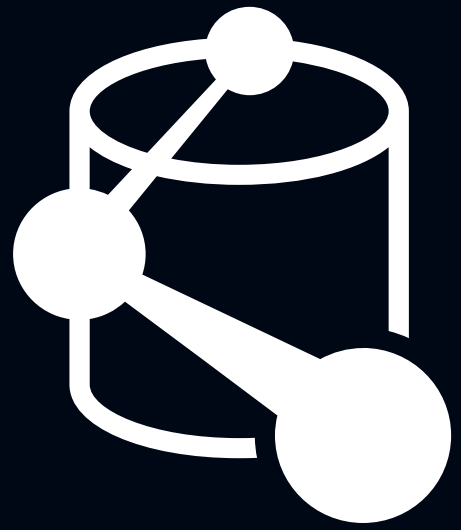
Powering automation, orchestration, prevention and protection

GRAIL™ WILL POWER ALL PLATFORM MODULES

LOG (R)EVOLUTION

Log Management and Analytics

powered by



Grail™

LOG MANAGEMENT AND ANALYTICS

powered by  Grail™

100TB/day per tenant
ingest performance towards Q1 CY23

MPP

Instant!
No rehydration
No index generation
No schema admin

up to... **100x faster***
Blazing Performance
~1TB in ~1s @ 1000 cores

Cost Efficient

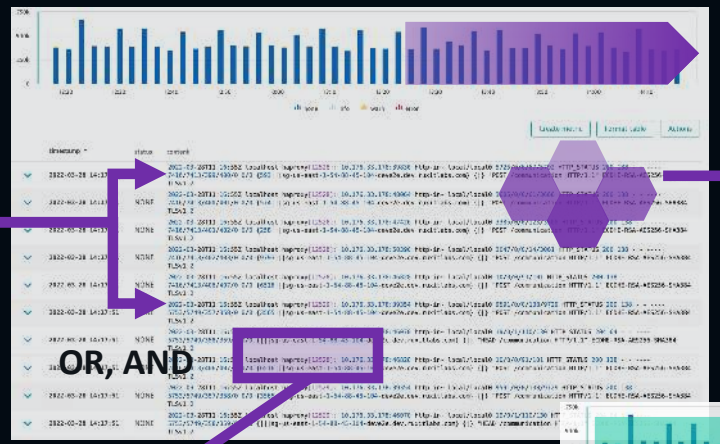
>100x
more scalable **

* for many advanced queries on large datavolumes

** compared to executing queries on open-source index based databases, are limited by expensive disks, compared to the built-in automatic multi-tiered approach of Grail

EXCERPT / IN-DEPTH
INDUSTRY STANDARD (INDEXED DATABASE)

4 Wait for new logs to arrive



1 Decide relevant indexes before ingest

2 Limited query operators

OR, AND

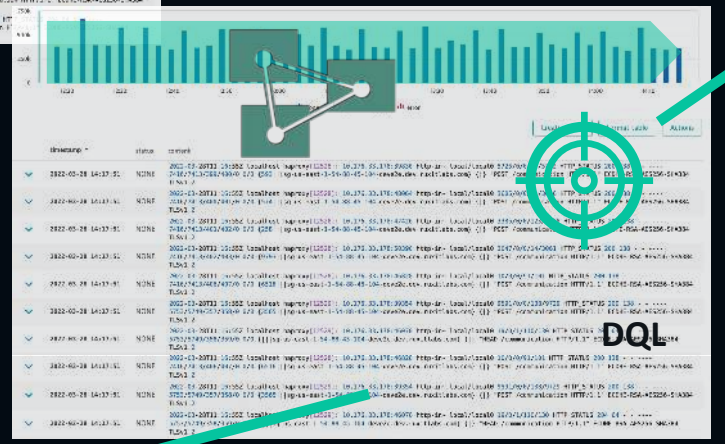
3 Simple keyword search

5 Statistical approximate (sampled) summaries

...vs Grail™

5 100% precision insight into all data stored

4 Full historical analysis



DQL

1 Decide at any time what is relevant in logs

2 Powerful data exploration & transformation with DQL

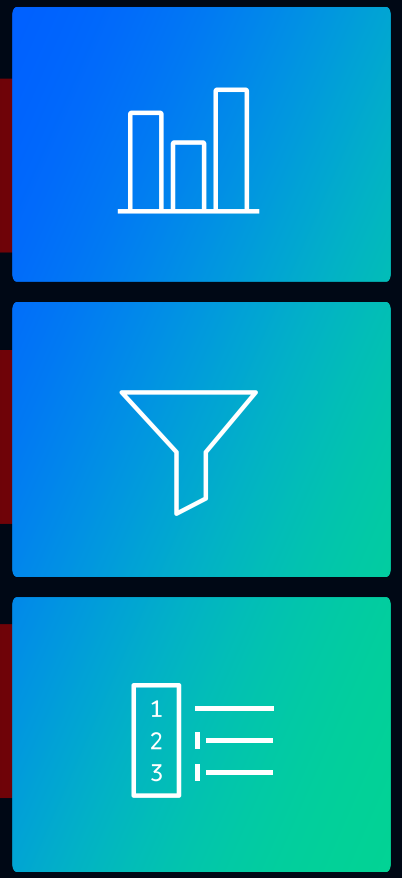
3 Unlimited transforming, restructuring of all raw logs

AUTOMATIC EVERYTHING, NOW WITH A LANGUAGE

Time consuming
writing and maintaining queries

Too complicated
for everyday users

Lack of instant access



DQL

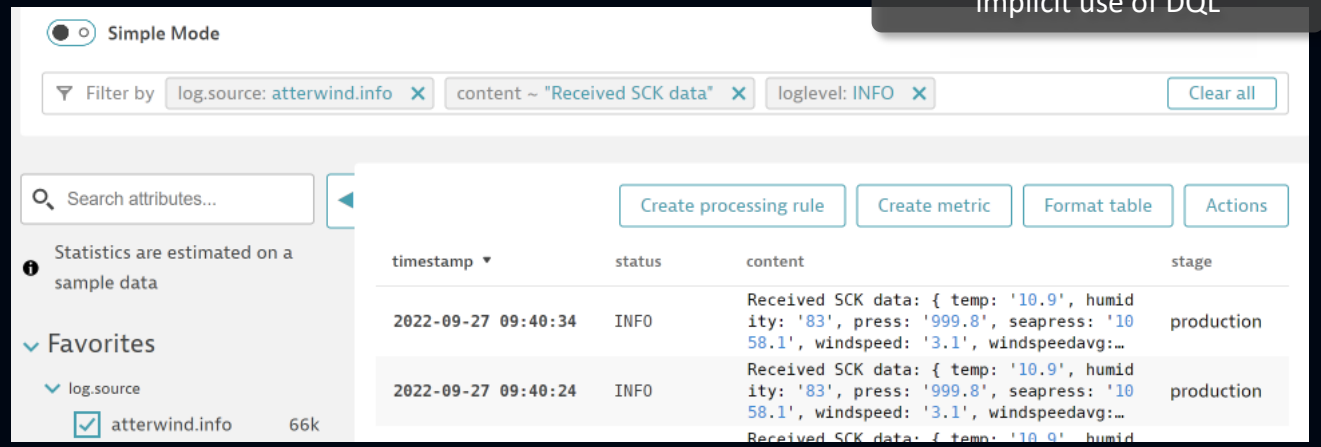
Continued automatic everything
Out of the box, for common use-cases

Easy visual query building
for simple customizations

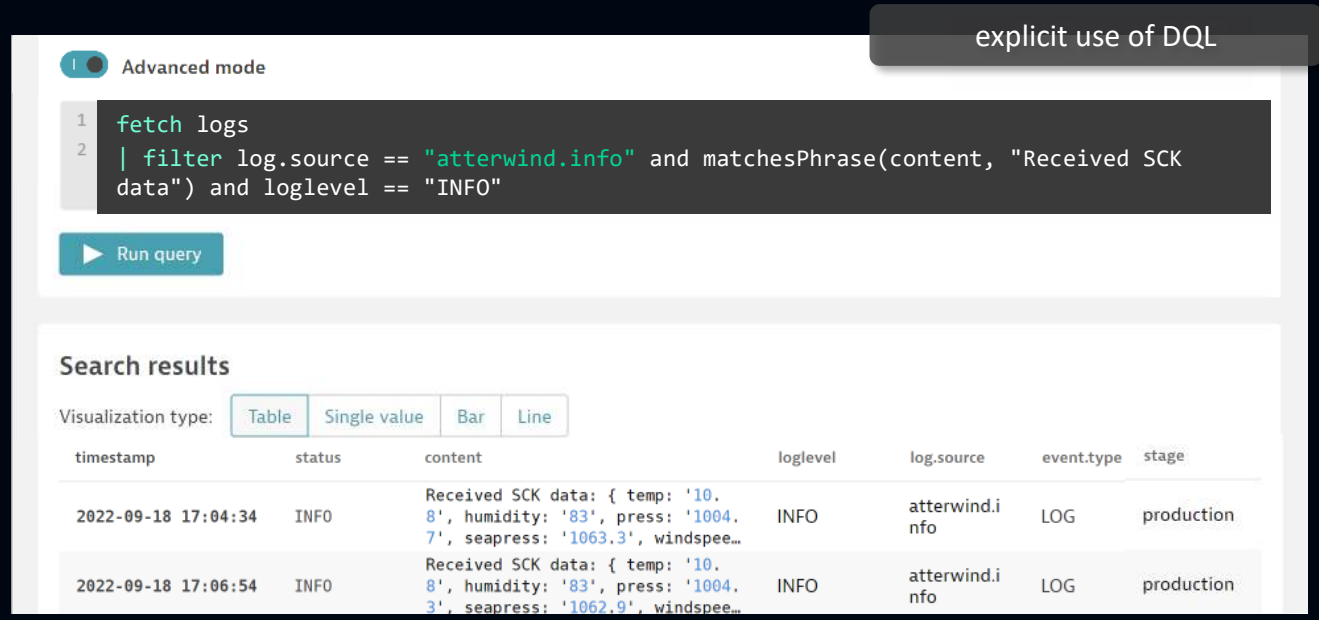
Powerful query language
for custom and extended use cases, instantaneously

DYNATRACE QUERY LANGUAGE (DQL)

implicit use of DQL



explicit use of DQL



Do more with less

- Simple visual query builder powered by DQL
- Ability to do advanced custom queries
- Instant access to your unknown unknowns

All in one

- Ubiquitous, analytics and power

BUILT WITH POWER AND PURPOSE

Effortless migration

- Minimal learning curve for new users – it just makes sense
- Familiar to power users – i.e. Splunk

Purpose build for observability and security

- 50% reduced time to query
- 5-10x faster parsing with DPL and (DPL = Dynatrace Pattern Language)
- 90% reduced trial-and-error compared to Regular Expressions

```
fetch    logs, from:now()-10m
| filter event.type == "K8S"
| filter dt.event.group_label == "Failed
         or k8s.container.name == "contr
         and not contains(log.source, "d
| fields timestamp, namespace = k8s.name
         workload = dt.kubernetes.worklo
Parse HERE E HRERERE HTERER HERHRERE
| filter namespace == "dps-ingest"
| sort   timestamp desc
| fields last_mountfail = timestamp, wor
| limit  1
```

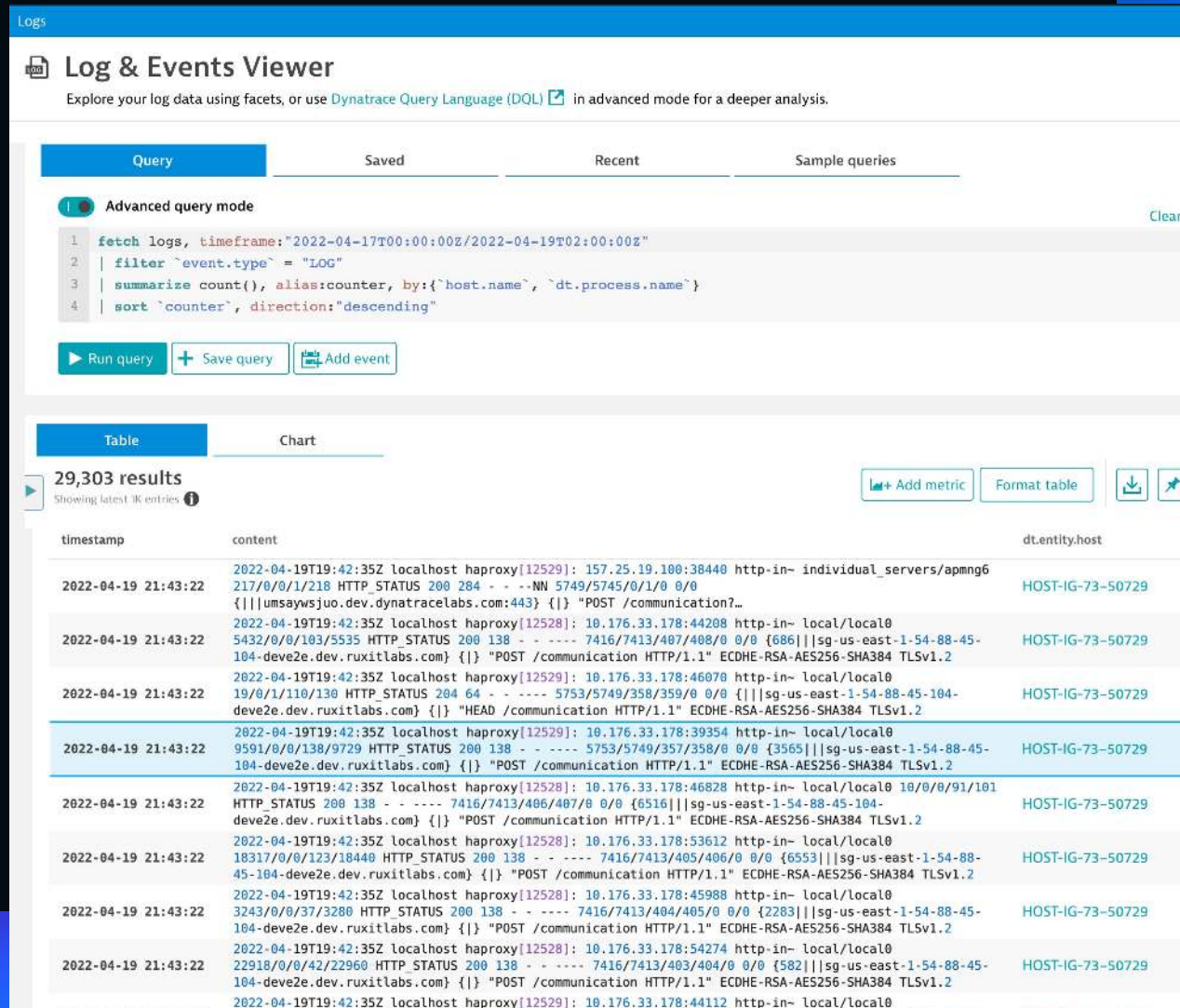
Readable

- 2-100x increase in productivity on query building and collaboration
- Step by step data processing

GRAIL™ WILL ENABLE MANY LOG USE-CASES

Exciting initial use cases

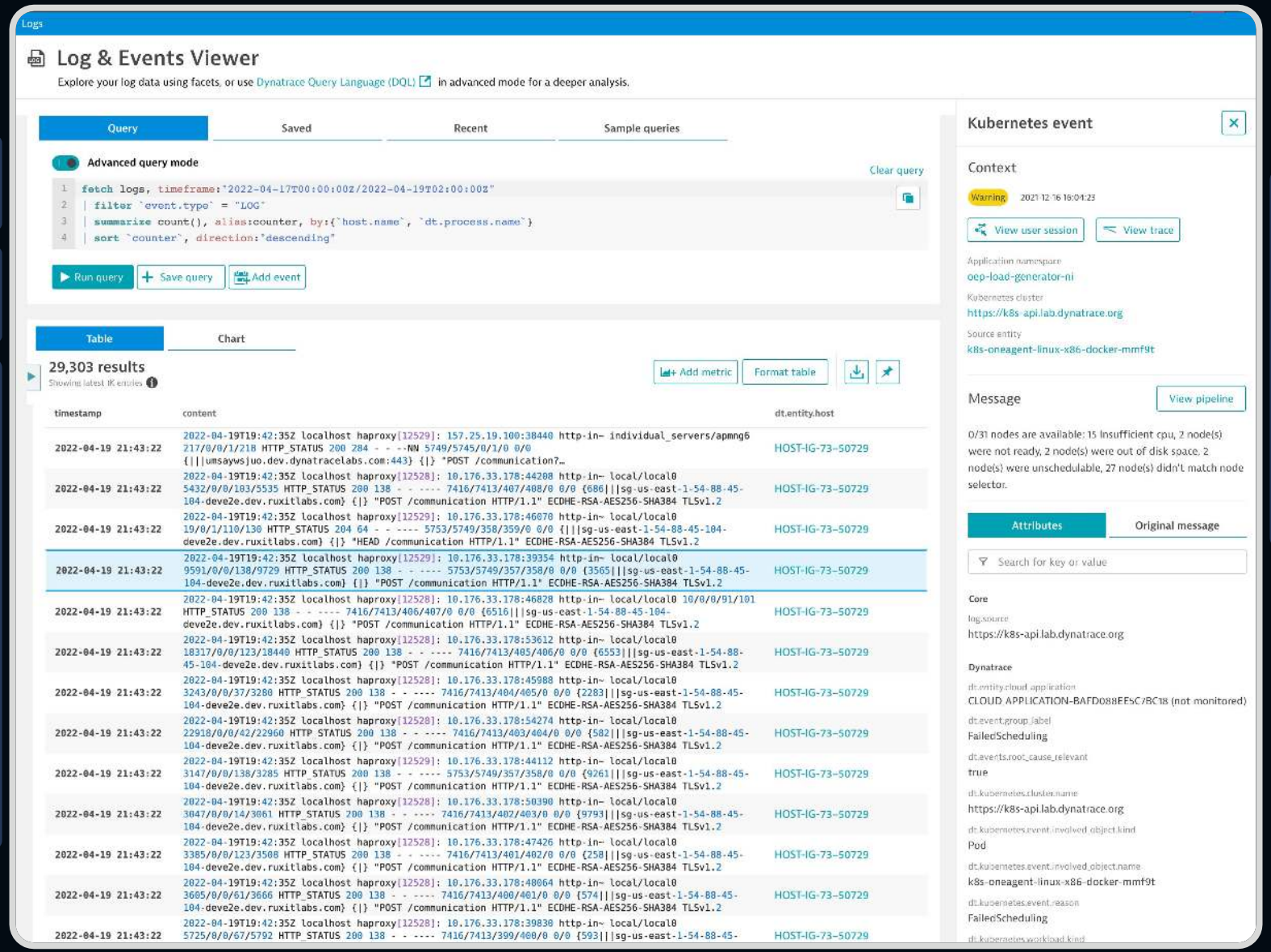
- **Log to Metrics**
- **Troubleshooting**
- **Application Optimization**
- **Audit and Forensics**



Simple mode:
quick search

Advanced mode:
DQL queries

Actionable results:
- create charts
- pin to dashboard



Meaningful context:
- linked traces
- sessions

Drill down to details

Deployment Verification example

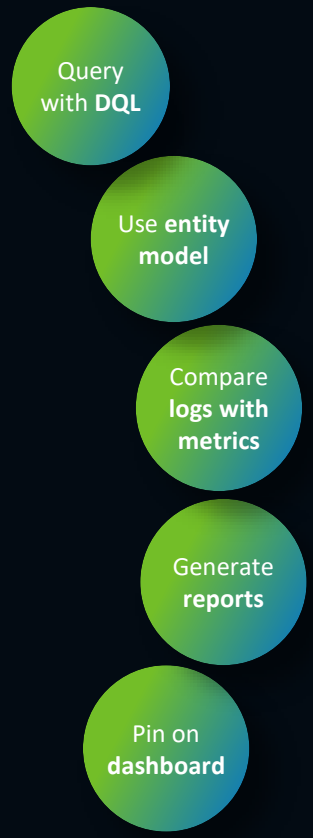
Questions are asked quickly….

Find all apps with failure trace logged more than 10x within a 5-minute interval.

… answers are even faster:

Logs powered by Grail

```
fetch logs
| filter contains(content,"failure")
| summarize failurecount=count(), by:{bin(timestamp, 5m), app}
| filter failurecount > 10
| summarize by:app
```



Troubleshooting - Fault isolation example

Questions are asked quickly….

Are our app error logs correlated with too many requests from the same IP?
(example where IP address is not extracted/indexed on ingest)

… answers are even faster:

Logs powered by Grail

```
fetch logs
| filter dt.process.name=="myApp" and status=="ERROR"
| parse content, "LD IPADDR:ip"
| summarize errorcount=count(), by:{ip}
| sort errorcount
```

- Query with **DQL**
- Use **entity model**
- Compare **logs with metrics**
- Generate **reports**
- Pin on **dashboard**

Audit & Forensics example

Questions are asked quickly….

Is that a malicious user? Check the logs for similar user input patterns for last 12 months.

… answers are even faster:

Logs powered by Grail

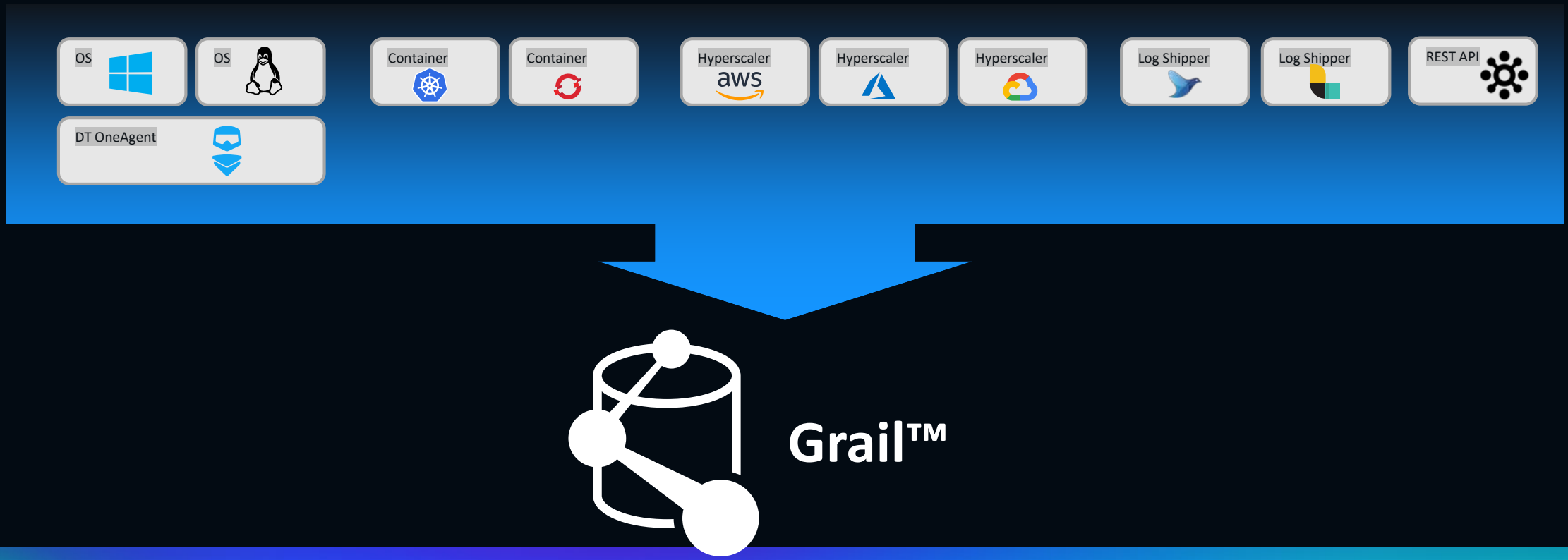
```
fetch logs, from:now()-1y
| filter endsWith(log.source,"audit.log") and
(action.type == "change_password" or action.type=="reset_password")
| parse content, "LD:userId ',' DATA:oldPW (',' !>>SPACE) LD:to EOL"
| summarize attempts=count(), by:{bin(timestamp,1d),userId}
| sort attempts desc
```

- Query with **DQL**
- Use **entity model**
- Compare **logs with metrics**
- Generate **reports**
- Pin on **dashboard**

Easy enablement, broad tech coverage

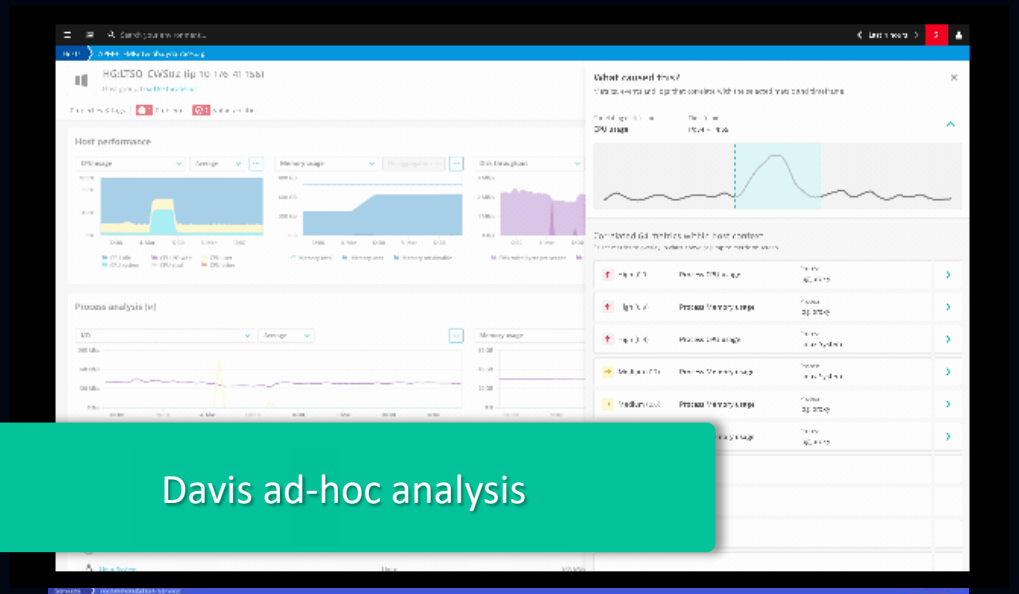
Install **OneAgent** or use the **REST API**
→ start ingesting data

No index creation
No data schema decisions
No ballooning of retention costs

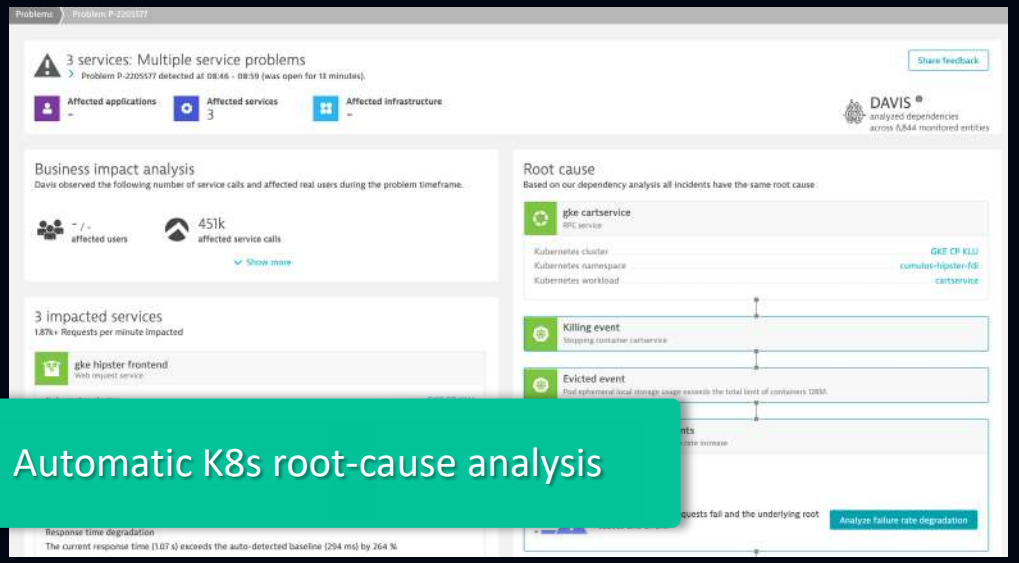


NOTHING BEATS DAVIS

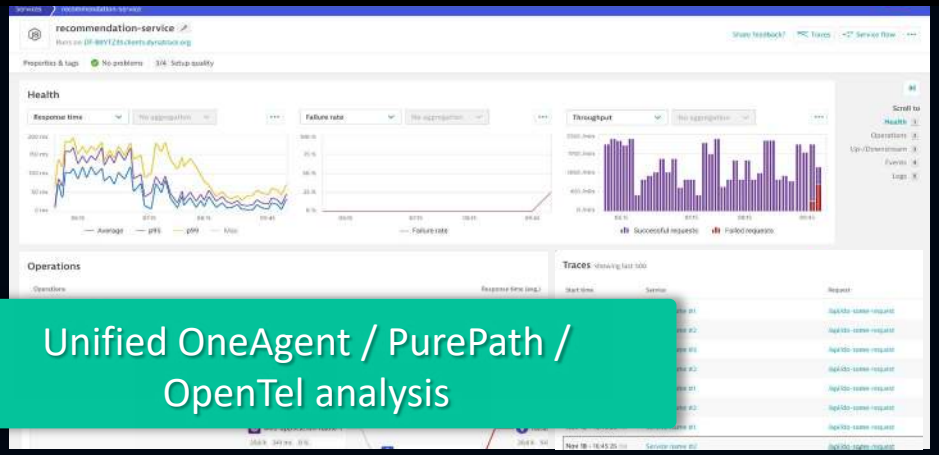
WE CONTINUE TO ENHANCE OUR CAUSAL AI ENGINE



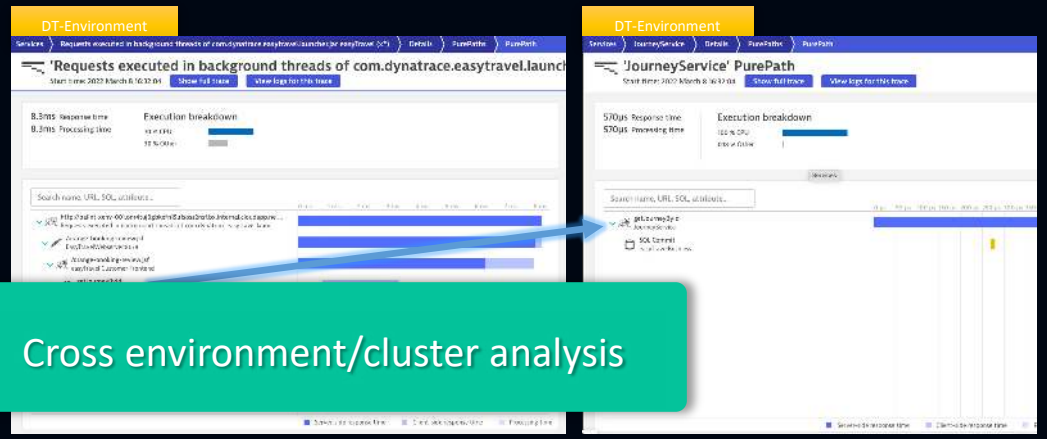
Davis ad-hoc analysis



Automatic K8s root-cause analysis

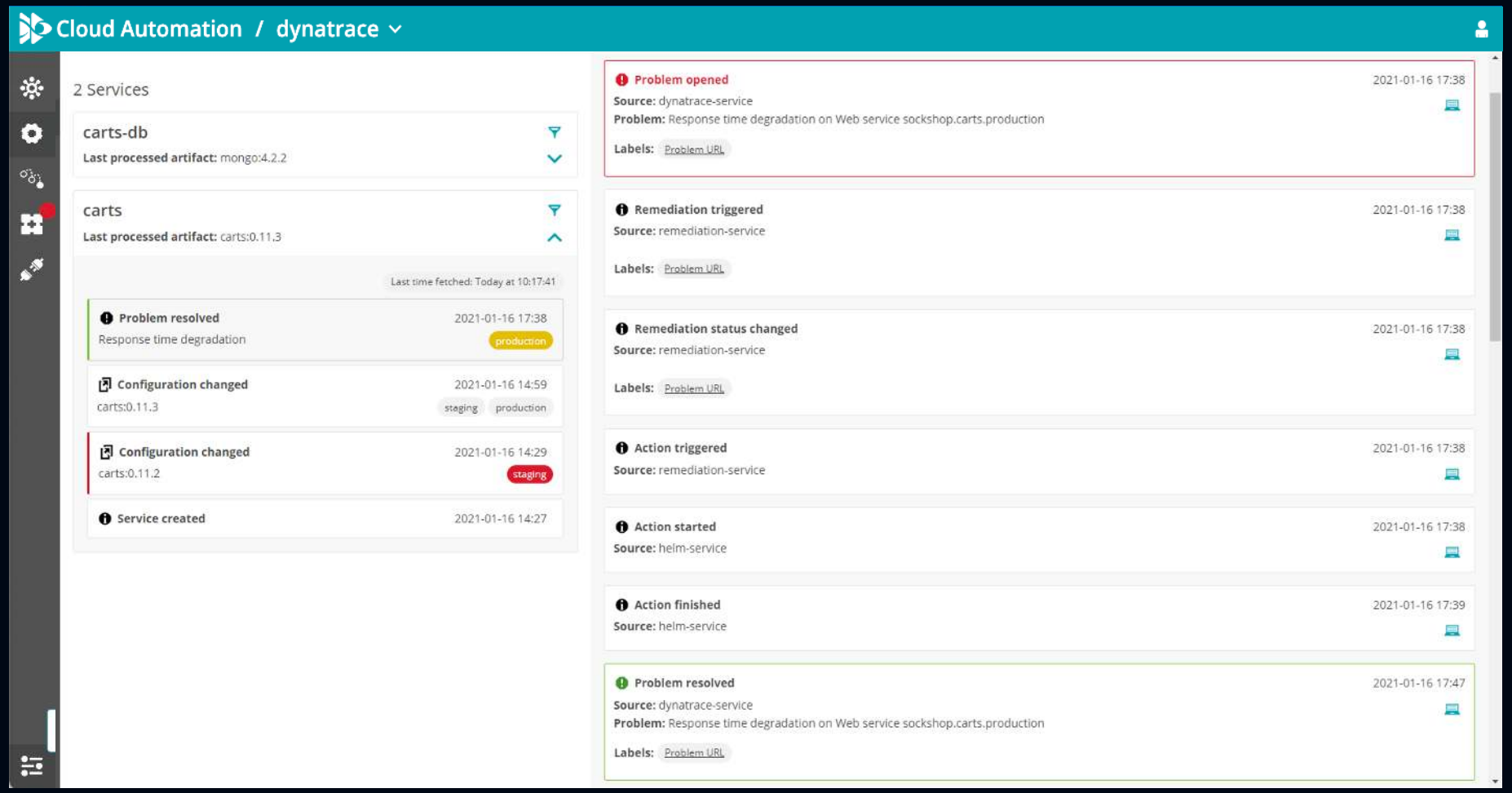


Unified OneAgent / PurePath / OpenTel analysis



Cross environment/cluster analysis

AND ULTIMATELY WE AUTOMATE, NOT-JUST-DATA ON GLASS



THE POWER OF THE PLATFORM IS EVOLVING

A WORLD WHERE SOFTWARE WORKS PERFECTLY

THANK YOU