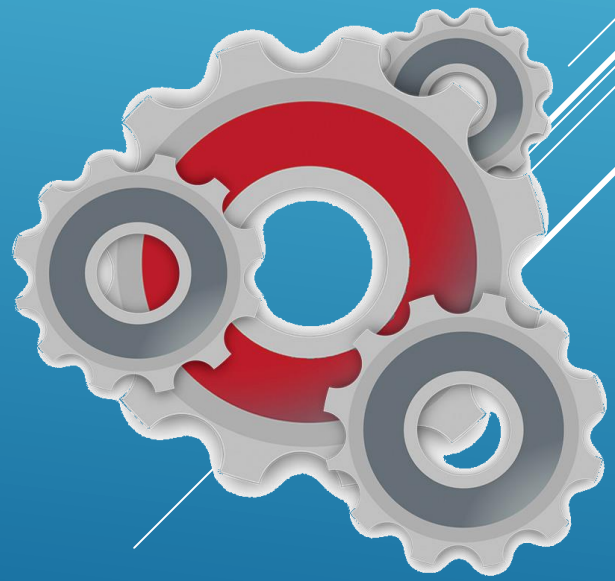
DELPHI METRICS

Recording Metrics Using the Elastic APM

- 20 years programming with relational databases
  - MSSQL, Interbase/Firebird, MySQL, Oracle
- Specialise in manipulating large data sets
- 30 months working with Elasticsearch
  - Developed Elastic Explorer

NIGEL TAVENDALE

www.allthingssyslog.com



ALL THINGS SYSLOG

CODE SAMPLES AVAILABLE AT:

https://github.com/ntavendale/opal

ELASTIC EXPLORER AVAILABLE FROM:

http://www.elasticexplorer.com/

NIGEL TAVENDALE

nigel.tavendale@allthingssyslog.com

ALL THINGS SYSLOG

WHY METRICS?

Monitor Performance

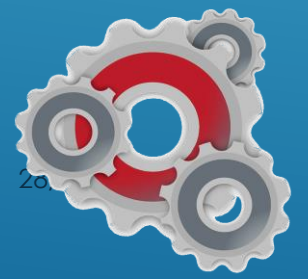
Monitor User Actions

Identify Bottlenecks

Check For Security Vulnerabilities

Gather Data On Interactions With External Components

Support In The Field



TRADITIONAL METRIC GATHERING

Logging

```
begin
  var LThen := Now;
  DoSomething;
  var LNow := Now;
  var LSpan := TTimeSpan.Subtract(LNow, LThen);
  WriteLn(String.Format('Operation Completed In %0.3f msec', [LSpan.TotalMilliseconds]));
end;
```

Easy to implement

Non visual

Can't really aggregate or correlate log data.

TRADITIONAL METRIC GATHERING

Windows Performance Counters

Reliable

Visual

Can record a period of time

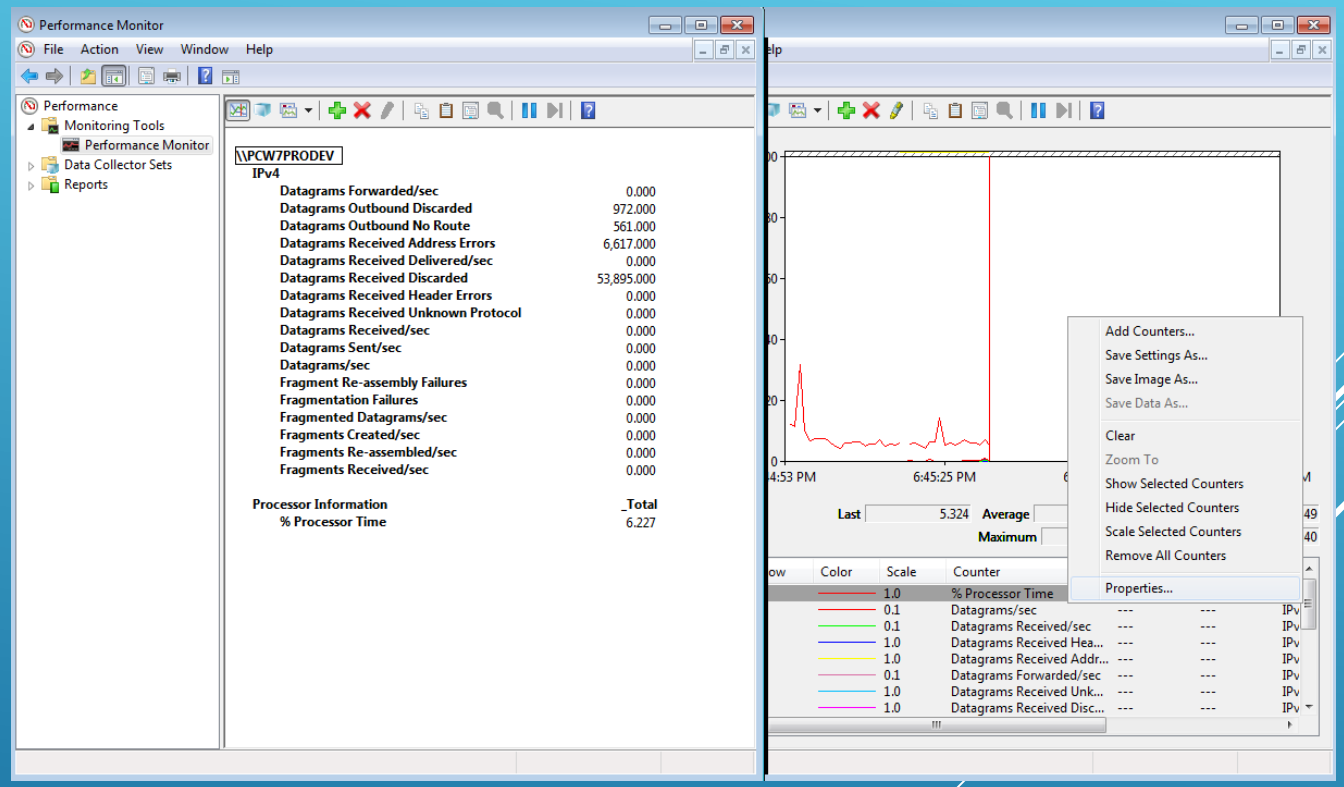
Can save data

Windows Only

Single Process Only

Cannot track a single piece of data

Implementation non trivial



WHAT WE NEED TODAY

Modern applications made up of loosely coupled services so we need to aggregate metrics over all parts of application.

Services can be written in different languages, but still need a common metrics strategy. Transforming a single dataset can involve many services on different hosts

- **Service Level**. Performance degradation in an individual service can cascade through system.
- **Transaction Level**. Applications consist of transactions and you must aggregate performance across all of them.
- **External Dependencies**. Outbound calls to systems like DNS, Databases, LDAP servers. Resolving issues with external dependencies different from resolving issues with application.

THE ELASTIC APM

APM Server

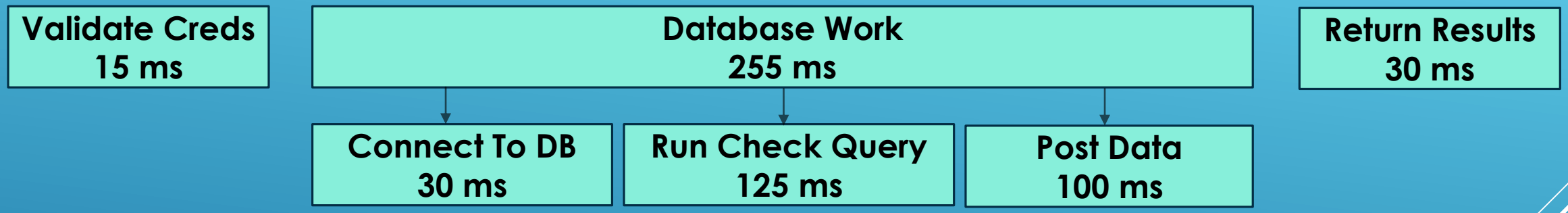
- REST API Using JSON
- Newline delimited json MIME type (“application/x-ndjson”)
- Translates APM data into elasticsearch documents, and stores them in an ES cluster.

APM Clients

- Create APM Events
- Written in various languages - .NET, go, java , javascript
- No agent in Pascal…. Yet!

ELASTIC APM - SPANS

Basic unit of performance. Measures a single activity from beginning to end.



name

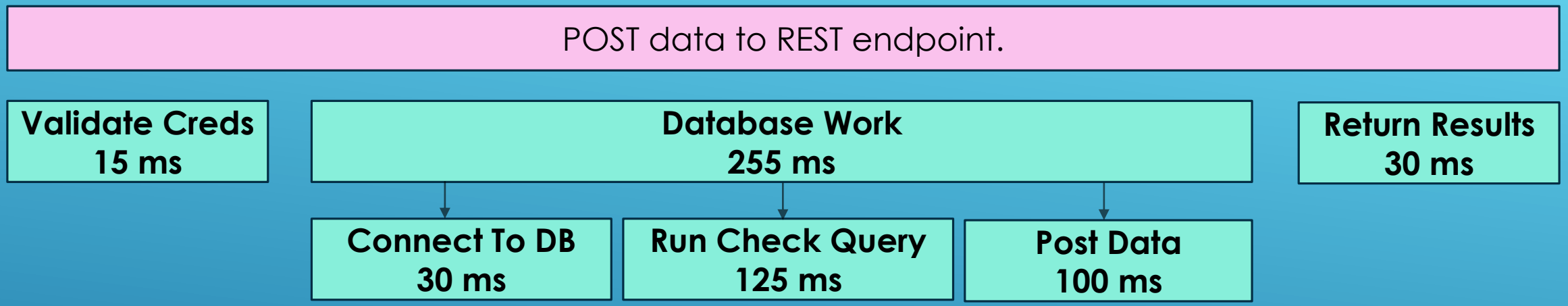
type – http request, db query, event log read

start time & duration

parent id – id of parent span

transaction id

ELASTIC APM - TRANSACTIONS



Special Type of span. Spans can have parent spans and parent transactions. A transaction is standalone. It has child spans, but no parent.

ELASTIC APM - TRANSACTIONS

Transactions need to contain extra data not found in spans. This information is information would be common to the spans.

Process & Service Information – ID, Name, Language & Framework

Host Information – Hostname, Architecture, IP

User Info – User Account process is running as

In a microservices architecture transactions occur within a service/process. They do not span across them.

ELASTIC APM - ERRORS

There Are Two Parts – Exceptions and Log Messages.

Exceptions

- Can include a stack trace – array of StackTrace frames**
- Error Code
- Exception Type
- Exception Message
- Was exception was handled?

Must have either a message or a type

ELASTIC APM - ERRORS

Log Messages

- Message (required)
- Parameterized Message (e.g. “error in in %s module”)
- Level
- Logger name

ELASTIC APM – METRIC SETS

Data determined by design of agent.

Values are numbers, but each agent decides which values (cpu, mem, to report) .

Agent implementation plays a role. A java agent might report garbage collector metrics. A C++ agent would not.

MORE INFORMATION

Elastic APM:

https://www.elastic.co/guide/en/apm/index.html

https://github.com/elastic/apm-server

Kibana:

https://www.elastic.co/guide/en/kibana/current/index.html

ALL THINGS SYSLOG