



# What is going on?

## Application Diagnostics using the cloud

Maarten Balliau  
@maartenballiau





# Agenda

What is going on?

Application Insights

the service

the developer side (SDK)

Application Insights Analytics



A photograph of a stone circle in a forest. The circle is constructed from large, flat, greyish-brown stones arranged in a ring. The interior of the circle is filled with a thick layer of light-colored wood chips or mulch. In the center of the circle stands a dark, rectangular stone marker. A small, rectangular wooden plaque is attached to the top of the marker, containing text in Vietnamese. The forest floor is covered with dry leaves and twigs. In the background, several tall, thin trees with green foliage are visible. The text "What is going on?" is overlaid in a large, white, sans-serif font across the middle of the image.

What is going on?



# Log all the things!

```
System.Diagnostics.Trace.TraceInformation(  
    "Something happened");
```

```
System.Diagnostics.Trace.TraceWarning(  
    "Error! " + ex.Message);
```

# And here's a typical log...

```
App.exe [12:13:03:985] Information: 0 : Customer address updated.
App.exe [12:13:04:011] Error: 8 : System.NullReferenceException occurred. Value
can not be null.
App.exe [12:13:04:567] Information: 0 : Machine policy value 'Debug' is 0
App.exe [12:13:04:569] Verbose: 9 : ***** RunEngine
        ***** Action:
        ***** CommandLine: *****
App.exe [12:13:04:578] Information: 9 : Entered CheckResult()
App.exe [12:13:04:689] Debug: 0 : Created. Req=0, Ret=8, Font: Req=null
```

Does this help troubleshooting? Improve the application? Analyze trends?

No.

# Log files suck.

Just stupid string data

Typical log file has no “context”

Typical log file has no correlation

Log files do not measure CPU, memory, I/O, ...

How to get data off our machines?

How to report/analyze?

# Log files suck.

Just stupid string data

Typical log file has no "context"

Typical log file has no correlation

Log files do not measure CPU, memory, I/O, ...

How to get data off our machines?

How to report/analyze?

Semantic Logging  
(e.g. ETW, Serilog, ...)



# Log files suck.

Just stupid string data

Typical log file has no “context”

Typical log file has no correlation

{ Log files do not measure CPU, memory, I/O, ... }

How to get data off our machines?

How to report/analyze?

Windows Server tooling,  
NewRelic, PRTG, SNMP, ...

# Log files suck.

Just stupid string data

Typical log file has no “context”

Typical log file has no correlation

Log files do not measure CPU, memory, I/O, ...

{ How to get data off our machines?  
How to report/analyze? }

Vendor tooling, NewRelic, PRTG,  
SNMP, ...



# Log files suck.

Just stupid string data

Typical log file has no “context”

Typical log file has no correlation

Log files do not measure CPU, memory, I/O, ...

How to get data off our machines?

{ How to report/analyze? }

Splunk, LogStash, Kibana, ...

# Analytics suck!

Either use vendor tooling

What with x-plat deployments?

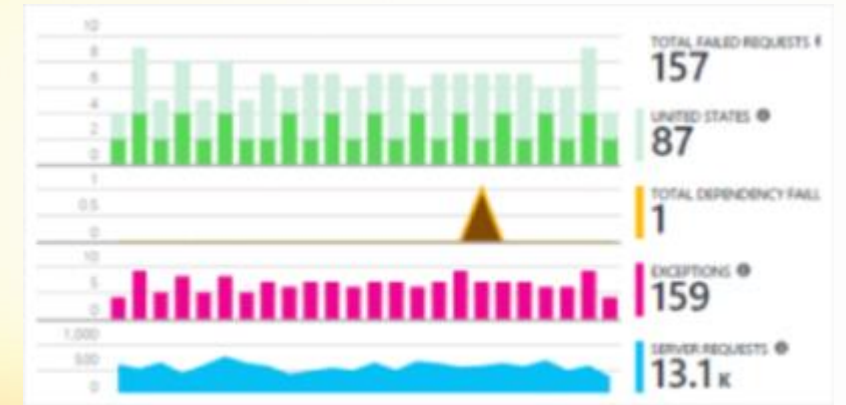
Either build it ourselves

Lots of components, management overhead, ...

All we wanted was logging and correlation...



# Application Insights



# Application Insights

"[Application Insights](#)" is an extensible Application Performance Management (APM) service for web developers on multiple platforms."



# Application Insights

Azure Service + Library/SDK

Solves “where to store”, “how to ship”, “how to analyze”

Enriches data with telemetry

Correlates data (e.g. client + server + dependency/DB/...)

Allows structured logging

Allows rich querying, alerting

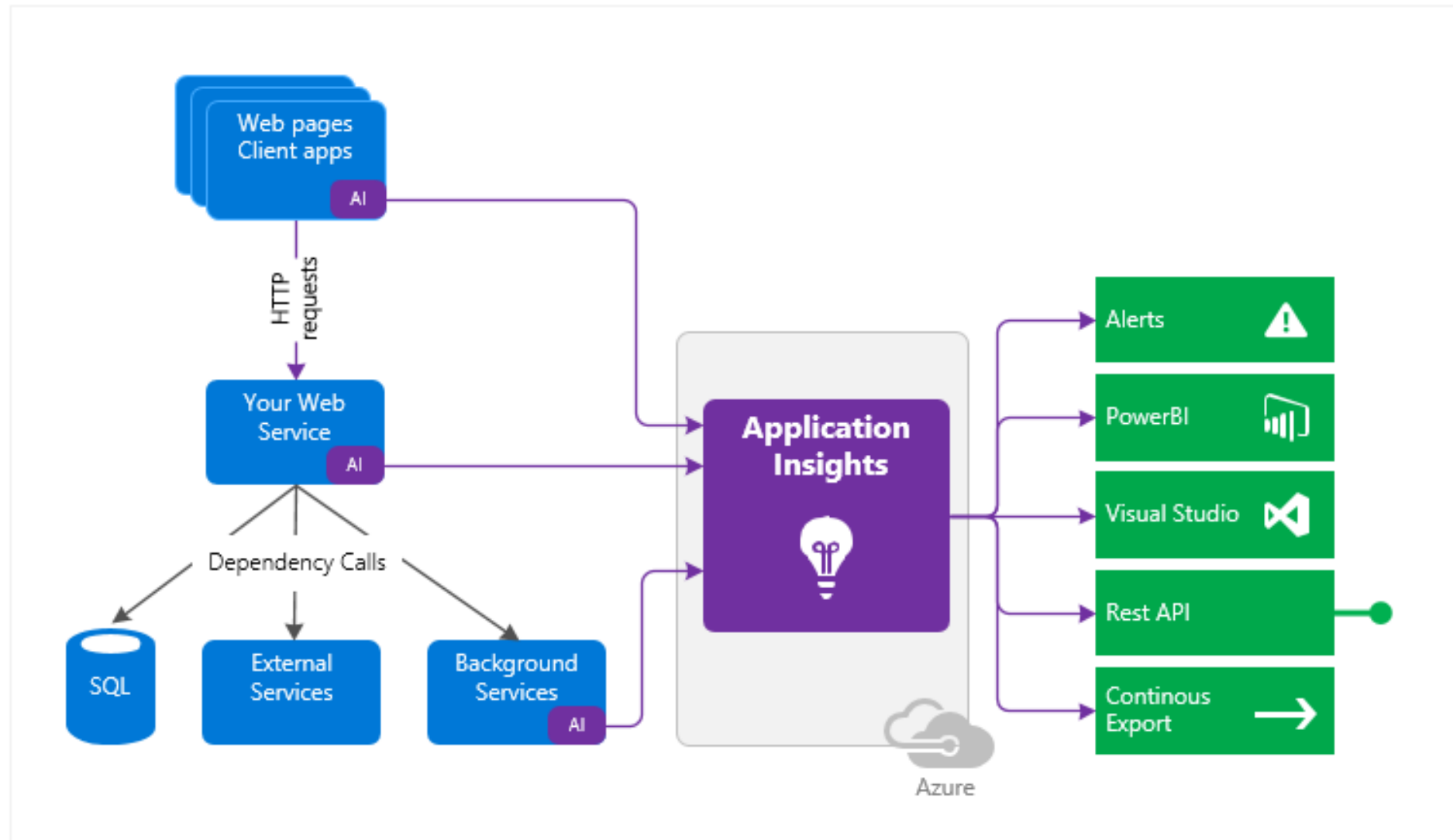
Works for many, many platforms and languages

Web, Windows, Xamarin, any application type really

.NET, Java, JavaScript, Objective-C, PHP, Python, Ruby, ...

Docker (host and container)

# Application Insights





# Getting Started

DEMO

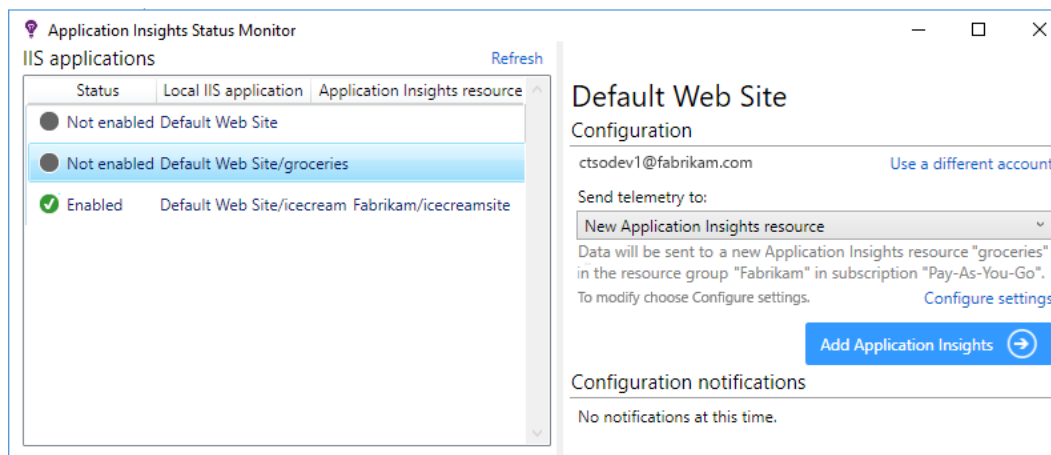
# Do I *have* to run on Azure?

REST API to send data to

Various tools and SDK's to collect data specific to language/platform

<https://docs.microsoft.com/en-us/azure/application-insights/app-insights-platforms>

## IIS



## Docker host and containers

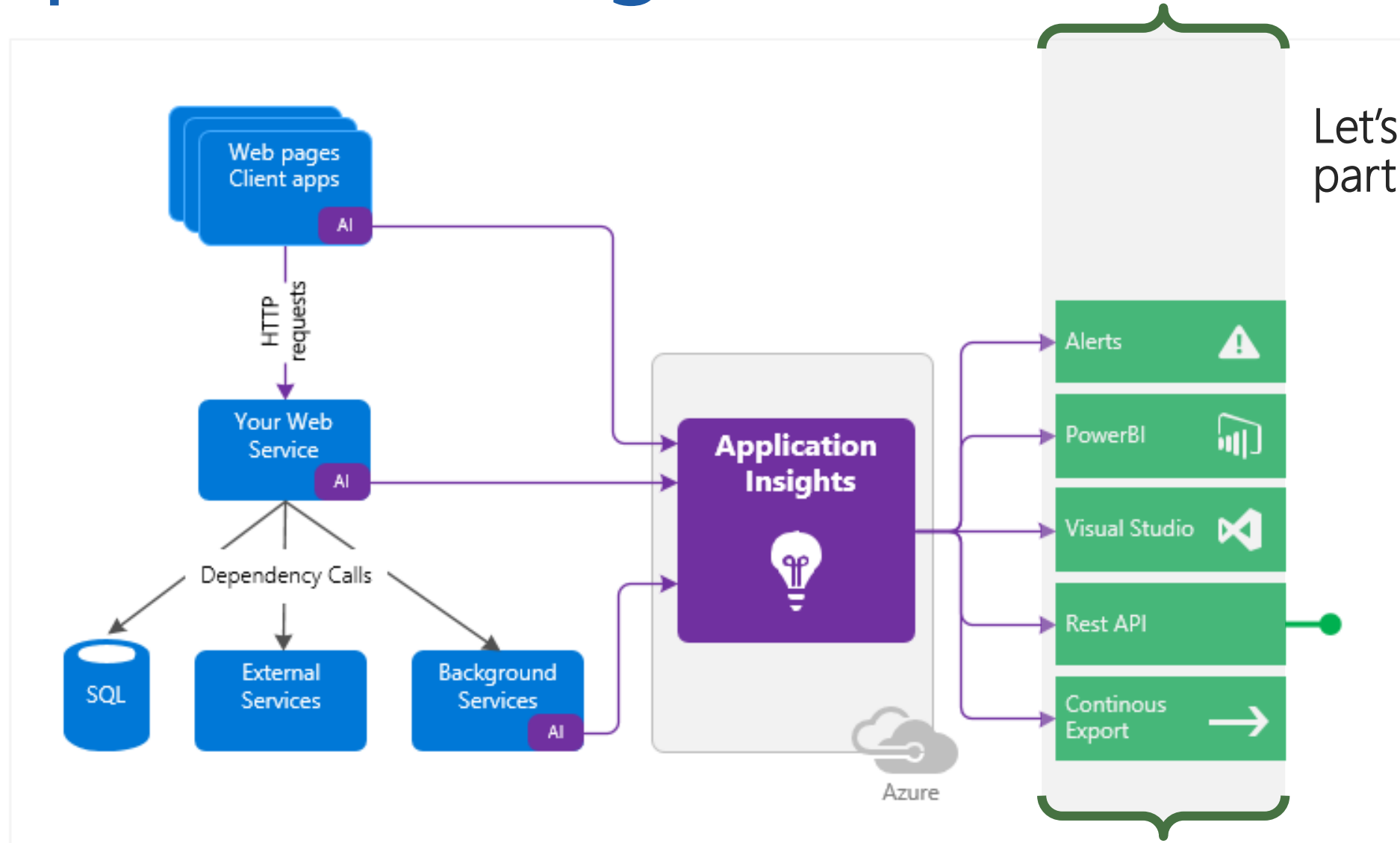
```
docker run -v  
/var/run/docker.sock:/docker.sock -d  
microsoft/applicationinsights  
ikey=000000-1111-2222-3333-4444444444
```

# Application Insights – the service





# Application Insights



# Monitoring, alerting, exporting

DEMO

# Input, processing, output

## DATA INGESTION

Performance Counters

Requests (both server/client side)

Traces

Exceptions

Dependencies

Custom Metrics & Events

...

## CAPABILITIES

Monitoring

Alerting

Querying

Exporting

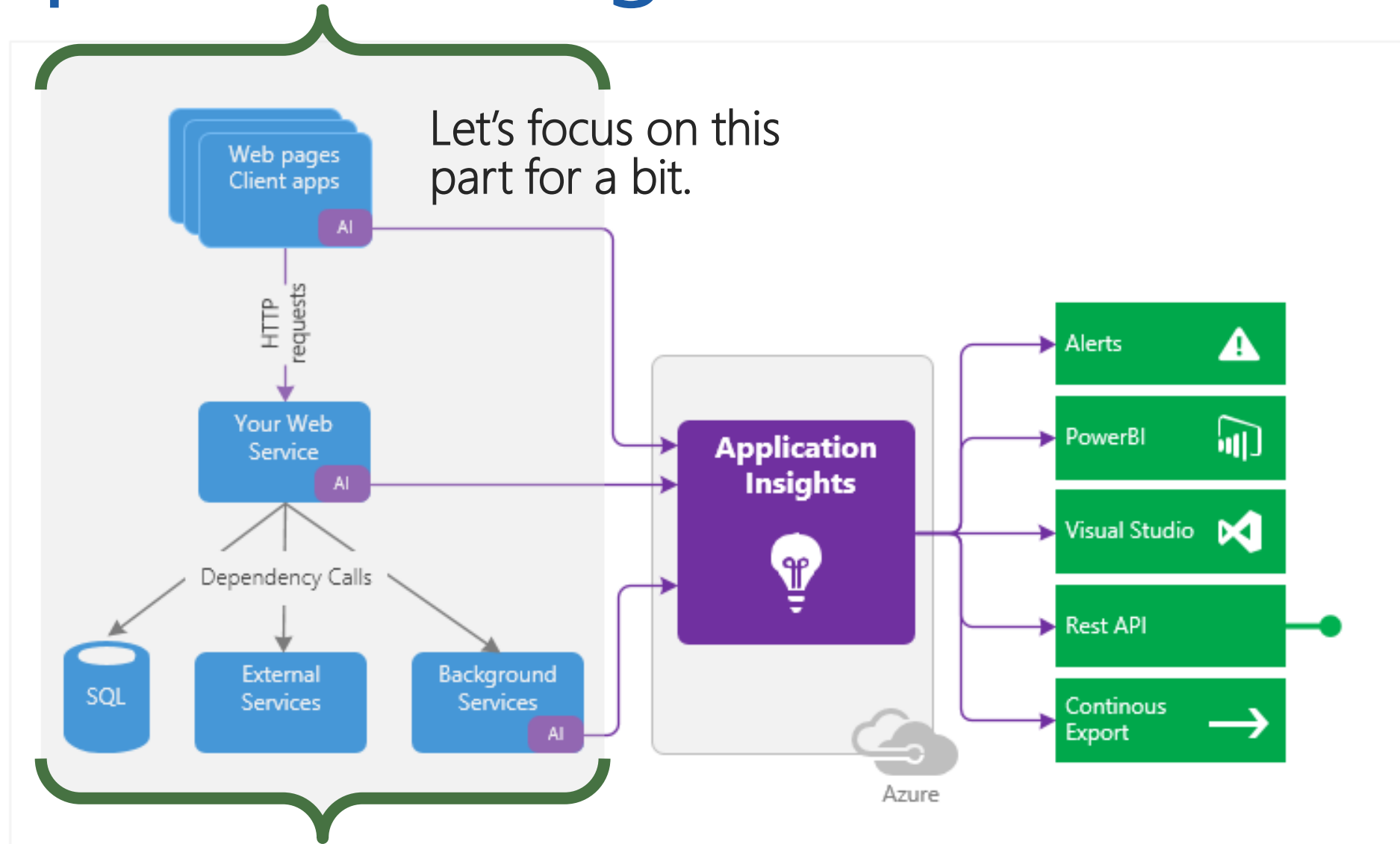




# Application Insights — the developer side



# Application Insights



# Data being collected

## AUTOMATICALLY

- Performance Counters
- Requests (both server/client side)
- Traces
- Exceptions
- Dependencies
- Custom Metrics & Events
- ...

## ENRICH MANUALLY

- Better tracing (semantic logging!)
- Events
- Metrics
- Exceptions
- Telemetry pipeline
- ...





# Better tracing

Don't do stupid strings.

`System.Diagnostics.Trace` is nice, but also not nice.

Do semantic logging!

Serilog, .NET Core logging have an Application Insights sink

All else: write a sink or log directly to **`TelemetryClient.Current`**

# Better tracing

DEMO

# Events and Metrics

Send custom events and metrics

Name of event/metric + a value

Events – help find how the application is used and can be optimized

User resized column in a grid

User logged in

User clicked “Share”

...

Metrics – help measuring quantifiable data

# Items in shopping cart

...

# Events and Metrics

DEMO



# Exceptions

Exceptions are unpleasant.

How to make them more pleasant?

Stack traces? MiniDump?

# Exceptions

DEMO

# Snapshots! (MiniDump)

NuGet package

MinidumpUploader.exe

*separate process which monitors snapshot requests*

SnapshotHolder\_x64.exe

*creates a shadow process in 10-20ms, captures the minidump, returns it to MinidumpUploader.exe*

<https://docs.microsoft.com/en-us/azure/application-insights/app-insights-snapshot-debugger>

# Enriching telemetry

Why?

Enriching telemetry – adding properties to all data

Obfuscating telemetry – GDPR, sensitive data, ...

How?

Directly adding data on **TelemetryClient.Current**

Using the telemetry pipeline (middlewares on sending data to AI service)

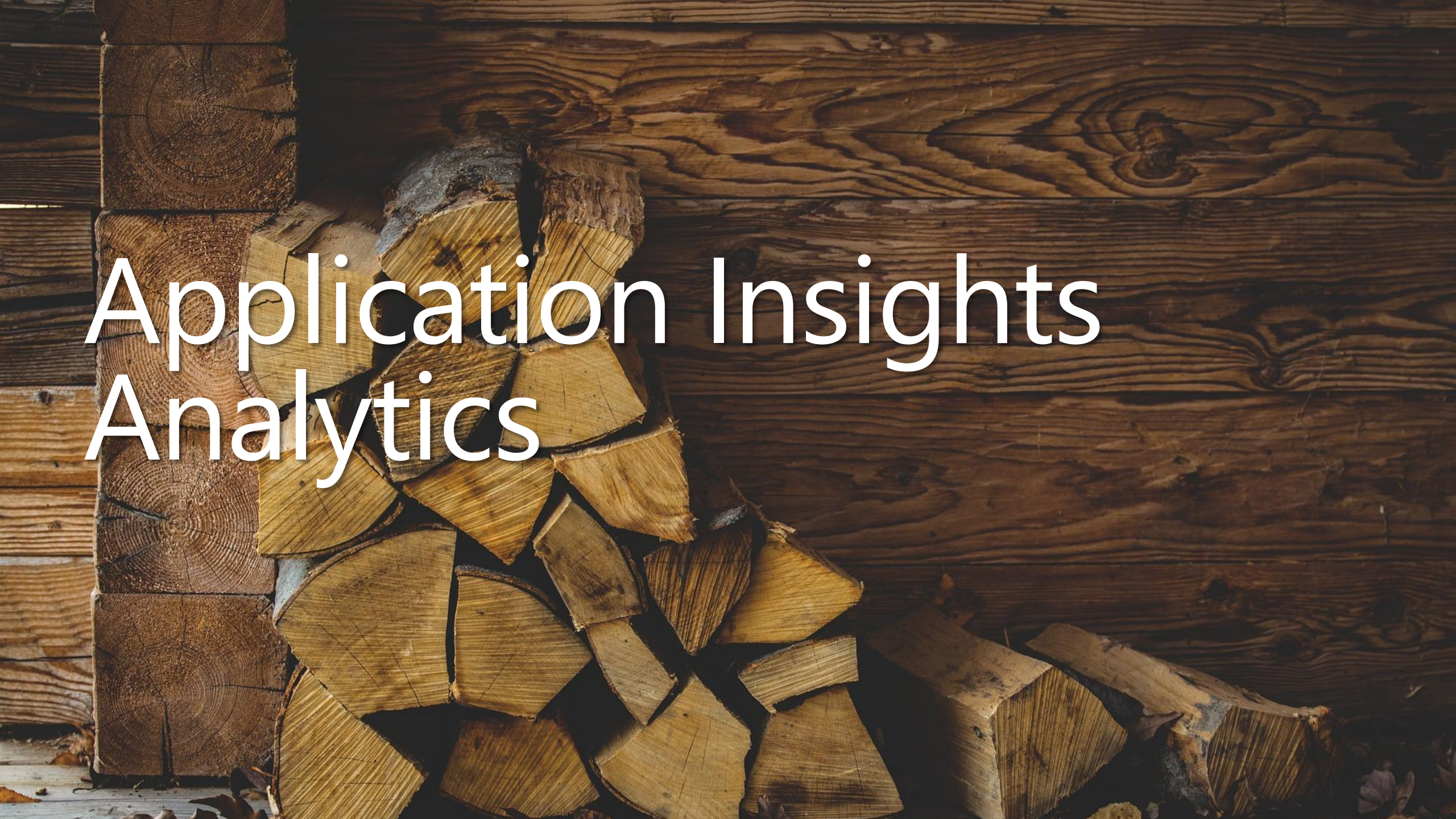
<https://blog.maartenballiauw.be/post/2017/01/31/application-insights-telemetry-processors.html>



# Enriching telemetry

DEMO





# Application Insights Analytics



# Application Insights Analytics

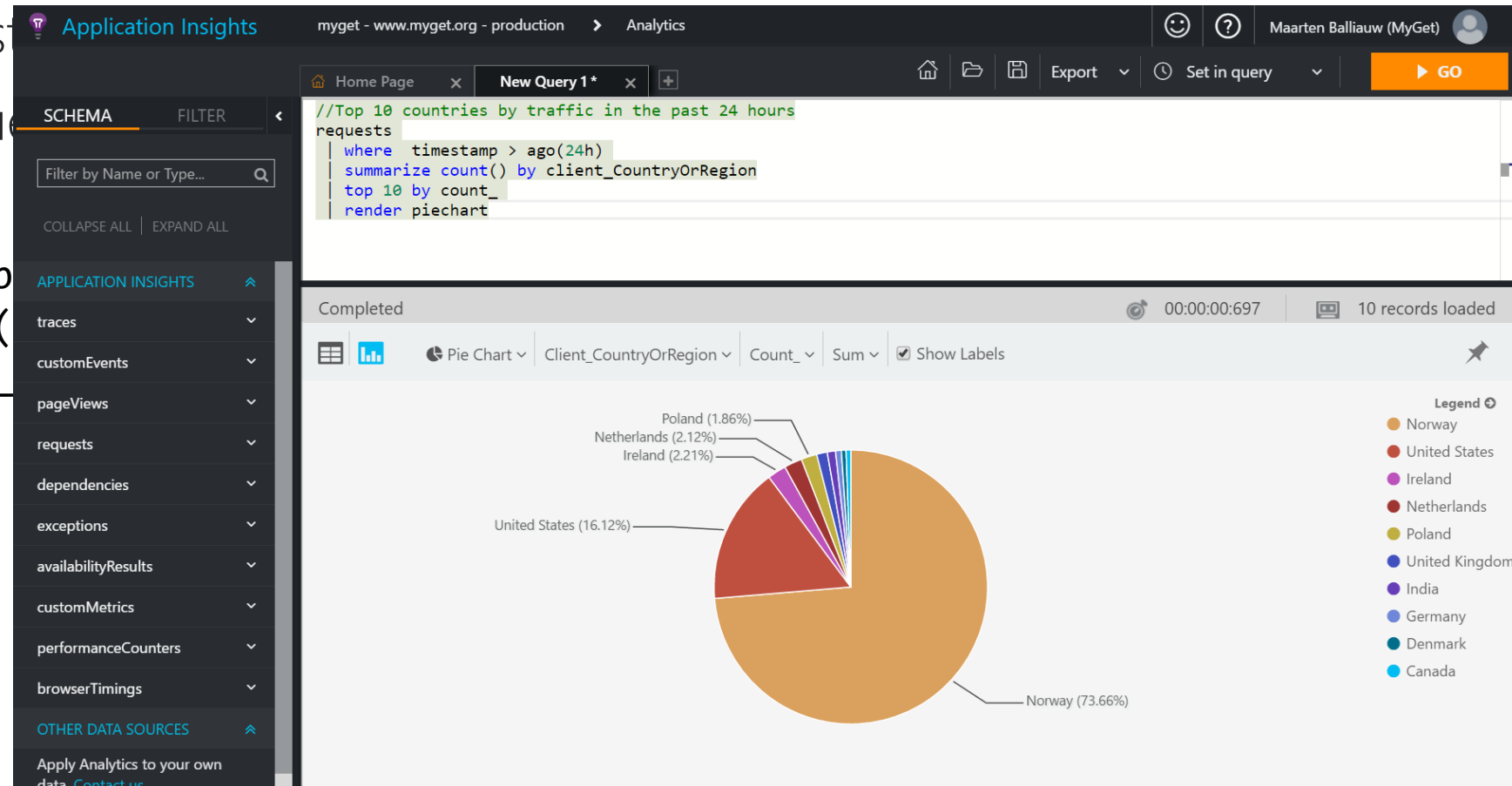
Former MS-internal tool "Kusto"

Near-realtime log ingestion

Lets you run custom queries

requests

```
| where timestamp > ago(24h)
| summarize count_ = count() by client_CountryOrRegion
| top 10 by count_
| render piechart
```



# Queries over logs!

<https://docs.microsoft.com/en-us/azure/application-insights/app-insights-analytics-reference>

## Input dataset

traces  
customEvents  
pageViews  
requests  
dependencies  
exceptions  
availabilityResults  
customMetrics  
performanceCounters  
browserTimings

## | Operators

where  
count  
project  
join  
limit  
order

## | Renderer

table  
chart (bar, pie, time, area, scatter)

# Smart Detection

Uses Application Insights Analytics data and searches for trends and anomalies

- Potential memleaks – memory increase

- Increase in specific exceptions

- Abnormal rise in failed requests

- ...

Each detection comes with a query!



# Application Insights Analytics

DEMO

# What else is there?

Other client types (mobile, Java, ...)

Sampling our data (in portal)

Do we need *all* data? Or just data that is representative enough?

Release annotations

Show a marker in the timeline when a release happens

<https://docs.microsoft.com/en-us/azure/application-insights/app-insights-annotations>

Profiler

Preview that I can't get to work, but should do profiling when interesting things happen

<https://docs.microsoft.com/en-us/azure/application-insights/app-insights-profiler>





# Conclusion



# Conclusion

Logging sucks

Application Insights

- the service – collect and ingest, analyze, different views

- the developer side (SDK) – enrich, customize, ...

Application Insights Analytics

- rich querying over all data

- base for automated insights/smart detection



# Thank you!

<http://blog.maartenballiauw.be>  
@maartenballiauw