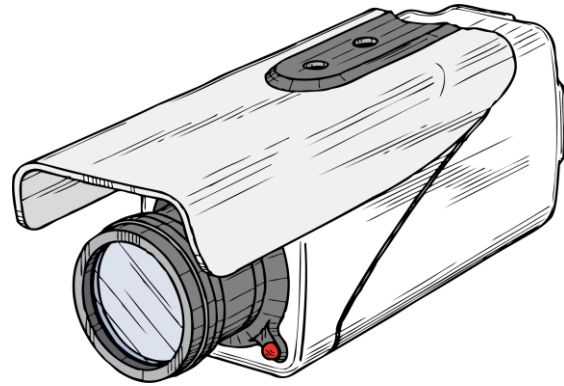


# Monitoring 2.0 – Alles im Lot?

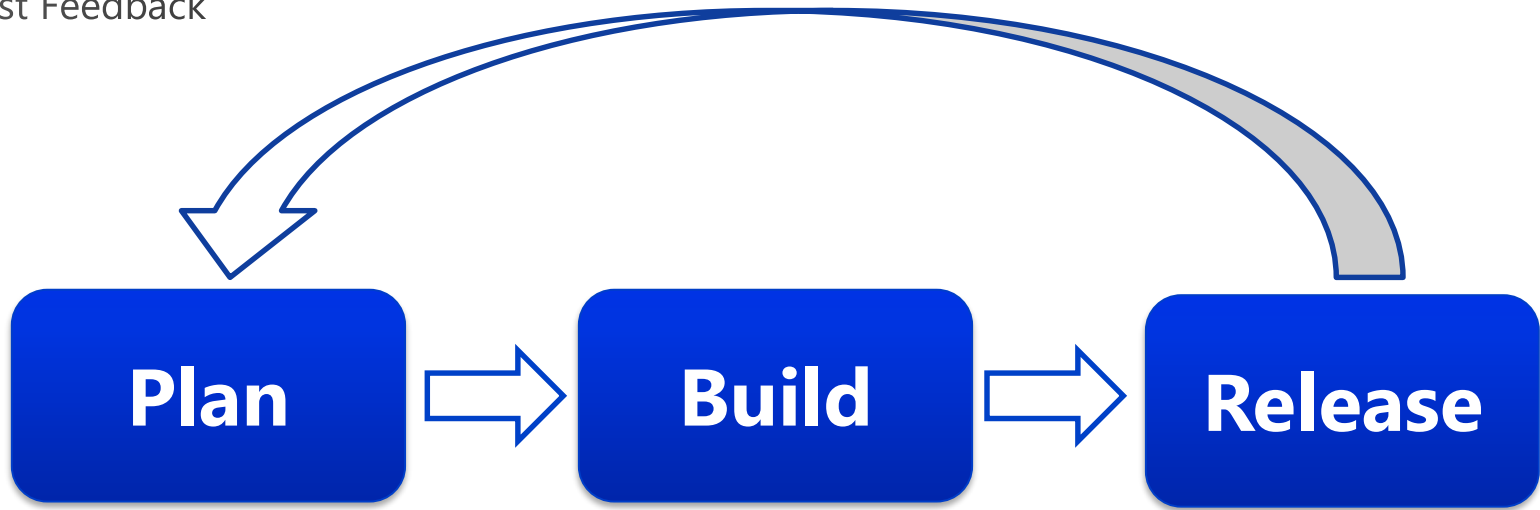


Nico Orschel, MVP @ AIT, DE

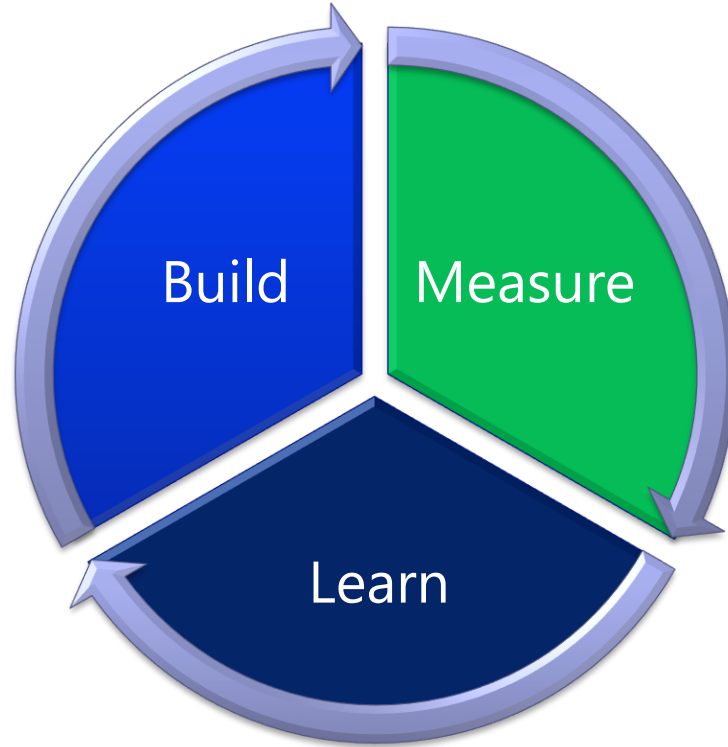
Marc Müller, MVP @ 4tecture, CH

# Ziele der modernen Software-Entwicklung

- Cycle-Times reduzieren
- Fast Feedback



# DevOps - BML



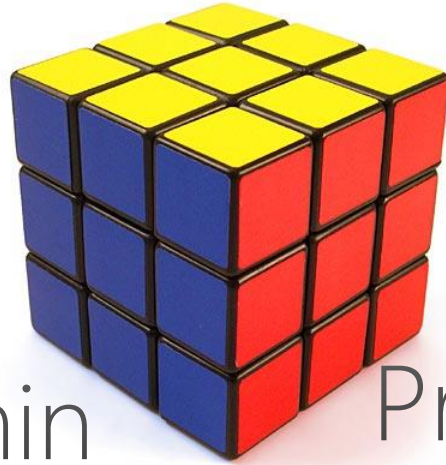
# Monitoring

... ist ein Überbegriff für alle Arten der unmittelbaren systematischen Erfassung (Protokollierung), Messung, Beobachtung oder Überwachung eines Vorgangs oder Prozesses mittels technischer Hilfsmittel (zum Beispiel Langzeit-EKG) oder anderer Beobachtungssysteme. Dabei ist die wiederholte regelmäßige Durchführung ein zentrales Element der jeweiligen Untersuchungsprogramme, um anhand von Ergebnisvergleichen Schlussfolgerungen ziehen zu können (siehe auch Längsschnittstudie).

Quelle: <https://de.wikipedia.org/wiki/Monitoring>

# Überwachung mehrere Dimensionen

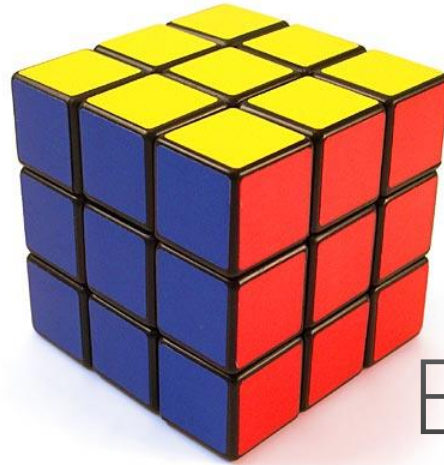
Entwickler



IT-Admin

Projektmanager

# Überwachung mehrere Dimensionen



Funktional

End-2-End

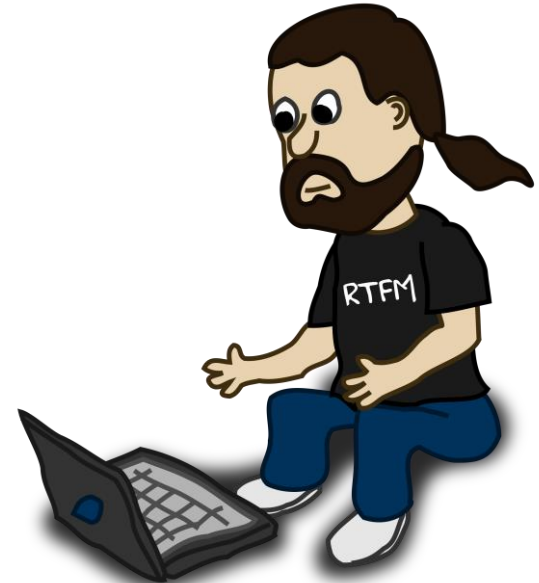
Nicht-  
Funktional

# Beispiele für nicht-funktionale Metriken (Der Admin)

Ziel: Primäre technische Kennzahlen vom Server

Beispiele:

- CPU / Memory Utilization
- Response Time
- Transferred Data
- Data Transfer Rate (e.g MBits/sec)
- Connections
- ...



# Beispiel Funktional (Der Entwickler)

Ziel: Primäre technische Kennzahlen aus der Applikation

Beispiele:

- Exceptions Client
- Exceptions Server
- Stacktraces
- Non responsive external Services (Salesforce, SAP, CRM, Internal Services, ...)





# Beispiel Business (Der Analyst)

Ziel: Überwachung der (Business-)Prozesse

Beispiele:

- Welche Features meiner Anwendung werden „tatsächlich“ genutzt?
- Wie intensiv werden Features genutzt?
- In welche Features müssen wir investieren?
- Wird die Anwendung aus Kundenperspektive langsamer?



# Arten an Monitoring



# Beispiel aus unserem Projektalltag

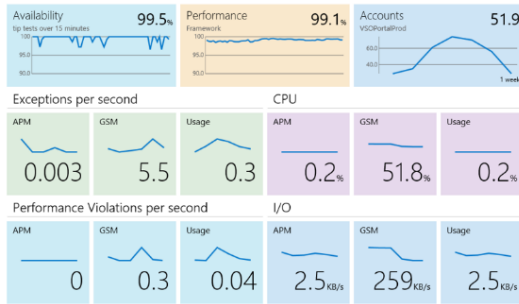
Unser Fokus heute: Microsoft Azure Application Insights

Framework zur Überwachung von Anwendungs-Telemetriedaten

Alternativen:

[PreEmptive Analytics](#), [New Relic Software Analytics](#)

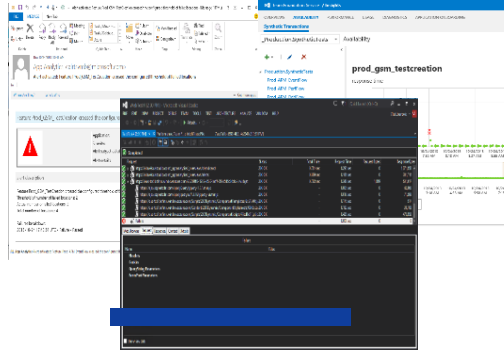
# Application Insights



Available

Health Dashboards

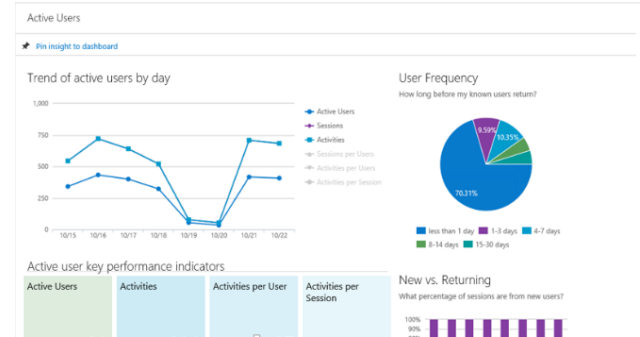
Is my application available and performing for users?  
One Dashboard



Performing

Notifications & Deep Insights

What's wrong?  
Show me suspicious code and test cases



Improving

Usage Dashboards

Where do we invest next?  
Show me top features and customer Usage patterns

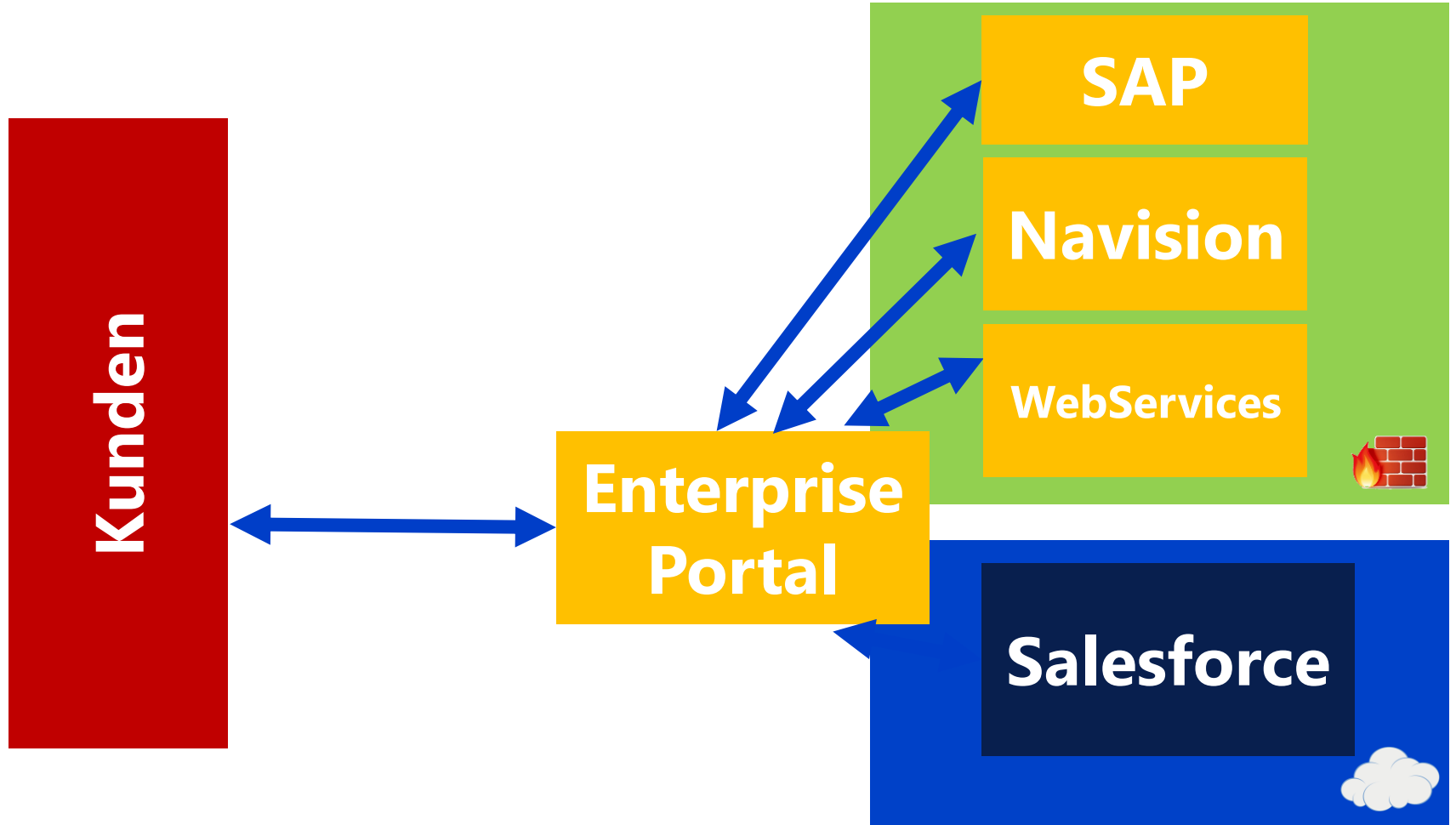


Theorie ist gut ...  
Praxis ist „anders“ ...

# Fallstudie

WebApp

# Problemstellung





# Überwachung externe Websites

## Problem:

- a) Zyklische Überwachung eines extern. erreichbaren Webportals
- b) Seite enthält viel JavaScript Logik sowie ext. Services

## Lösung:

Monitoring der Verfügbarkeit über WebTests

- Aufzeichnung via VS Ultimate, Fiddler
- Direkte Ausführung via Azure Portal

# Offene Punkte – Out of the box AI

Testen von komplexer JavaScript Logik

JavaScript Code teilweise voneinander abhängig

Externe Services überwachen (ohne die Möglichkeit AI zu integrieren)

# Überwachung v. int. Services

## Problem:

- a) Services ggf. schlecht via webtest überwachbar
- b) Daten sollen im globalen Monitoring überwacht werden

## Lösung:

Verwendung von anderen Testing Frameworks / CMD

- CodedWebTests
- Unit Tests (MSTest, Xunit,...)

Reporting und Fehleranalyse durch eigene Zusatzdaten via Azure AI SDK

Ausführung zyklisch via Task Scheduler/CMD

# Funktionales Monitoring

## Problem:

- a) Prüfen ob Kernszenarien (End2End) noch funktionieren
- b) WebApp setzt intensiv auf JavaScript

## Lösung:

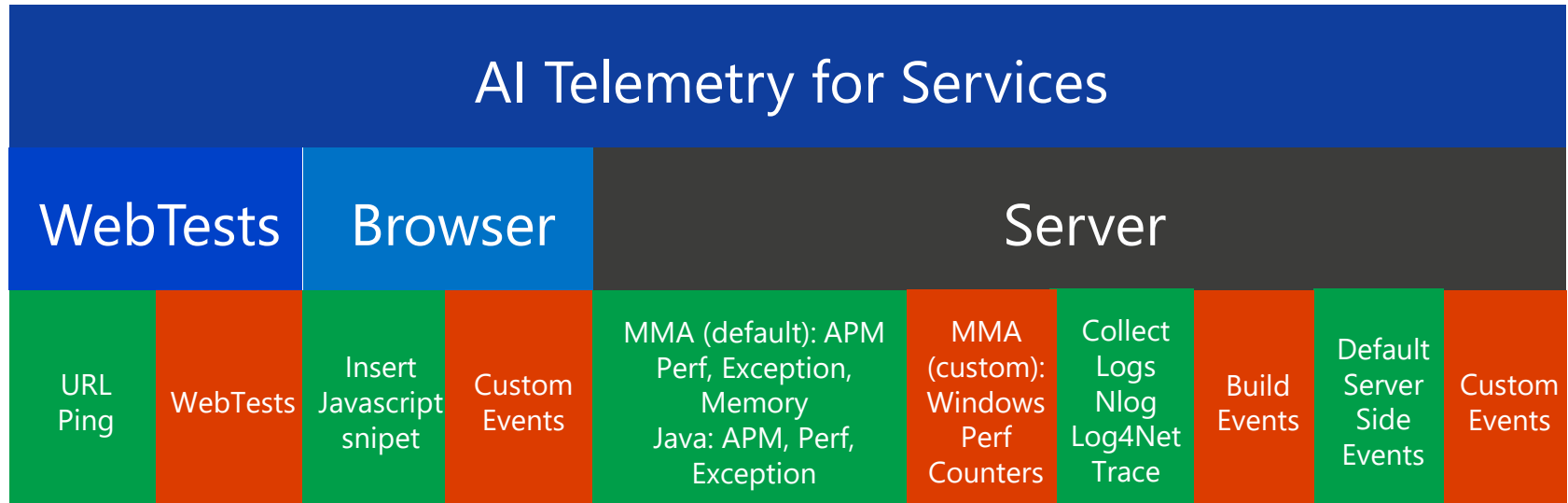
Funktionale Tests via UI oder Unit Tests (Selenium, CodedUI, ...)

Ausführung: Zyklisch via CMD / Task Scheduler

Reporting und Fehleranalyse durch eigene Zusatzdaten via Azure AI SDK

# Application Insights

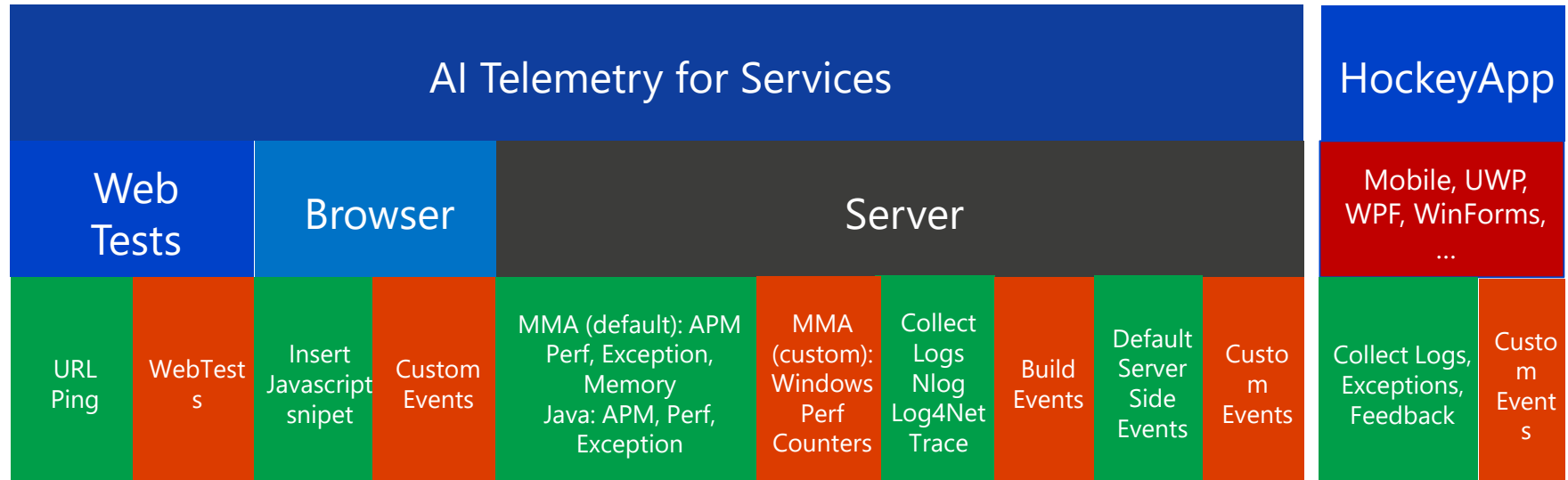
# Services and AI Telemetry



Easy to configure: No coding, simple, fast, few clicks.

Requires more time: could require coding, more clicks.

# Services and AI Telemetry and HockeyApp



Easy to configure: No coding, simple, fast, few clicks.

Requires more time: could require coding, more clicks.

# What is Application Insights?

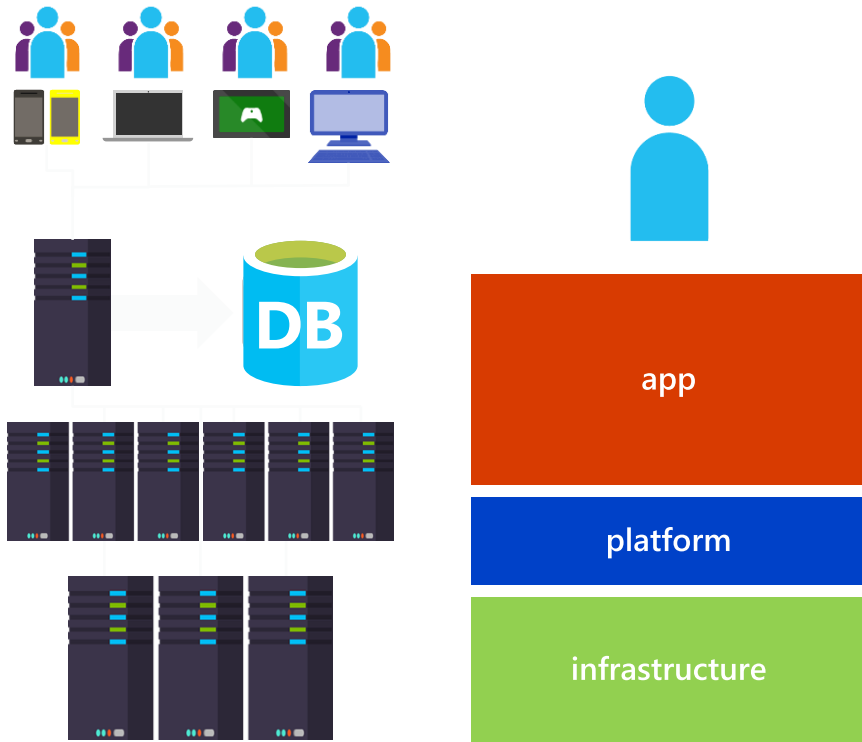
1

3





# Sources of Telemetry



Outside-in monitoring  
URL pings and web tests from 16 global points of presence



Observed user behavior  
How is the application being used?



Developer traces and events  
Whatever the developer would like to send to Application Insights



Observed application behavior  
No coding required – service dependencies, queries, response time, exceptions, logs, etc.



Infrastructure performance  
System performance counters

# Availability

# Availability

Ist meine Applikation / meine Services noch verfügbar?

## Http basierte Webtests

- Einfacher URL Call
- Komplexer Web-Test

## Alert

- Status Code
- Anzahl fehlerhafter Locations

# Availability

## Ping auf URL

Create test  
PREVIEW

\* Test name  
Simple Ping

Test type  
URL ping test

\* URL ⓘ  
http://PracticalDevOps-Dev.azurewebsites.net

Parse dependent requests ⓘ

Enable retries for web test failures. ⓘ

Test frequency ⓘ  
5 minutes

Test locations ⓘ  
5 location(s) configured

Success criteria ⓘ  
HTTP response: 200, Test Timeou...

Alerts ⓘ  
Alert if 3/5 locations fails in 5 mi...

## WebTests

Create test  
PREVIEW

\* Test name  
Complex Webtest

Test type  
Multi-step test

Upload a multi-step test  
GetBooksTest.webtest

Enable retries for web test failures. ⓘ

Test frequency ⓘ  
5 minutes

Test locations ⓘ  
5 location(s) configured

Success criteria ⓘ  
Criteria specified in test file

Alerts ⓘ  
Alert if 3/5 locations fails in 5 mi...

# Tracking

# User Tracking

## JavaScript bei Page Load

- Page Navigation Events
- Custom Events

## Async, Queued

```
<script type="text/javascript">
  var appInsights = window.appInsights || function (config) {
    function r(config) { t[config] = function () { var i = arguments; t.queue.push(function () { t[config].apply(t, i) }) } } var t = { config: config }, u = document, e =
    window, o = "script", s = u.createElement(o), i, f; for (s.src = config.url || "//az416426.vo.msecnd.net/scripts/a/ai.0.js",
    u.getElementsByTagName(o)[0].parentNode.appendChild(s), t.cookie = u.cookie, t.queue = [], i = ["Event", "Exception", "Metric", "PageView", "Trace"]; i.length;) r("track" +
    i.pop()); return r("setAuthenticatedUserContext"), r("clearAuthenticatedUserContext"), config.disableExceptionTracking || (i = "onerror", r("_" + i), f = e[i], e[i] = function
    (config, r, u, e, o) { var s = f && f(config, r, u, e, o); return s !== !0 && t["_" + i](config, r, u, e, o), s })), t
    }({
      instrumentationKey: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    });

    window.appInsights = appInsights;
    appInsights.trackPageView();
  }
</script>
```

# Backend Tracking

```
public class Startup
{
    1 reference | Marc Müller, 18 days ago | 1 author, 1 change
    private class AIInitializer : ITelemetryInitializer
    {
        0 references | Marc Müller, 18 days ago | 1 author, 1 change
        public void Initialize(ITelemetry telemetry)
        {
            telemetry.Context.Properties["Env"] = ConfigurationManager.AppSettings["InstrumentationEnvironment"];
            telemetry.Context.Component.Version = System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString();
        }
    }

    1 reference | Marc Müller, 23 days ago | 1 author, 1 change
    private class AiExceptionHandler : ExceptionLogger
    {
        1 reference | Marc Müller, 23 days ago | 1 author, 1 change
        public override void Log(ExceptionLoggerContext context)
        {
            if (context != null && context.Exception != null)
            {
                var ai = new TelemetryClient();
                ai.TrackException(context.Exception);
            }

            base.Log(context);
        }
    }

    0 references | Marc Müller, 18 days ago | 1 author, 4 changes
    public void Configuration(IApplicationBuilder app)
    {
        TelemetryConfiguration.Active.InstrumentationKey = ConfigurationManager.AppSettings["InstrumentationKey"];
        TelemetryConfiguration.Active.TelemetryInitializers.Add(new AIInitializer());

        // Allow CORS
        app.UseCors(CorsOptions.AllowAll);

        // Configure and add Web API
        configuration.Services.AddMvc();
    }
}
```

# Custom Events

Quelle: <https://azure.microsoft.com/en-us/documentation/articles/app-insights-api-custom-events-metrics/>

Beispiele: TrackPageView, TrackEvent, TrackMetric, TrackException, TrackRequest, TrackTrace, TrackDependency

Beispiel:

```
var telemetryClient = new TelemetryClient();  
telemetryClient.TrackEvent($"Generating {numberOfBooks} books");
```



# Performance

# Application Insights Agent

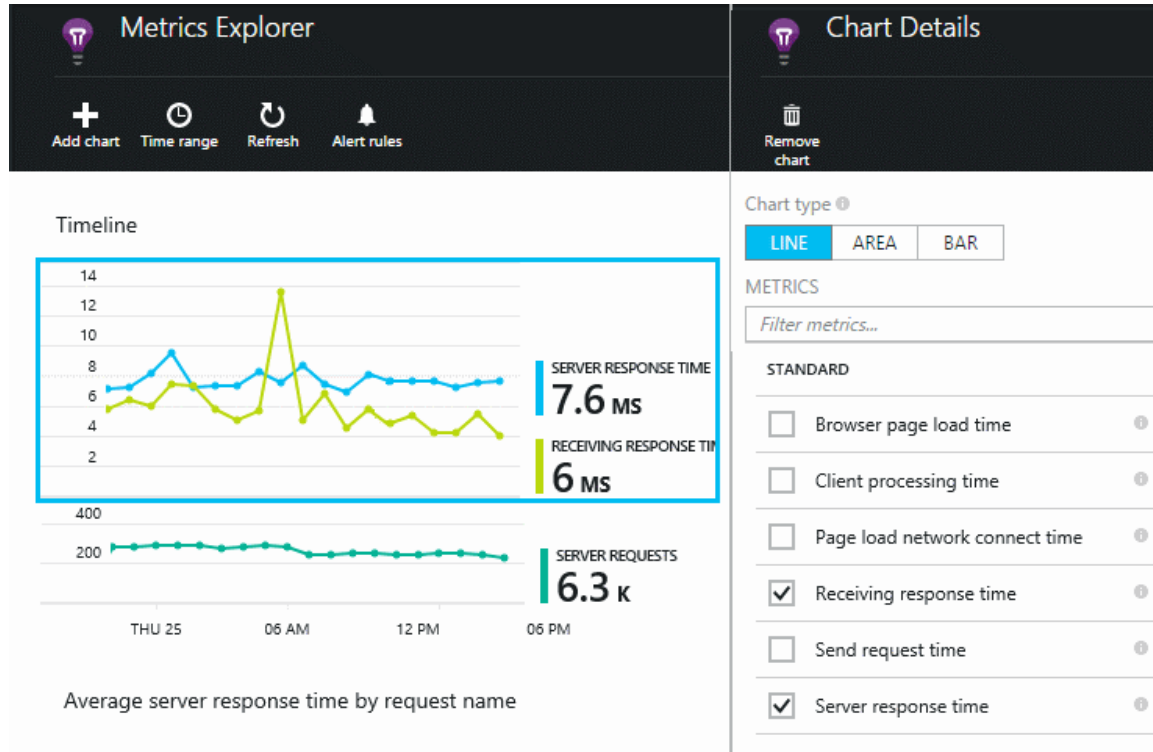
Überwachung des Ausführungsprozesses

Überwachung der Performance Indikatoren

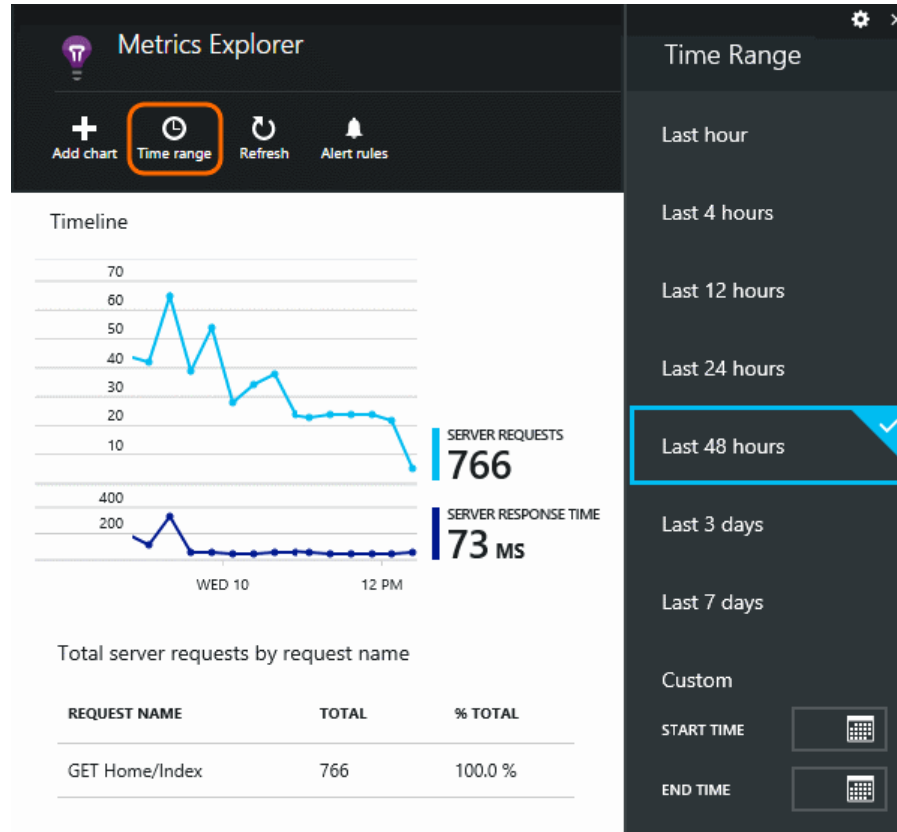
Installation auf Zielsystem notwendig

# Ergebnis – 360 Grad Monitoring

# Auswertung – Metric Explorer ( 1 / 2 )



# Auswertung – Metric Explorer ( 2 / 2 )

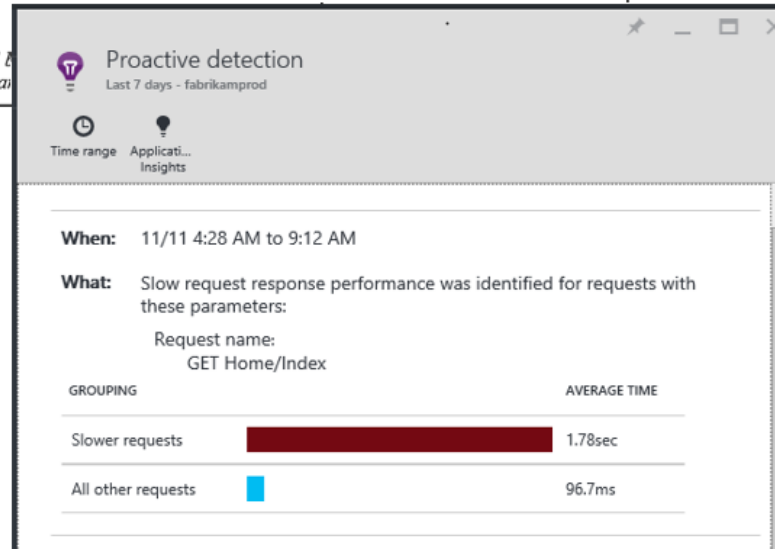


# Proactive Detection

Don't forget, page load times are critical to success for web applications. Slow pages are frustrating for your users.

[Check out the page view performance of fabrikamprod in Application Insights](#)  
[Learn how to monitor web page performance](#)

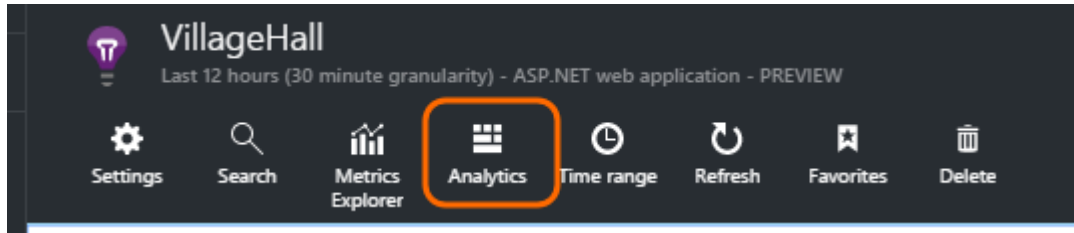
*You got this email because you used fabrikamprod, or another application.*



# Application Insights Analytics

Analyse von allen AI Daten über eigene Abfragen im Browser

Codename: Kusto





Export ▾

▶ GO

requests

```
| where timestamp > ago(7d) and toint(duration) > 0  
| summarize avg(duration), percentiles(duration,50,95) by bin(timestamp,2h)  
| order by percentile_duration_95 | render timechart
```

Completed

00:00:07 :7153

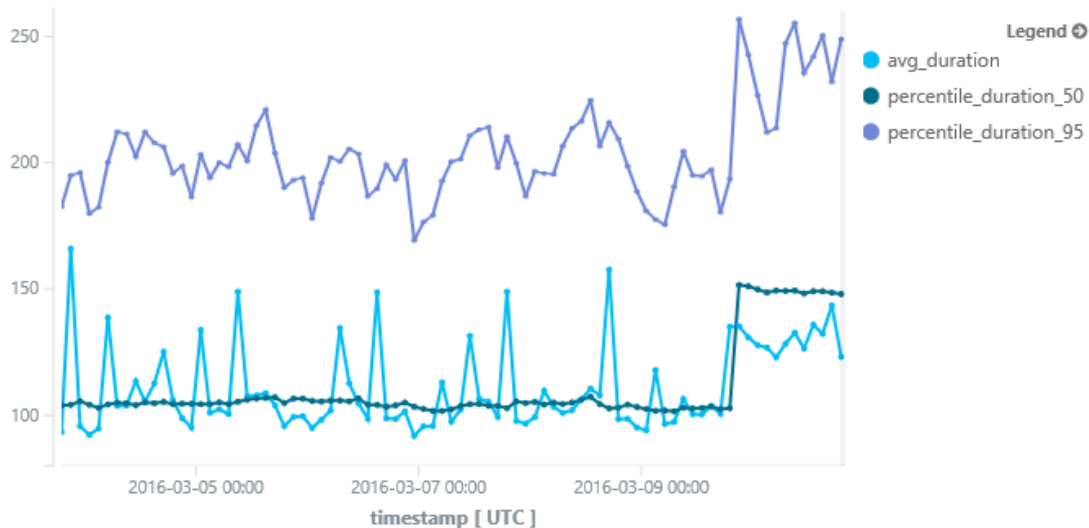
385 records matched



Line chart ▾

timestamp ▾

3 selected ▾





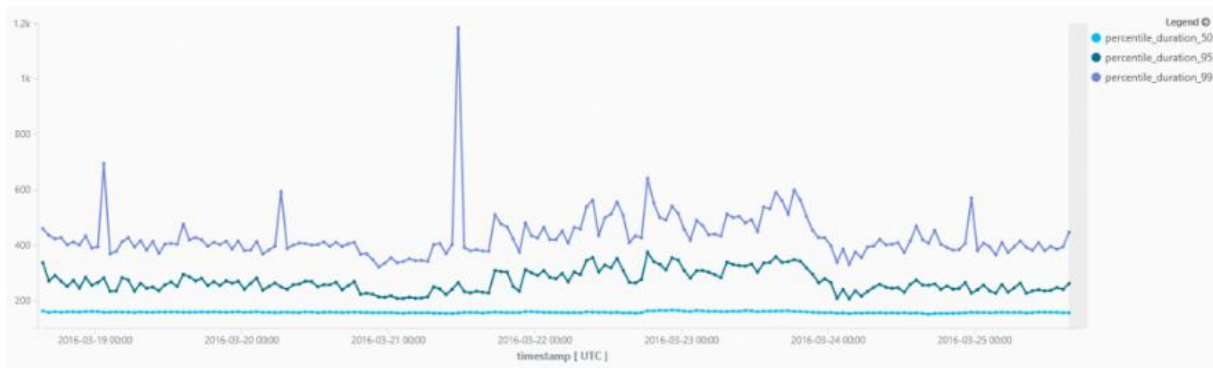
# Application Insights Analytics - Beispiele

requests

| where timestamp >= now(-7d)

| summarize percentiles(duration,50,95,99) by bin (timestamp, 1h)

| render timechart



# Application Insights Analytics - Beispiele

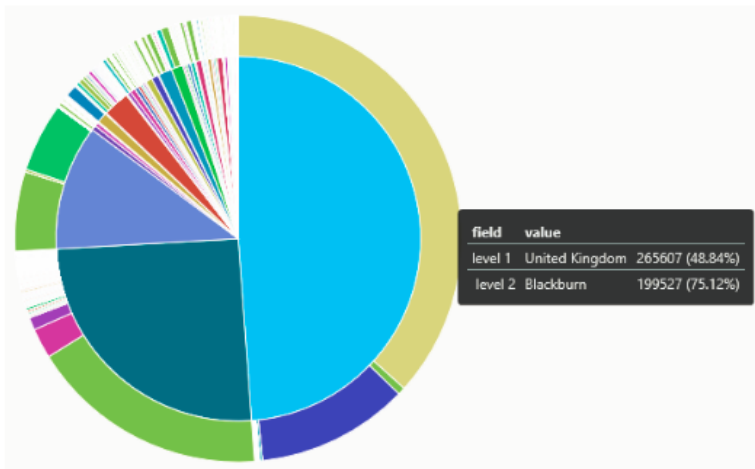
requests

| where timestamp >= ago(24h)

| summarize requestCount=count() by client\_CountryOrRegion, client\_City

| order by requestCount desc

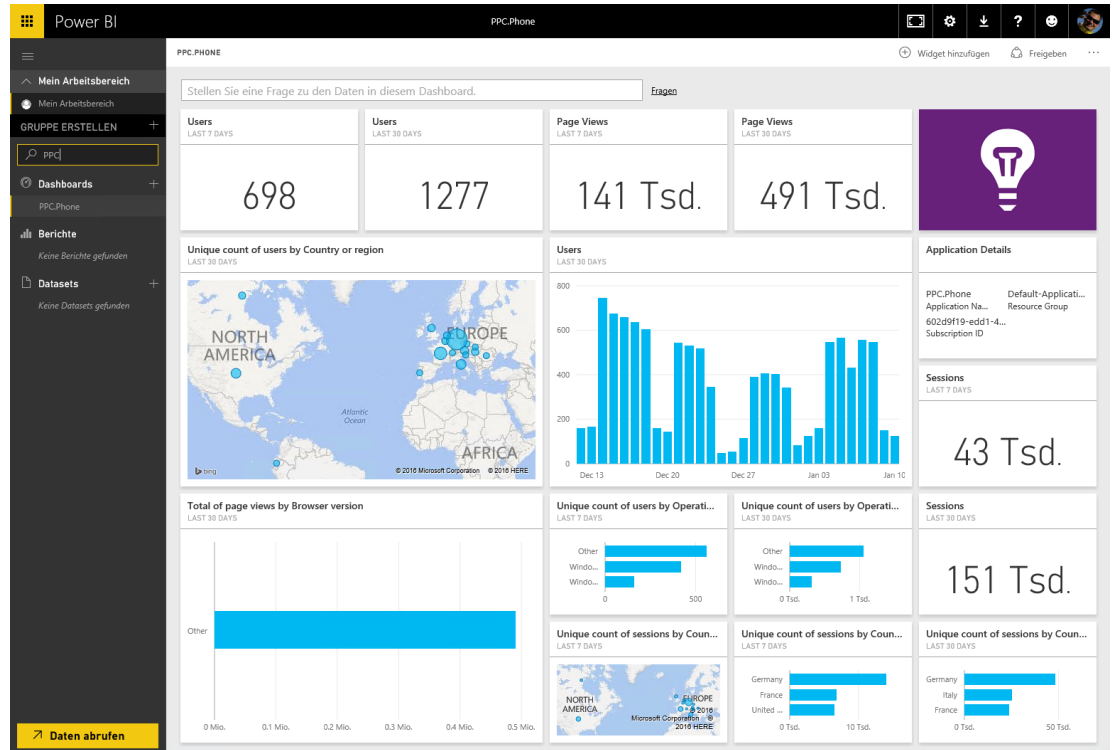
| render piechart



# Application Insights

## Power BI

- Business-Sicht
- Integration in SharePoint und Webseiten



# Wir unterstützen Sie

**KONTAKT**  
Nico.Orschel@aitgmbh.de  
+49 151 55052624



**AIT GmbH & Co. KG**  
Leitzstr. 45, 70469 Stuttgart  
www.aitgmbh.de

**BERATUNG**  
Agile ALM und TFS  
.NET und Architektur



**ENTWICKLUNG**  
Dienstleister für individuelle  
Lösungen mit .NET und Azure



© AIT GmbH & Co. KG – Alle genannten und gezeigten Marken oder Warenzeichen sind eingetragene Marken oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer und ggf. nicht gesondert gekennzeichnet. Aus dem Fehlen der Kennzeichnung kann nicht geschlossen werden, daß es sich bei einem Begriff oder einem Bild nicht um eine eingetragene Marke oder ein eingetragenes Warenzeichen handelt.

# 4tecture<sup>®</sup>

empower your software solutions

Marc Müller

Principal Consultant  
für DevOps, ALM, TFS /VS, .NET



4tecture<sup>®</sup>

E-Mail: [marc.mueller@4tecture.ch](mailto:marc.mueller@4tecture.ch)  
Webseite: <http://www.4tecture.ch>  
Schulungen: <http://4tecture.ch/trainings>  
Blog: <http://4tecture.ch/blog>  
Twitter: @muellermarc

4tecture GmbH  
Aathalstrasse 84  
CH-8610 Uster

+41 44 508 37 00  
[info@4tecture.ch](mailto:info@4tecture.ch)





4tecture<sup>©</sup>  
empower your software solutions

# Weiterführende Informationen

- <https://azure.microsoft.com/en-us/documentation/articles/app-insights-analytics/>
- <https://azure.microsoft.com/en-us/documentation/articles/app-insights-web-monitor-performance/>
- <https://azure.microsoft.com/en-us/documentation/articles/app-insights-proactive-detection/>