

# Web APIs auf dem Prüfstand

Volle Kontrolle oder fertig mit den Azure Mobile Services?



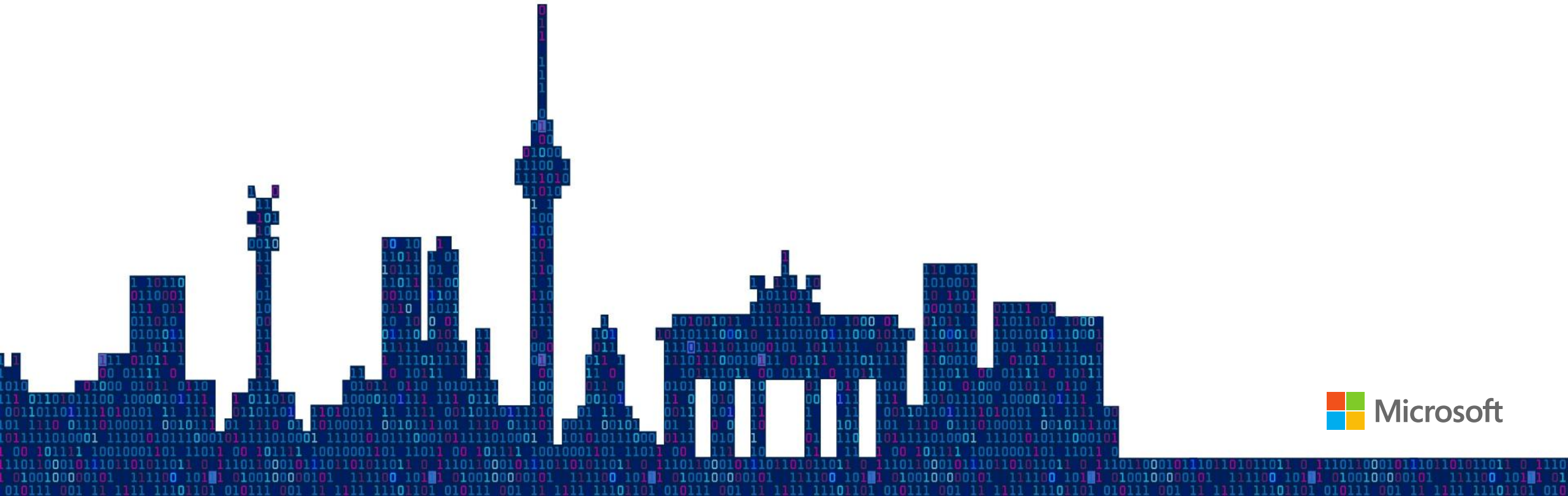
Peter Kirchner  
Microsoft Deutschland



Technical Summit 2014  
Pure Technology.

Die Microsoft Konferenz  
für Entwickler und IT Pros vom  
**11.–13. November in Berlin**

# Web APIs



# Wo kommen wir her?

Remote Procedure Calls (RPC)

Verben/Aktionen im Endpunkt enthalten

GetCustomer

InsertInvoice

DeleteCustomer

Jeder Endpunkt entspricht einem eigenen URI

# Representational State Transfer (REST)

REST is kein Standard, verwendet aber Standards

Architekturkonzept

Roy Fielding in 2000

“Architectural Styles and the Design of Network-based Software Architectures”

HTTP Stack, expliziter Gebrauch von HTTP

Methoden/Verben

# Grundlegende Entwurfsprinzipien von REST

Statuslos (stateless)

URIs mit Verzeichnisstruktur

XML, JavaScript Object Notation (JSON) oder beides

Übertragung von Snapshots der Daten im Body

Keine Anweisungen oder Methoden im Body

# Anatomie von HTTP-Nachrichten

## Anfrage (Request)

- Request Line
- Request Headers
- Request Body

## Antwort (Response)

- Status Line
- Response Headers
- Response Body

# Beispiel: GET

## Anfrage (Request)

GET http://localhost:21697/api/employees/12345 HTTP/1.1

User-Agent: Fiddler

Host: localhost:21697

## Antwort (Response)

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Date: Tue, 04 Nov 2014 15:48:03 GMT

Content-Length: 50

```
{"Id":12345,"FirstName":"John","LastName":"Human"}
```

# Beispiel: PATCH

## Anfrage (Request)

PATCH http://localhost:21697/api/employees/12345 HTTP/1.1

Content-Type: application/json

Host: localhost:21697

Content-Length: 23

```
{"LastName":"Kirchner"}
```

## Antwort (Response)

HTTP/1.1 204 No Content

Date: Tue, 04 Nov 2014 16:36:35 GMT



# HTTP-Methoden

Create, Read, Update und Delete-Operationen (CRUD) bilden auf HTTP-Methoden ab.

Erstellen (Create)	→ POST
Lesen (Read)	→ GET
Ändern (Update)	→ PUT / PATCH
Löschen (Delete)	→ DELETE

Generelles Entwurfsprinzip: Nomen statt Verben verwenden

# ASP.NET Web API

```
public class HelloController : ApiController
{
    // GET: api/Hello
    // references
    public IEnumerable<string> Get()
    {
        return new string[] { "value1", "value2" };
    }

    // GET: api/Hello/5
    // references
    public string Get(int id)
    {
        return "value";
    }

    // POST: api/Hello
    // references
    public void Post([FromBody]string value)
    {
    }

    // PUT: api/Hello/5
    // references
    public void Put(int id, [FromBody]string value)
    {
    }

    // DELETE: api/Hello/5
    // references
    public void Delete(int id)
    {
    }
}
```

ASP.NET MVC Modell

Controller → ApiController

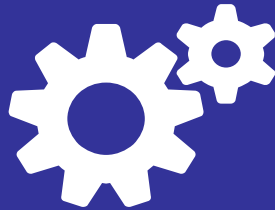
# Demo: ASP.NET Web API



# Microsoft Azure Mobile Services



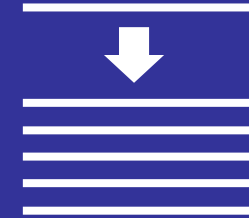
Authentifizieren



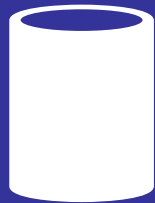
Server-Logik



Benachrichtigen  
(Push)



Protokollieren



Daten verwalten



Diagnose



Skalieren



# Demo: Azure Mobile Services

# Entwicklungsumgebung

## ASP.NET MVC Web API

- .NET-Sprachen wie C#, VB.NET, ...
- MVC-Architektur
- I.d.R. Remote-Debuggen möglich
- Lokales Testen möglich

## Azure Mobile Services

### Mit JavaScript-Backend

- JavaScript und node.js
- Kein lokales Testen
- Kein Remote-Debuggen

### Mit .NET-Backend

- ASP.NET MVC Web API  
Siehe linke Seite ☺
- Integrierte Features von AMS

# Deployment

## ASP.NET MVC Web API

- I.d.R. IIS in vielfältiger Umgebung  
Eigener Server  
Hoster  
Azure Websites  
Azure Cloud Services
- Übertragung abhängig von Umgebung  
FTP  
Web Deploy  
Git, TFS

## Azure Mobile Services

- Git

## Mit JavaScript-Backend

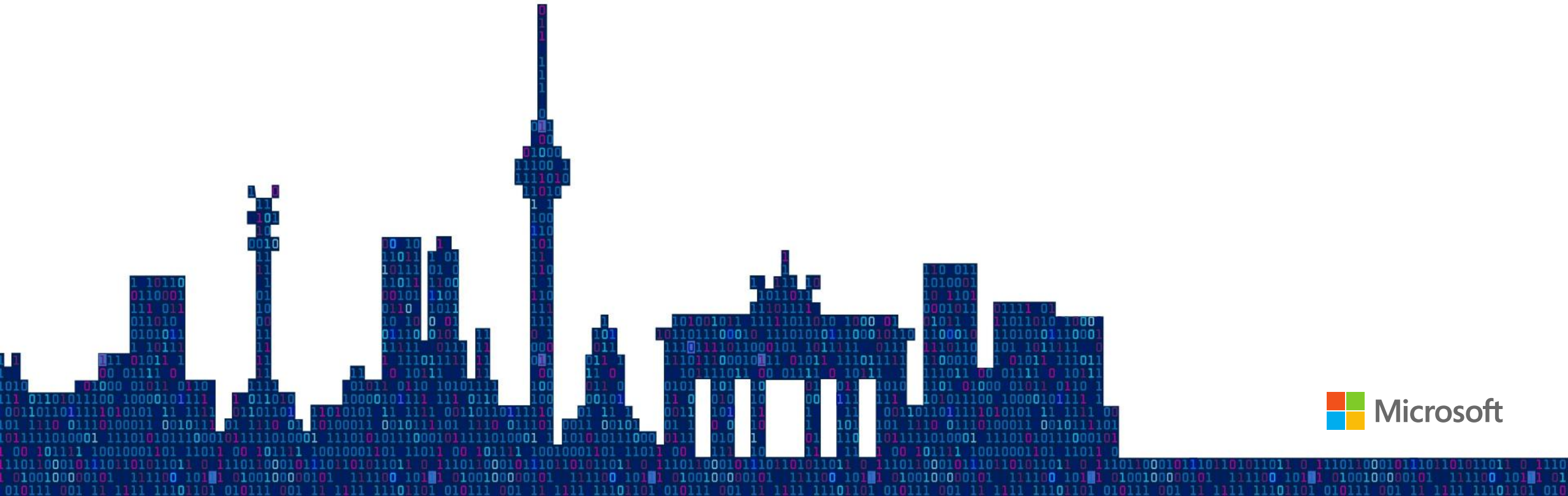
- Manuelles Editieren im Verwaltungsportal

## Mit .NET-Backend

- Ausführung auf Azure Websites  
Übertragung dadurch mit Web Deploy oder FTP



# Daten speichern





# Daten

## ASP.NET MVC Web API

- Datensinke erstellen, verbinden und verwenden

## Azure Mobile Services

- Azure SQL Database
- Offline-Synchronisation möglich
- Datenbankzugriff abstrahiert  
Client- und serverseitig

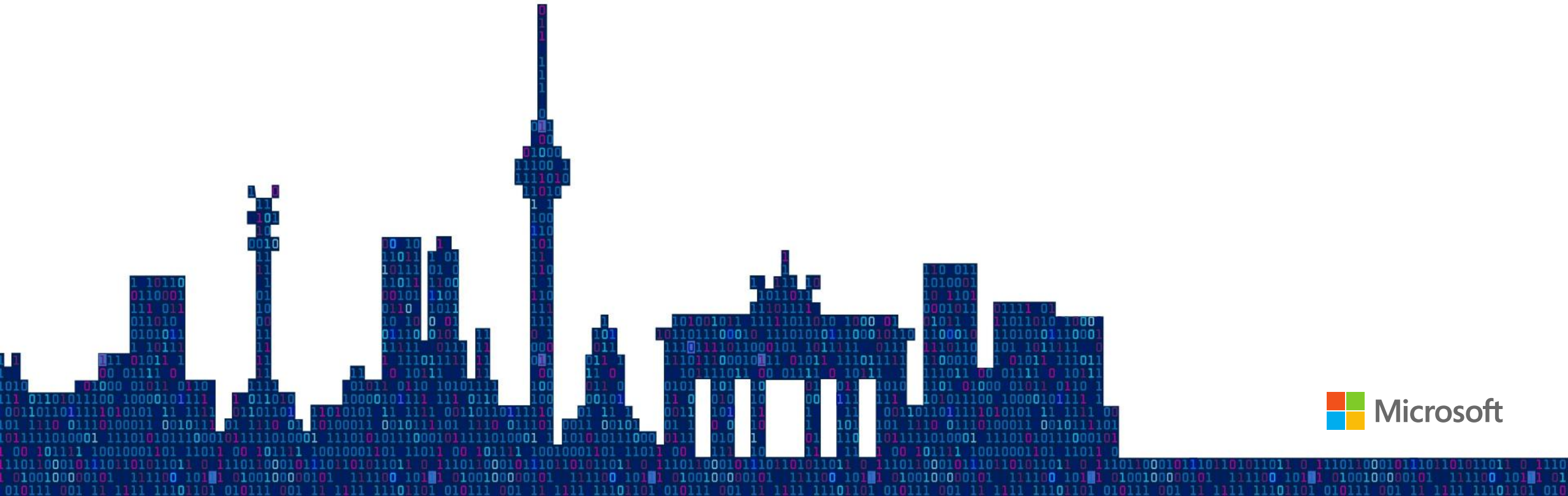
## Mit JavaScript-Backend

- Dynamisches Schema

# Demo: Datenspeichern



# Integration



# Integration von REST-Diensten

## Unterstützung für REST durch viele Bibliotheken

Alternativ: separate Unterstützung für HTTP, JSON und XML

## Breite Tool-Unterstützung zum Testen

GET-Anfragen mit Browsern

Auswahl von HTTP-Methoden in Tools wie Fiddler oder Runscope

## Entwicklerunterstützung

SDKs für Client-Entwicklung

Azure API Management

# SDKs in Azure Mobile Services

## SDKs in Azure Mobile Services für Vielzahl an Plattformen

Windows Universal Apps

Android

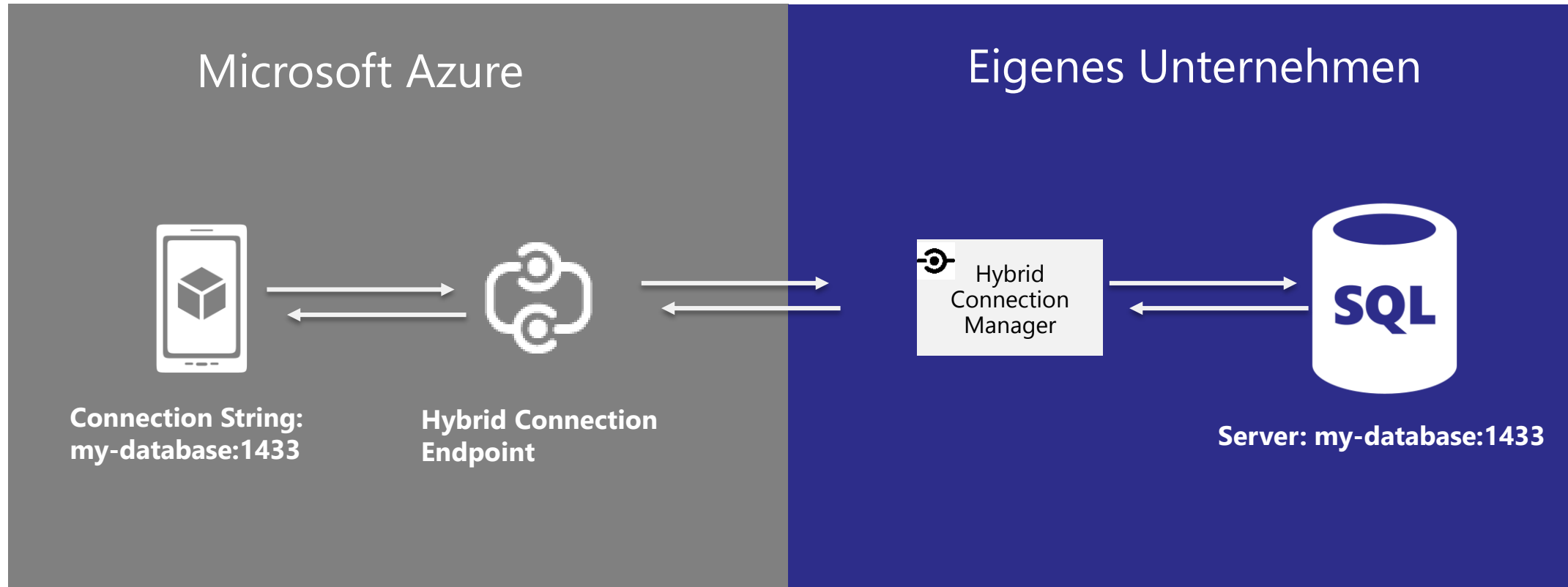
iOS

HTML / JavaScript

Xamarin

PhoneGap

# Verbindung mit lokaler Infrastruktur

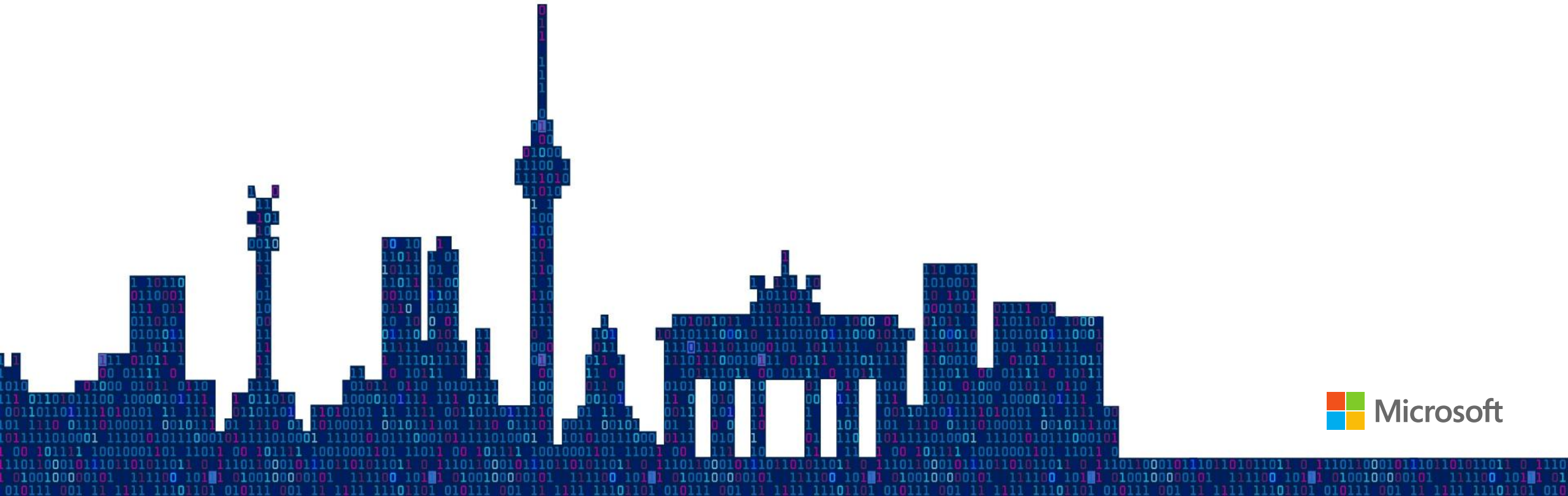




# Demo: Integration



# Absicherung





# TLS / SSL

Konfiguration entspricht der Einrichtung von Web-Servern

Unabhängig von REST oder Web API

Konfiguration abhängig von der Installation

Hoster

VMs

Azure Website

Cloud Service

Azure Mobile Service



pkwp

## zertifikate



BETREFF	ABLAUFDATUM	FINGERABDRUCK
www.pi-ter.de	19.09.2014	FF62413E0C04A80CC0944D17D9C1CE656763FD6E

zertifikat hochladen

## domänennamen



www.pi-ter.de
pkwp.azurewebsites.net

domänen verwalten

## ssl-bindungen



www.pi-ter.de	(Abgelaufen) www.pi-ter.de	SNI SSL
<input type="text" value="Einen Domänennamen wählen"/> ▼	<input type="text" value="Ein Zertifikat wählen"/> ▼	<input type="text" value="SNI SSL"/> ▼

# Authentifikation

## ASP.NET MVC Web API

- Implementierung bei ASP.NET MVC Web API benutzerdefiniert

## Azure Mobile Services

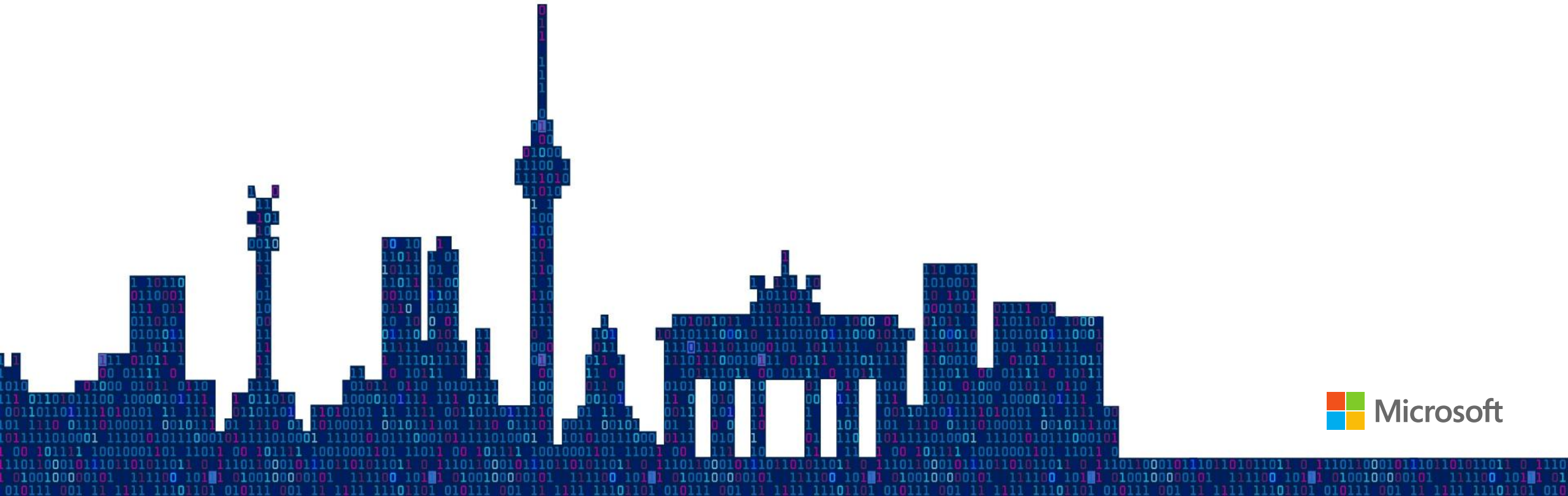
### Beinhaltet SSO-Provider

- Twitter
- Google
- Facebook
- Microsoft-Konto
- Azure Active Directory

# Demo: Absicherung



# Resumé



# Auswahl des Frameworks und der Plattform

	Web API On-Prem	Web API auf Azure	Azure Mobile Services
HTTP Stack	Web API	Web API	Web API
Data Stack	EF, nach Belieben	EF, Azure Tables, nach Belieben	EF, Azure Tables etc. Automatische SQL-Provisionierung
Hosting	BYO	PaaS	PaaS + seamless framework updates and bugfixes
Device SDK	BYO	BYO	Windows, iOS, Android, HTML, Xamarin, PhoneGap
Auth/Offline	NuGet	NuGet	Integriert für Social SSOs und AAD Offline basiert auf SQLite
Push	NuGet	Notification Hubs	Integriert mit Notification Hubs mit eingebauter Authentifizierung und Analytics

# Vielen Dank



Peter.Kirchner  
@Microsoft.com



[blogs.msdn.com/  
pkirchner](https://blogs.msdn.com/pkirchner)



Twitter:  
@peterkirchner

© 2014 Microsoft Corporation. All rights reserved. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.