

# Windows Azure Services Plattform

Basis für Software + Services Lösungen

Holger Sirtl

Architect Evangelist, S+S Lead

Microsoft Deutschland GmbH

# Agenda

- Definition von Software-plus-Services
- Microsoft Azure – ein Überblick
- Herausforderungen bei Software-plus-Services  
Lösungen
- Antworten mit Microsoft Azure

# Software + Services

Lokal betriebene Software kombiniert mit Cloud Services

- User Experience
- Verschiedene Endgeräte
- Offline-Fähigkeit

Betrieb  
**vor-Ort**



**Client**

- Flexible Konfiguration (Funktionalität)
- Vertrauen
- Regularien
- Unternehmenskritische Anwendungen



**Server**

Vor-Ort  
Software



Betrieb  
bei einem  
**Dienstleister**



**Services**

Cloud  
Services

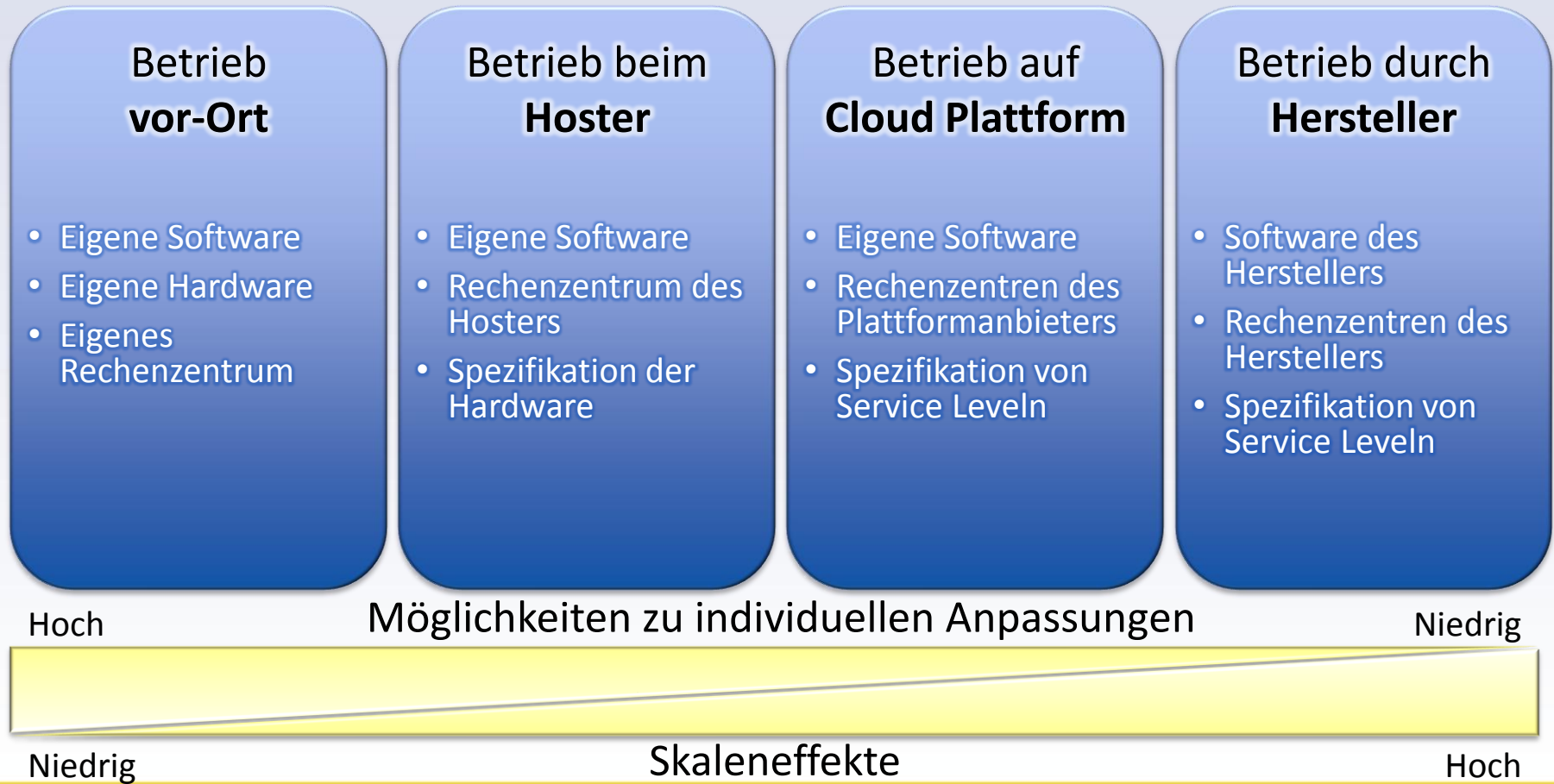
- Skalierbarkeit
- Flexible Bezahlung / alternative Bezahlmodelle
- Flexible Bereitstellung
- Kurze Time-to-Market

Microsoft

**Technical Summit**  
2008

# Alternativen für den Softwarebetrieb

## Möglichkeiten zu Anpassungen vs. Skalierbarkeit



# Microsoft Azure

## Plattform für Partner- und Microsoft-Dienste

Partner Lösungen

Windows Live  
Microsoft  
Office Live

Microsoft  
Dynamics CRM Online  
Microsoft  
Exchange Online  
Microsoft  
SharePoint Online

### Azure™ Services Platform

 Live Services

 Microsoft  
.NET Services

 Microsoft  
SQL Services

Microsoft  
SharePoint Services

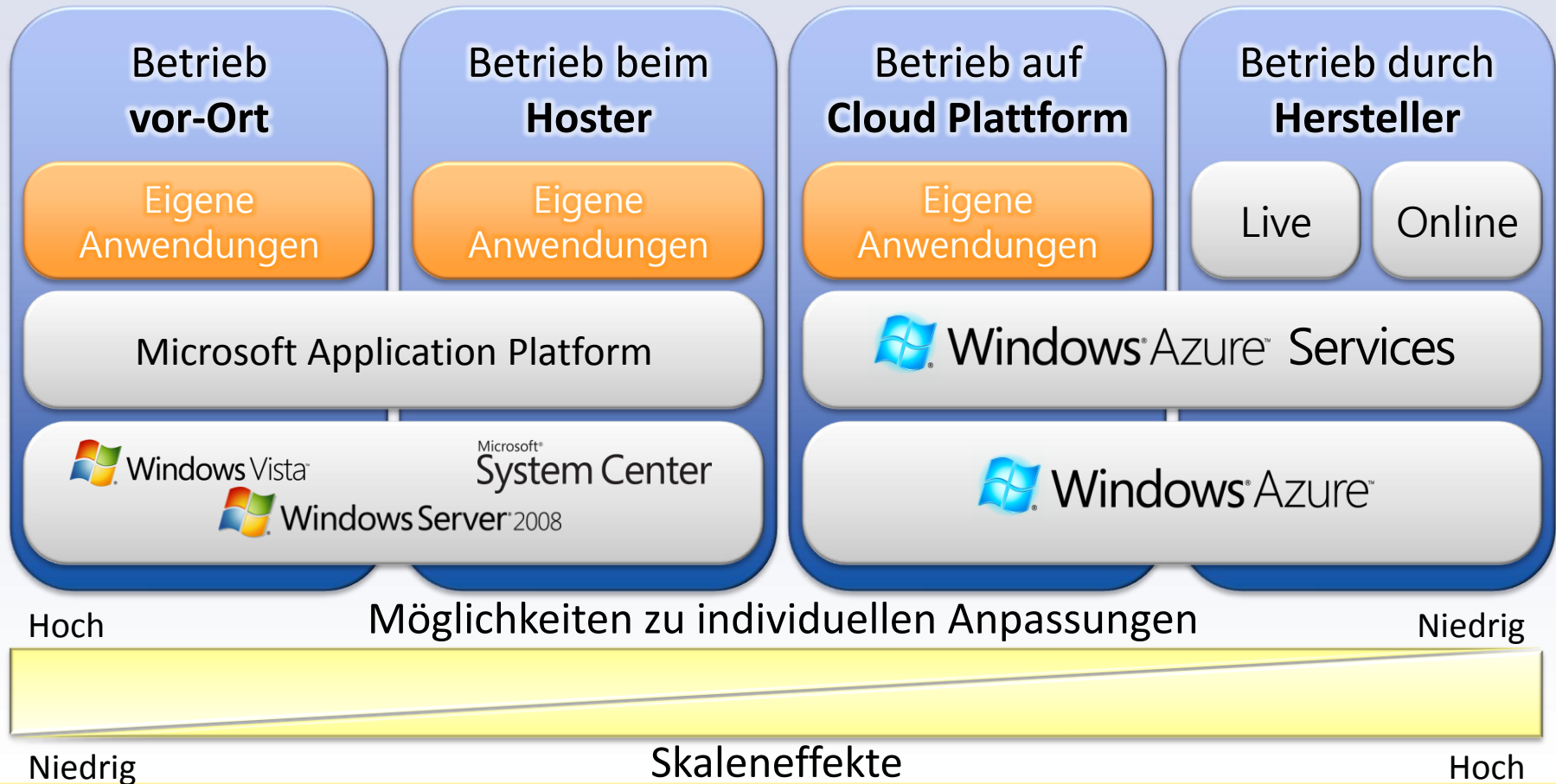
Microsoft  
Dynamics CRM Services

 Windows® Azure™

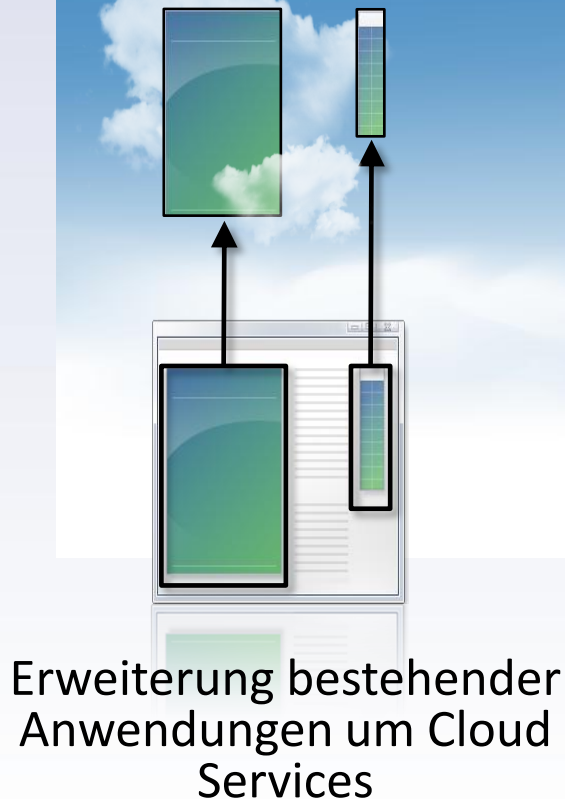
Microsoft Infrastruktur (Rechenzentren etc.)

# Microsofts S+S Plattform

## Flexibilität im IT-Portfolio



# Möglichkeiten mit Azure





# Herausforderungen der Cloud

- Eigenschaften der Cloud:
  - Scale-out  
(Einsatz massiv-redundanter Standard-Hardware)
  - Hardwarekomponenten können ausfallen  
(Ausfallsicherheit muss in die Software)
  - Hohe Verfügbarkeitsanforderungen (24 x 7)
- Auswirkungen auf:
  - Softwarearchitektur
  - Softwareaktualisierungen
  - Betrieb  
(Überwachung und Fehlersuche)





# Windows Azure

## Betriebssystem für die Cloud

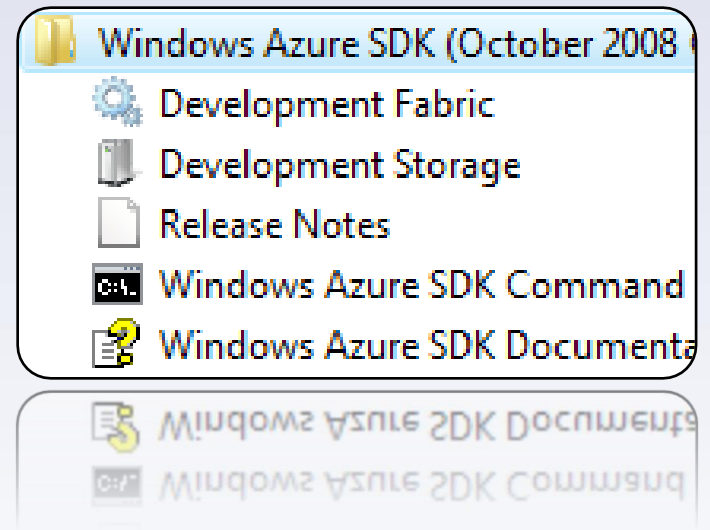


- Gleiche Funktionalitäten, die ein Desktop-OS bietet, nur auf einer **Menge vernetzter Server**:
  - Ausführungsumgebung für Anwendungen
  - Dateisystem, das sich mehrere Anwendungen teilen
  - Ressourcenzuordnung
  - Programmierumgebung
- Und darüber hinaus: **Utility Computing**
  - 24\*7 Betrieb
  - “Pay for what you use”
  - Einfache, intuitive Administration

# Betrieb in der Cloud

## Debugging und Fehlersuche

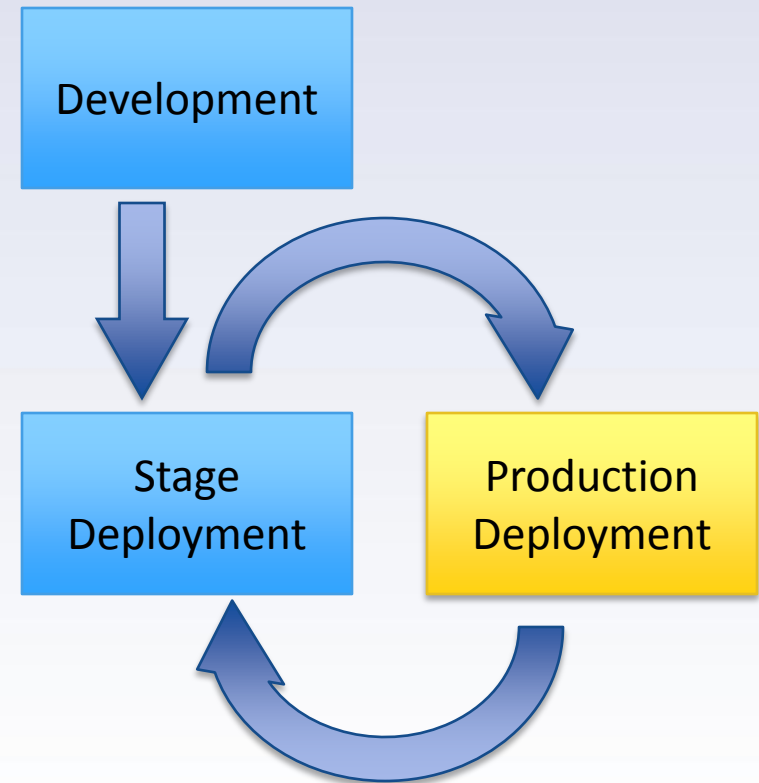
- Herausforderung:
  - Kein direkter Zugriff auf die Umgebung in der Cloud
- Lösung:
  - Logging mit Informationen zu Timestamp, Version, Umgebung, ...
  - **Azure: Windows Azure SDK mit lokaler Testumgebung („lokale Cloud“)**



# Softwareaktualisierung in der Cloud

## Aktualisierungen ohne Downtime

- Herausforderung:
  - 24 x 7 Anforderungen
- Lösung:
  - Stop + Start, bei größeren Änderungen oder wenn Downtimes akzeptabel sind
  - **Azure: Rolling Updates und Staging-Umgebung**



# Betrieb in der Cloud

## Konfigurierbares Logging und Tracing mit Alerts

- Herausforderung:
  - Kein direkter Zugriff auf die Umgebung in der Cloud
- Lösung:
  - Separierung von Konfigurationsinformationen vom Code
  - **Azure: Management Konsole mit Möglichkeit zur Konfigurationsänderung zur Laufzeit**
  - **Azure: Alerts**



# Windows Azure

# *Demo*

# Softwarearchitektur für die Cloud

## Unzuverlässige Komponenten

- Herausforderung:
  - Komponenten können ausfallen
- Lösung:
  - Lose Koppelung
  - Vermeidung von „Shutdown“-Code
  - Einsatz von Queues für asynchrone Aufrufe
  - **.NET Services**
  - **Azure: Azure Queues**



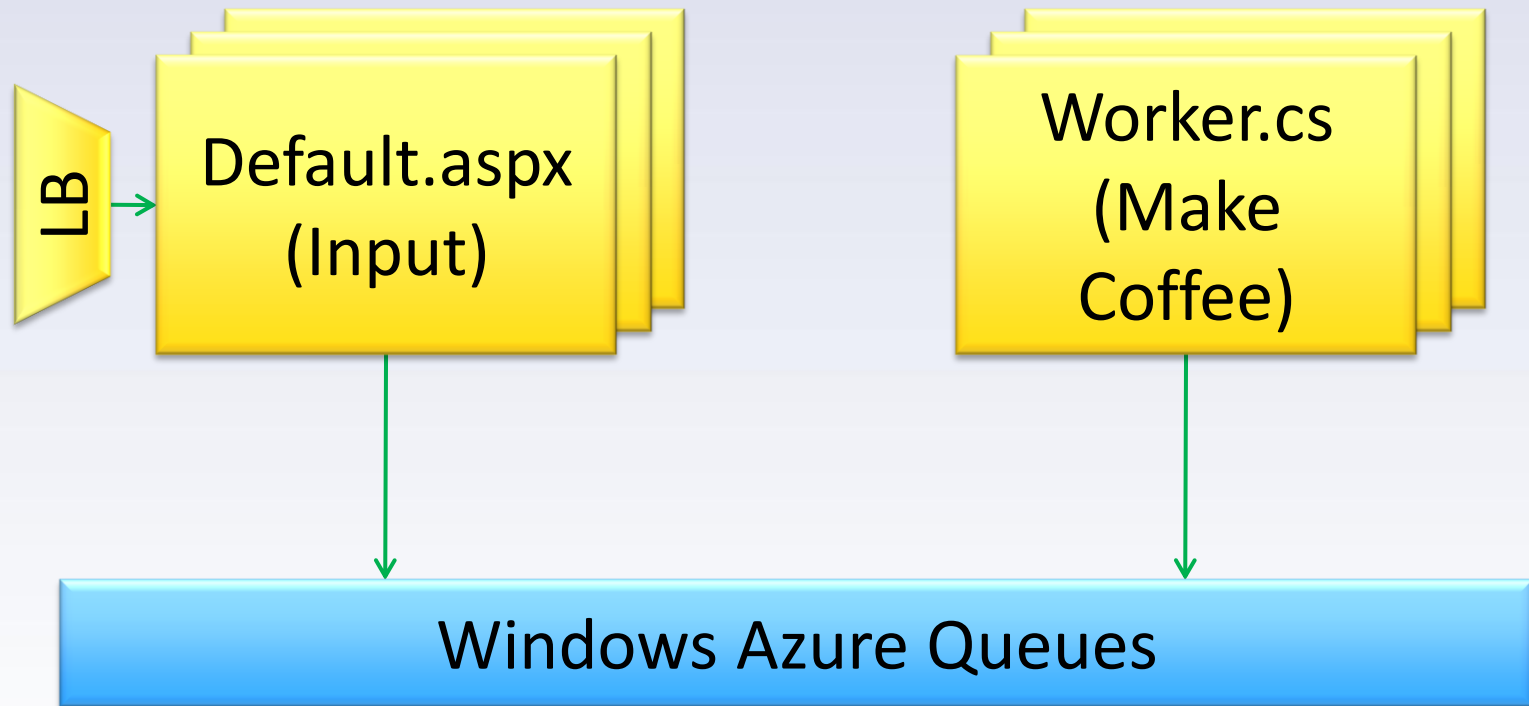
# Enge Kopplung: Default.aspx.cs

```
public partial class _Default : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        var order = txtOrder.Text;
        ProcessOrder(order);
    }

    protected void ProcessOrder(string order)
    {
        //Make some coffee!
        ...
    }
}
```



# Einsatz von Azure Queues



# Lose Kopplung: Default.aspx.cs

```
public partial class _Default : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        var order = txtOrder.Text;

        QueueStorage qStore = QueueStorage.Create(_account);
        MessageQueue orderQ = qStore.GetQueue("OrderQueue");
        orderQ.PutMessage(new Message(order));
    }
}
```

# Lose Kopplung: WorkerRole.cs

```
public class WorkerRole : RoleEntryPoint
{
    public override void Start()
    {
        QueueStorage qStore = QueueStorage.Create(_account);
        MessageQueue orderQ = qStore.GetQueue("OrderQueue");
        while (true)
        {
            Message msg = orderQ.GetMessage();
            if(msg != null)
                ProcessOrder(msg.ContentAsString());
        }
    }

    protected void ProcessOrder(string order)
    {
        //Make some coffee!
        ...
    }
}
```

# Live Services

Anwendungsdienste für die Cloud



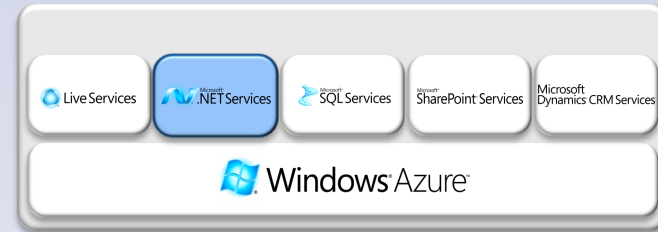
- Identity: Anwender, Geräte, Anwendungen
- Zugriff auf Verzeichnisdienste
- Kommunikation und Präsenzinformation
- Suche & ortsbezogene Dienste
- Verschiedene Geräte
- Synchronisation

# Live Mesh

# *Demo*

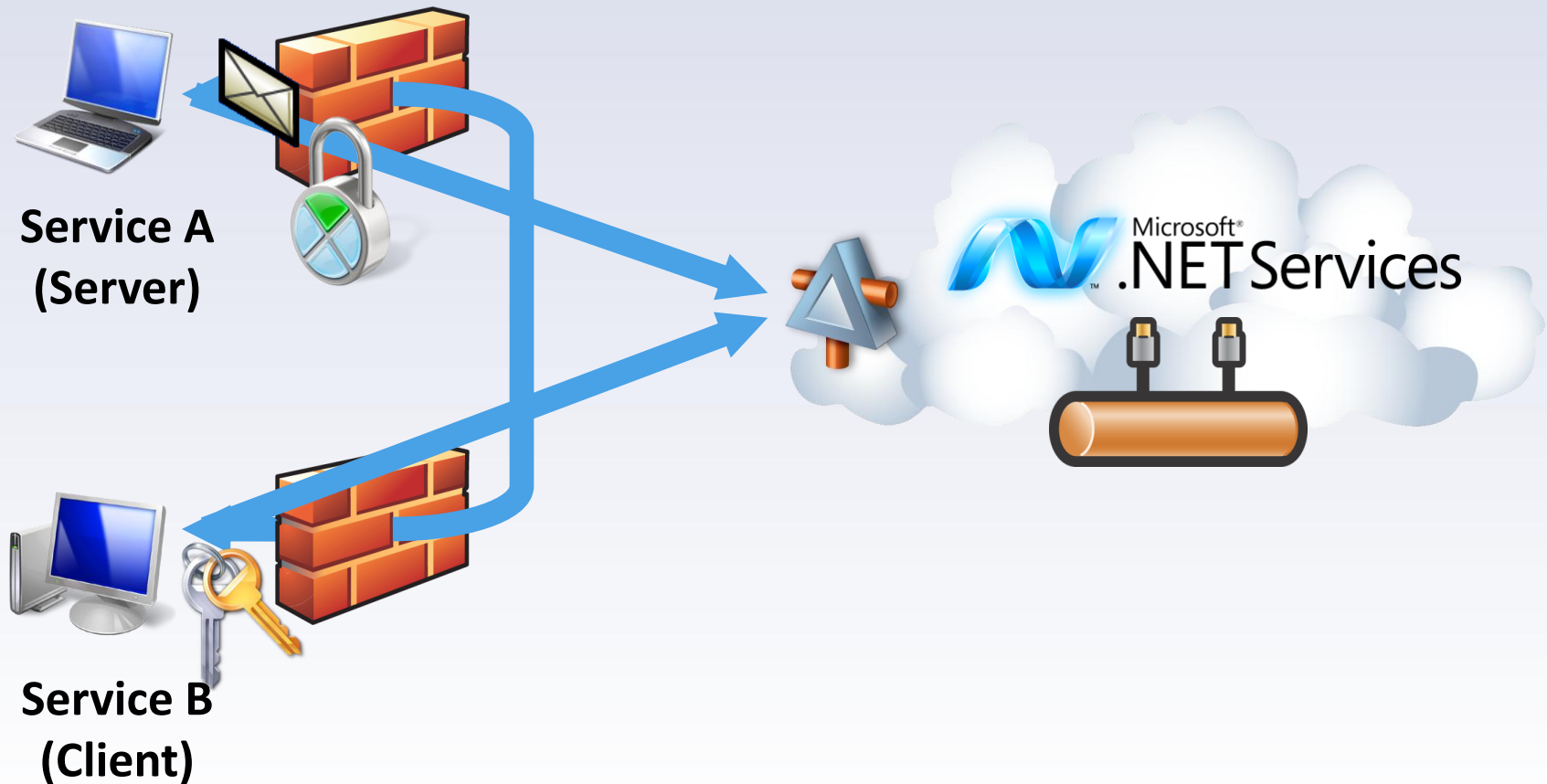
# .NET Services

## Integration für die Cloud



- Erweiterung von .NET Technologien in die Cloud
- Offene Zugriffsstandards
  - REST, SOAP, RSS, ...
  - Klassen für Java, PHP, Ruby, ...
- Fokus auf **drei zentrale Herausforderungen** bei verteilten Anwendungen:
  - Integration
  - Zugriffskontrolle
  - Orchestrierung

# Beispiel: EchoService



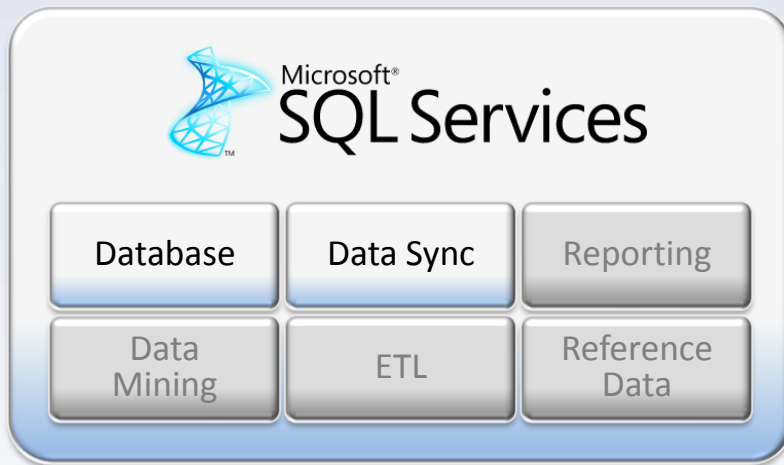


# .NET Services

# *Demo*

# SQL Services

## Daten für die Cloud



- Datenbankdienste für die Cloud
- Relationale Datenverarbeitung über strukturierte und unstrukturierte Daten
- Aktuelle Dienste:
  - Datenbank, Synchronisation
- Zukünftige Dienste:
  - Reporting, Data Mining, ETL, Referenzdaten

# Weitere Services

...in Zukunft noch vieles mehr



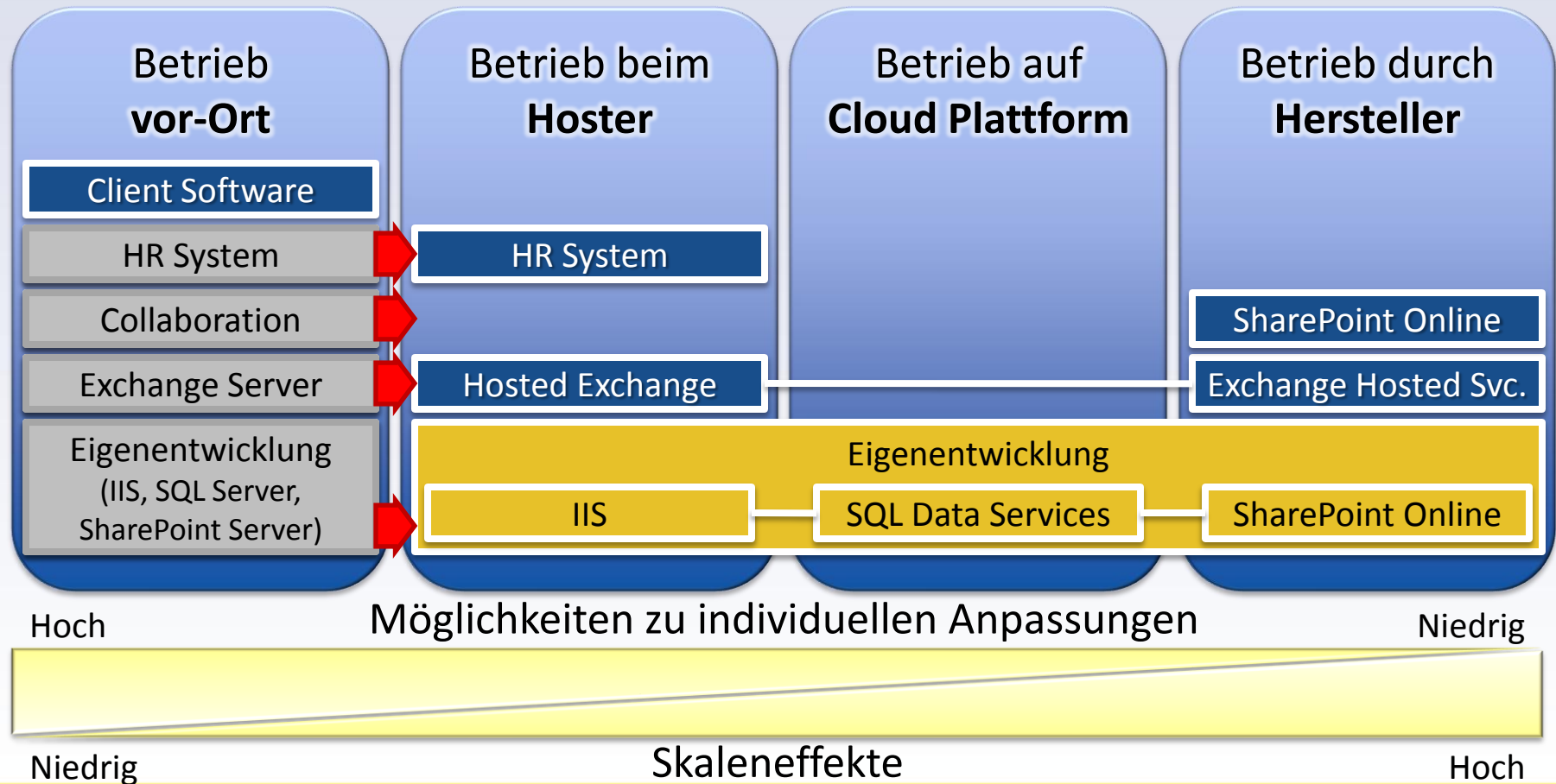
Microsoft  
SharePoint Services

Microsoft  
Dynamics CRM Services

- Funktionalität von Microsoft SharePoint® Services bzw. Microsoft Dynamics® CRM auf Basis der Azure Services Platform
- Entwicklung mit Microsoft Visual Studio
- SharePoint- bzw. CRM-Funktionalität als Bestandteile eigener Anwendungen

# Software-plus-Services

Architekturentscheidung: lokal oder als Service?



# Get Started...

- Azure Homepage  
<http://www.azure.com/>
- Azure Services Platform Developer Center  
<http://msdn.microsoft.com/en-us/azure/>
- PDC2008  
<http://www.microsoftpdc.com/>
- Mein Blog  
<http://blogs.msdn.com/hsirtl/>

# Zusammenfassung

- Software-plus-Services kombiniert die Stärken lokaler Software mit Cloud Services
- Die Microsoft-Plattform umfasst
  - eine On-premises Infrastruktur- und Anwendungsplattform
  - Windows Azure als Betriebssystem für die Cloud
  - Cloud Services, die in eigene Anwendungen integriert werden können
  - Wahlfreiheit des Deploymentmodells auf der Ebene einzelner Anwendungskomponenten

# Vertiefende Vorträge

- 20.11. 14:00 – 15:00Uhr  
**„Cloud-Computing mit Windows Azure: Verfügbare und wirklich skalierbare Web-Anwendungen und Services entwickeln und ganz einfach betreiben“**  
Tim Fischer
- 20.11. 15:30 – 16:30Uhr  
**“Software Entwicklung mit der Windows Azure Services Plattform”,**  
Dariusz Parys
- 21.11. 08:30 – 09:30Uhr  
**“Oslo – Modell-Driven Development”,**  
Dariusz Parys



# Ihr Potenzial. Unser Antrieb.