



# 10 Things You Can Do to Use Windows Azure More Effectively

Rainer Stropek

software architects gmbh

[rainer@software-architects.at](mailto:rainer@software-architects.at)

# Abstract

You are already using Windows Azure or you are planning to take the step into the cloud? In this session Rainer Stropek, MVP for Windows Azure, presents some of the most important tips to get the most out of your investment in Microsoft's cloud computing platform. Did you know that you can host multiple websites in a single Azure web role? Rainer will cover things like this during his session. You will learn about the theory behind them and see them presented in practical examples.

# Introduction

- [software architects gmbh](http://www.software-architects.gmbh)
- **Rainer Stropek**  
Developer, Speaker, Trainer  
MVP for Windows Azure  
[rainer@timecockpit.com](mailto:rainer@timecockpit.com)



@rstropek



<http://www.timecockpit.com>

<http://www.software-architects.com>



IT&  
Dev CONNECTIONS powered by Microsoft®





Common Creative License  
Source: <http://www.flickr.com/photos/romainguy/230416692>

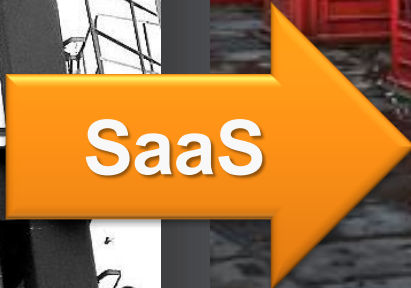


## Strategy

„How a battle is fought is a matter of tactics: the terms and conditions that it is fought on and whether it should be fought at all is a matter of strategy” (Source: [Wikipedia](#))



# 1. Change Your Business Model?!



Common Creative License  
Source: <http://www.flickr.com/photos/28481088@N00/621194808/>

Common Creative License  
Source: <http://www.flickr.com/photos/monster/310530160/>

IT & Dev CONNECTIONS  
powered by Microsoft®

# 1. Change Your Business Model?!

## Classical ISV

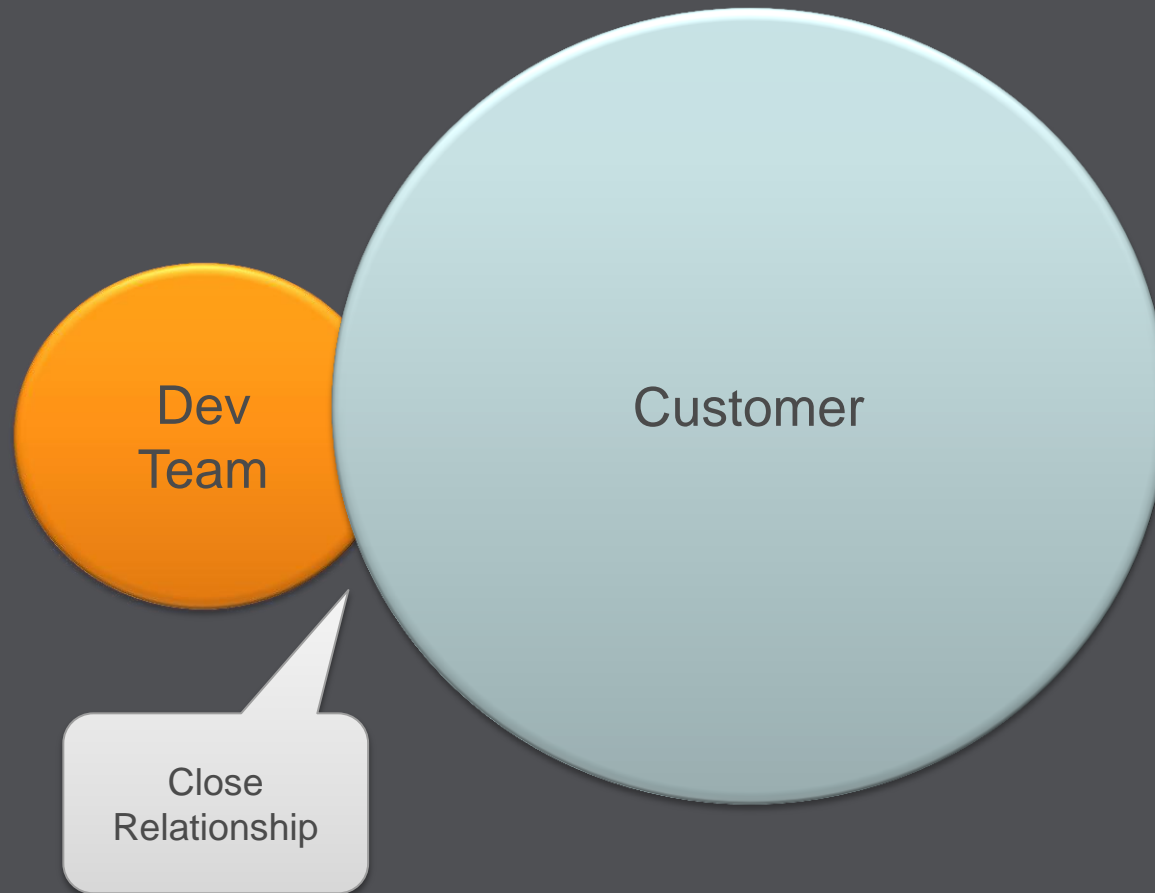
- Complexity is visible
  - Sales process
  - Project planning
  - Deployment
- Revenue
  - Per project
- Costs
  - Development, maintenance
  - Necessary infrastructure

## SaaS + Cloud ISV

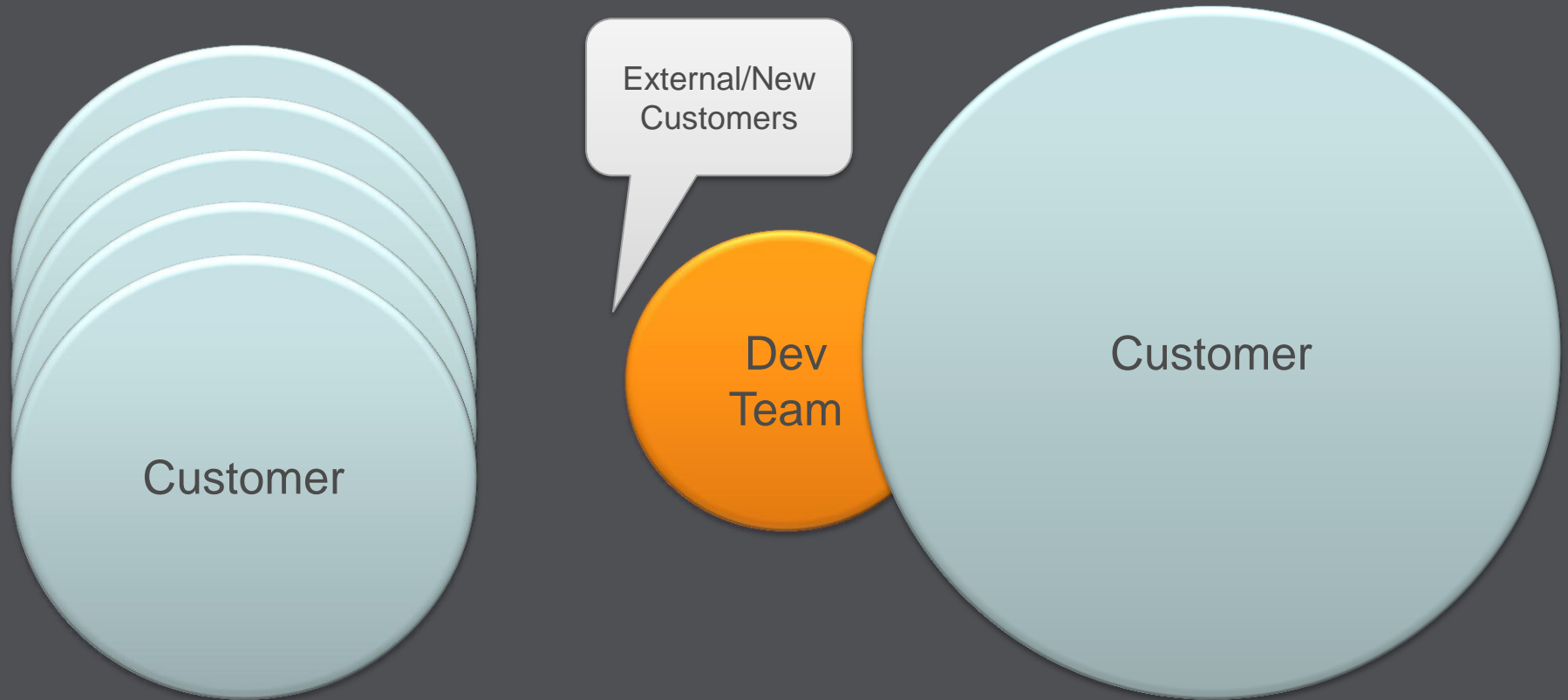
- Hidden Complexity
  - Simple or self-service sales
  - No classical „project“
  - Minimal or no deployment
- Revenue
  - ARPU, RGU, etc.
- Costs
  - Depreciation, capital costs
  - Total cost



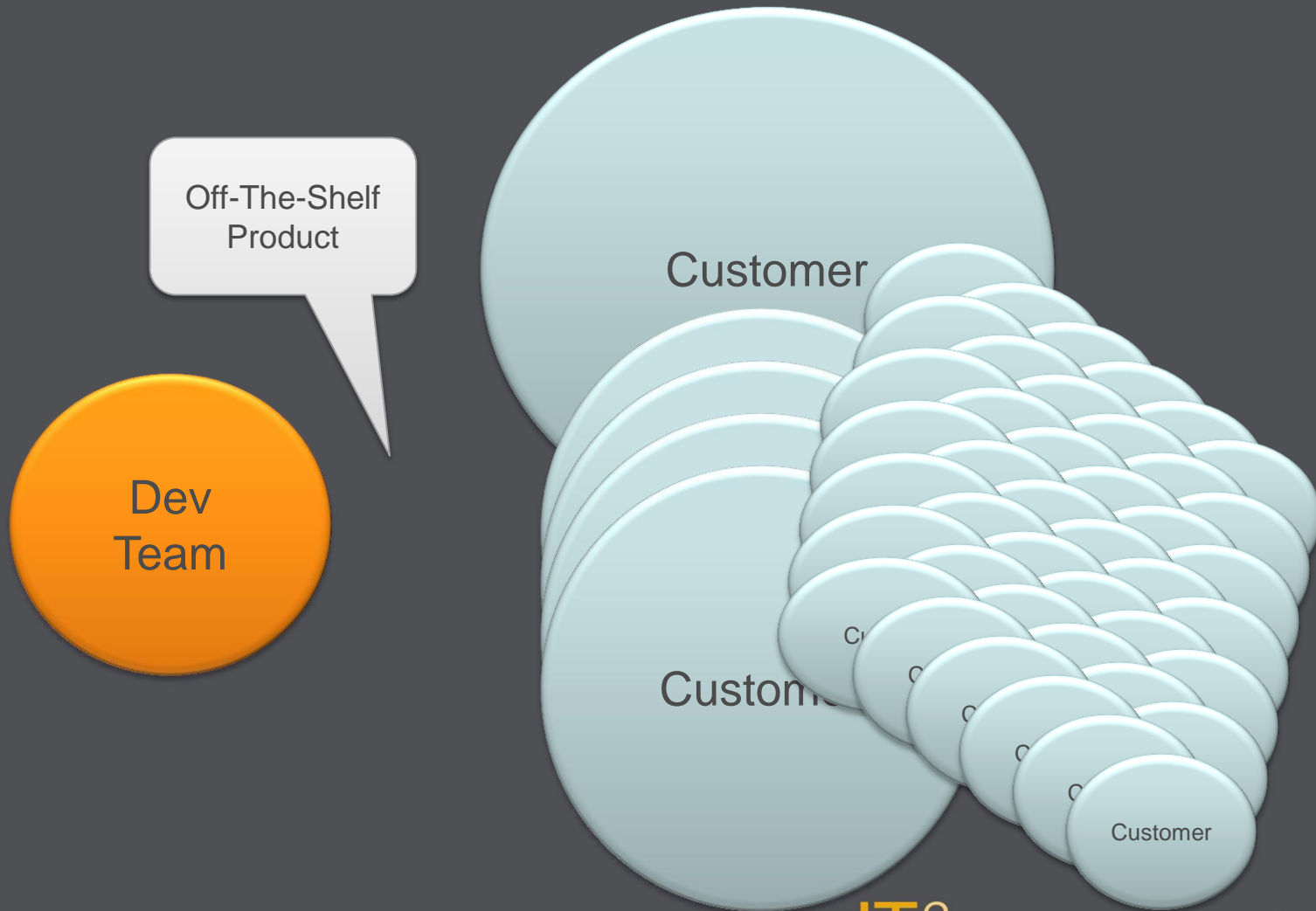
# Where Most Of Us Come From



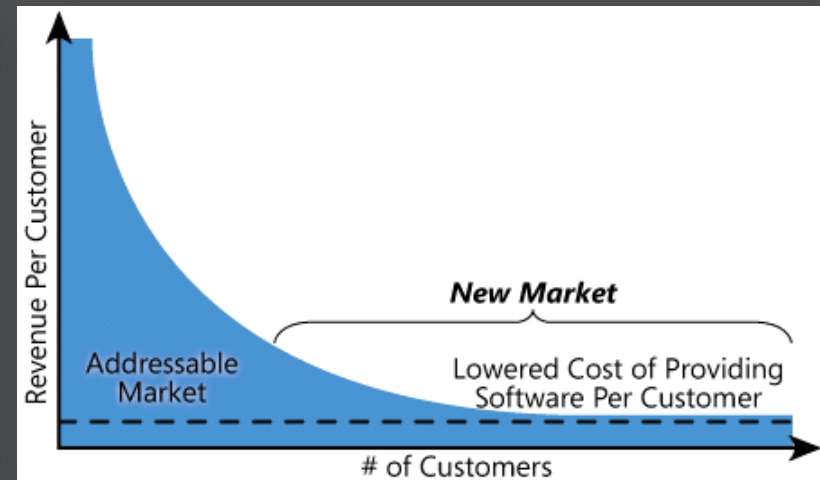
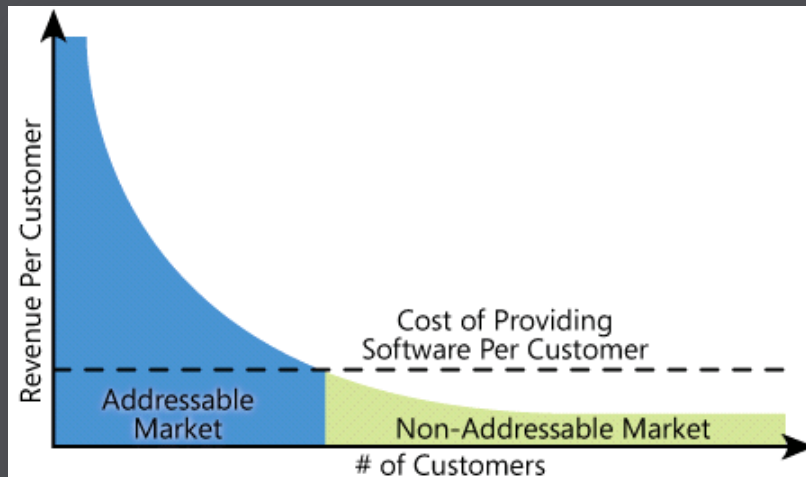
# Where Many Of Us Are Today



# Where We Want To Be



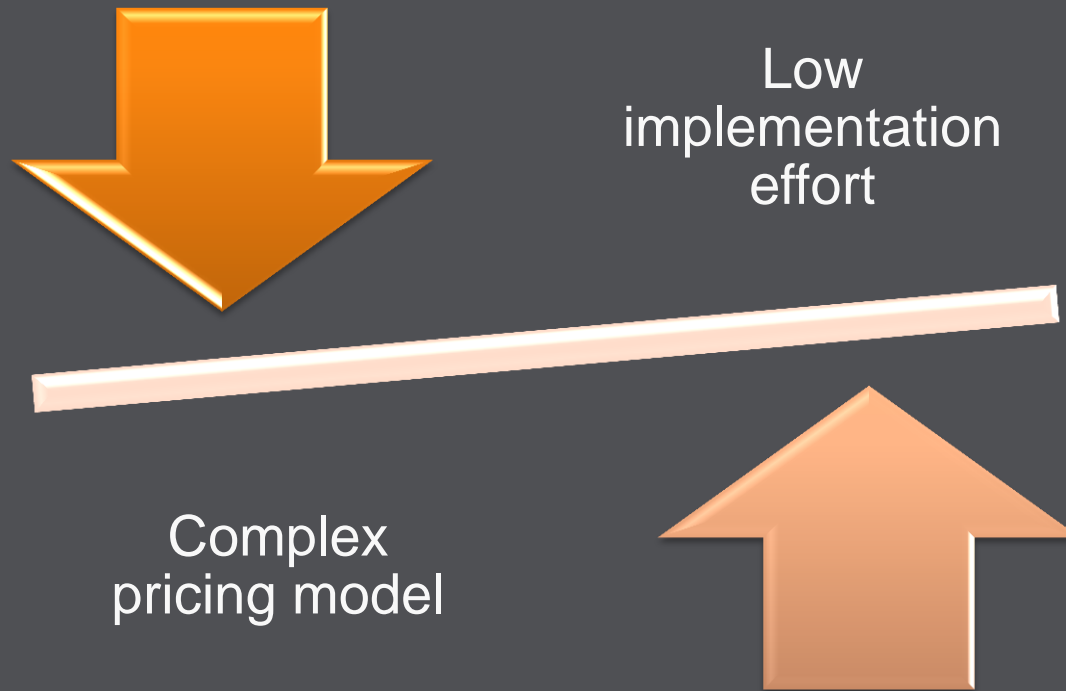
# Catching The Long Tail



The Long Tail

<http://www.wired.com/wired/archive/12.10/tail.html>





# Rating/Billing/Payment

- **Web portal for...**
  - ...registration of new users (has to handle very sensitive credit card data → has to be stored safely and securely, too!)
  - ...maintaining customer billing data (e.g. customer changes address or credit card provider)
  - ...cancelling accounts or users
- **Infrastructure for rating**
  - Measure usage
  - Rate usage with rates/unit
  - Additional rating logic will be necessary; examples:
    - Mechanisms for sales discounts (e.g. introductory offer)
- **Infrastructure for billing**
  - Creation of invoices (in total for accounting and itemized bills for checking the correctness)
  - Sending out invoices (electronically with/without signature, printed)
  - Storing invoices internally.
- **Infrastructure for payment; options could be:**
  - Credit card payments including mechanisms for handling payment cancellations, invalid credit cards (e.g. expired cards), etc.
  - Payments via online payment solution like PayPal
  - Payment with vouchers (e.g. voucher that an end user received with hardware)
- Account/balance infrastructure (debit system instead of credit system; example: Skype)
  - E.g. what to do with remaining balance if user cancels his account?
- **Fraud detection**
  - Infrastructure for credit checking and fraud detection; examples are...
    - ...credit card checking
    - ...check of credit ranking for users
- **Support infrastructure for rating/billing/payment issues**
- **Alternatives**
- **Alternatives for building such a system from scratch could be:**
  - Buying an existing billing system (expensive; does only make sense if customer needs such a system in other areas, too)
  - Handle payments with payment partners (e.g. payment via phone invoice → difficult because such contracts have to be maintained country by country)
  - Use existing SaaS payment solutions (e.g. PayPal; difficult with pay-per-use)

Common Creative License  
Source: <http://www.flickr.com/photos/rpenalozan/5057881708/>

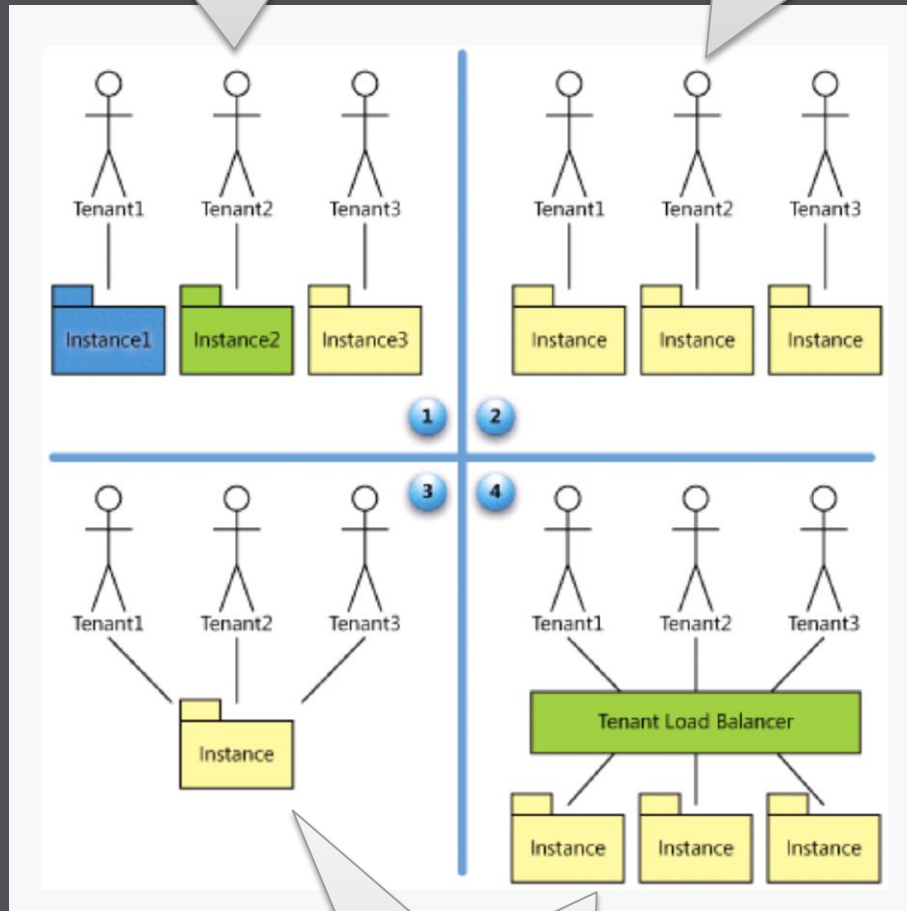


## 2. Multi-Tenancy

To be or not to be...

Custom Solution With Common Components

Multi-Instance (ASP)









Multi-Tenancy

## SaaS Maturity Levels

Kwok, Nguyen, Lam: A Software as a Service with Multi-tenancy Support for an Electronic Contract Management Application, IEEE International Conference on e-Business Engineering, pp. 179-186, 2008



Level of Competency	Description	Approach	Variance Level Supported
Entry	Highly standardized offering without any configuration and customization support	Well design the functionalities as standardized offering to cover targeted customers	None 
Aware	Relatively standardized offering with pre-defined variance points	Offer parameterized configuration	Low 
Capable	Relatively standardized offering with user defined configuration	Offer self serve configuration tool to empower customers	Medium 
Mature	Base offering with programable enviroment to enable user preferred customization	Offer scripting based programming for very flexible customization	High 
World Class	Offer a platform supported by programming model and tools to enable extremely strong customization or even new application development	Offer well defined programming model and tools to enable extensive customization and new application development	Extremely High 



Completely Standardized Offering

Fully Tenantized Offering

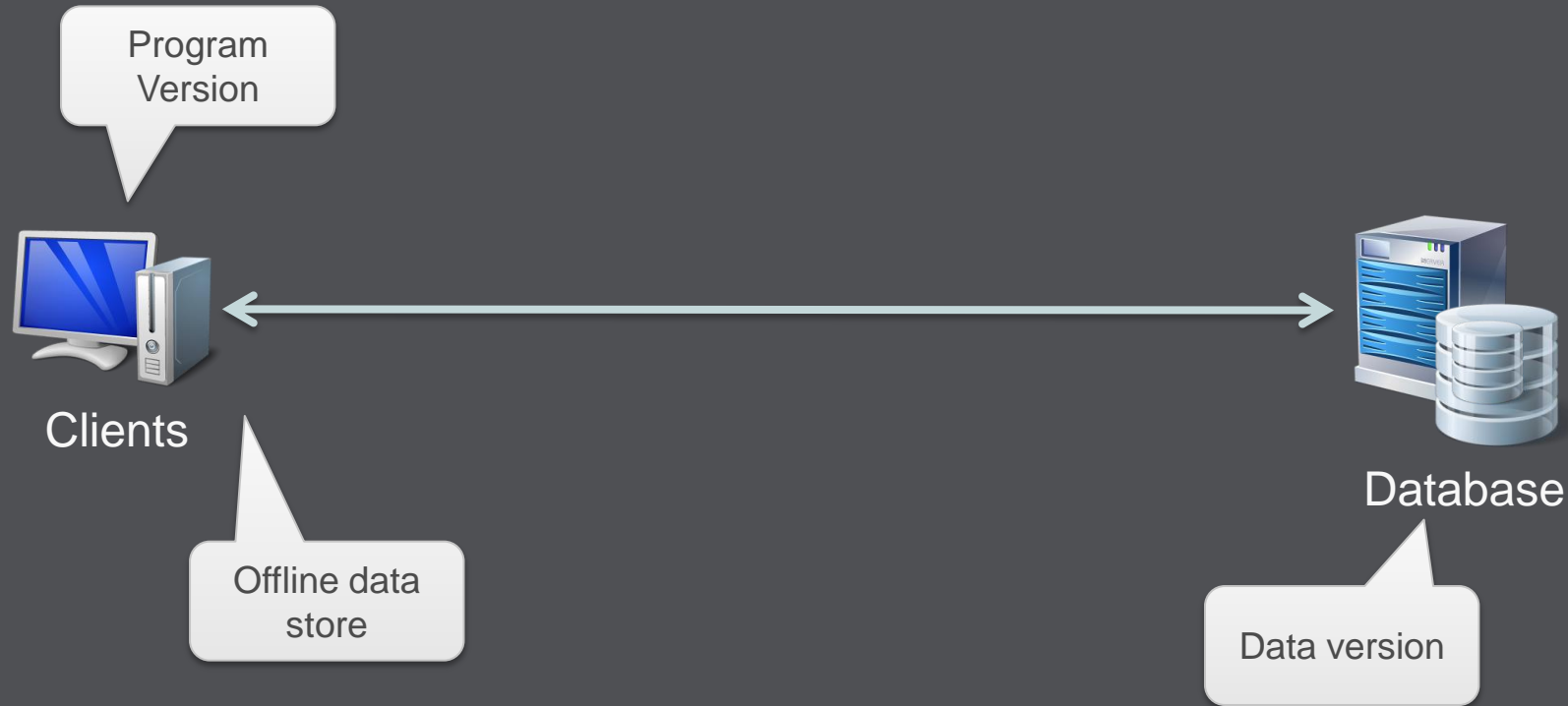
## SaaS Configuration Competency Model

Wei Sun, Xin Zhang, Chang Jie Guo, Pei Sun, Hui Su: Software as a Service: Configuration and Customization Perspectives, in Proceedings of IEEE Congress on Services Part II, 2008

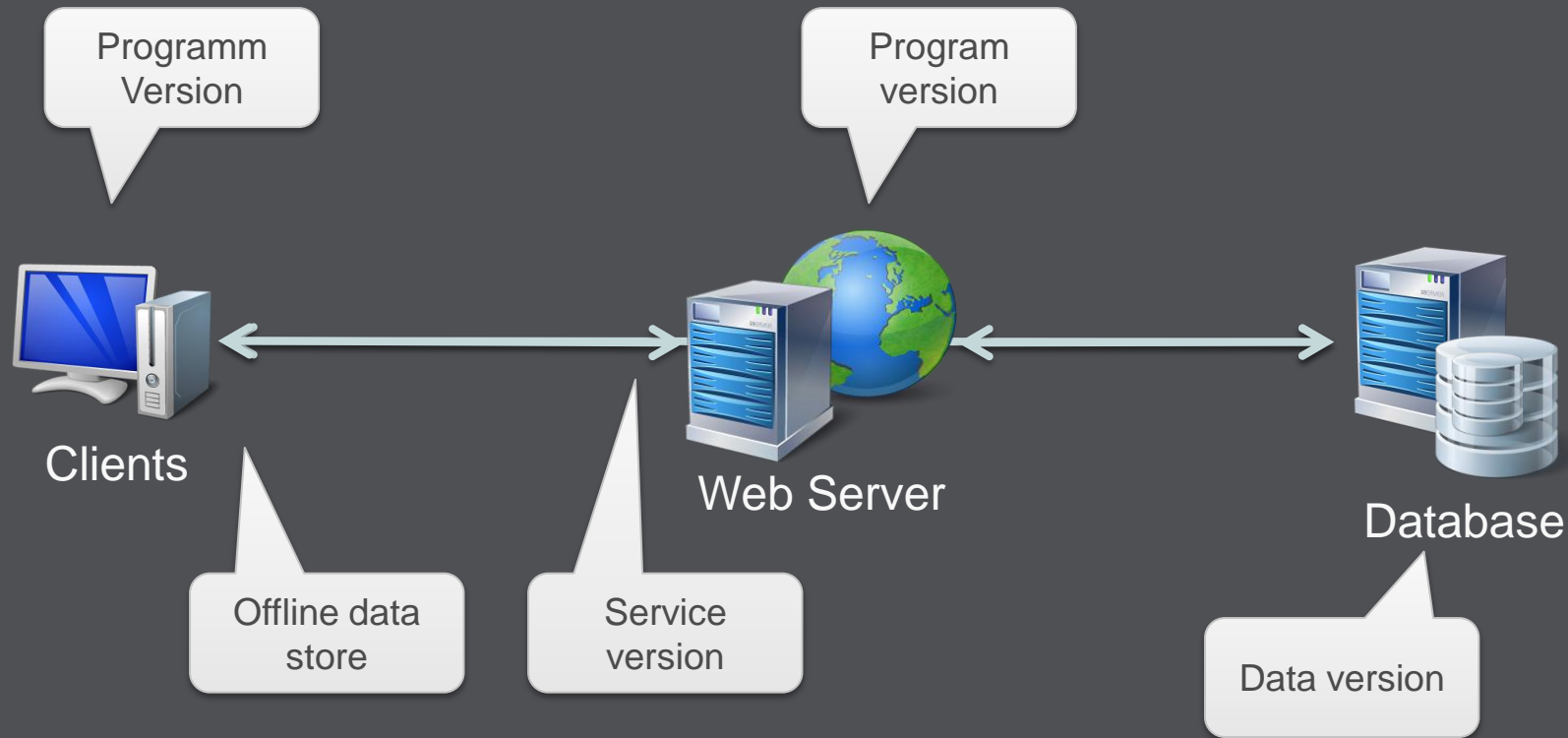
# Multi-Tenency

- How to handle tenant separation?
  - Check out upcoming [federation feature in SQL Azure](#)
- How to handle per-tenant customization?
  - Data Model, behavior, forms, lists, workflows, reports, permissions, etc.
  - Sandboxing (extensibility)
- Programming model
  - Database access
  - API

# Good Old Client-Server Model

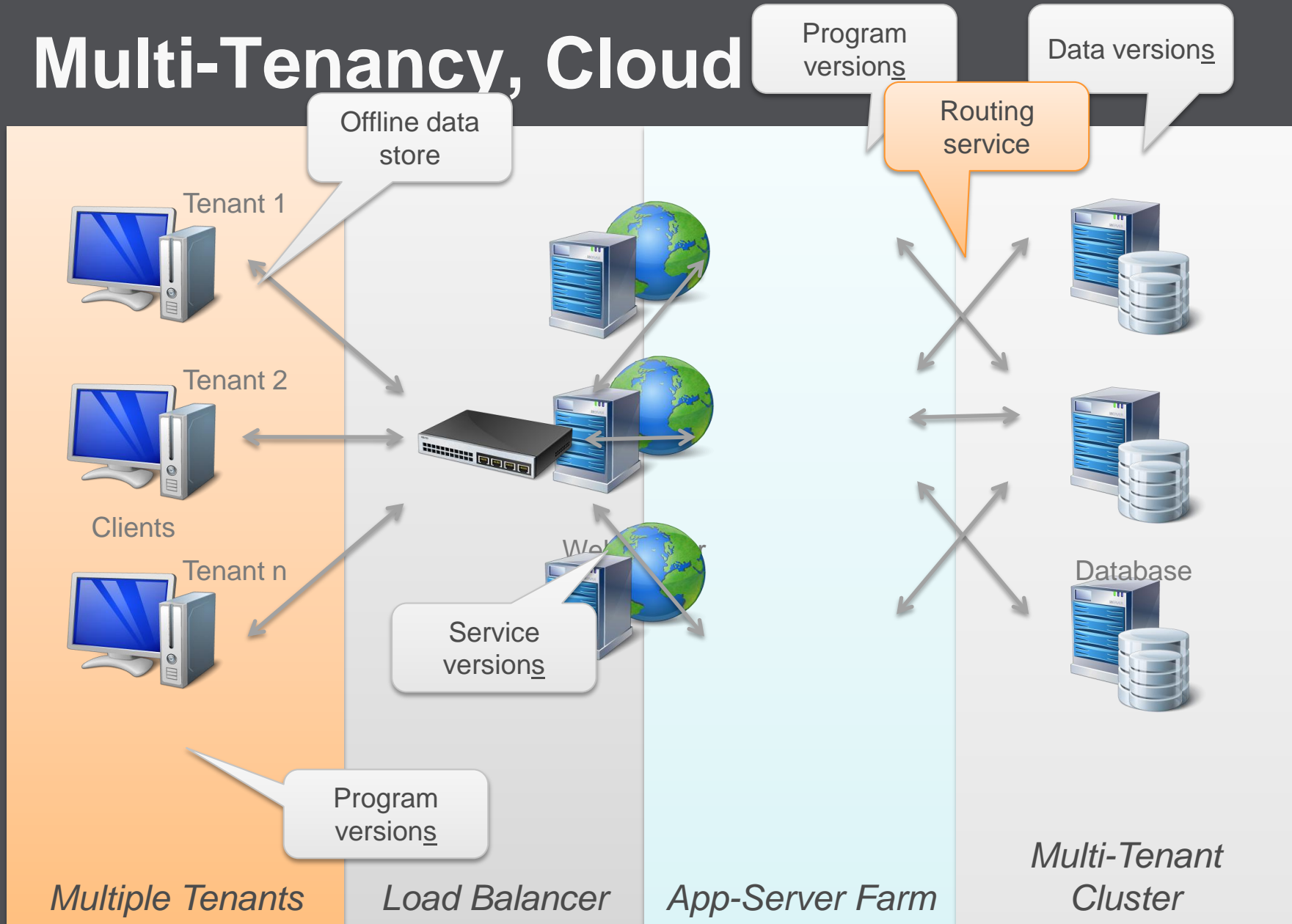


# Three-Tier Modell





# Multi-Tenancy, Cloud





Common Creative License  
Source: <http://www.flickr.com/photos/rpenalozan/5057881708/>

## 3. Make or Buy

Gotcha!

# Make or Buy

- Same general issues as always
- PLUS
  - Use service or deploy with your app?
    - Service → consider SLAs of 3<sup>rd</sup> parties
    - Multi-Tenancy down to 3<sup>rd</sup> party services
    - Authentication, delegation
  - Is 3<sup>rd</sup> party cloud-ready?
    - Don't build your business based on announcements
  - Licensing
    - What is a „server“?
    - General: Prefer e.g. per-developer licensing
  - Caution: Not only SW components, same issues with data

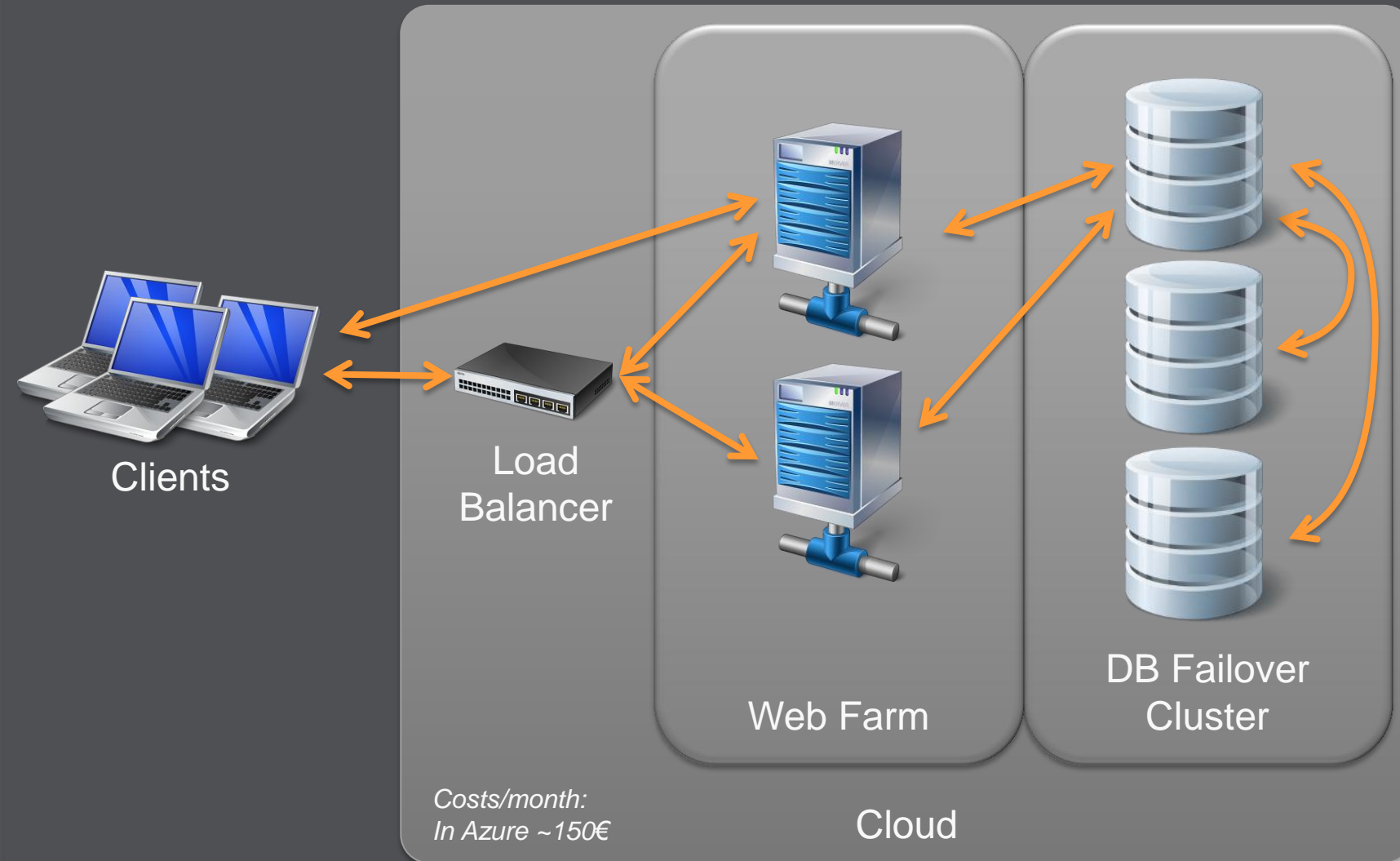


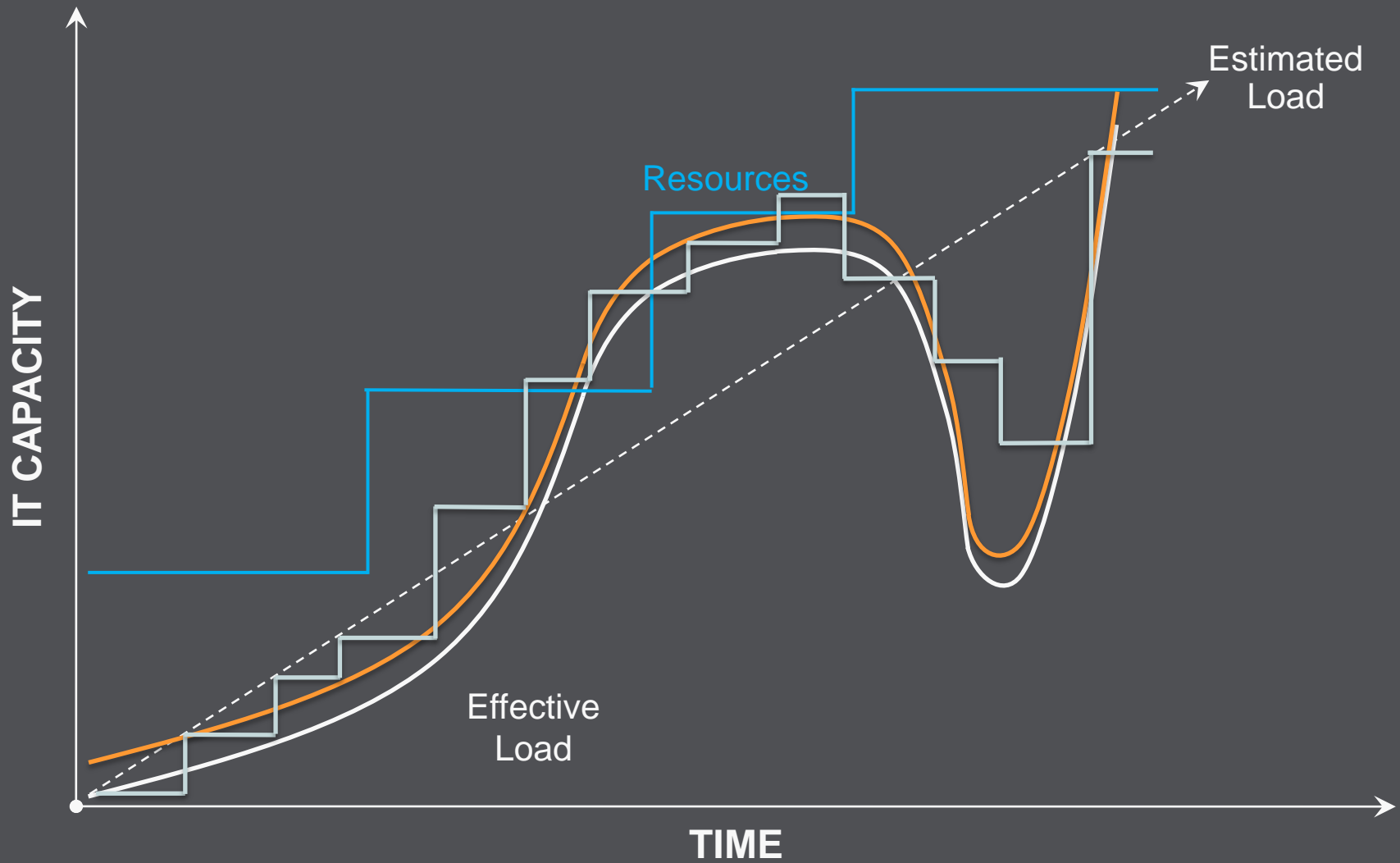
## Tactics

„the art of organizing an army, are the techniques for using weapons or military units in combination for engaging and defeating an enemy in battle” (Source: Carl Clausewitz: On War, 1832)



# 4. Scale Out Not Up

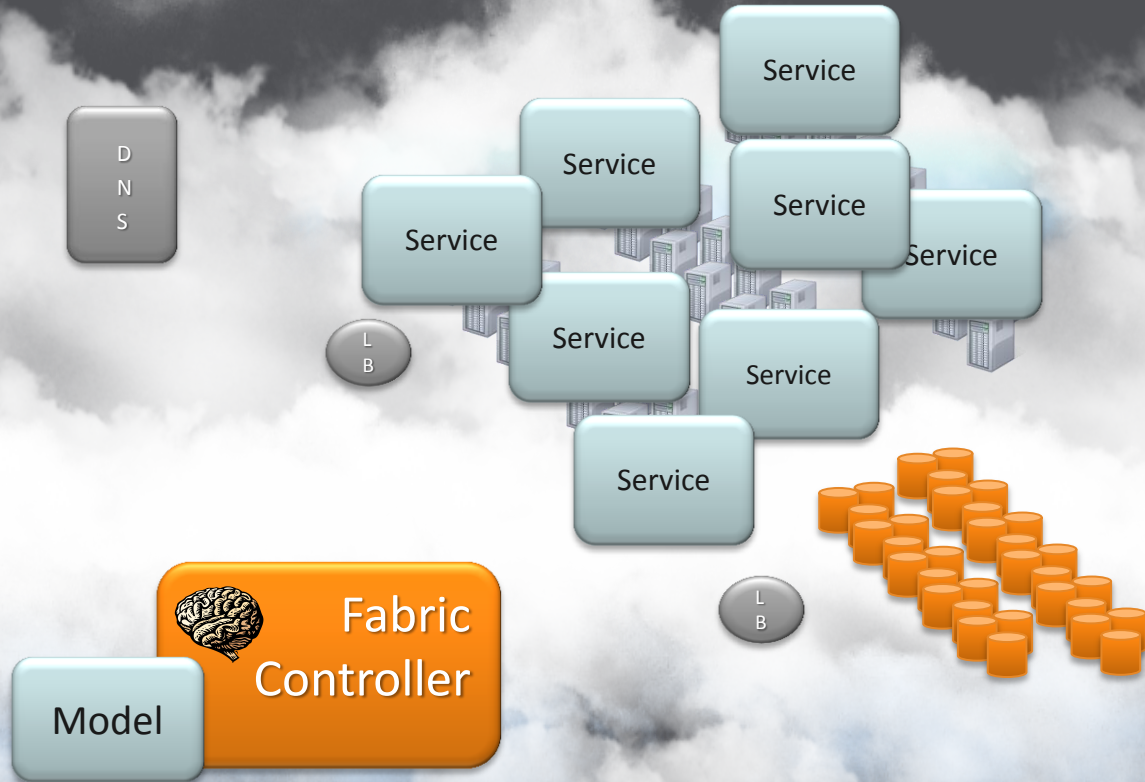




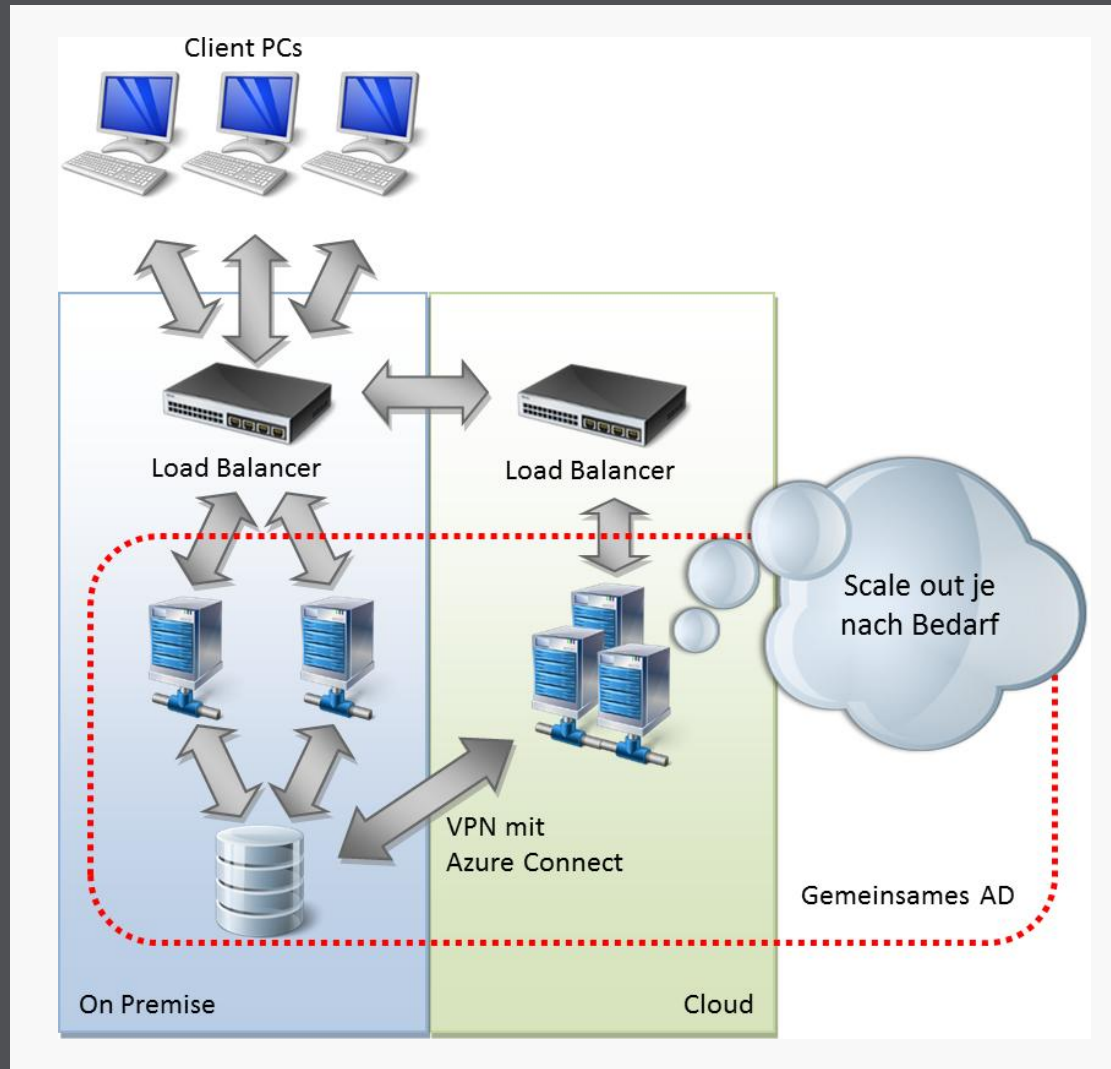
# Azure Scaling

Your Service

Web Portal (API)



# Scale Into The Cloud



powered by Microsoft®

Dev CONNECTIONS

# Azure Scaling

- Scaling by schedule
  - [Azure PowerShell Cmdlets](#)
    - [Example](#) on StackOverflow.com
  - See also [Windows Azure Dynamic Scaling Sample](#)
- Auto-scaling
  - Based on collected metrics (see also later)
  - Pre-built scaling services (e.g. [AzureWatch](#))



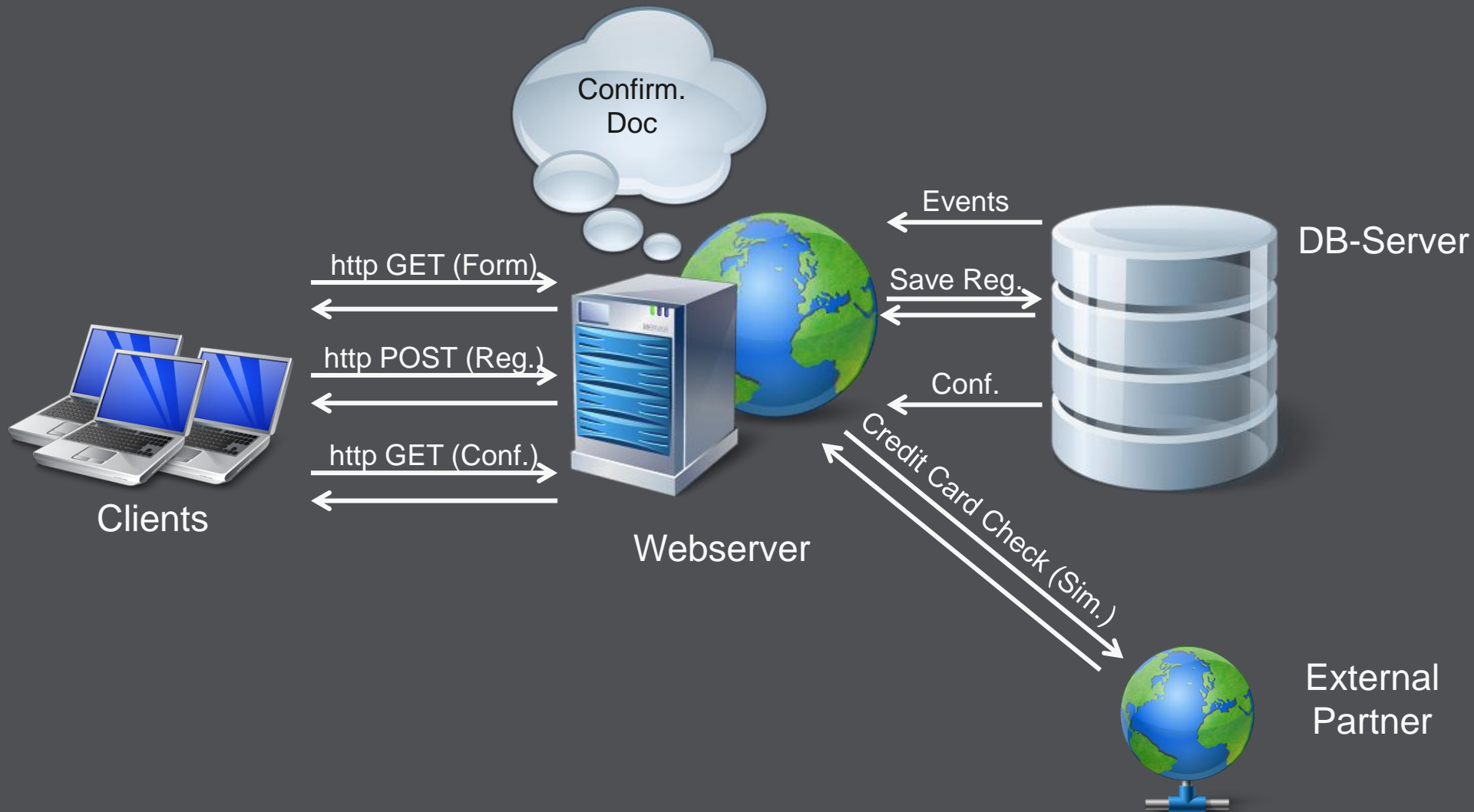
# Azure Scaling

- Some services provide auto-scaling out of the box
  - E.g. Table storage, blob storage
- Auto-scaling of RDBMS-layer is still hard
  - Sharding (see also [Sharding Wiki](#) on TechNet)
  - Check out upcoming [federation feature in SQL Azure](#)



Common Creative License  
Source: <http://www.flickr.com/photos/steffenz/3893797302/sizes/l/in/photostream/>

## 5. Async Rulez!



```
// Initialize runtime statistics
var statistic = new RegistrationStatistic();
```

Build registration object from http parameters

```
// Remember the time when processing has started
statistic.RegistrationProcessingStartTime = DateTime.Now;
```

```
#region Processing
```

```
// Check credit card with external service
```

```
if (BusinessLogic.TryCheckCreditCard(registration))
```

```
{
```

```
    // Generate confirmation document
```

```
    registration.ConfirmationDocument = BusinessLogic.GenerateConfirmationDocument(registration);
```

```
}
```

```
else
```

```
{
```

```
    registration.Status = "InvalidCreditCard";
```

```
}
```

```
// Write registration to database
```

```
BusinessLogic.AddRegistrationToDatabase(registration);
```

```
#endregion
```

```
// Remember the time when processing has finished and write statistic data to table
```

```
statistic.RegistrationProcessingFinishedTime = DateTime.Now;
```

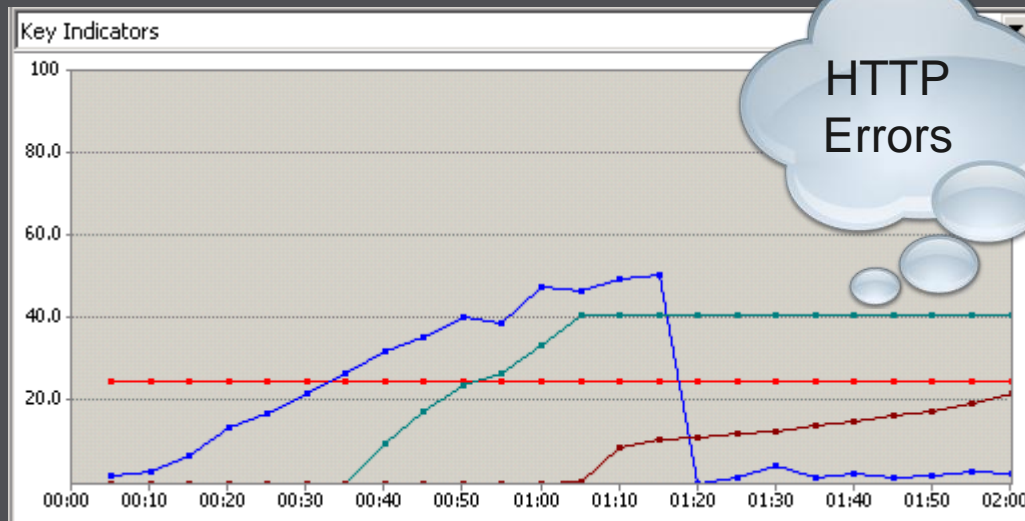
```
var cloudStorage = new CloudStorageConnection();
```

```
cloudStorage.Open();
```

```
RegistrationStatisticManagement.UpdateStatisticInTable(cloudStorage, registration.RegistrationId, statistic);
```

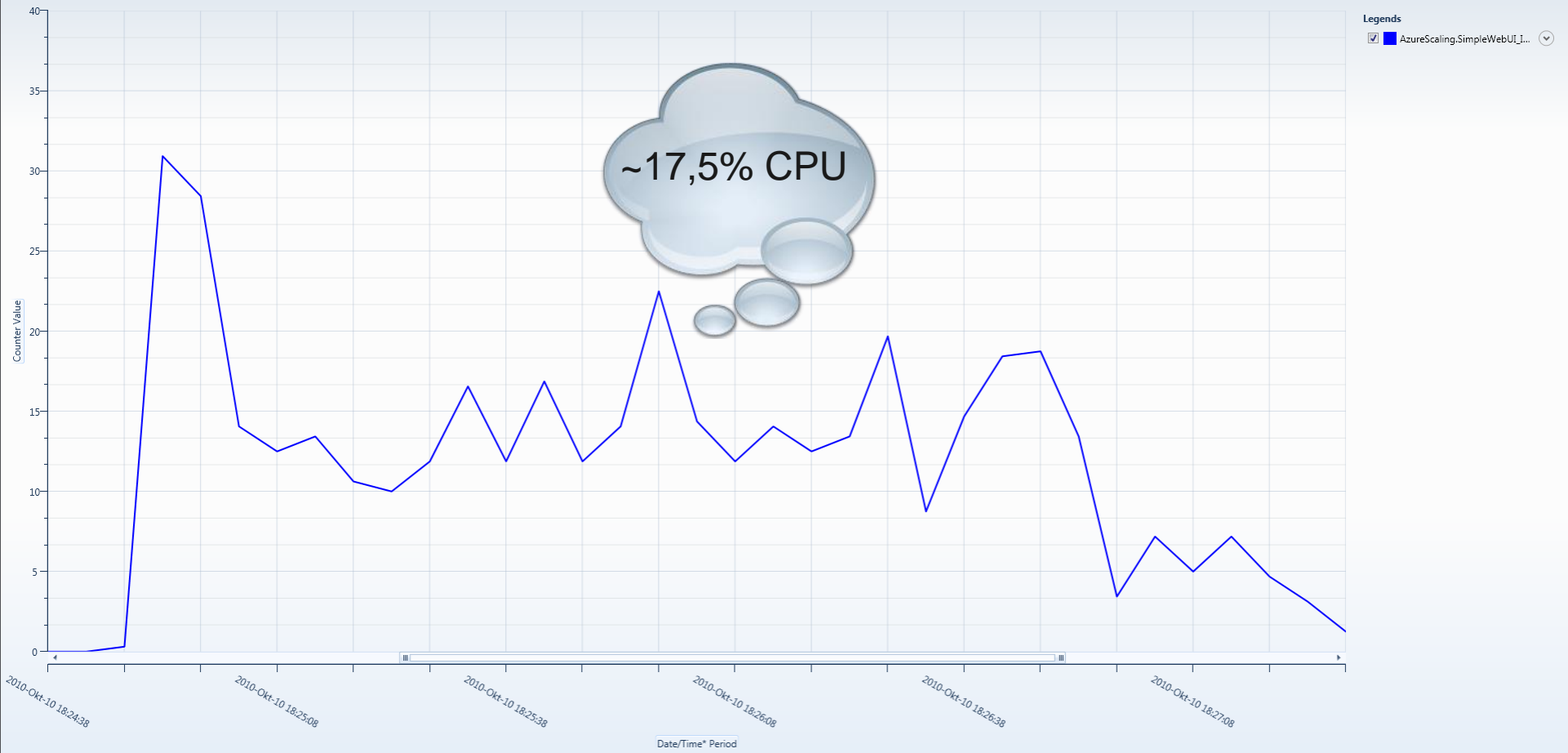
Display registration status



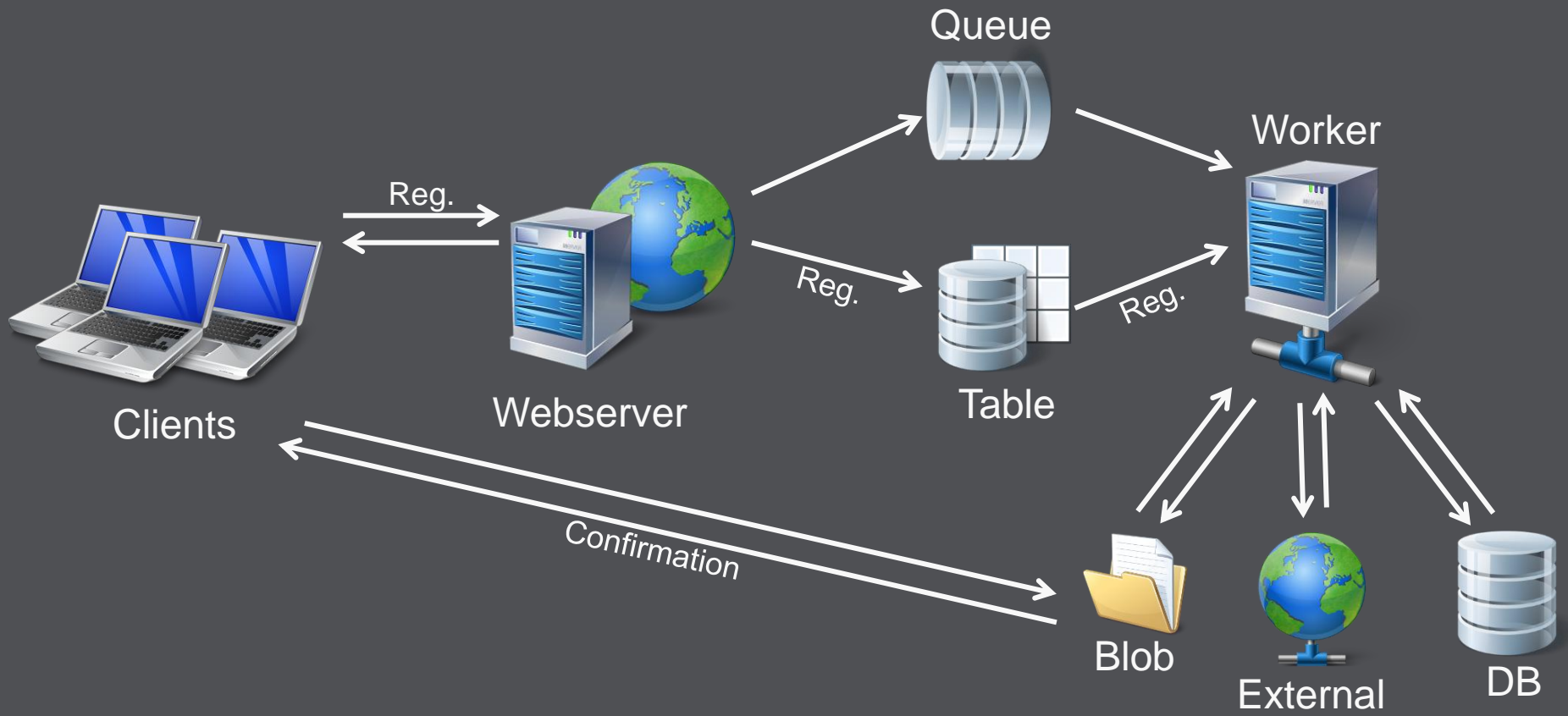


Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
<b>Key Indicators</b>								
<input checked="" type="checkbox"/> User Load	_Total	LoadTest:Scenario	IP-0AE33654		1,000	250	250	250
<input checked="" type="checkbox"/> Avg. Page Time	_Total	LoadTest:Page	IP-0AE33654		100	0.41	51.1	14.4
<input checked="" type="checkbox"/> Http Errors	_Total	LoadTest:Errors	IP-0AE33654		100	0	41	25
<input checked="" type="checkbox"/> Exceptions	_Total	LoadTest:Errors	IP-0AE33654		1,000	0	222	68

Performance counter chart for \Processor(\_Total)\% Processor Time







```
ClientScript.GetPostBackEventReference(this, string.Empty);
```

```
if (!Page.IsPostBack)
```

```
{
```

```
    Guid registrationId = Guid.NewGuid();
```

```
    Save time when request has arrived
```

```
    Build registration object from http parameters
```

```
    #region Add registration in processing queue
```

```
    // Add registration request to queue
```

```
    cloudStorage.RegistrationQueue.AddMessage(new CloudQueueMessage(registration.RegistrationId.ToString()));
```

```
    // Add registration payload to table
```

```
    cloudStorage.RegistrationPayloadContext.AddObject(CloudStorageConnection.RegistrationTableName, registration);
```

```
    cloudStorage.RegistrationPayloadContext.SaveChanges();
```

```
    #endregion
```

```
    this.RegistrationId.Value = registrationId.ToString();
```

```
    this.WelcomeMessage.Text = "Registration is currently processed, will take a while...";
```

```
    RegisterRefreshScript();
```


```
}
```

```
else
```

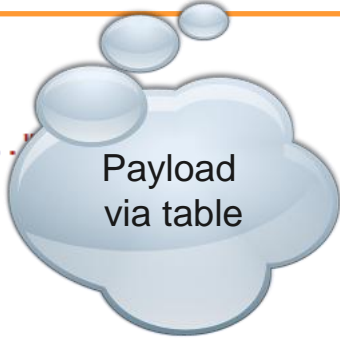
```
{
```

```
    Display registration status
```

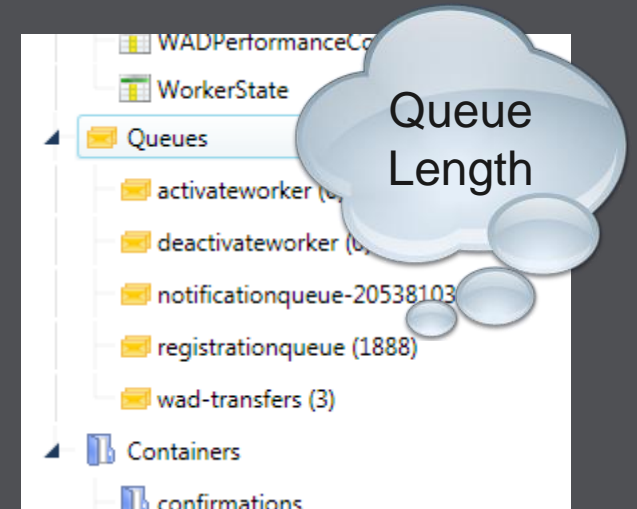
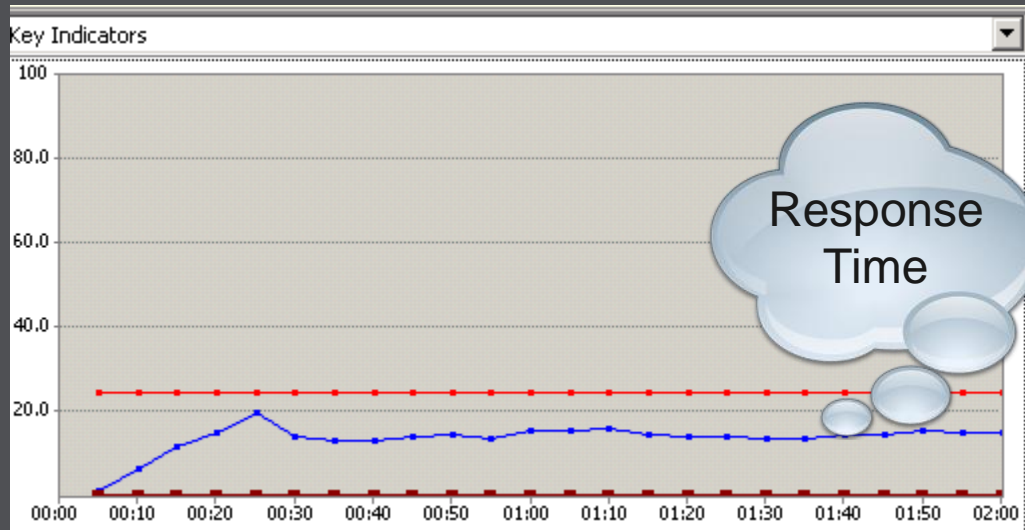
```
}
```



Notify  
worker via  
queue

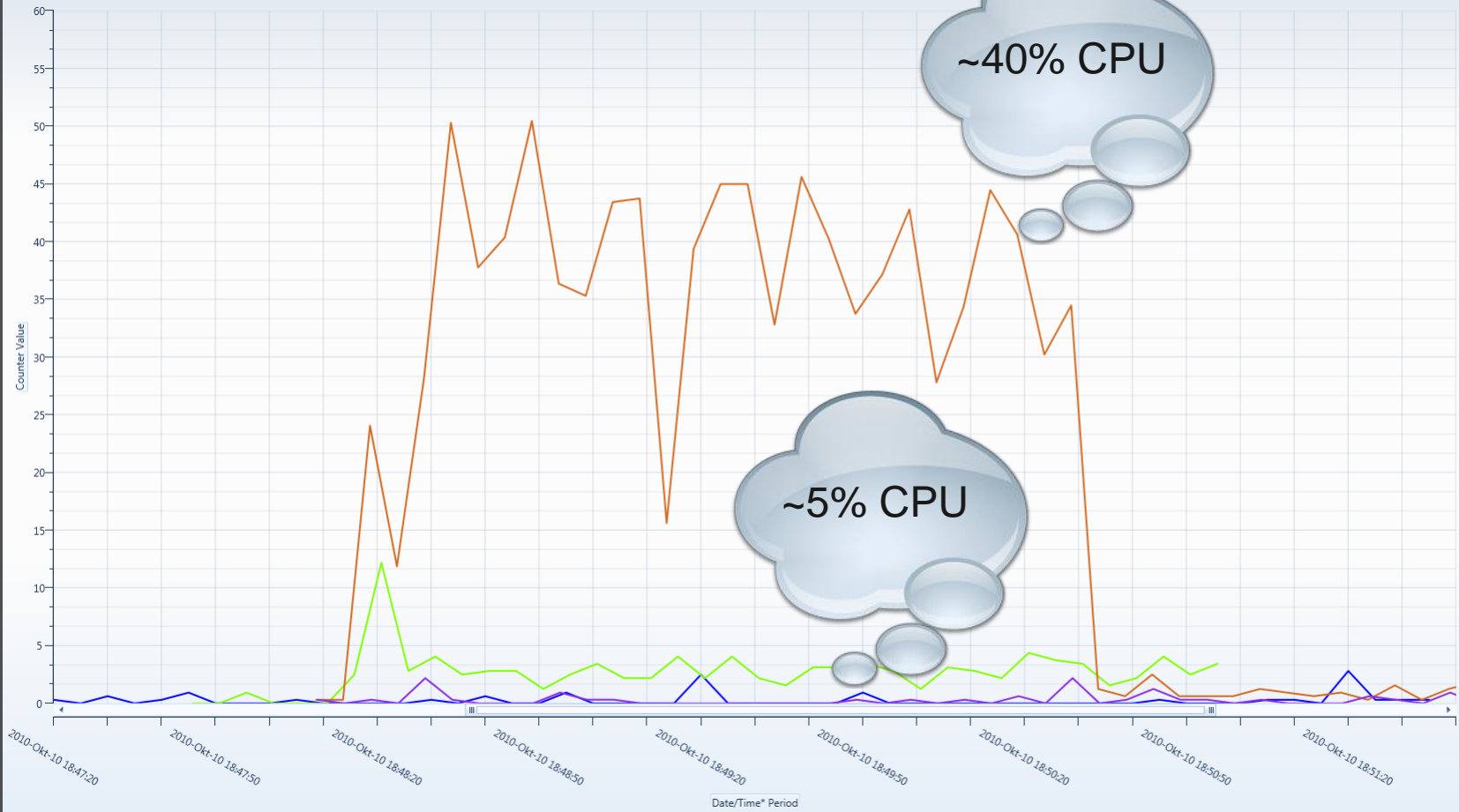


Payload  
via table



Counter	Instance	Category	Computer	Color	Range	Min	Max	Avg
<b>Key Indicators</b>								
<input checked="" type="checkbox"/> User Load	_Total	LoadTest:Scenario	IP-0AE33654		1,000	250	250	250
<input checked="" type="checkbox"/> Avg. Page Time	_Total	LoadTest:Page	IP-0AE33654		100	1.86	20.3	14.9
<input checked="" type="checkbox"/> Exceptions	_Total	LoadTest:Errors	IP-0AE33654		0	0	0	0
<input checked="" type="checkbox"/> Http Errors	_Total	LoadTest:Errors	IP-0AE33654		0	0	0	0

Performance counter chart for \Processor(\_Total)\% Processor Time



~40% CPU

~5% CPU

- Legends
- AzureScaling.EnhancedWork... (Blue)
  - AzureScaling.EnhancedWork... (Green)
  - AzureScaling.EnhancedWebU... (Orange)
  - AzureScaling.EnhancedWork... (Purple)

```
/// <summary>
/// Number of threads per worker role
/// </summary>
private int DegreeOfParallelism { get; set; }

/// <summary>
/// Indicates whether smart seat checking is used
/// </summary>
private bool UseSmartSeatChecking { get; set; }

private CancellationTokenSource cancellationTokenSource;
private CancellationToken cancellationToken;
private Task[] workerThreads;


private void StartWorkerThreads()
{
    Trace.WriteLine("Starting worker threads", "Information");

    // Get cancellation token
    this.cancellationTokenSource = new CancellationTokenSource();
    this.cancellationToken = this.cancellationTokenSource.Token;

    this.workerThreads = ParallelEnumerable.Range(0, this.DegreeOfParallelism)
        .Select(i => Task.Factory.StartNew(this.WorkerThread))
        .ToArray();
}

private void StopWorkerThreads()
{
    Trace.WriteLine("Stopping worker threads", "Information");

    this.cancellationTokenSource.Cancel();
    Task.WaitAll(this.workerThreads);
    this.cancellationTokenSource.Dispose();
}
```

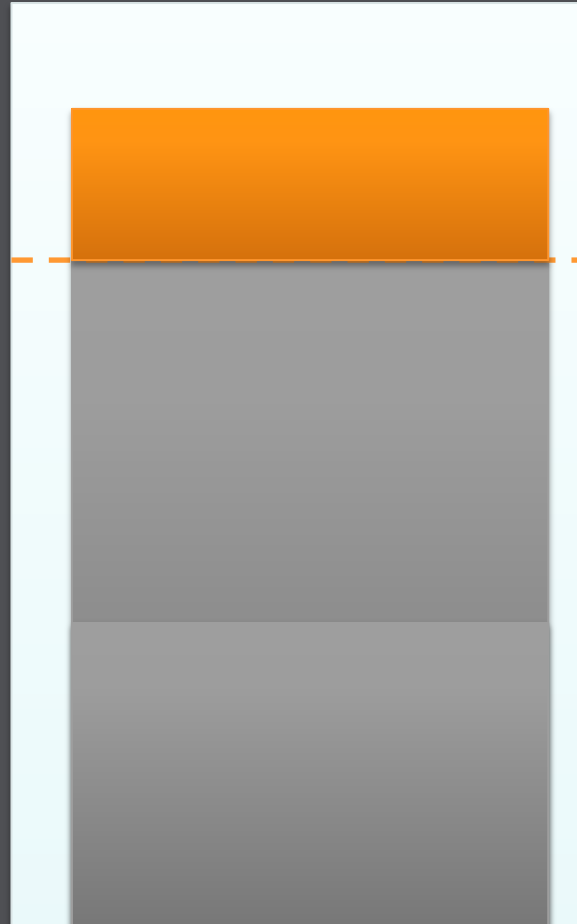


Start mult.  
worker threads

**Avoid  
Locks!!**

powered by Microsoft®  
**ONNECTIONS**

Transactionally safe  
checking (i.e. locking)

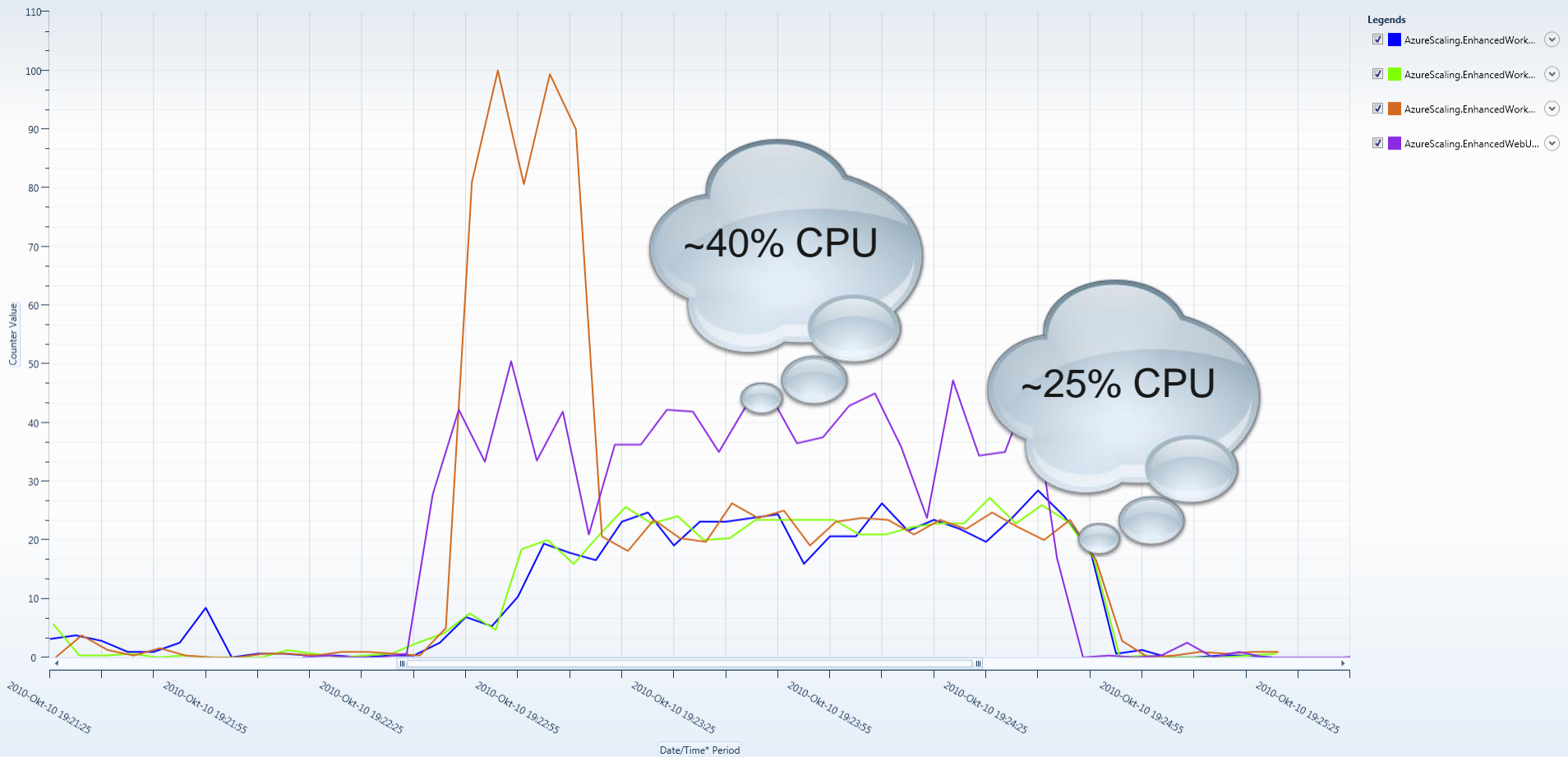


High Watermark

Simple checking without  
transactions (i.e. no locking)



Performance counter chart for \Processor(\_Total)\% Processor Time



# 6. Care About Storage Types, you must!



## Storage

Per GB stored and transactions  
**\$0.15 GB/month**  
\$0.01/10k transactions

## Web Edition

Per database/month  
**\$9.99/month**  
(1-5 GB DB/month)

## Business Edition

Per database/month  
**Starting at \$99.99/month**  
(10-50 GB DB/month)

- **Think about...**
  - ...storage volume needed
  - ...number of transactions
  - ...programming effort
- **Background information**  
for Azure Storage Pricing

- ...client capabilities
- ...necessary performance
- ...necessary throughput
- ...static/dynamic nature

# When To Use What??

## SQL Azure

- Strong programming model needed
- Need for complex ACID transactions

KO:

- Restricted storage amount acceptable (currently max. 50GB/DB)
- TDS is possible (soon no-code OData will be an option)

## Windows Azure Storage

- Price sensitive (~1/65<sup>th</sup> compared to SQL Azure)
- Auto-scale out → Fast
- Large storage volumes (many, many TBs)
- REST/HTTP needed
- CDN needed
  - Possible with SQL Azure + web role, too – not without code
- NTFS needed (Drives)
- Queues needed

# 7. Microsoft's SLAs → No Silver Bullet

You get credits, not compensation for damage

$$\frac{\text{Maximum Connectivity Minutes} - \text{Connectivity Downtime}}{\text{Maximum Connectivity Minutes}} = \text{Monthly Connectivity Uptime Percentage}$$

## ii. Monthly Connectivity Uptime Service Levels

Monthly Uptime Percentage	Service Credit*
<99.95%	10%
<99%	25%

\*Service credit applies only to Windows Azure Compute Services (i.e., not Windows Azure Storage or other Windows Azure platform services)

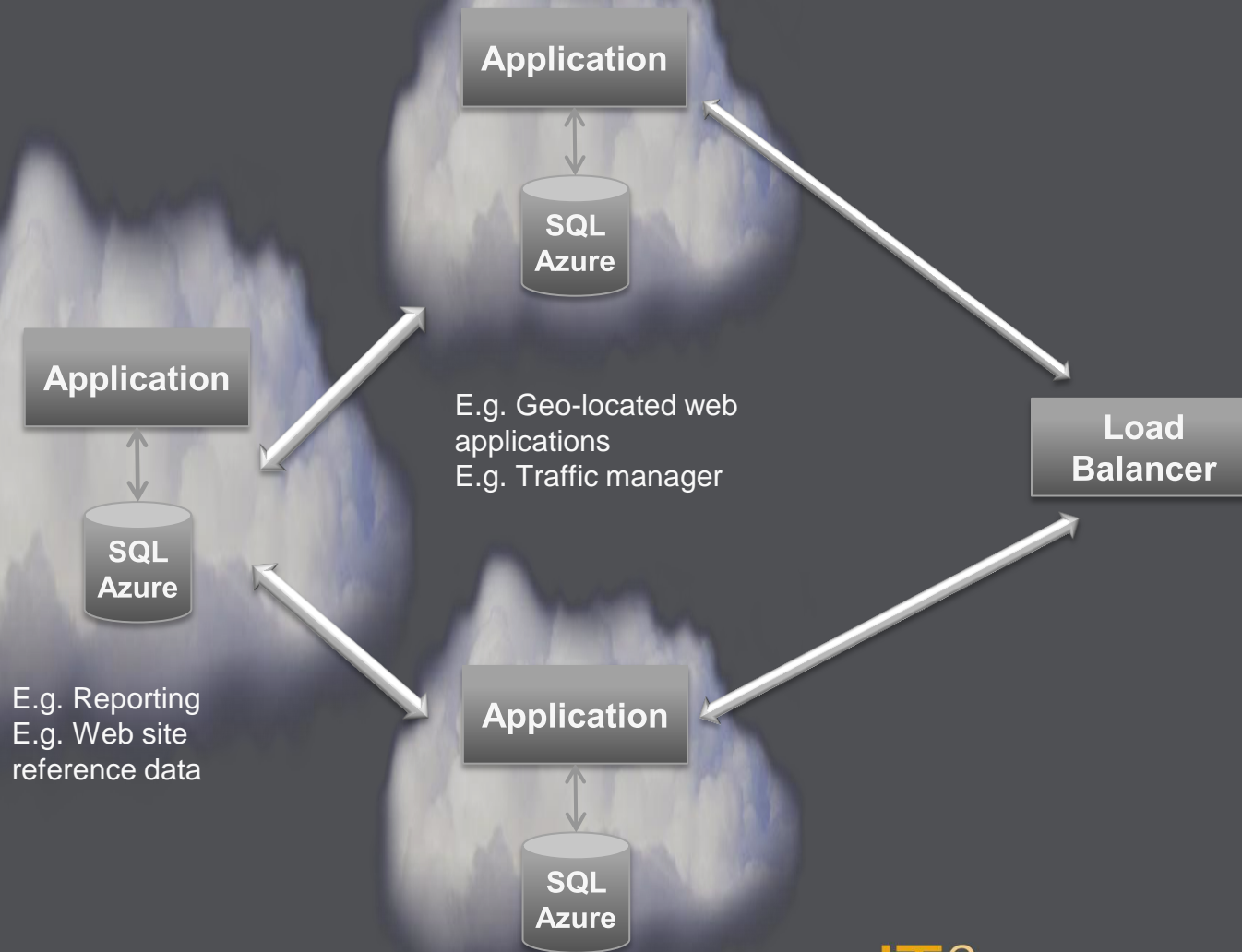
## iii. Monthly Role Instance Uptime Service Levels

Monthly Uptime Percentage	Service Credit*
<99.9%	10%
<99%	25%

\*Service credit applies only to Windows Azure Compute charges (i.e., not Windows Azure Storage or other Windows Azure platform services)

Backup still important!

# Scenarios



# Windows Azure Traffic Manager

**Beta Programs**

Thank you for your interest in Windows Azure Beta Programs. Access to Windows Azure beta programs is by invite only, and is being allocated on a first-come-first-serve basis.

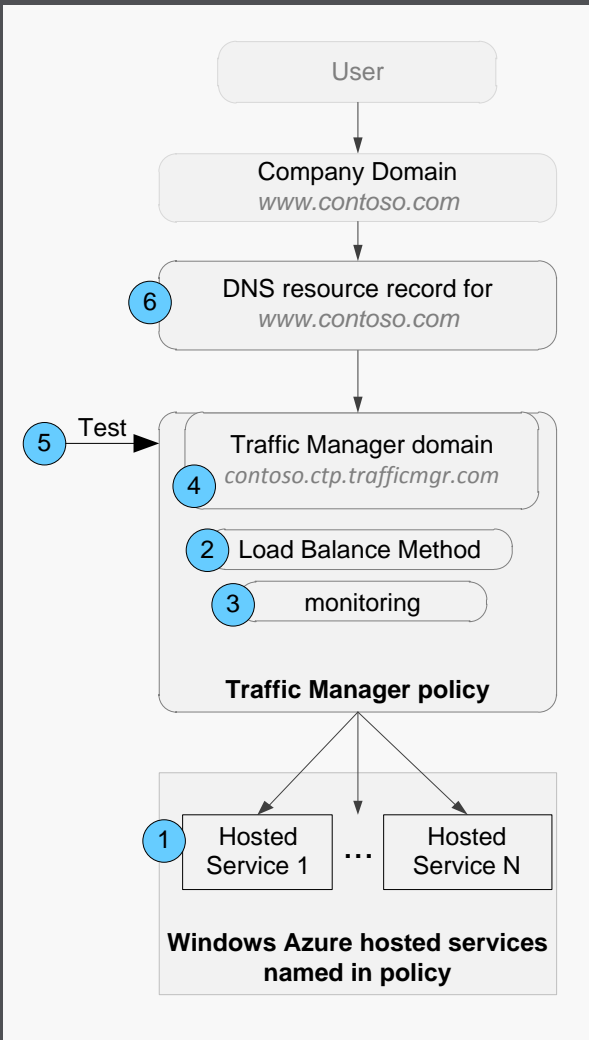
To apply for a beta program, select the program's checkbox and click "Apply for Access".

We appreciate your patience.

Name	Description	Status
Windows Azure Connect	The Windows Azure Connect CTP provides secure IP-level connectivity between Windows Azure and on-premises IT resources. Please see <a href="#">Virtual Network</a> for more details.	Active
Traffic Manager	The Windows Azure Traffic Manager CTP is a new feature that allows you to load balance traffic to multiple hosted services. You can choose from three load balancing methods: Performance, Failover, or Round Robin. Traffic Manager will monitor your hosted service on any http or https port you choose. If it detects your service is offline it will send traffic to the next best available service. Please see <a href="#">Virtual Network</a> for more details.	Active
MVP MSDN Abc Special: a-alejim		Active
VM Role	The VM Role Beta Program includes a new Windows Azure role that allows you to upload a custom virtual hard disk image of a Windows Server 2008 R2 virtual machine and run it in Windows Azure. By checking the box to opt-in to the VM Role Beta program, you accept the <a href="#">license terms</a> for your use of the Windows Server 2008 R2 software in the VM Role Beta Program.	Active



# Windows Azure Traffic Manager



### Create Traffic Manager policy

**Choose a subscription**  
Special: a-alejim

**Choose a load balancing method**

- Performance: Detects the location of the user traffic to route it to the best online hosted service based on network performance.
- Failover: Create an ordered list of hosted services. All traffic is routed to the online service highest on the list.
- Round Robin: Equally distributes traffic to all hosted services.

**Select hosted services to include in policy**

Type to filter DNS names

Available DNS names:	Region
customodataprovider.cloudapp.net	West Europe
devconnections2011.cloudapp.net	West Europe
importosmlarge.cloudapp.net	West Europe
loadtestingdemo.cloudapp.net	West Europe

Selected DNS names:

**Specify a monitoring endpoint**

Protocol	Port	Relative path and filename
HTTP	80	/

**Specify DNS settings**

Traffic Manager DNS prefix: [ ] .ctp.trafficmgr.com

DNS time to live (TTL): [ 300 ] seconds



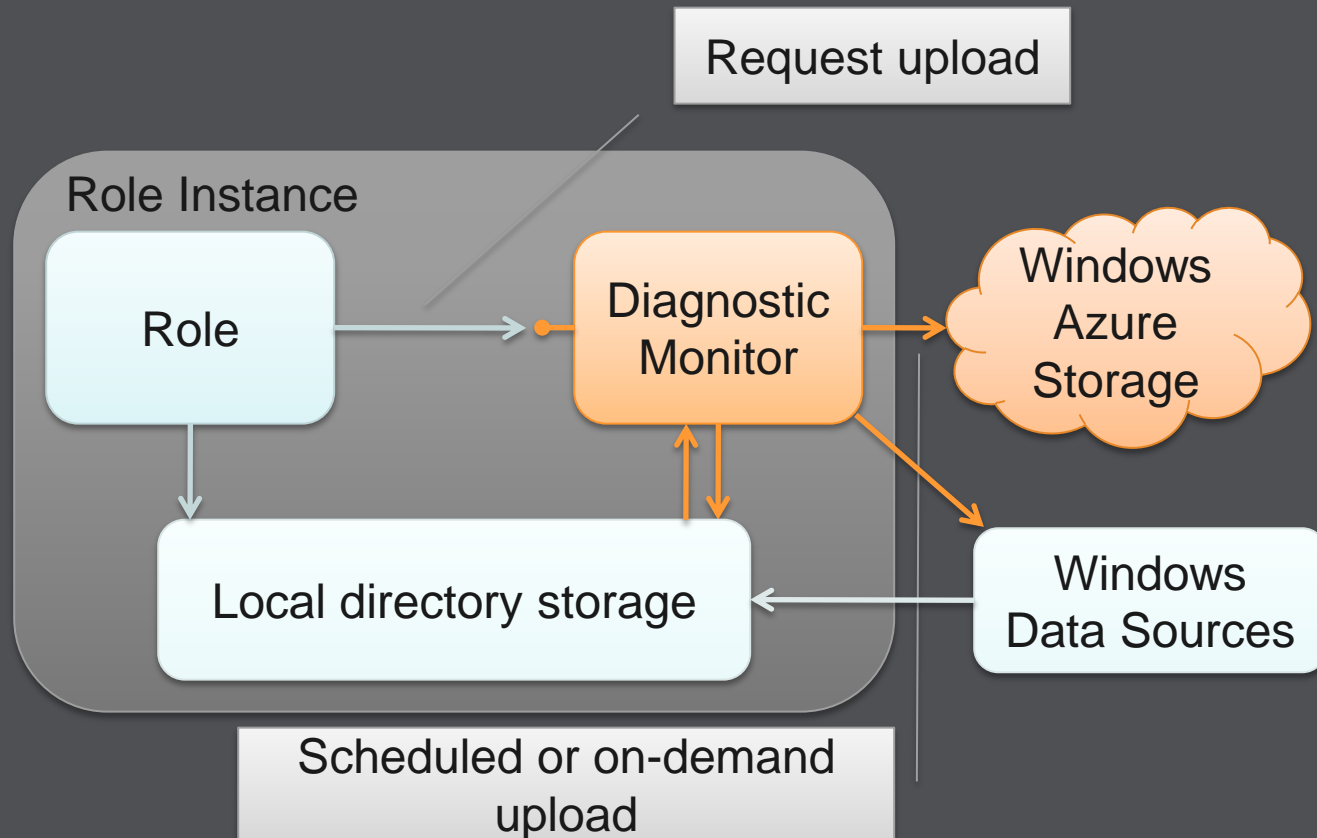
## Operational

“represents the level of command which coordinates the minute details of tactics” (Source: [Wikipedia](#))

# 8. Monitoring is Crucial

- Optimization is SEXY again!
- Windows Azure Diagnostics
  - „Telemetry“ for your productive Azure roles
  - Long-term monitoring (service health, auto-scaling, etc.)
- Remote Desktop
  - Difficult because of dynamic nature of the cloud
  - Troubleshooting, development
- IntelliTrace
  - Troubleshooting, development

# Windows Azure Diagnostics



```
public static void ApplyDefaultAzureDiagnosticConfiguration()
{
    // Get default initial configuration.
    var config = DiagnosticMonitor.GetDefaultInitialConfiguration();

    // Adding performance counters to the default diagnostic configuration
    ConfigureDiagnostics(config);
    ScheduleTransfer(config, TimeSpan.FromMinutes(1));

    // Start the diagnostic monitor with the modified configuration.
    DiagnosticMonitor.Start("Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString", config);
}
```

```
public static void ConfigureDiagnostics(DiagnosticMonitorConfiguration config)
{
    config.PerformanceCounters.DataSources.Add(
        new PerformanceCounterConfiguration()
        {
            CounterSpecifier = @"\Processor(_Total)\% Processor Time",
            SampleRate = TimeSpan.FromSeconds(5)
        });
}
```

```
public static void ScheduleTransfer(DiagnosticMonitorConfiguration config, TimeSpan transferPeriod)
{
    config.PerformanceCounters.ScheduledTransferPeriod =
        config.DiagnosticInfrastructureLogs.ScheduledTransferPeriod = transferPeriod;
}
```

```
public static DeploymentDiagnosticManager GetRemoteDiagnosticsManager()
{
    return CloudAccountDiagnosticMonitorExtensions.CreateDeploymentDiagnosticManager(
        CloudStorageAccount.FromConfigurationSetting("StorageConnectionString"),
        ConfigurationManager.AppSettings["DeploymentID"]);
}
```

Setup  
Diagnostics

Remote  
Diagnostics  
config.

# *demo*

## Monitoring

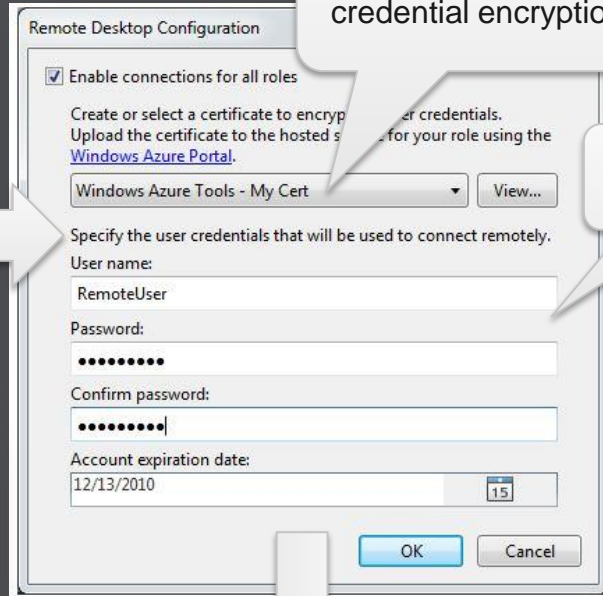
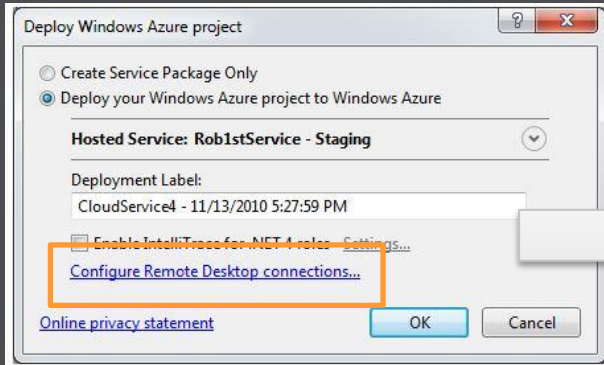
Windows Azure  
Diagnostics and RDP



# Demo Content

- Show diagnostics in storage
- Show cerebrata diagnostics manager
- Show RDP into role

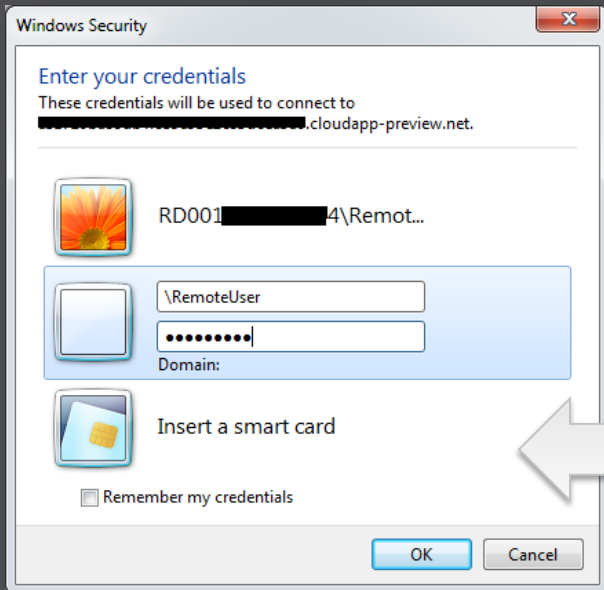
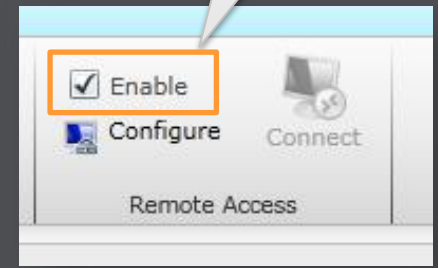
# Remote Connection (RDP)



Certificate for credential encryption

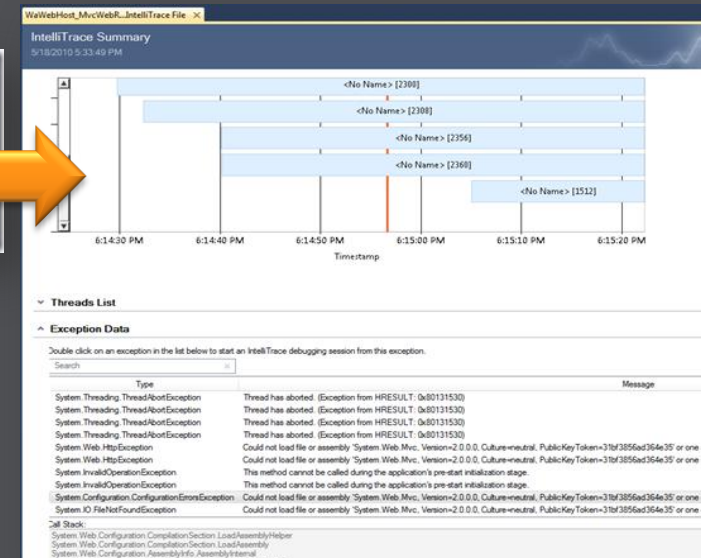
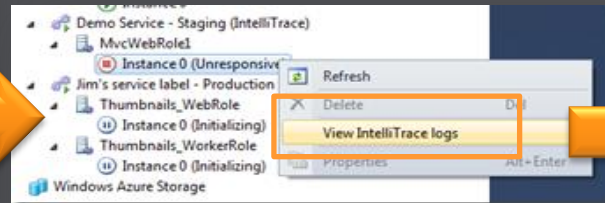
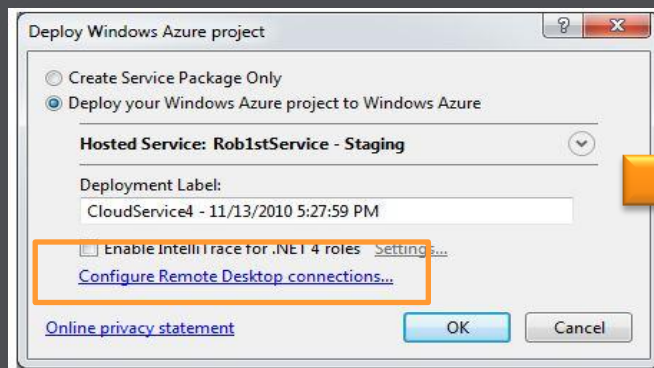
Credentials

Don't forget to enable RDP!



# IntelliTrace in Azure

- Collect data about events that happened in Azure
- Open data in VS and see e.g. exceptions, call flow, etc.
- IntelliTrace data is collected in Windows Azure Storage



# 9. Occupy Roles Wisely

## Web Role

- IIS (since 1.4)

## Worker Role

- `While (true)`  
`{ }`

## VM-Role

- Custom

- **Web Role for...**
  - ...anything that should be hosted in IIS
  - ...multiple web sites in a single role
- **Worker Roles for...**
  - ...any kind of background work
  - Remember: Can be combined with web role
- **VM-Role only if...**
  - ...you are unable to automate role setup (startup tasks)
  - ...software necessary for role setup is unstable
  - ...software necessary for role setup needs UI

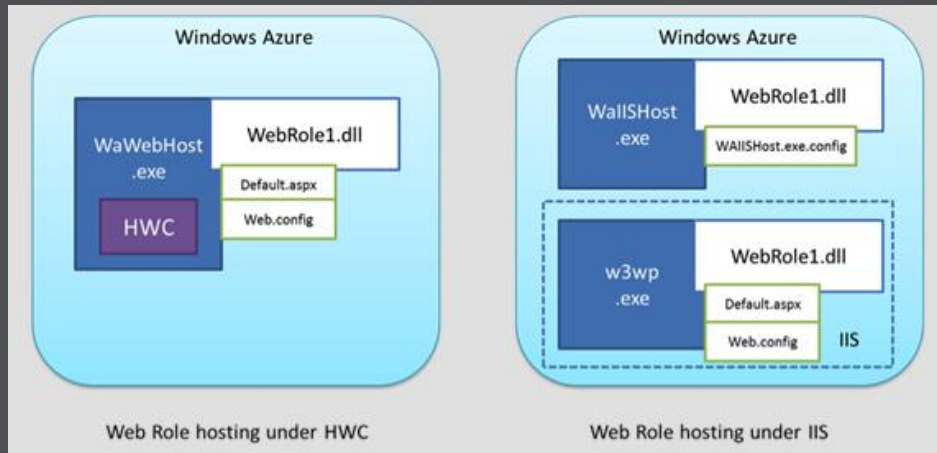
# *demo*

## Advanced Roles

## Full IIS, Startup Tasks

# Demo Content

- Show web role with multiple sites
  - Point out differences with entry point code



- Show worker role with complex startup task
  - Show project configuration
  - RDP to role, show folders



## ServiceDefinition.csdef

```
<Startup>  
  <Task commandLine="DACFramework\install_dac.cmd" executionContext="elevated" taskType="simple" />  
</Startup>
```

## Install\_dac.cmd

```
REM Cleanup old log files (just in case)  
erase DACFramework\install*.txt  
erase DACFramework\install_SQLSysClrTypes.txt > DACFramework\install_log.txt  
  
REM Install DAC components using MSIEXEC  
msiexec /i DACFramework\SQLSysClrTypes.msi /qn /!* DACFramework\install_SQLSysClrTypes.txt  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\SharedManagementObjects.msi /qn /!* DACFramework\install_SharedManagementObjects.txt  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\DACFramework.msi /qn /!* DACFramework\install_DACFramework.txt  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\SqlDom.msi /qn /!* DACFramework\install_SqlDom.txt  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\TSqlLanguageService.msi /qn /!* DACFramework\install_TSqlLanguageService.txt  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\SqlCmdLnUtils.msi /qn /!* DACFramework\install_SqlCmdLnUtils.txt  
if ERRORLEVEL 1 goto InstallError  
  
REM Success  
echo INSTALLATION SUCCESSFULLY COMPLETED >> DACFramework\install_log.txt  
goto EndOfScript  
  
:InstallError  
REM Handle installation error  
echo INSTALLATION ERROR (ERRORLEVEL=%ERRORLEVEL%) >> DACFramework\install_log.txt  
goto :EndOfScript  
  
:EndOfScript
```

## ServiceDefinition.csdef

```
<Startup>  
  <Task commandLine="DACFramework\install_dac.cmd" executionContext="elevated" taskType="simple" />  
</Startup>
```

## Install\_dac.cmd

```
REM Cleanup old log files (just in case)  
erase DACFramework\install*.txt  
erase DACFramework\install_SQLSysClrTypes.txt > DACFramework\install.log  
  
REM Install DAC components using MSIEXEC  
msiexec /i DACFramework\SQLSysClrTypes.msi /qn /!* DACFramework\install.log  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\SharedManagementObjects.msi /qn /!* DACFramework\install.log  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\DACFramework.msi /qn /!* DACFramework\install.log  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\SqlDom.msi /qn /!* DACFramework\install.log  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\TSqlLanguageService.msi /qn /!* DACFramework\install.log  
if ERRORLEVEL 1 goto InstallError  
msiexec /i DACFramework\SqlCmdLnUtils.msi /qn /!* DACFramework\install.log  
if ERRORLEVEL 1 goto InstallError  
  
REM Success  
echo INSTALLATION SUCCESSFULLY COMPLETED >> DACFramework\install.log  
goto EndOfScript  
  
:InstallError  
REM Handle installation error  
echo INSTALLATION ERROR (ERRORLEVEL=%ERRORLEVEL%) >> DACFramework\install.log  
goto :EndOfScript  
  
:EndOfScript
```

Solution 'BackupSample' (2 projects)

- Backup.Worker
  - Properties
  - References
  - DACFramework
    - DACFramework.msi**
    - DacImportExportCli.exe
    - DacImportExportCli.exe.config
    - install\_dac.cmd
    - SharedManagementObjects.msi
    - SqlCmdLnUtils.msi
    - SqlDom.msi
    - SQLSysClrTypes.msi
    - TSqlLanguageService.msi

Properties

**DACFramework.msi** File Properties

Build Action	None
Copy to Output Directory	Copy if newer
Custom Tool	
Custom Tool Namespace	
File Name	DACFramework.msi
Full Path	T:\TimeCockpit.Prototypes\C



## 10. Automation Saves Time

Automate your build process including deployment to Azure

# Build Automation with Azure

- **Goals**
  - Create a build process template for customization
  - Build cspkg in Build Process
  - Deploy to Azure using Azure Powershell Cmdlets
  - Run Unit test against newly deployed service
  - Remove Hosted Service
    - Unless you have too much \$

# Goal: Building cspkg in Build Process

- `Microsoft.CloudService.targets` provides a „Publish“ target
  - used by default for cloud projects
  - call the target additionally to the normal build
- Creates the cspkg in the Publish folder
  - Is automatically copied to Drop location

# Goal: Building cspkg in Build Process

- Edit
- Go
- Edit
- /t
- Save

General  
Trigger  
Workspace  
Build Defaults  
**Process**  
Retention Policy

Team Foundation Build uses a build process template defined by a Windows Workflow (XAML) file. The behavior of this template can be customized by setting the build process parameters provided by the selected template.

Build process template:  
**BuildDeployTestAzure.xaml** Show details

Build process parameters:

<b>1. Required</b>	
Items to Build	Build 1 project(s) for 1 platform(s) and configuration(s)
<b>2. Basic</b>	
Automated Tests	Run tests in assemblies matching <code>**\*test*.dll</code>
Build Number Format	<code>\$(BuildDefinitionName)_\$(Date:yyyyMMdd)\$(Rev:.r)</code>
Clean Workspace	All
Logging Verbosity	<b>Diagnostic</b>
Perform Code Analysis	AsConfigured
Source And Symbol Server Settings	
<b>3. Advanced</b>	
Agent Settings	Use agent where Name=* and Tags is empty; Max Wait Tim
Analyze Test Impact	True
Associate Changesets and Work Items	True
Copy Outputs to Drop Folder	True
Create Work Item on Failure	<b>False</b>
Disable Tests	False
Get Version	
Label Sources	True
MSBuild Arguments	<b>/t:Build;Publish</b>
MSBuild Platform	Auto
Private Drop Location	
<b>4. Misc</b>	
AzureCertificateThumbprint	993C21CE392234A6EFC3E6A344D64175A154776
AzureHostedServiceName	TFSAzureDeployTest
AzureStorageName	oop2011
AzureSubscriptionID	501a9957-7343-41f9-bef8-d3af3c8e4bfd

Windows Workflow (XAML) file. The behavior of this template can be customized by setting the build process parameters provided by the selected template.

Show details

Build 1 project(s) for 1 platform(s) and configuration(s)

Run tests in assemblies matching `**\*test*.dll`

`$(BuildDefinitionName)_$(Date:yyyyMMdd)$(Rev:.r)`

All

**Diagnostic**

AsConfigured

Use agent where Name=\* and Tags is empty; Max Wait Tim

True

True

True

**False**

False

True

Auto

**4. Misc**

993C21CE392234A6EFC3E6A344D64175A154776

TFSAzureDeployTest

oop2011

501a9957-7343-41f9-bef8-d3af3c8e4bfd



# PowerShell Script for Deployment

```
# certificatethumb subscriptionId servicename package config
$certTP = $args[0]
$cert = Get-Item cert:\CurrentUser\My\$certTP
$sub = $args[1]
$storageAccount = $args[2]
$serviceName = $args[3]
$package = $args[4]
$config = $args[5]
$label = $args[6]
Add-PSSnapin AzureManagementToolsSnapIn
New-Deployment -serviceName $serviceName -storageserviceName $storageAccount -subscriptionId $sub -
certificate $cert -slot 'Staging' -package $package -configuration $config -label $label | Get-OperationStatus -
waitToComplete
Get-HostedService $serviceName -Certificate $cert -SubscriptionId $sub | Get-Deployment -Slot 'Staging' | Set-
DeploymentStatus 'Running' | Get-OperationStatus -waitToComplete
Get-Deployment staging -subscriptionId $sub -certificate $cert -serviceName $serviceName | Move-Deployment | Get-
OperationStatus -waitToComplete
Get-HostedService $serviceName -Certificate $cert -SubscriptionId $sub | Get-Deployment -Slot 'Staging' | Set-
DeploymentStatus 'Suspended' | Get-OperationStatus -waitToComplete
Get-HostedService $serviceName -Certificate $cert -SubscriptionId $sub | Get-Deployment -Slot 'Staging' | Remove-
Deployment | Get-OperationStatus -waitToComplete
Get-HostedService $serviceName -Certificate $cert -SubscriptionId $sub | Get-Deployment -Slot 'Production' | Set-
DeploymentStatus 'Running' | Get-OperationStatus -waitToComplete

$ready = $False
while(!$ready)
{
    $d = Get-HostedService $serviceName -Certificate $cert -SubscriptionId $sub | Get-Deployment -Slot 'Production'
    $ready = ($d.RoleInstanceList[0].InstanceStatus -eq "Ready") -and ($d.Label -eq $label)
}
```

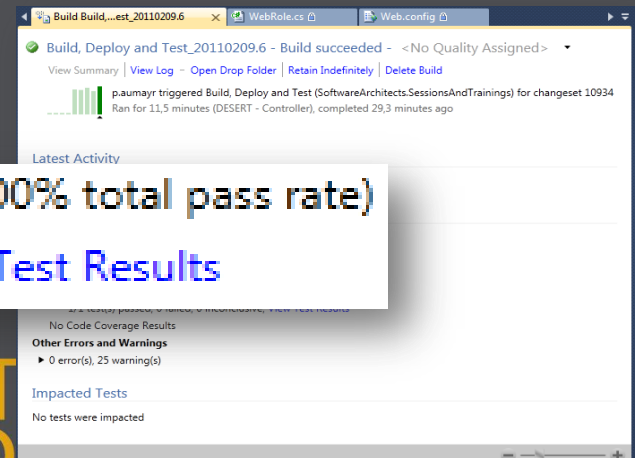
# Run Unit test against newly deployed service

- Unit test currently runs against a local service
- Change endpoint in test configuration
  - Service name you chose(!)

```
<endpoint address="http://TFSAzureDeployTest.cloudapp.net/ToUpper.svc" binding="basicHttpBinding"
  →  →  bindingConfiguration="BasicHttpBinding_IToUpper" contract="ToUpperService.IToUpper"
  →  name="BasicHttpBinding_IToUpper_Azure" />
```

- Start a new build
- Open build

▼ 1 test run(s) completed - 100% average pass rate (100% total pass rate)  
1/1 test(s) passed, 0 failed, 0 inconclusive, [View Test Results](#)



# PowerShell Script to Remove Deployment

```
# certificatethumb subscriptionId servicename
$certTP = $args[0]
$cert = Get-Item cert:\CurrentUser\My\$certTP
$sub = $args[1]
$servicename = $args[2]
Add-PSSnapin AzureManagementToolsSnapIn

Get-HostedService $servicename -Certificate $cert -SubscriptionId $sub |
  Get-Deployment -Slot 'Production' |
  Set-DeploymentStatus 'Suspended' |
  Get-OperationStatus -waitToComplete

Remove-Deployment -Slot 'Production' -ServiceName $servicename -SubscriptionId $sub -Certificate $cert |
  Get-OperationStatus -waitToComplete
```

# Summary

- **Strategy**
  - Business Model
  - Multi-Tenancy
  - Make or Buy
- **Tactical**
  - Scale Out Not Up
  - Async Rulez!
  - Storage Types
  - SLAs
- **Operational**
  - Monitoring
  - Choose Roles Carefully
  - Automation Saves Time

# Your Feedback is Important

Please fill out a session evaluation form.

Thank you!