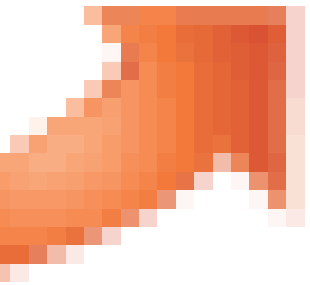
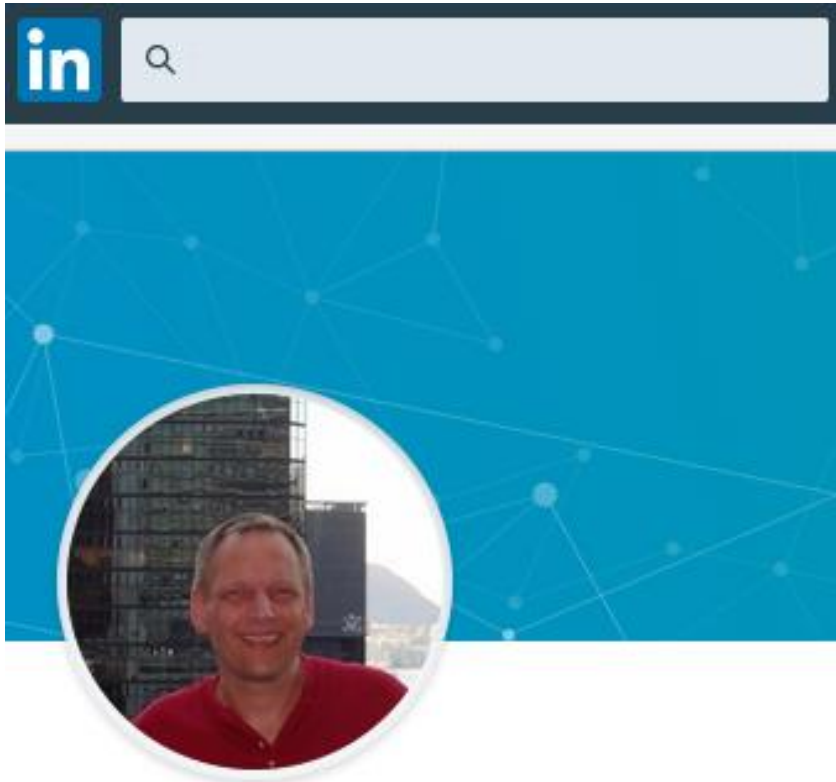


SQL-CLR mit "strict security" im
SQL-Server und in der Azure-Cloud

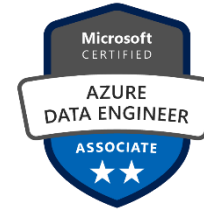


about me



Alexander Karl

<https://www.linkedin.com/in/alexander-karl-44561012a/>



.. heutiges Programm

? was ist "SQL-CLR"

? was ist besonders an der "strict security"

? Was sind die Feinheiten in der Azure-Cloud

was ist "SQL-CLR"

seit der Version SQL-Server 2005 können Programmierungen zusätzlich zu T-SQL auch mit .net-Sprachen erfolgen.

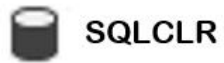
- Data Types
- Stored Procedures
- Triggers
- Functions
 - Scalar
 - Aggregate

“SQL-CLR” Anwendungen ...

- Stringvalidierungen mit **Regular Expressions**
- **Hashstrings** (SHA1, SHA256, ...)
- Stringvergleich (**Levenshtein**)



„build“



„deploy“

```
use SQLCLR  
GO  
  
CREATE ASSEMBLY ...  
  
CREATE FUNCTION ...
```

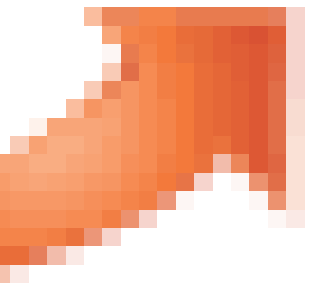
C# code

```
1 using System;
2 using System.Data;
3 using System.Data.SqlTypes;
4 using System.Data.SqlClient;
5 using Microsoft.SqlServer.Server;
6
7 public partial class StoredProcedures
8 {
9     [Microsoft.SqlServer.Server.SqlProcedure]
10    public static void SaveXML(SqlXml XmlData, SqlString FileName)
11    {
12        System.Xml.XmlDocument xmlDoc = new System.Xml.XmlDocument();
13        xmlDoc.LoadXml(XmlData.Value);
14        xmlDoc.Save(FileName.Value);
15    }
16};
```

TSQL code

```
1  -- Registrieren der Assembly
2  CREATE ASSEMBLY xmlSaveProc
3  FROM 'C:\ <folder> \xmlSaveProc.dll'
4  WITH Permission_Set = External_Access
5
6  -- überprüfung
7  SELECT * FROM sys.assemblies
8  SELECT * FROM sys.assembly_files
9
10 -- Erzeugen der Proc aus der Assembly
11 CREATE PROCEDURE dbo.SaveXML @xmlIn xml, @fileOut nvarchar(255)
12     as external name xmlSaveProc.StoredProcedures.SaveXML
13     --          DLL          .Class          .Proc name
```


Demo



SQL CLR seit 2017

SQLQuery1.sql

```
1 use master
2 GO
3
4 SELECT *
5 FROM sys.configurations
6 WHERE name like '%clr%'
7
```

100 %

Results Messages

	configuration_id	name	value	minimum	maximum	value_in_use	description	is_dynamic	is_advanced
1	1562	clr enabled	1	0	1	1	CLR user code execution enabled in the server	1	0
2	1587	clr strict security	1	0	1	1	CLR strict security enabled in the server	1	1

handling der "strict security"

- deaktivieren `exec sp_configure 'clr strict security', 0 -- 0=disabled`
- `sp_add_trusted_assembly` `exec sys.sp_add_trusted_assembly @hash = 0x8893AD6D78D14EE43DF4...`
- Verwenden der .dll und eines asymmetrischen Schlüssels
<https://techcommunity.microsoft.com/t5/sql-server-support/deploying-sql-clr-assembly-using-asymmetric-key/ba-p/316727>
- Signieren der Assembly mittels Zertifikat

handling der "strict security" sp_add_trusted_assembly

- wie lässt sich der als Parameter erforderliche Hashstring (SHA2_512) erzeugen.

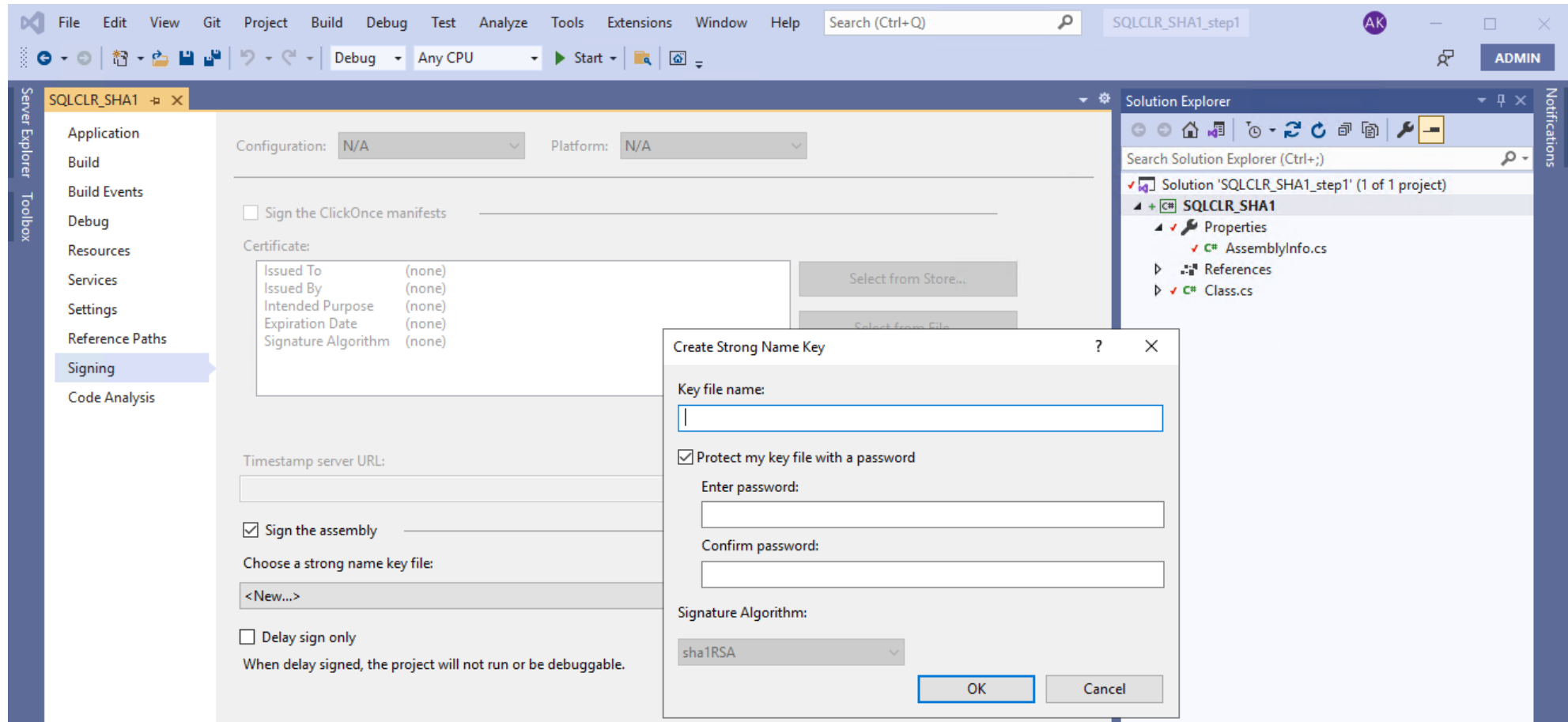
syntaxsql

```
HASHBYTES ( '<algorithm>', { @input | 'input' } )
```

```
<algorithm> ::= MD2 | MD4 | MD5 | SHA | SHA1 | SHA2_256 | SHA2_512
```

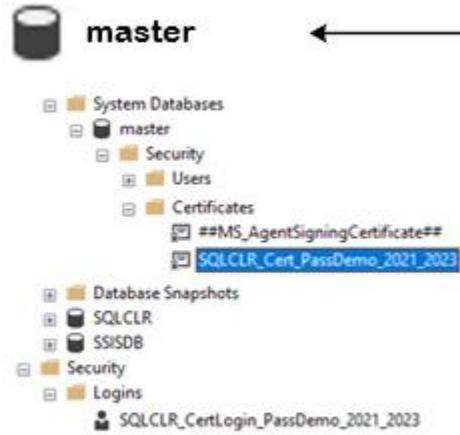
- <https://docs.microsoft.com/de-de/sql/t-sql/functions/hashbytes-transact-sql?view=sql-server-ver15>
- Achtung, ältere Versionen
Bei SQL Server 2014 (12.x) und früher sind zulässige Eingabewerte auf 8.000 Byte beschränkt.

Verwenden der .dll und eines asymmetrischen Schlüssels



<https://techcommunity.microsoft.com/t5/sql-server-support/deploying-sql-clr-assembly-using-asymmetric-key/ba-p/316727>

ha



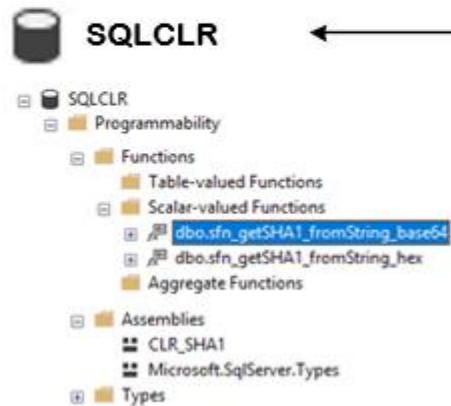
1. makecert.exe



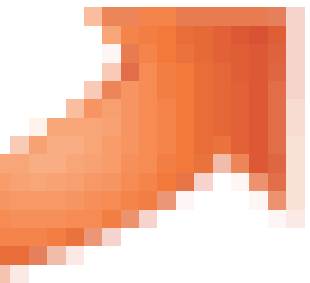
2. pvk2pfx.exe



3. signtool.exe



Demo



in Azure ...



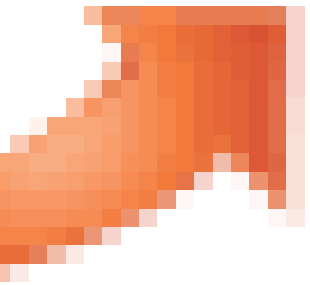
Virtual machine (IaaS)



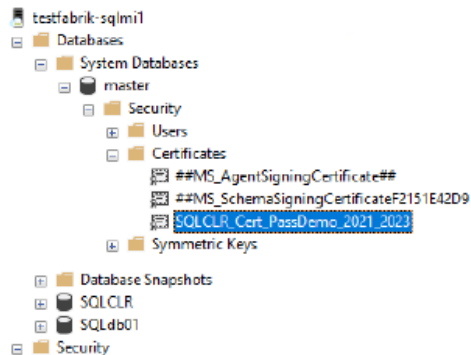
SQL Managed Instance(PaaS)



SQL Database (PaaS)



,hex_Content



```

use master
GO

CREATE CERTIFICATE SQLCLR_Cert_PassDemo_2021_2023
FROM BINARY = 0x3082034B30820233A00302010202109...

CREATE LOGIN SQLCLR_CertLogin_PassDemo_2021_2023
FROM CERTIFICATE SQLCLR_Cert_PassDemo_2021_2023
GO

GRANT UNSAFE ASSEMBLY
TO SQLCLR_CertLogin_PassDemo_2021_2023

```

```

use SQLCLR
GO

```

```

CREATE ASSEMBLY CLR_SHA1
FROM 0x4D5A90000300000004000000FFFF0000B800...
WITH Permission_Set = EXTERNAL_ACCESS
GO

```

```

CREATE FUNCTION dbo.sfn_getSHA1_fromString_base64
    ( @inputString nvarchar(4000) )
RETURNS nvarchar(400)
AS
EXTERNAL NAME
    CLR_SHA1.[SQLCLR_SHA1.UserDefinedFunctions].getSHA1_fromString_base64;
---- Assembly.[Namespace .Class] .Function Name

```

```

use master
GO

SELECT *
FROM sys.certificates
WHERE name like '%PassDemo%'

-- 1. CertEncoded >>
SELECT CERTENCODED( 259 ) as 'hex_Cert'
-->> 0x3082034B30820233A00302010202109...

-->> switch Instance

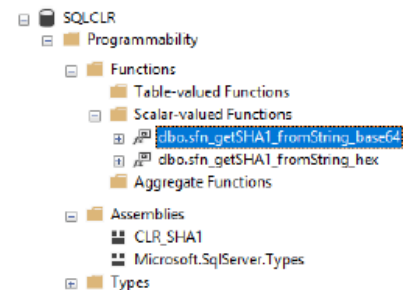
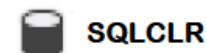
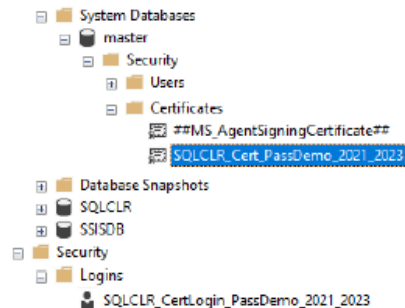
```

```

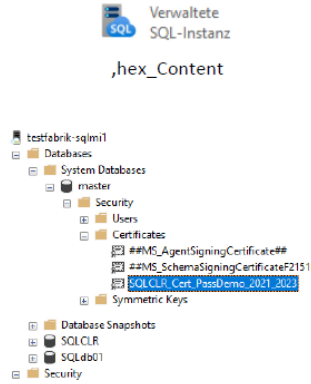
use SQLCLR
GO

SELECT a.name, f.content as 'hex_Content'
FROM sys.assemblies a
join sys.assembly_files f
ON a.assembly_id = f.assembly_id

```



Deployment von der Virtual Machine in die SQL Managed Instance



Verwaltete SQL-Instanz
,hex_Content

```
use master
GO

CREATE CERTIFICATE SQLCLR_Cert_PassDemo_2021_2023
FROM BINARY = 0x3082034B30820233A00302010202109...

CREATE LOGIN SQLCLR_CertLogin_PassDemo_2021_2023
FROM CERTIFICATE SQLCLR_Cert_PassDemo_2021_2023
GO

GRANT UNSAFE ASSEMBLY
TO SQLCLR_CertLogin_PassDemo_2021_2023
```

```
use SQLCLR
GO

CREATE ASSEMBLY CLR_SHA1
FROM 0x4D5A90000300000004000000FFFF0000B800...
WITH Permission_Set = EXTERNAL_ACCESS
GO

CREATE FUNCTION dbo.sfn_getSHA1_fromString_base64
( @inputString nvarchar(4000) )
RETURNS nvarchar(400)
AS
EXTERNAL NAME
CLR_SHA1.[SQLCLR_SHA1.UserDefinedFunctions].getSHA1_fromString_base64;
---- Assembly.[Namespace .Class ].Function Name
```

```
use master
GO

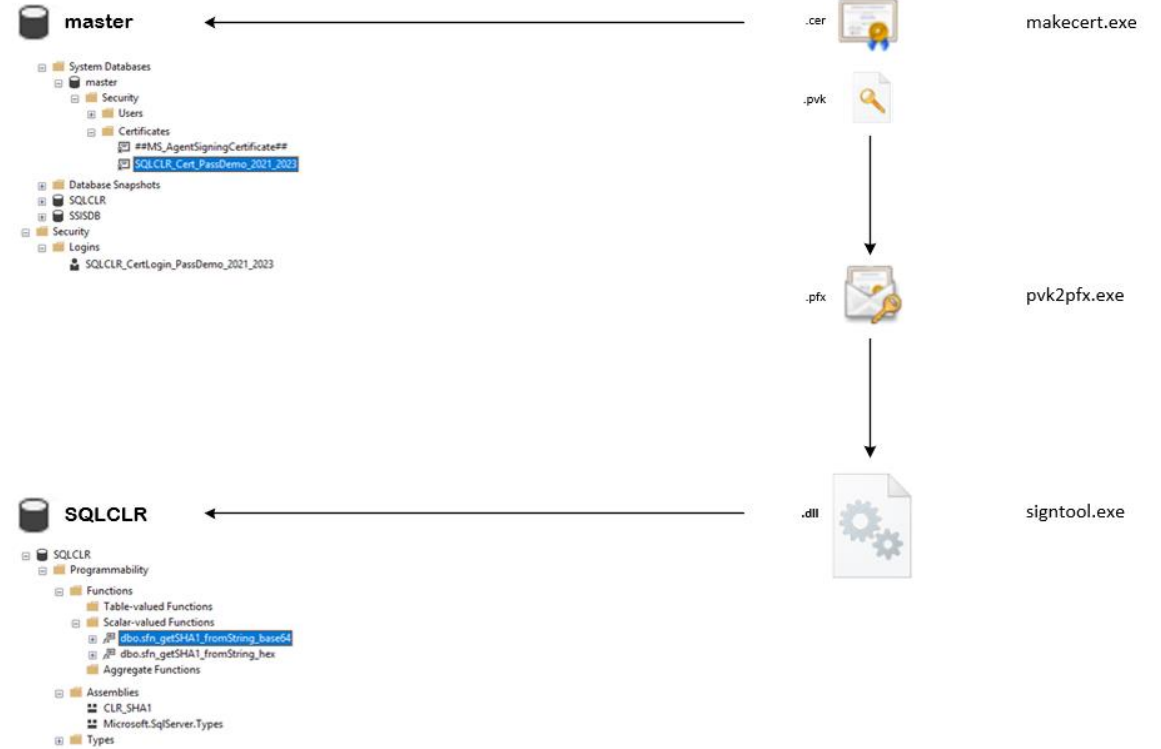
SELECT *
FROM sys.certificates
WHERE name like '%PassDemo%'

-- 1. CertEncoded >>
SELECT CERTENCODED( 259 ) as 'hex_Cert'
-->> 0x3082034B30820233A00302010202109..

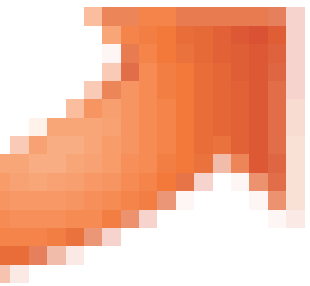
-->> switch Instance
```

```
use SQLCLR
GO

SELECT a.name, f.content as 'hex_Content'
FROM sys.assemblies a
join sys.assembly_files f
ON a.assembly_id = f.assembly_id
```



Demo



Literaturempfehlung



[Dieses Bild anzeigen](#)

Dem Autor folgen



Thorsten Kansy

+ Folgen

SQL Server 2008-Programmierung mit der CLR und .NET: Datenbankseitige Programmierung ausreizen Gebundene Ausgabe – 20. April 2009

von [Thorsten Kansy](#) (Autor)

★★★★☆ 1 Sternebewertung

[Alle Formate und Editionen anzeigen](#)

Gebundenes Buch

35,90 €

2 Gebraucht ab 0,99 €

2 Neu ab 35,90 €

Dieses Buch behandelt in vier übersichtlichen Teilen die datenbankseitige Programmierung des SQL Server 2008 mit Hilfe der integrierten .NET CLR (Common Language Runtime): Der erste Teil erklärt die Grundlagen, die für eine effektive Programmierung benötigt werden. Die Themen Sicherheit und Überlegungen zum sinnvollen Einsatz finden dabei besondere Beachtung. Darauf aufbauend zeigt der zweite Teil anhand praktischer Beispiele, wie gespeicherte Prozeduren, Trigger, Funktionen und benutzerdefinierte Typen mit .NET für den SQL Server entwickelt werden können. Im dritten Teil wird die Integration in eine leistungsfähige Anwendung behandelt. Als Abschluss zeigt der vierte Teil die Arbeit mit den Typen `XML`, `GEOMETRY` und `GEOSPATIAL` des Microsoft SQL Server .NET < [Mehr lesen](#)

Seitenzahl der Print-Ausgabe



352 Seiten

Sprache



Deutsch

Herausgeber



Microsoft

Erscheinungstermin



20. April 2009

ISBN-10



3866454368

ISBN-13

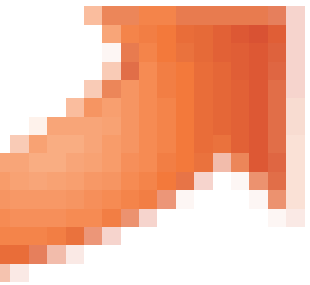


978-3866454361

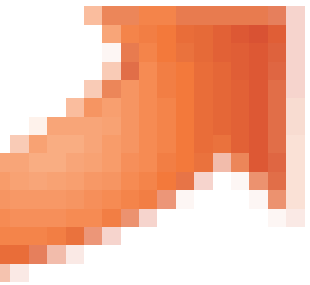


die aktuelle und die bisherigen Präsentationen
sind hier

<http://www.net-cde.de/presentations/>



.. bis zum nächsten ...





.. jetzt einen schönen
Feierabend

Sponsors

You Rock! Sponsor



Gold Sponsor



Silver Sponsor



Bronze Sponsor

