


# ETL für Faulpelze Einführung in Biml für SSIS

WEITER BLICKEN. MEHR ERKENNEN. BESSER ENTSCHIEDEN.



- › DBA?
- › DEV?
- › Management?
- › Sonstige?
- › SSIS User?
- › Biml Erfahrungen?

- › Ben Weissman, Solisyon, Nürnberg
- ›  @bweissman
- › [b.weissman@solisyon.de](mailto:b.weissman@solisyon.de)
- › SQL Server seit Version 6.5
- › Zu blöd für C#



**Microsoft**  
**CERTIFIED**

Solutions Expert

---

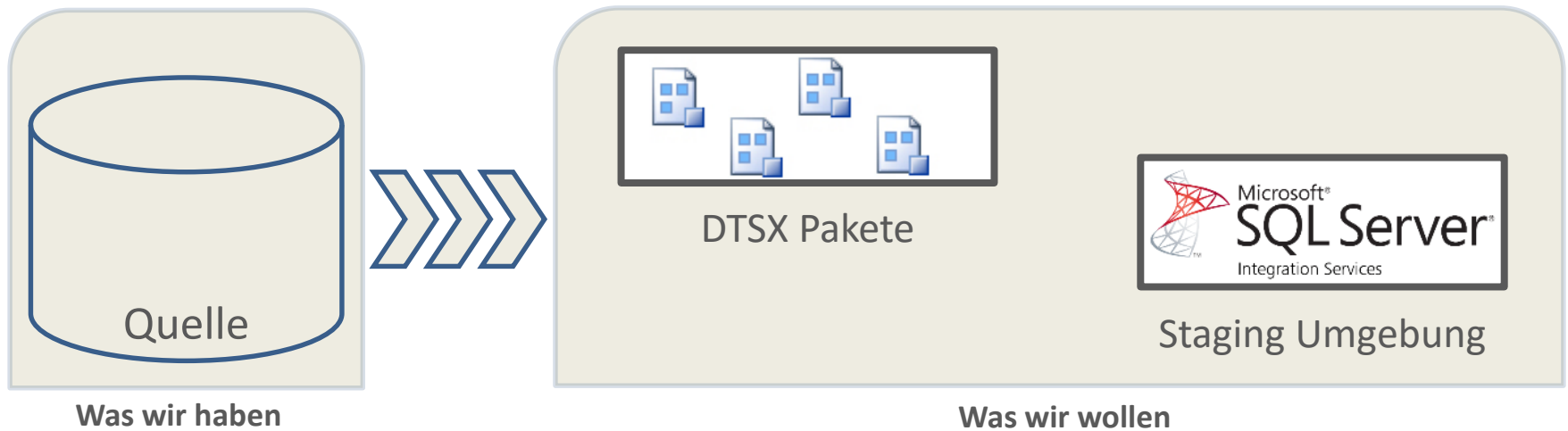
Business Intelligence

**Microsoft**  
**CERTIFIED**

Solutions Expert

---

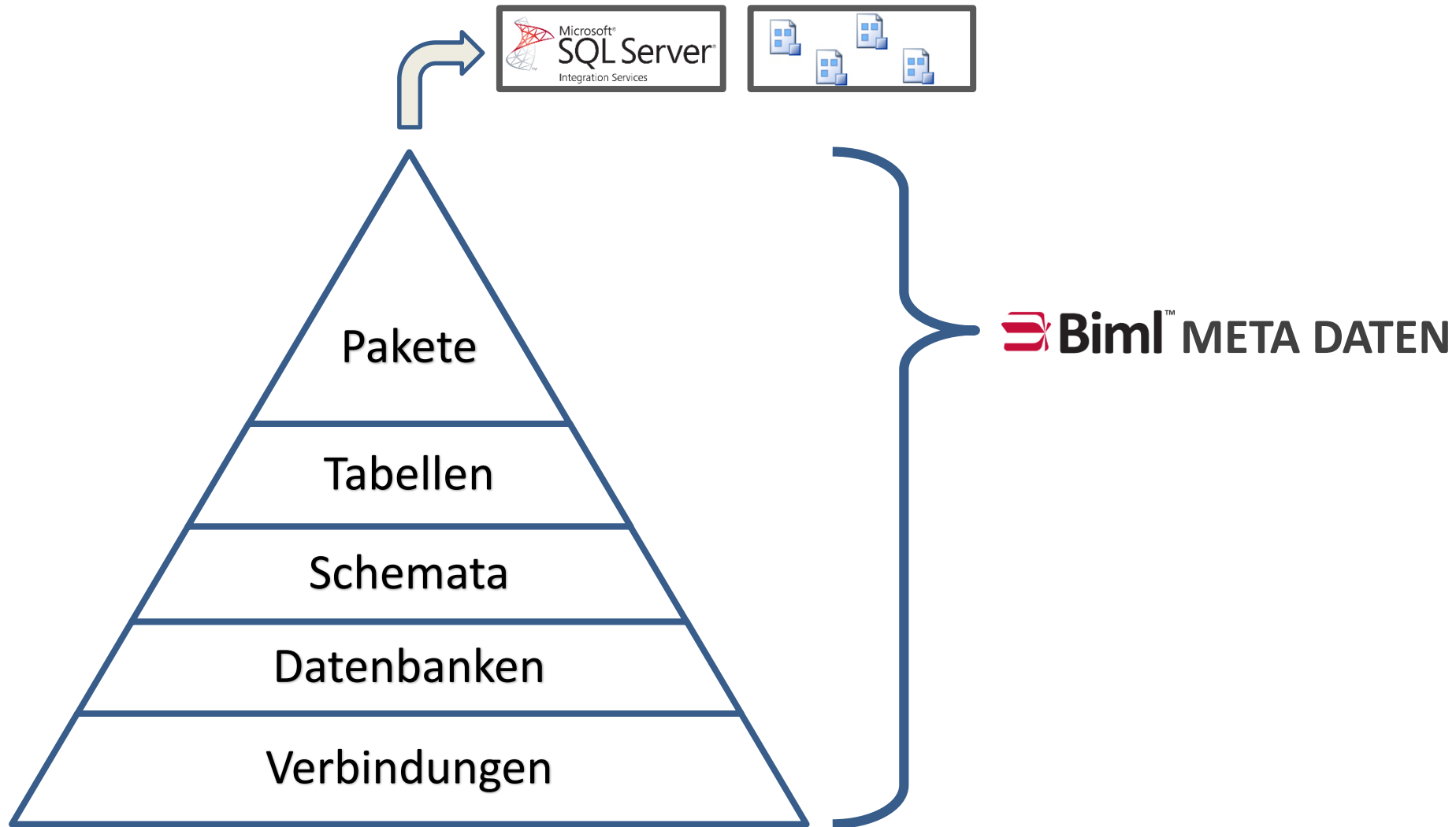
Data Platform

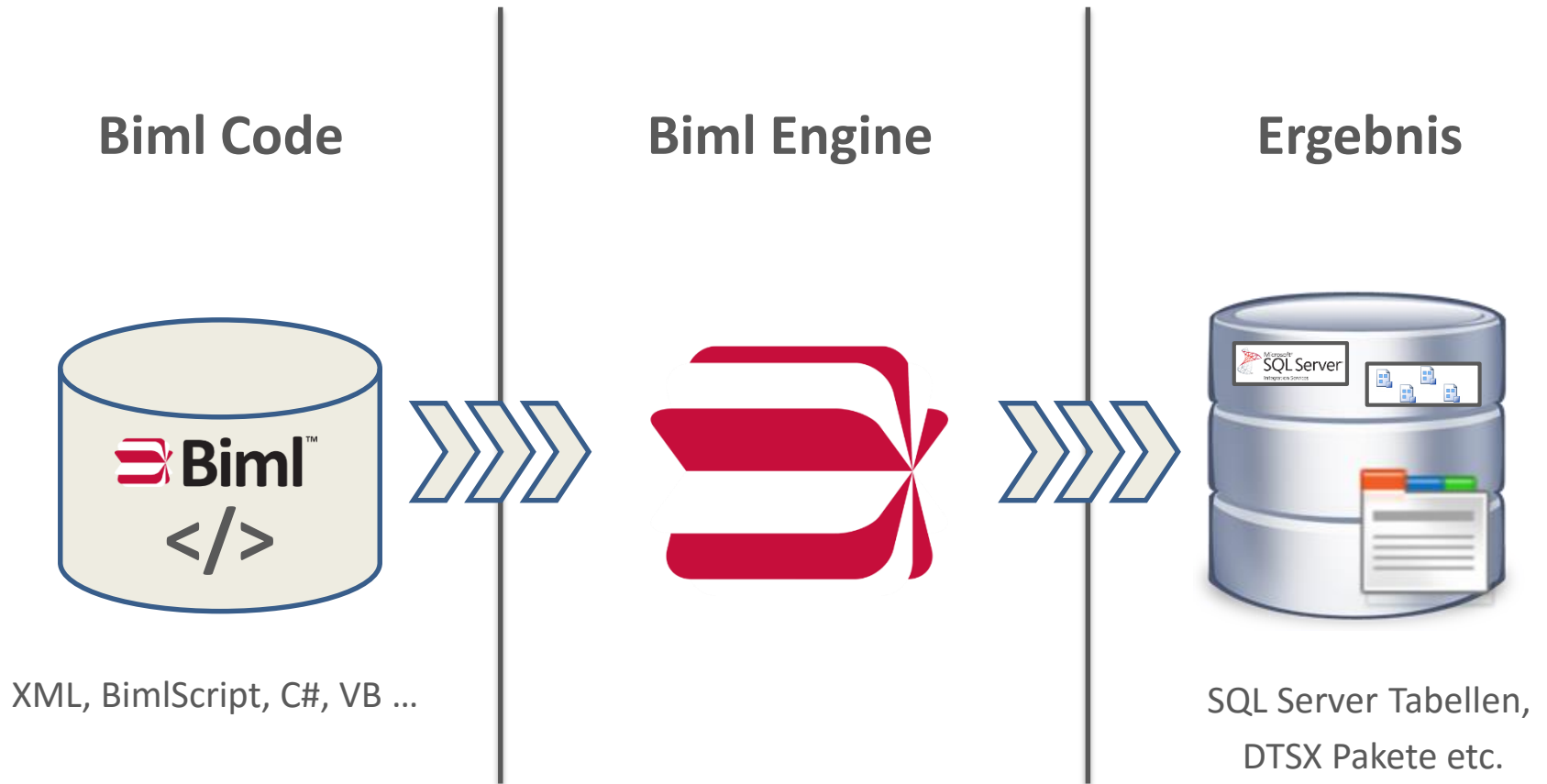


- › Biml ist eine Markup Sprache – also: XML
- › Erfunden/Entwickelt von Varigence
- › So sieht Biml aus:

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">  
  <Packages>  
    <Package Name="HelloBiml"/>  
  </Packages>  
</Biml>
```
- › Wird erst in Verbindung mit BimlScript/APIs mächtig
- › Kann neben SSIS auch für T-SQL und SSAS genutzt werden
- › Verschiedenste Frontends

- › Kompatibilität zwischen SQL Versionen wird belanglos
- › Gesteigerte Produktivität durch Zeitersparnis bei “dummen” Datenbeladungen – Suche nach Mustern!
- › Dreht er jetzt völlig durch? Damit gibt der Berater ja sein Brot und Butter Geschäft auf.
- › Nein – es bleibt nur mehr Zeit für die spannenden Themen!
- › Je besser ein Task standardisierbar ist (z.B. Beladung einer Staging Umgebung), desto höher der Zeitgewinn





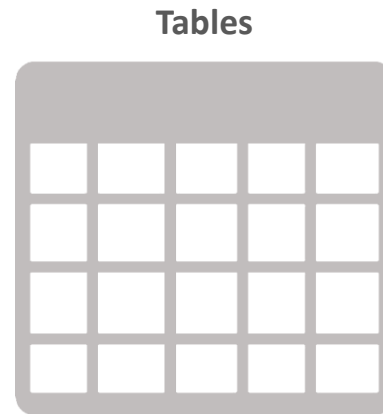


 **BimlOnline** **BI** BIDS Helper **BimlExpress** **BimlFlex** **BimlStudio**

# Demo

- › Auswahl eines Frontends
- › Erstellen einer Zieldatenbank mit entsprechenden Steuertabellen
- › Befüllen der Steuertabellen
  - › Welche Daten sollen von wo geladen werden?
- › Erstellen einer Biml Lösung auf dieser Basis

- › Im einfachsten Fall gibt es nur eine Steuertabelle:



Hier definieren wir einfach alle Tabellen (per Name), die wir importieren wollen

# Demo

01\_Environment.biml

Verbindungsdaten  
(statisch)

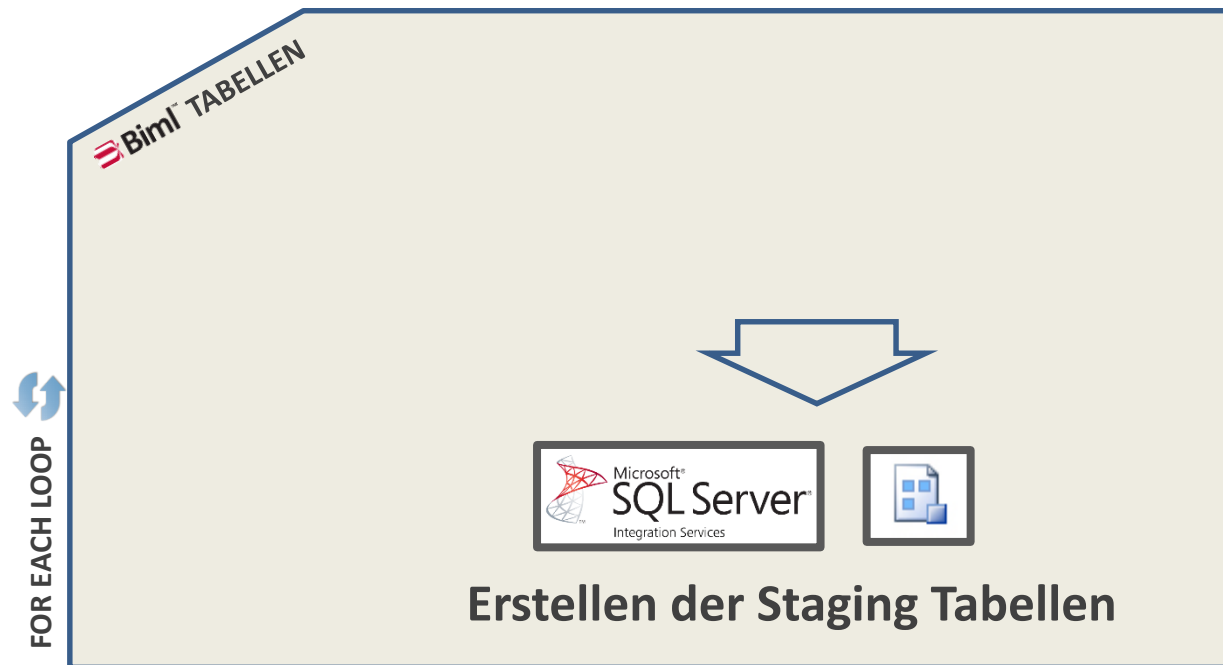
 **Biml™** META DATEN

```
1 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
2   <Connections>
3     <OleDbConnection Name="Target" ConnectionString="Provider=SQLNCLI11;Server=NB-BWEISS
4     <OleDbConnection Name="AdventureWorks" ConnectionString="Provider=SQLNCLI11;Server=N
5   </Connections>
6   <Databases>
7     <Database Name="MySimpleBiml_Destination" ConnectionName="Target" />
8   </Databases>
9   <Schemas>
10    <Schema Name="dbo" DatabaseName="MySimpleBiml_Destination" />
11  </Schemas>
12 </Biml>
```



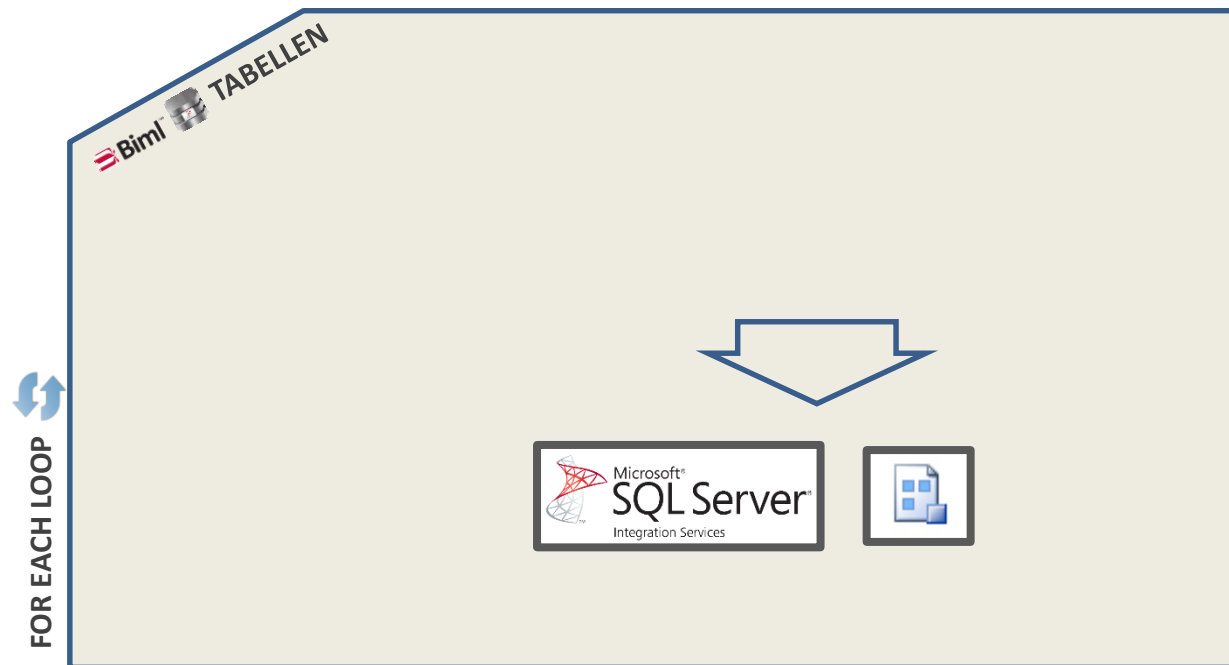


```
1 <#@ template language="VB" tier="2" #>
2 <#@ code file="../code/BimlFunctions.vb" #>
3 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
4   <Tables>
5     <# dim AW as astdbconnectionnode = RootNode.Connections("AdventureWorks")
6       dim sql as string = "select [TableName] from [MyBimlMeta_Tables]"
7       dim importResult as Importresults = AW.GetDatabaseSchema(nothing,bimlfunctions.GetNonEmptyList(RootNode.Connections("Target"),sql), _
8         ImportOptions.ExcludeIdentity or ImportOptions.ExcludePrimaryKey or ImportOptions.ExcludeUniqueKey or ImportOptions.ExcludeColumnDefault _
9         or ImportOptions.ExcludeIndex or ImportOptions.ExcludeCheckConstraint or ImportOptions.ExcludeForeignKey)
10      for each table as asttablenode in importResult.TableNodes #>
11        <Table Name="AW_<#=table.Schema.Name#>_<#=table.Name#>" SchemaName="MySimpleBiml_Destination.dbo">
12          <Columns>
13            <#= table.columns.getbiml #>
14          </Columns>
15          <Annotations>
16            <Annotation AnnotationType="Tag" Tag="SourceSchemaQualifiedName">
17              <#=table.SchemaQualifiedNames#>
18            </Annotation>
19          </Annotations>
20        </Table>
21        <# next #>
22      </Tables>
23 </Biml>
```



```
1  <#@ template tier="3" language="VB" #>
2  <Biml xmlns="http://schemas.varigence.com/biml.xsd">
3  <Packages>
4  <Package Name="01_CreateStaging" ConstraintMode="Parallel" PackageSubpath="MySimpleBiml">
5  <Tasks>
6  <# for each table as asttablenode in RootNode.Tables #>
7  <ExecuteSQL Name="Create <#=table.Name#>" ConnectionName="Target">
8  <DirectInput>
9  <#=table.GetDropAndCreateDdl()#>
10 </DirectInput>
11 </ExecuteSQL>
12 <# next #>
13 </Tasks>
14 </Package>
15 </Packages>
16 </Biml>
```

04\_PopulateStaging.biml



```

1  <#@ template tier="4" language="VB" #>
2  <Biml xmlns="http://schemas.varigence.com/biml.xsd">
3    <Packages>
4      <Package Name="02_Populate Tables" ConstraintMode="Parallel" ProtectionLevel="EncryptSensitiveWithUserKey" PackageSubpath="MySimpleBiml">
5        <Tasks>
6          <# for each table as asttablenode in RootNode.Tables #>
7            <Container Name="Copy <#=table.Schema.Name#> <#=table.Name#>" ConstraintMode="Linear">
8              <Tasks>
9                <ExecuteSQL Name="Truncate" ConnectionName="Target">
10                  <DirectInput>truncate table <#=table.ScopedName#></DirectInput>
11                </ExecuteSQL>
12                <Dataflow Name="Copy <#=table.Schema.Name#>_<#=table.Name#>">
13                  <Transformations>
14                    <OleDbSource Name="Retrieve Rows" ConnectionName="AdventureWorks">
15                      <DirectInput>SELECT <#=table.GetColumnList()#> FROM <#=table.GetTag("SourceSchemaQualifiedNames")#></DirectInput>
16                    </OleDbSource>
17                    <OleDbDestination Name="Load Rows" ConnectionName="Target">
18                      <TableOutput TableName="<#=table.ScopedName#>" />
19                    </OleDbDestination>
20                  </Transformations>
21                </Dataflow>
22              </Tasks>
23            </Container>
24          <# next #>
25        </Tasks>
26      </Package>
27    </Packages>
28  </Biml>

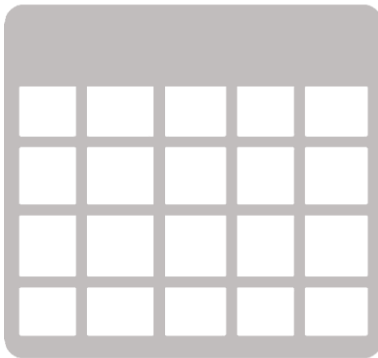
```

# Zeit für eine richtige Demo!

- › Mit nur 1 Tabelle haben wir die Staging Umgebung erstellt und befüllt
- › Sehr überschaubarer Code (< 80 Zeilen XML/BimlScript in 4 Dateien)
- › Für einfache Tasks ist das ggf. schon sehr hilfreich
- › Nicht sehr flexibel und, gerade bei großen Tabellen, viel Overhead

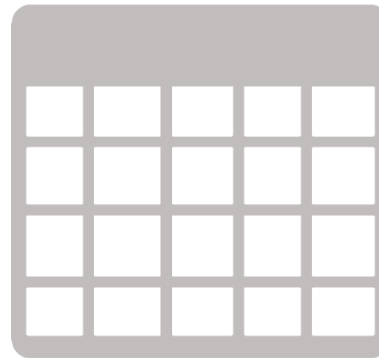
## › Wir brauchen etwas mehr Steuerinformationen

### Connections



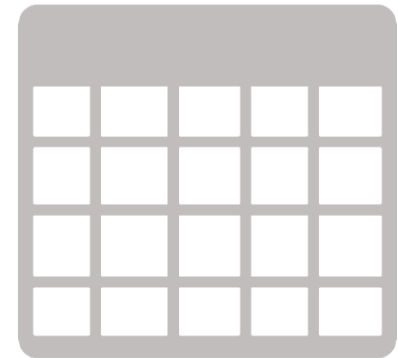
Aus welchen Quellen  
beziehen wir Daten?

### Tables



Aus welchen Tabellen  
beziehen wir welche  
Spalten?

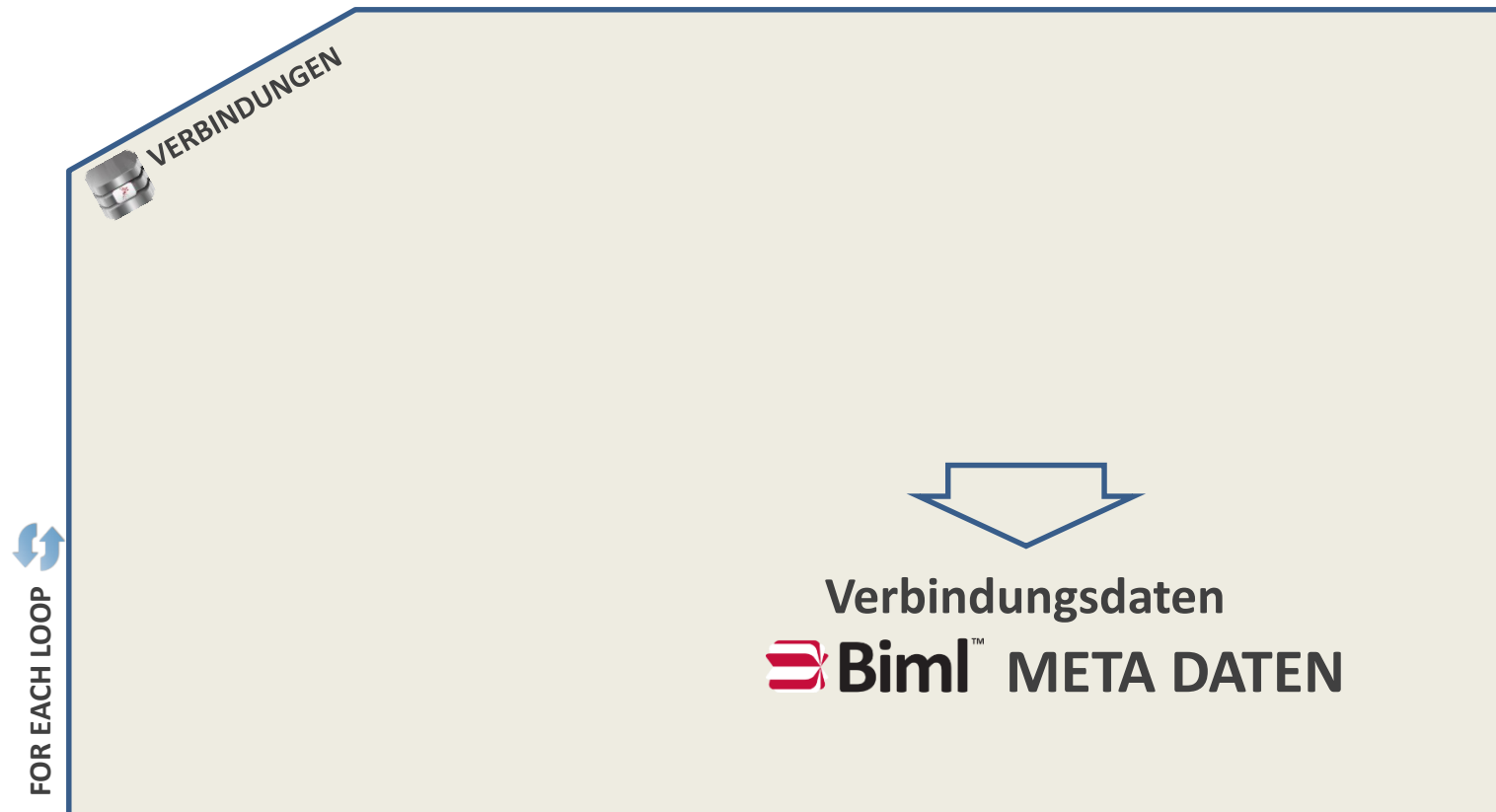
### Packages

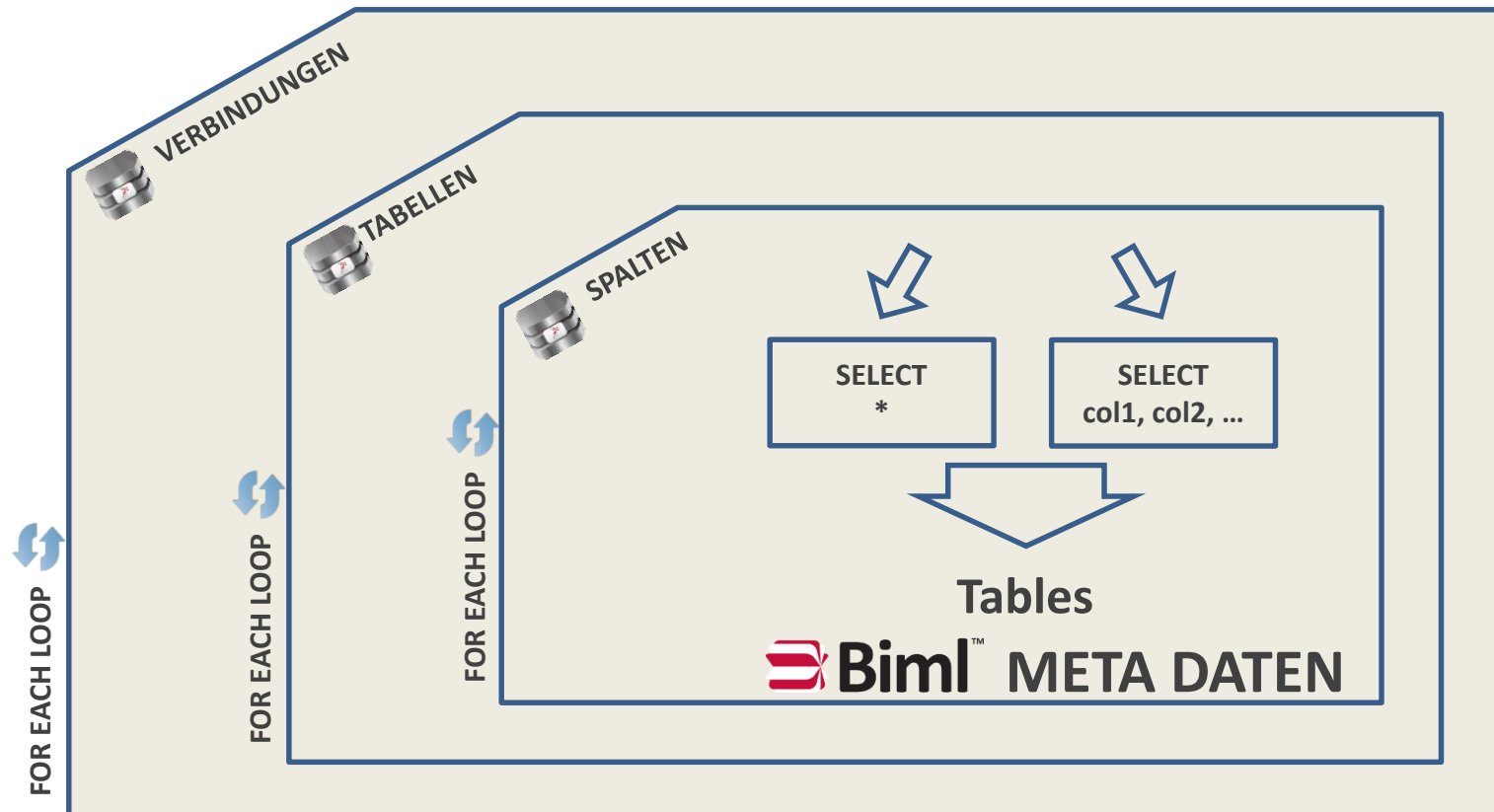


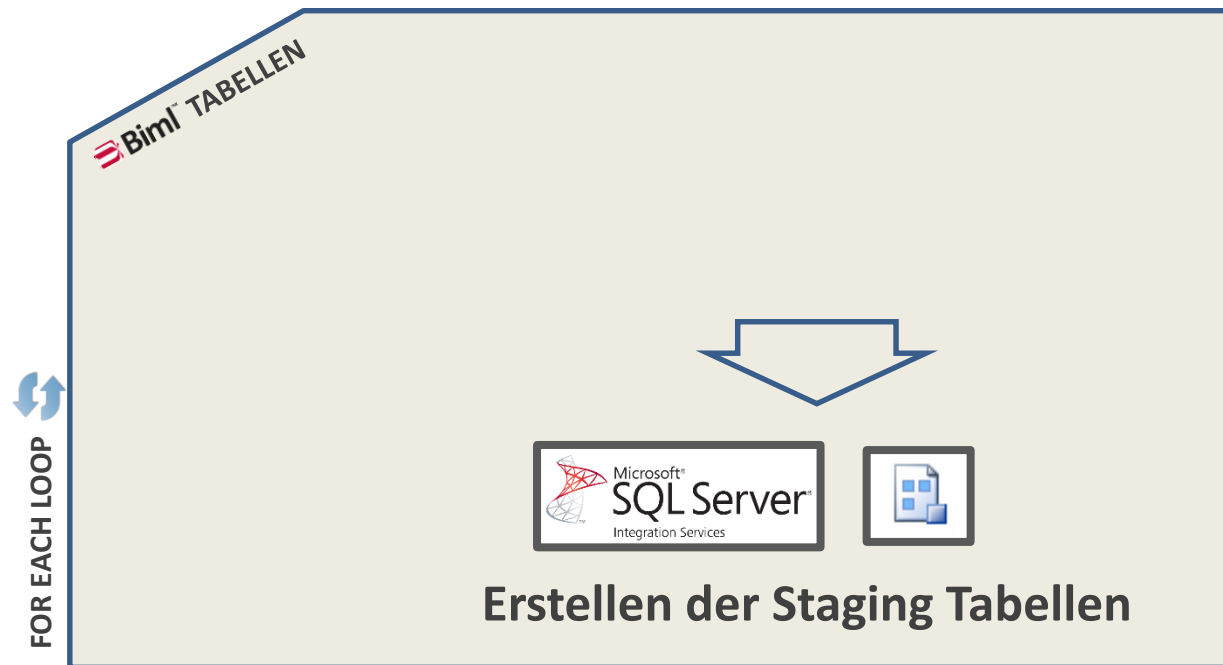
In welchen Unterpaketen  
wollen wir die Tabellen  
aufteilen?



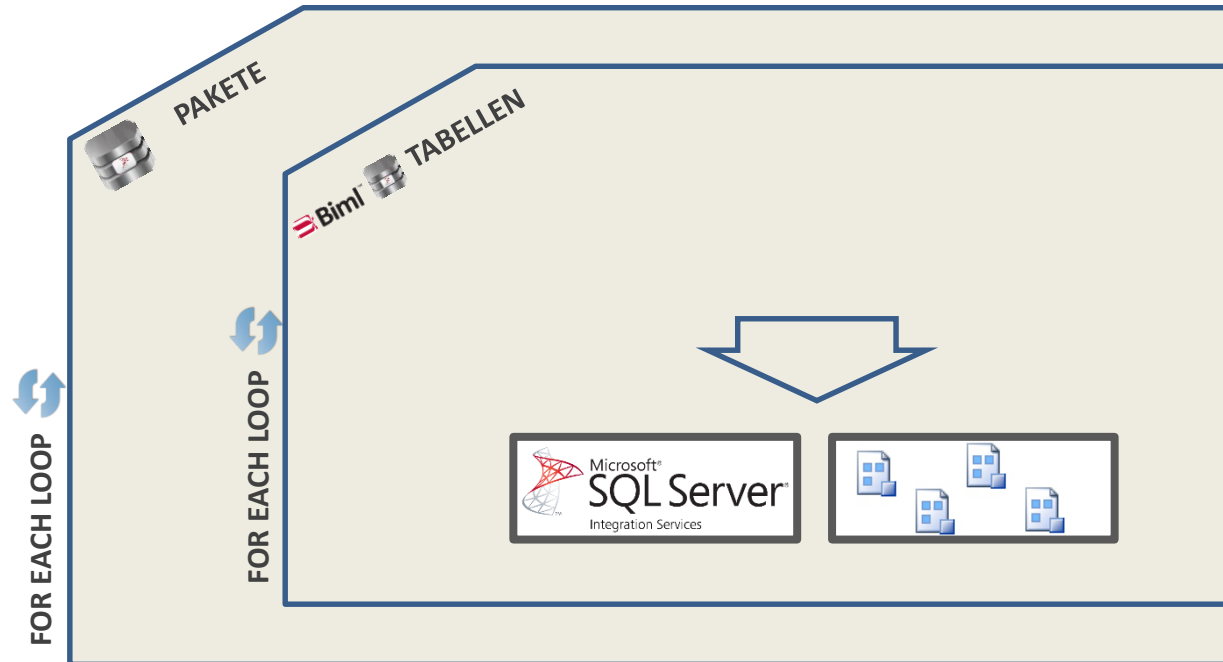
01\_Environment.biml







04\_PopulateStaging.biml / 05\_PopulateTable.biml



**Nun kommt die noch  
viel coolere Demo!**

- › Index management – nicht einfach nur „drop and recreate“
- › „Housekeeping“ – Automatisches löschen von nicht mehr benötigten Tabellen
- › Inkrementelle Beladung / Erstellung
- › Timestamps zu Dataloads hinzufügen

```
<DerivedColumns Name="TS">
```

```
  <Columns>
```

```
    <Column Name="LoadDate" DataType="DateTime">@[System::StartTime]</Column>
```

```
  </Columns>
```

```
</DerivedColumns>
```

- › Includes, Reference Scripts etc. nutzen um Redundanzen zu minimieren und Lesbarkeit zu erhöhen
- › Reverse engineering von bestehenden Warehouses
- › ...

**...kommt jetzt die  
coolste Demo!**

- › Biml Tutorials
  - › <https://www.bimlscript.com/>
  
- › Stairway to Biml
  - › <http://www.sqlservercentral.com/stairway/100550/>
  
- › Cathrine Wilhelmsen
  - › <http://www.cathrinewilhelmsen.net/biml/>
  
- › Solisyon Biml Blog
  - › <http://Biml-Blog.de>



# Fragen?

Gerne auch im Nachgang  
per Mail ([b.weissman@solisyon.de](mailto:b.weissman@solisyon.de))

oder

Twitter (@bweissman)

Vielen Dank!

[www.solisyon.de](http://www.solisyon.de)

