

# De administratie staat in een dag

AAN DE SLAG MET WINDOWS AZURE IN BELANG VAN DE KLANT

Aarjan Meirink, Antoon Tuijl, Dave Ruijter, Erik Proost, Floor Verheul en Robert de Ruiter

De boodschap van de cloudwereld lijkt wel heel erg cloudy, maar het is bepaald geen rocketscience. Als je met de Microsoft technologie overweg kunt, zul je weinig moeite hebben met de technieken die voor Windows Azure worden gebruikt. Maar het is even wennen. Bij Macaw Business Solutions deden ze dat onlangs in de vorm van een Rapid Development-dag, waarbij alle medewerkers werden betrokken.

**De RAD-race begint** met een algemene instructie over SI, CRM en BI. Vervolgens wordt uiteengezet wat de 'klant' verwacht en kan het modelleren- en programmerenwerk beginnen. De opdracht luidt: Zet een administratie op voor de interne beheer afdeling, waarbij per medewerker wordt bijgehouden wie welke auto, laptop(s) en telefoon(s) heeft, inclusief aanvullende informatie zoals soort gebruiker, basisvergoeding, serienummers, softwareversies, etc.

Op de checklist voor de jurering staan zaken als: Hoe zijn de teams omgegaan met projectmanagement? Zijn er momenten op de dag gepland om de voortgang door te nemen? Is er afgestemd met de business owner? Is er een rolverdeling gemaakt? Deze algemene aspecten worden aan het eind van de dag meegenomen in de beoordeling van de jury.

Daarnaast worden de verschillende technieken beoordeeld. Voor BI wordt gekeken naar aspecten als: Is gebruik gemaakt van een snapshot oplossing? Is de basisoplossing zelf gemaakt of gekocht? Zijn niet-BI'ers betrokken geweest bij het ontwikkelen van de oplossing? Zijn de gevraagde dashboards opgeleverd? Is er een alternatieve oplossing?

Op het vlak van System Integration wordt gekeken naar: Is gebruik gemaakt van Azure? Is de basisoplossing zelf gemaakt of gekocht? Is er documentatie van de oplossing? En voor CRM: Is er naast de gevraagde entiteiten verfraaiing aangebracht? Is de basisoplossing zelf gemaakt of gekocht? Is er voor de entiteit contactpersonen een apart formulier voor assetmanagement gemaakt?

## Oplossing

Het winnende team heeft een oplossing die bestaat uit de volgende onderdelen, opgesplitst per discipline:

### ♦ System Integration

- Koppeling van on-premise data met Microsoft Dynamics CRM Online
- Koppeling van Microsoft Dynamics CRM Online data met ODS (Operational Data Store)

### ♦ Customer Relationship Management

- Microsoft Dynamics CRM Online omgeving, met custom entiteiten voor Laptops, Telefoons, Auto's en Brandstofverbruik (dit laatste is een toevoeging van het team)
- Importeren van Excel documenten met gegevens over laptops, telefoons, auto's en het brandstofverbruik
- Ook door ons team toegevoegd: Operational BI Dashboard in CRM
- ♦ **Business Intelligence**
  - Operational Data Store in Microsoft SQL Azure Database
  - Analyse-omgeving in PowerPivot for Excel
  - Rapportage in Microsoft SQL Azure Reporting

**System Integration** is de linking pin in een hybride oplossing met on-premise en cloud data. De System Integration oplossing bestaat uit twee onderdelen, het uitlezen van het on-premise systeem - waar de gegevens van de medewerkers zijn opgeslagen



DE SNELSTE ONTWIKKELAARS OP AZURE VAN MACAW BUSINESS SOLUTIONS: AARJAN MEIRINK, ANTOON TUIJL, DAVE RUIJTER, ERIK PROOST EN FLOOR VERHEUL.

- naar CRM, en het vullen van het ODS vanuit CRM. Het uitlezen van de gegevens uit het on-premise systeem gebeurt via een gateway, deze bevat methodes om te kunnen inloggen en gegevens op te halen. Vervolgens moeten deze gegevens naar CRM worden geschreven, dit gebeurt via een CRM gateway. Om gegevens naar CRM te kunnen exporteren, moet er een vertaling worden gemaakt tussen de entiteiten die in de SI oplossing gehanteerd worden en de entiteiten die in het CRM systeem zijn aangemaakt. Er zijn entiteiten aangemaakt voor o.a. medewerkers, telefoons, auto's, brandstofverbruik en laptops. Vervolgens moeten er in de CRM gateway functies worden aangemaakt die de aangemaakte entiteiten vertalen naar de entiteiten zoals deze in CRM gedefinieerd zijn. Hierbij wordt een "Employee" entiteit welke uit het on-premise systeem komt, gemapt naar een "Contact" entiteit in CRM.

Als voorbeeld kunnen we het uitlezen van werknemers gebruiken, we lezen eerst de gegevens uit het on-premise systeem en vullen een lijst met Employees met deze gegevens, zie onderstaand code-fragment:

```
public List<Employee> GetEmployees()
{
    var retVal = new List<Employee>();
    var thisConnection = CreateConnection();
    SqlDataReader thisReader = ExecuteRead(thisConnection,
    "SELECT [WorkgroupName], [FirstName], [LastName], [MiddleName],
    [UserDefinedResourceId] FROM [Changepoint].[dbo].[DS_Resource]");
    while (thisReader.Read())
    {
        retVal.Add(new Employee
        {
            //new _medewerkernummer

            EmployeeID = thisReader["UserDefinedResourceId"].ToString(),

            FirstName = thisReader["FirstName"].ToString(),

            LastName = thisReader["LastName"].ToString(),

            SolutionCenter = thisReader["WorkgroupName"].ToString()
        });
    }
    thisReader.Close();
    thisConnection.Close();
    return retVal;
}
```

Vervolgens wordt deze lijst doorlopen en per Employee wordt er een vertaling gemaakt naar de Contact entiteit in CRM en het contact wordt opgeslagen in CRM, zie onderstaand code-fragment:

```
public Guid CreateNewContact(string firstname, string
lastname, int employeenumber, string solutionCenter)
{
    var contact = new Entity(ContactColumns.EntityName);
    contact[ContactColumns.Firstname] = firstname;
    contact[ContactColumns.Lastname] = lastname;
    contact[ContactColumns.Medewerkernummer] =
employeenumber;
    contact[ContactColumns.Solutioncenter] =
solutionCenter;
    return Service.Create(contact);
}
```

Op dit moment hebben we de medewerkers uit het on-premise systeem in het CRM systeem staan en zijn de gegevens daarmee in de cloud. De volgende System Integration taak voor het team is het vullen

van de Operational Data Store met de medewerkegegevens uit CRM, incl. laptops, telefoons, auto's en het brandstofverbruik. Dit wordt gedaan door een gateway te maken voor de Operational Data Store (ODS), deze gateway bevat functies om entiteiten in het ODS toe te voegen. Daarbij moet rekening gehouden worden dat er ook logica bij moet zitten om te controleren of de entiteit reeds aanwezig is, om duplicaten te voorkomen. Aangezien het ODS een SQL Azure database betreft, is gebruik gemaakt van de functionaliteit uit de System.Data.SqlClient namespace in het .NET Framework.

In het voorbeeld wordt het inlezen van medewerkers naar het ODS gebruikt. In de eerste plaats moeten de medewerkers uit het CRM systeem worden gelezen, dit gebeurt met de CRM gateway, deze bevat een functie om de Contact entiteiten uit te lezen, zie onderstaand code fragment:

```
public List<Contact> GetContacts()
{
    RetrieveMultipleRequest request =
    RequestMessageCreator.CreateContactRequest();
    var response =
    (RetrieveMultipleResponse)Service.Execute(request);
    return response.EntityCollection.Entities.Select(
    entity =>
    new Contact
    {
        Employeeenumber = EntityAttributes.GetIntFromEntity(entity,
        ContactColumns.Medewerkernummer),

        Firstname = EntityAttributes.GetStringValueFromEntity(entity,
        ContactColumns.Firstname),

        Lastname = EntityAttributes.GetStringValueFromEntity(entity,
        ContactColumns.Lastname),

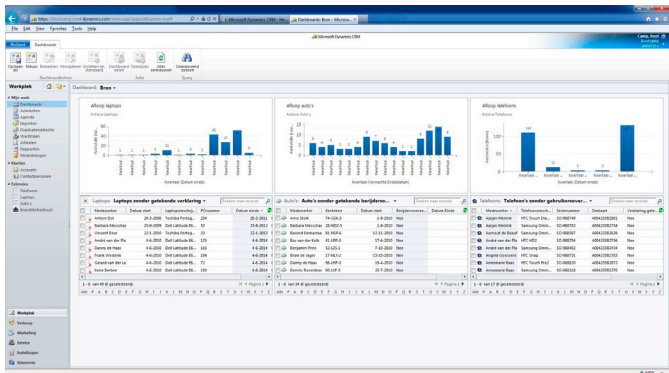
        Id = EntityAttributes.GetGuidValueFromEntity(entity,
        ContactColumns.Id),

        SolutionCenter =
        EntityAttributes.GetStringValueFromEntity(entity,
        ContactColumns.Solutioncenter)
    }).ToList();
}
```

Nu we de CRM Contacten hebben, kunnen we per contact deze naar het ODS exporteren, wat we doen via een Insert functie in de ODS gateway, deze insert functie gaat eerst na of de medewerker bestaat in het ODS. De functie LookupEmployee gaat na of er reeds een medewerker bestaat met het gegeven Employeeenumber, deze krijgt de GUID van de bestaande medewerker terug indien de medewerker bestaat in het ODS, zie onderstaand code fragment:

```
public void InsertEmployee(Contact contact)
{
    SqlConnection connection = null;
    try
    {
        // Controleer voor bestaande Medewerker in ODS
        Guid medewerkerId =
        LookupEmployee(contact.Employeeenumber);
        if (medewerkerId == Guid.Empty)
        {
            // Medewerker bestaat nog niet.
            connection = CreateConnection();

            ExecuteInsert(connection, "INSERT INTO [Team1].[dbo].[Medewerker]
            ([Naam] ,[Medewerkernummer] ,[SolutionCenter]) VALUES
            (' + contact.Firstname + " " + contact.Lastname + "', ' +
            contact.Employeeenumber + "', ' + contact.SolutionCenter + "')");
        }
    }
}
```



GRAFIEKEN EN TABELLEN GEVEN INZICHT IN HET AANTAL LAPTOPS EN TELEFOONS DAT IN DE KOMENDE PERIODE ZAL AFLOPEN.

```

}
catch (Exception ex)
{
    Trace.WriteLine(ex);
}
finally
{
    if (connection != null) connection.Close();
}
}

```

Nadat de gegevens uit CRM naar het ODS zijn geschreven, zijn we klaar om de gegevens te gebruiken in PowerPivot for Excel en de SQL Azure rapportages.

**CRM: Custom Entities & Operational BI Dashboard**

Een belangrijk onderdeel van de gehele oplossing was de Microsoft Dynamics CRM Online omgeving. Deze bevat standaard al vele entiteiten, waarvan we de Contacts vanuit een on-premise systeem vulden. Daarnaast moest er een aantal custom entities gemaakt worden, om de laptops, telefoons, auto's en brandstofverbruik te kunnen beheeren.

De benodigde entiteiten voor de basisopdracht zijn snel gereed. Het duurde even voordat de medewerkergegevens uit het on-premise systeem geladen konden worden, maar daarna was het mogelijk om de aangeleverde Excel documenten met laptop-, telefoon- en autogegevens te importeren. Deze hadden uiteraard allemaal een relatie met de Medewerker entiteit. Vervolgens kon de custom entiteit voor het brandstofverbruik

aangemaakt worden, waarna hiervan de gegevens vanuit een CSV bestand geïmporteerd konden worden. Na een paar aanpassingen op de data vanuit het bijbehorende formulier was ook dit gereed. Nadat alle entiteiten correct gevuld waren zijn we gestart met het maken van een Operational BI Dashboard, binnen CRM Online. Dit dashboard gaf de klant een idee van de dashboard en rapportage mogelijkheden voor de gebruiker van de CRM omgeving. We kozen ervoor om een aantal grafieken en tabellen te tonen die o.a. inzicht gaf in het aantal laptops en telefoons dat zou aflopen in de komende periode.

**Business Intelligence: Brandstofverbruik, SQL Azure en PowerPivot for Excel**

Als toevoeging op de casus hebben we het brandstofverbruik van iedere medewerker in de oplossing opgenomen. Dit paste mooi bij de entiteit Auto en je kunt hier uitstekende BI analyses op loslaten.

Voor een aantal medewerkers werd een CSV bestand met verbruiksgegevens gemaakt en in CRM geïmporteerd. We hebben aan de klant voorgesteld dat iedere medewerker een dergelijk bestand zelf zou kunnen downloaden van de website van de leasemaatschappij (of zelf zou kunnen aanmaken indien het geen leaseauto betreft), waarna de medewerker deze op zijn of haar MySite in SharePoint plaatst. Vervolgens worden de bestanden automatisch geïmporteerd, waardoor er geen tussenkomst van een stafmedewerker nodig is.

De verbruiksgegevens in CRM Online worden vervolgens doorgekoppeld naar een Operational Data Store, samen met de overige CRM entiteiten, waardoor het - naast het beschikbaar stellen van standaardrapportages - ook mogelijk werd om (ad-hoc) BI analyses op de gegevens uit te voeren.

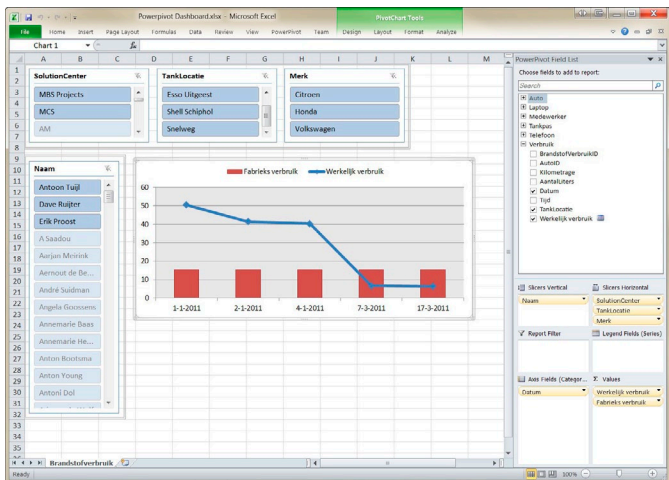
Het ODS wordt gebouwd in een SQL Azure database. Deze wordt gebruikt als bron voor een SQL Azure Reporting rapport en in PowerPivot for Excel. Het inladen van de gegevens in Excel was een koud kunstje, waarna we als voorbeeld een BI analyse op de verbruiksgegevens in beeld brachten, zoals je in onderstaand screenshot kunt zien:

Het verversen van de gegevens is niet noodzakelijk, dus je zou bij wijze van spreken het Excel document in het vliegtuig kunnen meenemen en doorgaan met je analyse!

**Conclusie**

De deelnemers zijn zonder uitzondering enthousiast over het Bootcamp. Zowel vakinhoudelijk als sociaal vinden ze het een prima initiatief.

Enkele teams hebben naast de basisopdracht ook extra functionaliteit kunnen inbouwen, zoals een overzicht van de brandstofkosten en het gemiddelde verbruik per werknemer. In het laatste half uur hebben drie teams kans gezien razendsnel functionaliteit toe te voegen, zowel on-premise als op Windows Azure.



HET BRANDSTOFVERBRUIK VAN DE LEASEAUTO'S IN BEELD GEBRACHT.

