SharePoint 2010 Developer's

Debunking myths for SharePoint Developers

About Mirjam

Microsoft CERTIFIED Master

MVP Microsoft Most Valuable Professional

Macaw

DIWUG

Blog: http://sharepointchick.com

Email: mirjam@macaw.nl

Twitter: @mirjamvanolst

Agenda

- Devs don’t need to know about SQL
- Service Apps is not for me
- SPWebConfigModifications are tha bomb
- Client Development = Server Development
- Upgrade is easy
- Web Templates are for end users

Devs don’t need to know SQL

SQL Server and SharePoint

- SQL is an essential part of a SharePoint environment
- SharePoint can’t be faster than SQL can handle the requests
- SQL installation and maintenance should be high priorities
  - Even on your development environment!

SQL Server and SharePoint

- A messed up SQL install…
- …means SharePoint is slow…
- …which your custom code could take the blame for.
- If you understand SQL…
- …you might be able to pinpoint the problem to something other than your code!

SQL Maintenance

- Don’t shrink
- Database Fragmentation:
  - Defragment the index, not the table
- Reorganize (online) or Rebuild (offline)
- Think about what SQL backup strategy to use
  - Full
  - Incremental
  - Log
- Setting backup to full, but not doing a full backup will cause the log to grow indefinitely

SQL Server while developing

- If a SQL install on your dev environment is bad…you will be waiting!
- A lot!
- Consider your SQL backup strategy
  - What do you need to backup?
  - How often do you need to do a backup?
  - Where do you want to backup that to?

Service Apps are not for me

Service Application model

- Replacing SharePoint 2007 SSPs
- Service Applications can easily be scaled out
- Service Applications can be shared across farms
- Multiple instances of the same Service Application can be deployed
- Web apps can consume services on an individual basis

“Service Applications”

- **Service Instance**
  the implementation of the functionality provided
- **Service Machine Instance**
  the machine or machines in the farm that run the “service”
- **Service Application Endpoint**
  an IIS Virtual Application created within the SharePoint Web Services IIS Web on the machine or machines running the “service”
- **Service Application**
  the logical container of the service, what is exposed through Manage Service Applications
- **Service Connection (aka Service Application Proxy)**
  a virtual link between consumers (e.g. Web Applications) and the service
- **Consumers**
  any component that uses the service

http://www.harbar.net/articles/sp2010sa2.aspx

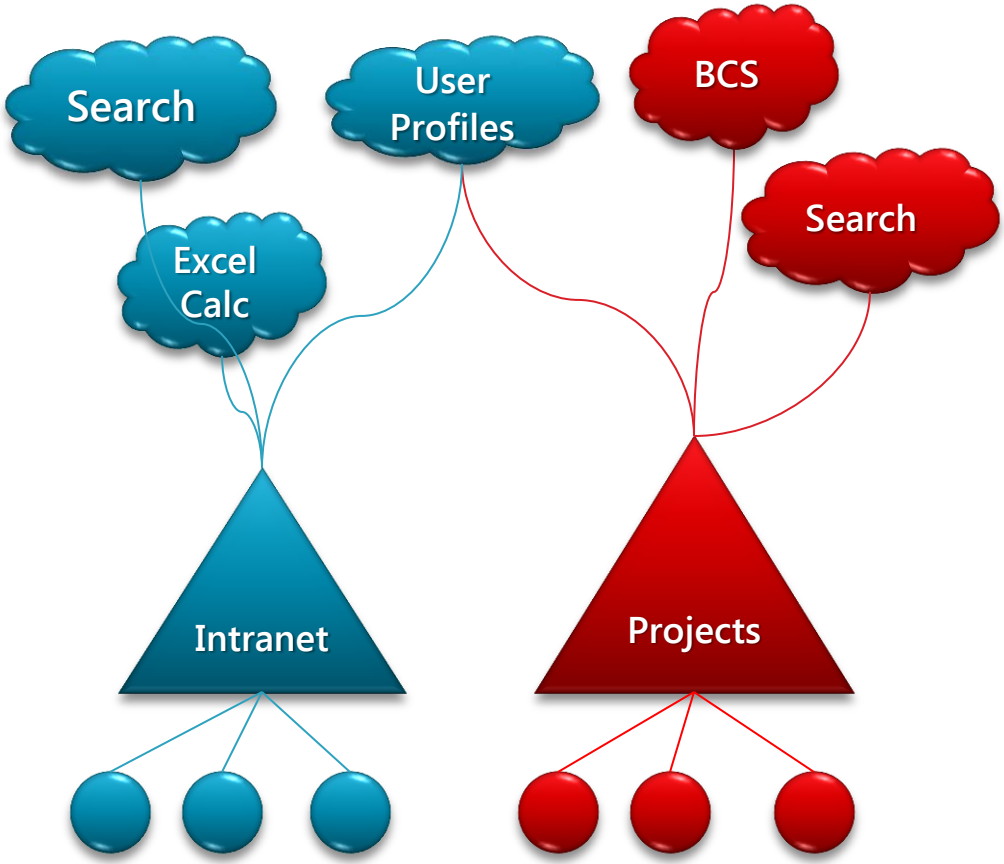
Service Application model

- Service Isolation
  - Each Service Application uses separate database
  - Optionally can use a separate Application Pool
- Multi-Tenancy
  - Some Service Applications can be partitioned to serve multiple tenants
- Simplified Administration Model
  - Central Administration and PowerShell

Proxy Groups

- A proxy group is a group of SA proxies (connections) that are selected for a web app
- By default, all SA proxies are included in the default proxy group
- When you create a web app you can:
  - select the default proxy group
  - create a custom proxy group by selecting which SA proxies should be included
- The custom proxy group for one web app cannot be reused with a different web app

Flexible Deployment



Deploying Service Applications

- Farm Configuration Wizard
  - Creates all Service Applications with default settings
  - Do not use it!
- Manually
  - Go to the Manage Service Application page in CA
  - Creates service apps and their proxies
- For maximum control, use PowerShell
- Don’t create a Service Application you don’t need
- Be aware that for some Service Application you need to start a service as well

Support for custom Service Apps

When to create a custom service application:

- If providing specialized and heavy computations & analytics
- If sharing data across site collections & web applications
- If executing long running operations
- If a robust scale out strategy is required

Support for custom Service Apps

When NOT to create a custom service application:

- If data and / or features are specific to a particular site collection
- If data and / or features are specific to a particular site template

- Most of the time you should build features, not service applications

Demo

SERVICE APPLICATIONS

SPWebConfigModifications are tha bomb

SPWebConfigModifications

- The short story?

- Don’t use it!

SPWebConfigModifications

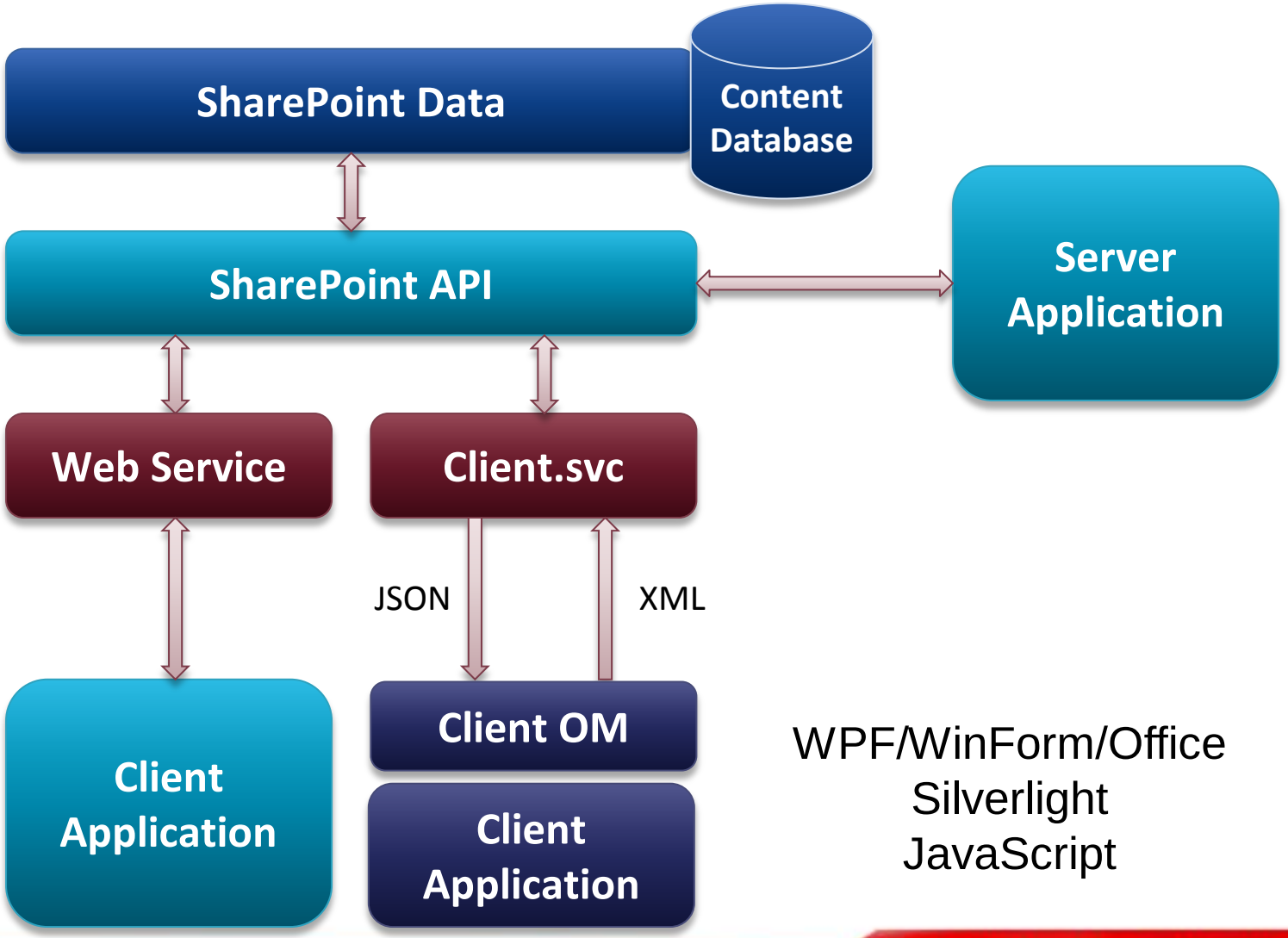
- You can only make changes to the SharePoint element
- You can't influence the order of elements properly
- It’s unreliable if you have a farm with more than one web server

Client Development

=

Server Development

Retrieving data using Client OM



WPF/WinForm/Office
Silverlight
JavaScript

Client OM Unified Clients

- Unified object model across all clients
- Three different APIs:
  - .NET CLR
  - Silverlight CLR
  - JavaScript (ECMA)
- All APIs offer a subset of the server API

Key differences on the client

- Limited API's
- Explicit ExecuteQuery calls
- Only async calls for JavaScript and Silverlight
- Context needs to be established
  - Full url context for .Net and Silverlight
  - Server relative context for Javascript

Client API locations

- .Net
  - [[SharePointRoot]]\ISAPI
- Silverlight
  - [[SharePointRoot]]\TEMPLATE\LAYOUTS\ClientBin
- JavaScript
  - [[SharePointRoot]]\LAYOUTS

Client API files

- .Net
  - Microsoft.SharePoint.Client.dll
  - Microsoft.SharePoint.Client.Runtime.dll
- Silverlight
  - Microsoft.SharePoint.Client.Silverlight.dll
  - Microsoft.SharePoint.Client.Silverlight.Runtime.dll
- Javascript
  - SP.js
  - SP.Core.js
  - SP.Runtime.js

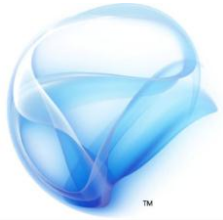
Example – Server API

```
SPWeb site = SPContext.Current.Web;
string title = site.Title;
site.Title = title + " and Client OM";
site.AllowUnsafeUpdates = true;
site.Update();
```

Example – .Net CLR

```
ClientContext clientContext =
    new ClientContext(
        "http://intranet.contoso.com");
Web site = clientContext.Web;
clientContext.Load(site);
clientContext.ExecuteQuery();
site.Title += " and Client OM";
site.Update();
clientContext.ExecuteQuery();
```

Example – Silverlight CLR

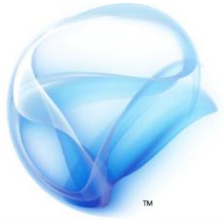


```
Web site;
ClientContext context;

void MainPage_Loaded(object sender,
                     RoutedEventArgs e)
{
    context = new ClientContext(
        "http://intranet.contoso.com");

    site = context.Web;
    context.Load(site);
    //Call the SharePoint Server
    context.ExecuteQueryAsync(
            Succeeded, Failed);
}
```

Example – Silverlight CLR



```
void Succeeded(object sender,
    ClientRequestSucceededEventArgs args)
{
    site.Title = site.Title + " with Client OM";
    site.Update();
    context.ExecuteQueryAsync(
            UpdateSucceeded, Failed);
}

void Failed(object sender,
    ClientRequestFailedEventArgs args)...
void UpdateSucceeded(object sender,
    ClientRequestSucceededEventArgs args)...
```

Example – JavaScript


```
<script language="javascript" type="text/javascript">

    _spBodyOnLoadFuntionNames.push("Initialize");

    var site;
    var context;

    function Initialize()
    {
        context = SP.ClientContext.get_current();
        site = context.get_web();
        context.load(site, "Title");
        //Call the SharePoint Server
        context.executeQueryAsync(
                  Succeeded, Failed);
    }
```

Example – JavaScript


```
function Succeeded(sender,args)
{
    site.set_title(
        site.get_title() + 'from js');
    site.update();
    context.executeQueryAsync(
            Succeeded2, Failed);
}

function Failed(sender, args)
{
    alert('request failed ' +
            args.get_message() + '\n' +
            args.get_stackTrace());
}
```

Demo

USING THE SHAREPOINT CLIENT OBJECT MODELS

Upgrade is easy

Upgrade is easy

- SharePoint feature and solution framework now supports upgrade
- Plan for upgrade of features and solutions beforehand
- Implementation isn’t that easy
- Coming up with the right strategy is even more difficult!

Upgrading Features

- Feature Version Attribute
- ActivationDependency - MinimumVersion attribute
- Declarative feature upgrade elements
- FeatureUpgrading event
  - New Feature Receiver event
- Object Model Changes
  - SPSite.QueryFeatures method
  - Upgrade Method

Upgrade a feature

When a new version of a Feature is created, this may involve:

- Incrementing the version number of an existing Feature (this is mandatory for Feature upgrade to happen)
- Adding some new XML to define new items for the Feature
- Writing code to execute in the FeatureUpgrading event in the Feature receiver

Feature Upgrade XML

```
<Feature xmlns="http://schemas.microsoft.com/sharepoint/" Version="1.0.0.0">
  <UpgradeActions>
    <VersionRange BeginVersion="0.0.0.0" EndVersion="1.0.0.0">
      <ApplyElementManifests>
        <ElementManifest Location="MyUpgrade\Elements.xml" />
      </ApplyElementManifests>

      <AddContentTypeField ContentTypeId="0x010073f25e2ac37846bb8e884770fb7307c7"
          FieldId="{536DC46C-DC26-4DB0-A97C-7C21E4362A85}" PushDown="TRUE"/>

      <CustomUpgradeAction Name="RunSomeUpgradeCode">
        <Parameters>
          <Parameter Name="LibraryToUpdate">My Shared Documents</Parameter>
        </Parameters>
      </CustomUpgradeAction>
    </VersionRange>
</Feature>
```

Upgrade a farm solution

- Stsadm –o upgrade solution –name solution.wsp –filename solution.wsp
- Farm wide:
  - Psconfig –cmd upgrade –inplace b2b
  - Run SharePoint configuration wizard
- More granular:
  - Feature.Upgrade()
  - Possibly use QueryFeatures() method
  - Upgrade sites one by one

Upgrade a sandboxed solution

- Upload new .wsp file
  - New file name
  - Same solution Id
- Press the Upgrade button in the solution gallery to upgrade the solution
- The new solution and features will be activated
- The old solution will be deactivated

Be aware!

- Feature upgrade does NOT happen automatically for Farm Solutions!
- Upgrade behavior for farm solutions is different than for sandboxed solutions
- On the VersionRange element, BeginVersion is inclusive but EndVersion is not
  - A Feature instance will be upgraded if the current version number is equal to or greater than BeginVersion , and less than EndVersion

Web Templates are for end users

Web Templates

- The new Site Templates
- “Save site as template” creates a .wsp file
- No more .stp files for sites!
- Site definitions and web templates look the same to end users

Web Templates for developers

- You can import a saved site into Visual Studio 2010
- You can also create web templates from scratch in Visual Studio
- A web template inherits from an existing site definition
- No need for custom site definitions
  - This means easier upgrades!

Creating Web Templates

- Feature Framework extended with WebTemplate element
- Can be deployed at Site and Farm scope
- Build using an elements.xml file and an onet.xml
  - No webtemp.xml file!


```
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
    <WebTemplate
            Name="ContosoProjectSiteWebTemplate"
            BaseTemplateName="STS"
            BaseTemplateID="1"
            BaseConfigurationID="0"
            Title="Contoso Project Site"
            Description="My Test Site Template"
            ImageUrl="/_layouts/images/contosointranet/ProjectSite.PNG"
            DisplayCategory="Contoso Collaboration"/>
</Elements>
```

Creating Web Templates

- Can be called from code by using the feature GUID and web template name
  - [GUID]#Name

```
Guid myFeatureId = new Guid("47f055d2-3c9b-458d-813a-553dd8d95076");

//format the web template name like so -> "{featureGuid}#Web Template Name
string myTemplateName = String.Format("{{{0}}}#My Web Template", myFeatureId.ToString());

using (SPWeb newWeb = site.RootWeb)
{
    newWeb.ApplyWebTemplate(myTemplateName);
}
```

Web Template limitations

- Module element
  - Use features to provision your (default.aspx) pages
- Components
  - FileDialogPostProcessor –class used to modify the file open and save dialogs on a document library
  - ExternalSecurityProvider –interface that returns custom info about the security used in SharePoint Foundation for indexing by a search crawler
- ServerEmailFooter
  - Defines footer section for server e-mail
- Feature Stapling

Demo

WEB TEMPLATES

SharePoint 2010 Developer's

Please be sure to fill out your session evaluation!