

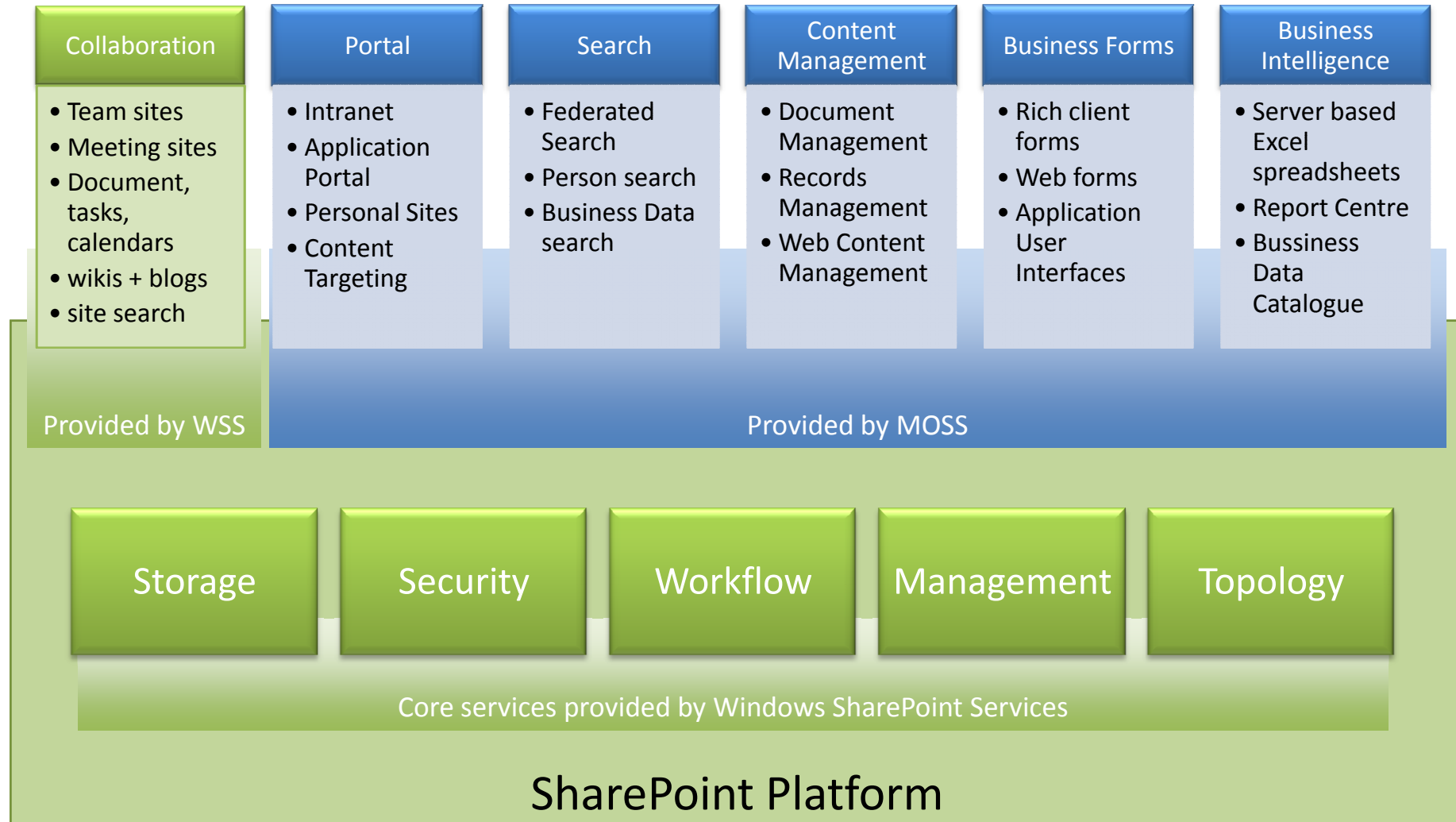
SharePoint Content Types

The backbone of good information management

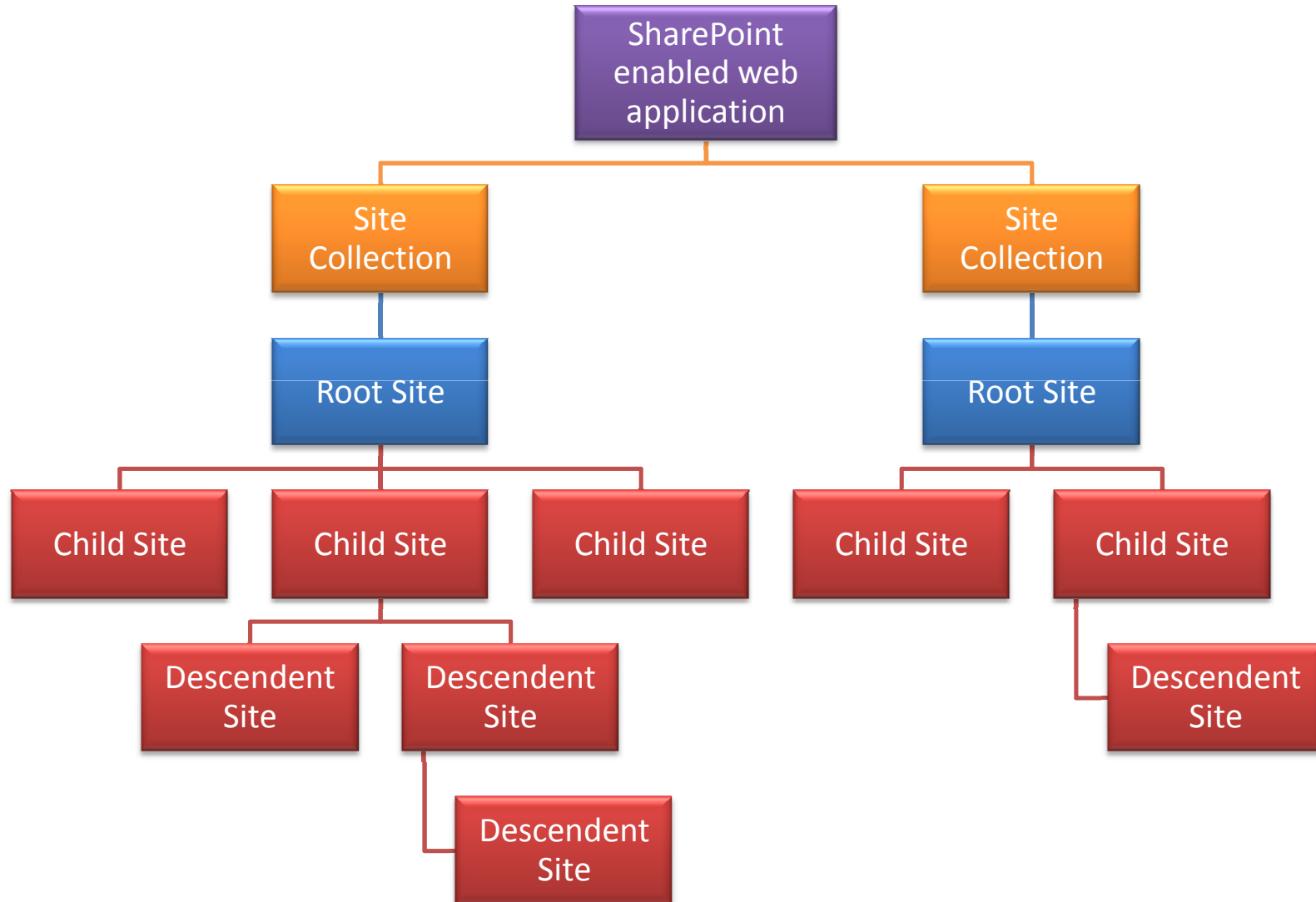
Mark Orange
Solution Consultant
Intergen



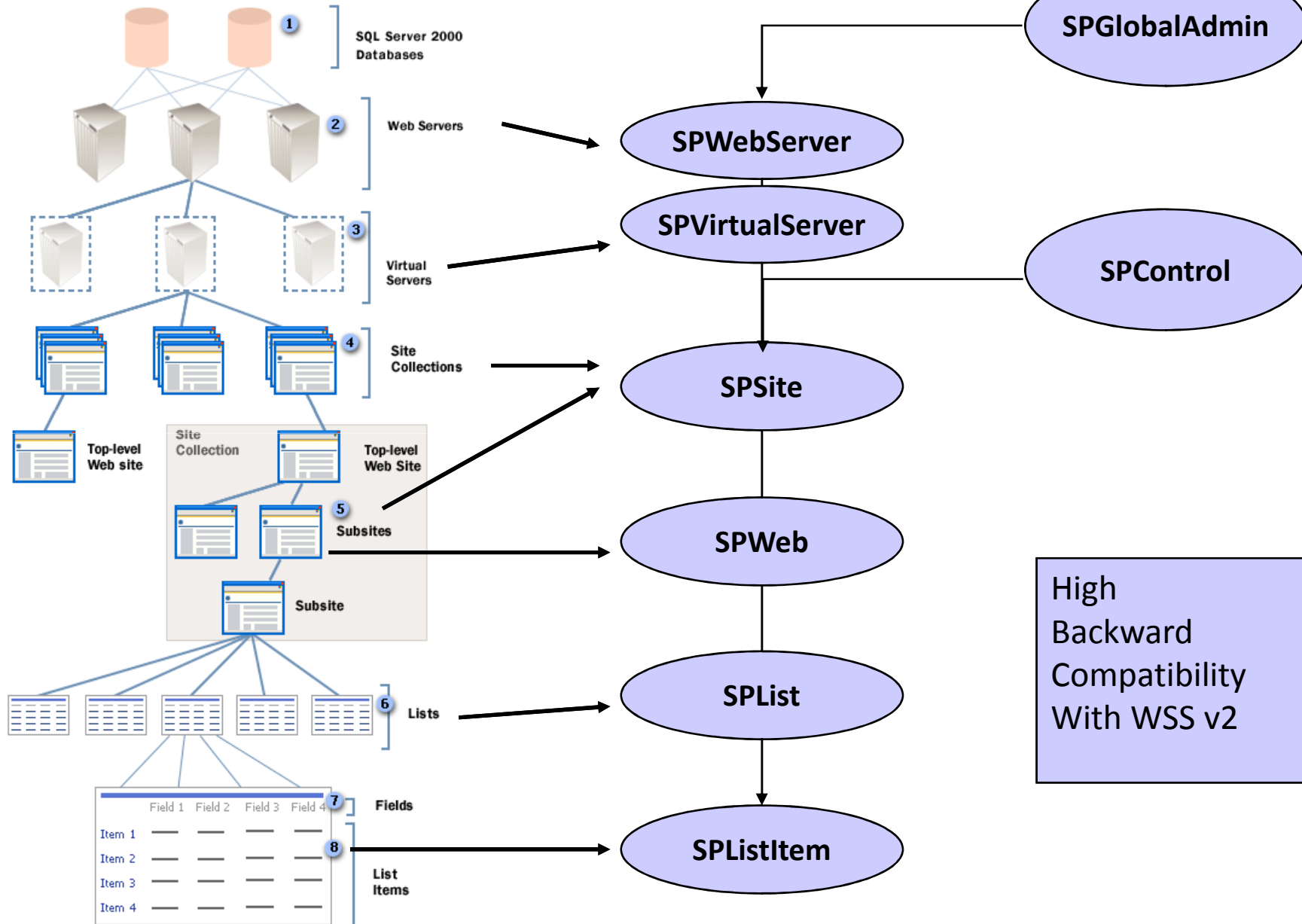
Solutions built on SharePoint Platform



Sites Architecture



Server and Site Architecture



Tools of the trade

- WSS SDK
- MOSS SDK

- Visual Studio 2005
- VSeWSS, version 1.1
(Visual Studio extensions for Windows SharePoint Services)

- Visual Studio 2008
- VSeWSS, version 1.2

VSeWSS

Project Templates

- Team Site
- Blank Site
- List Definition
- Web Part
- Empty

VSeWSS

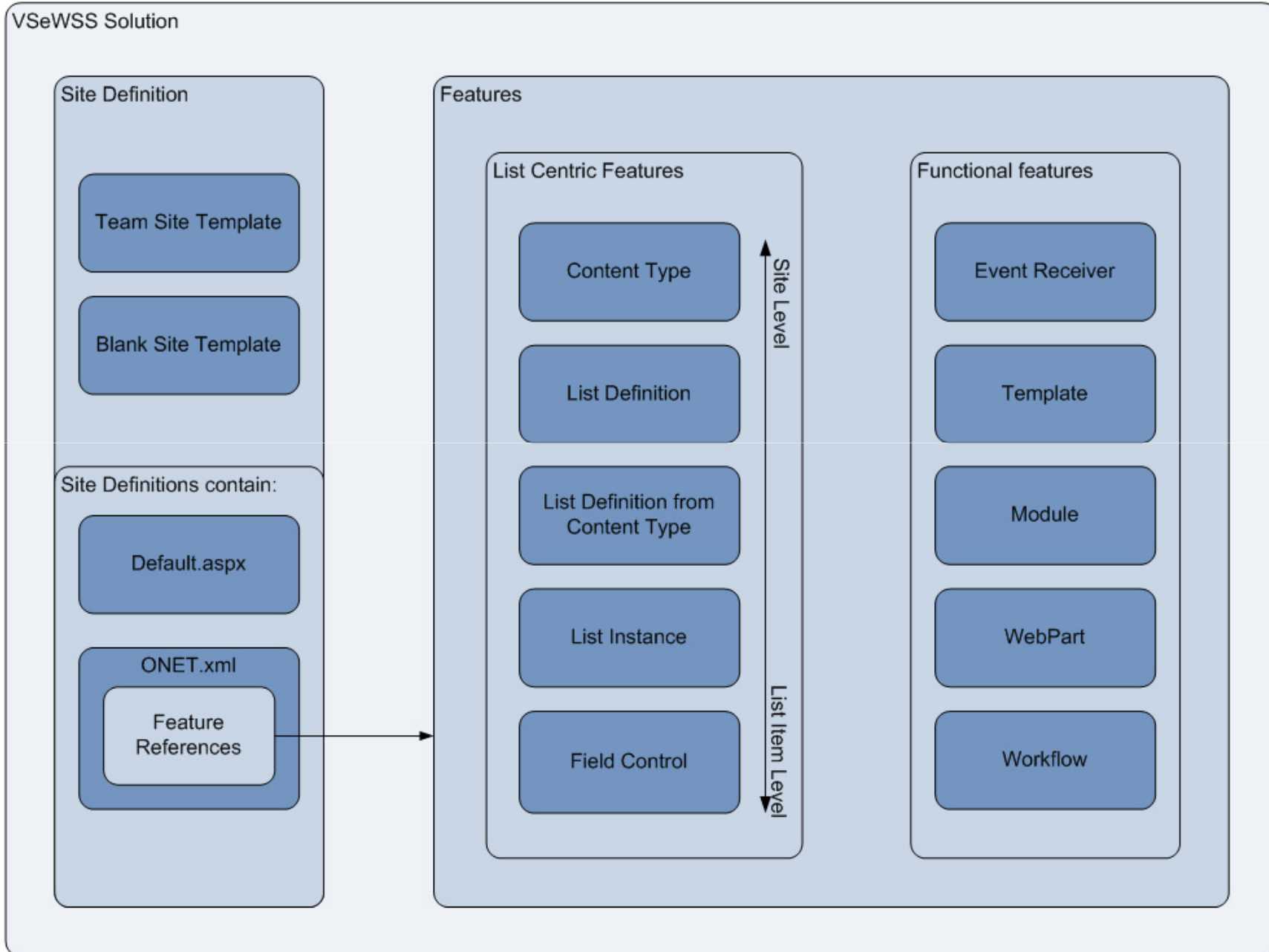
Item Templates

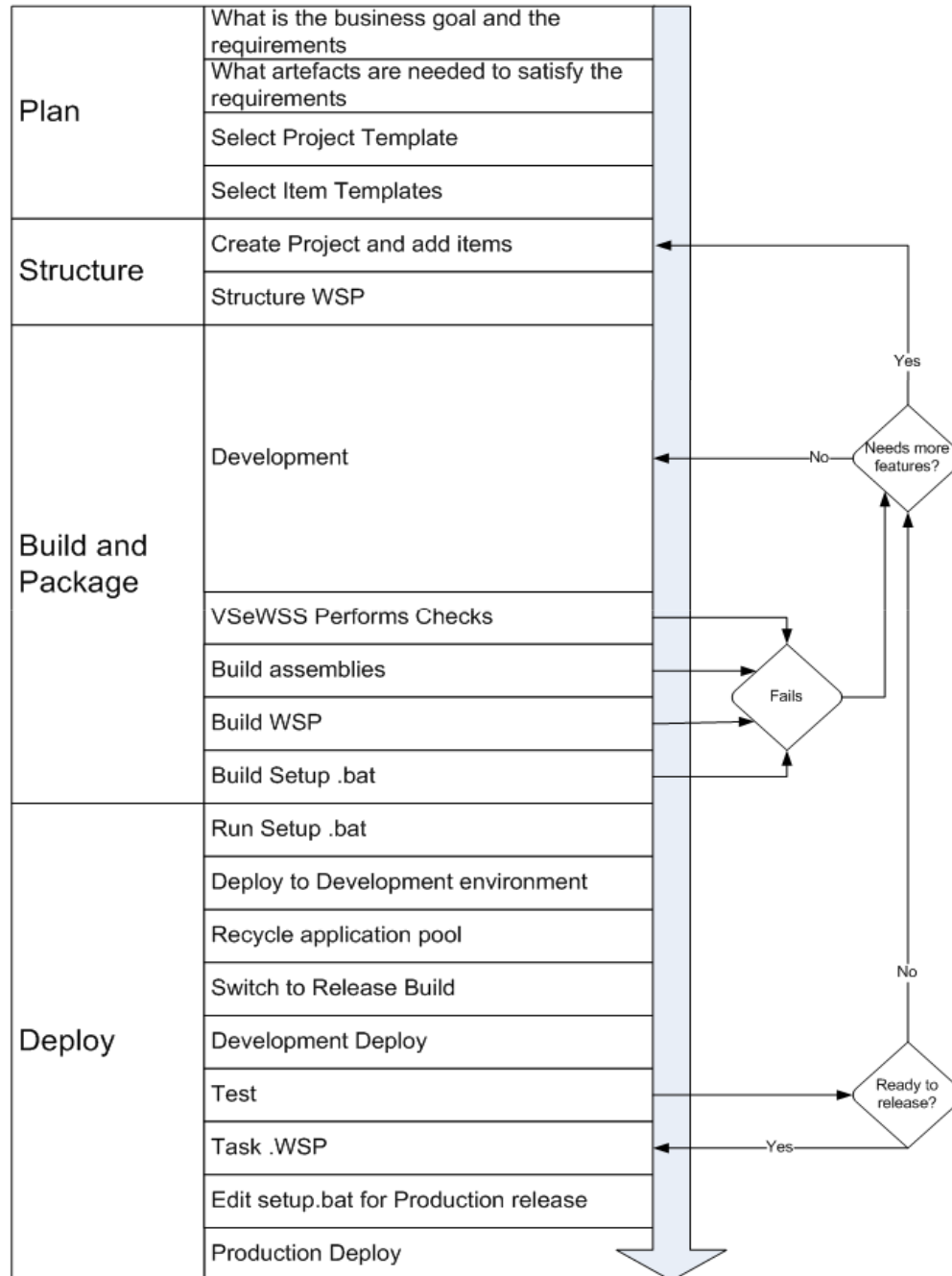
- List Definition
- List Definition from Content Type
- Content Type
- Field Control
- Web Part
- Module
- Template
- Event Receiver
- List Instance

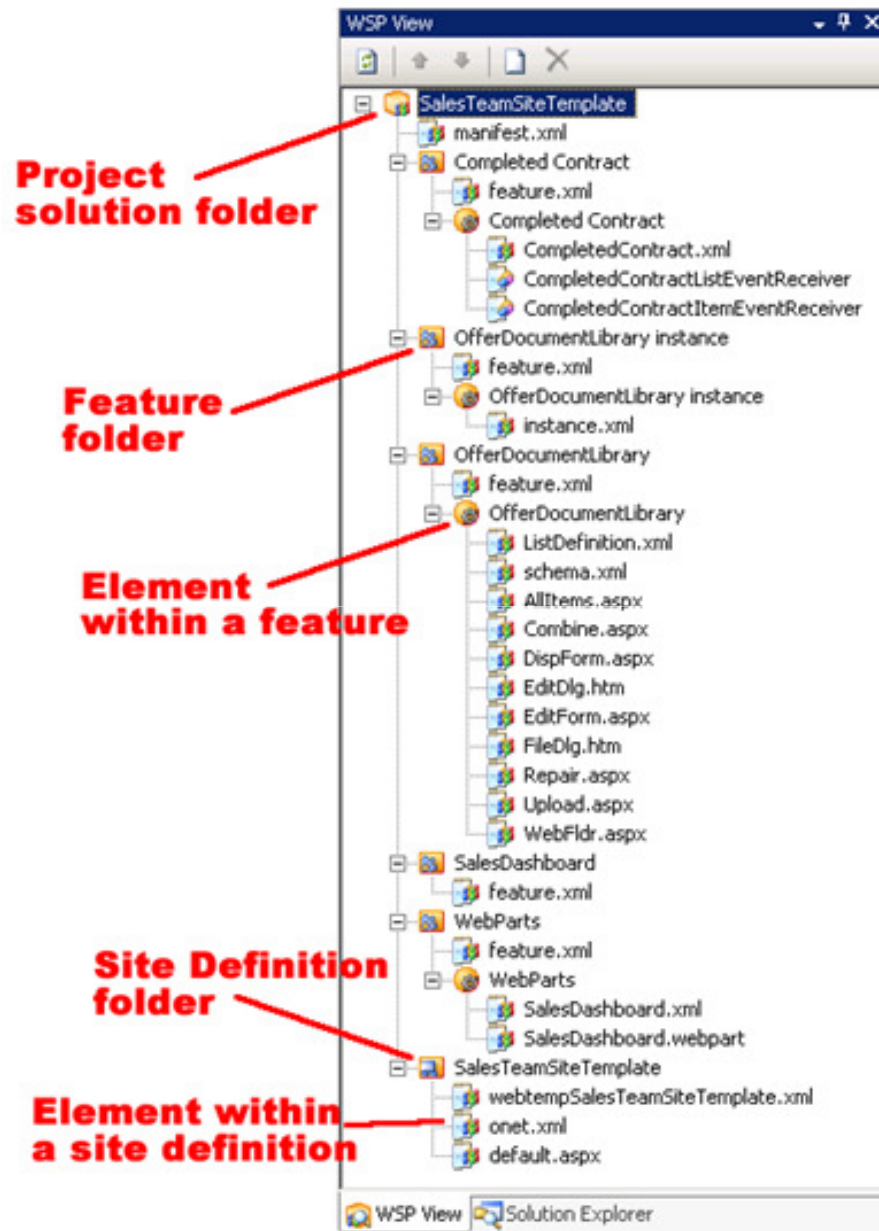
VSeWSS

WSP View

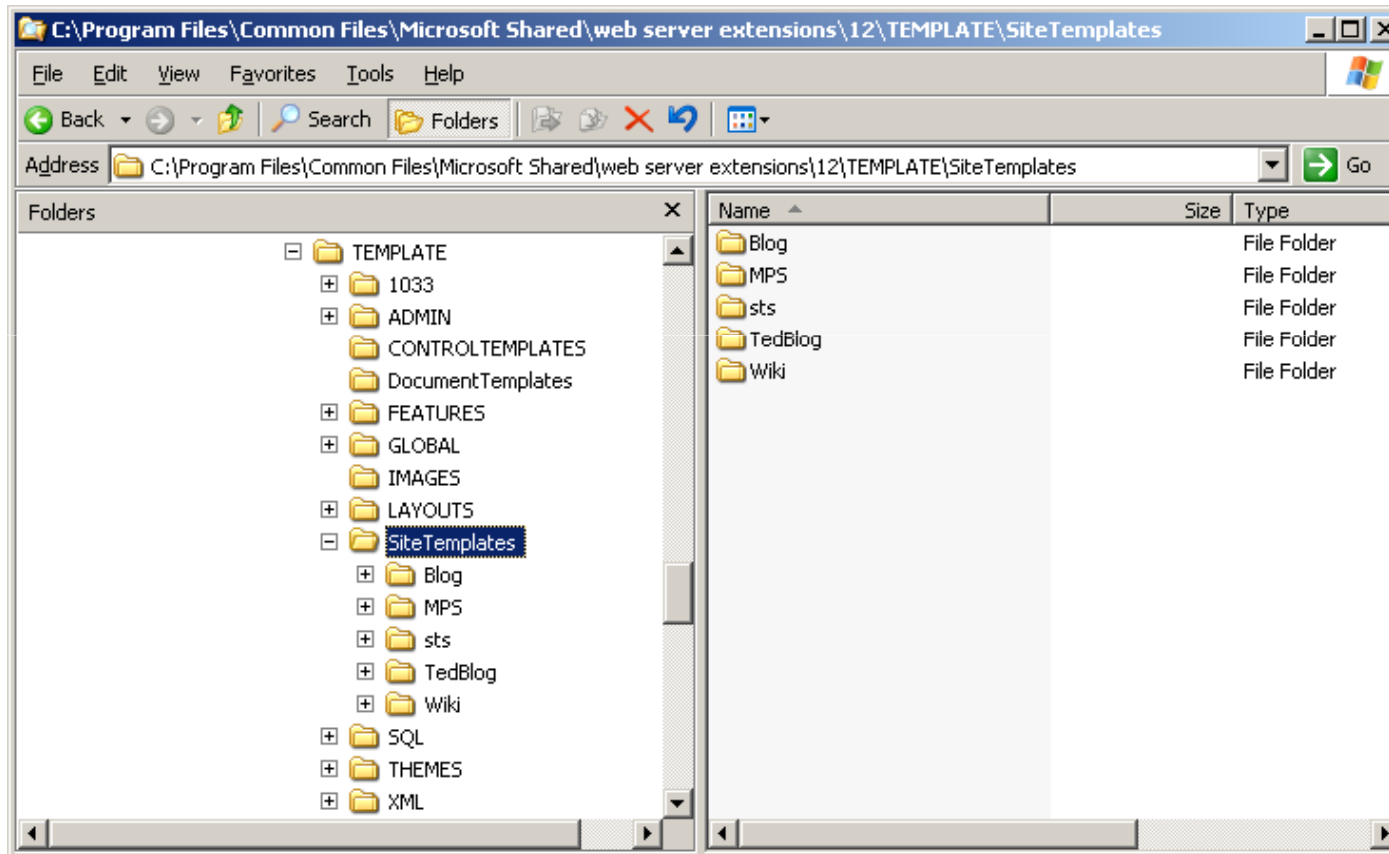
- Auto build and deploy locally
- Enhances the developer's visibility and control of the package and deploy process





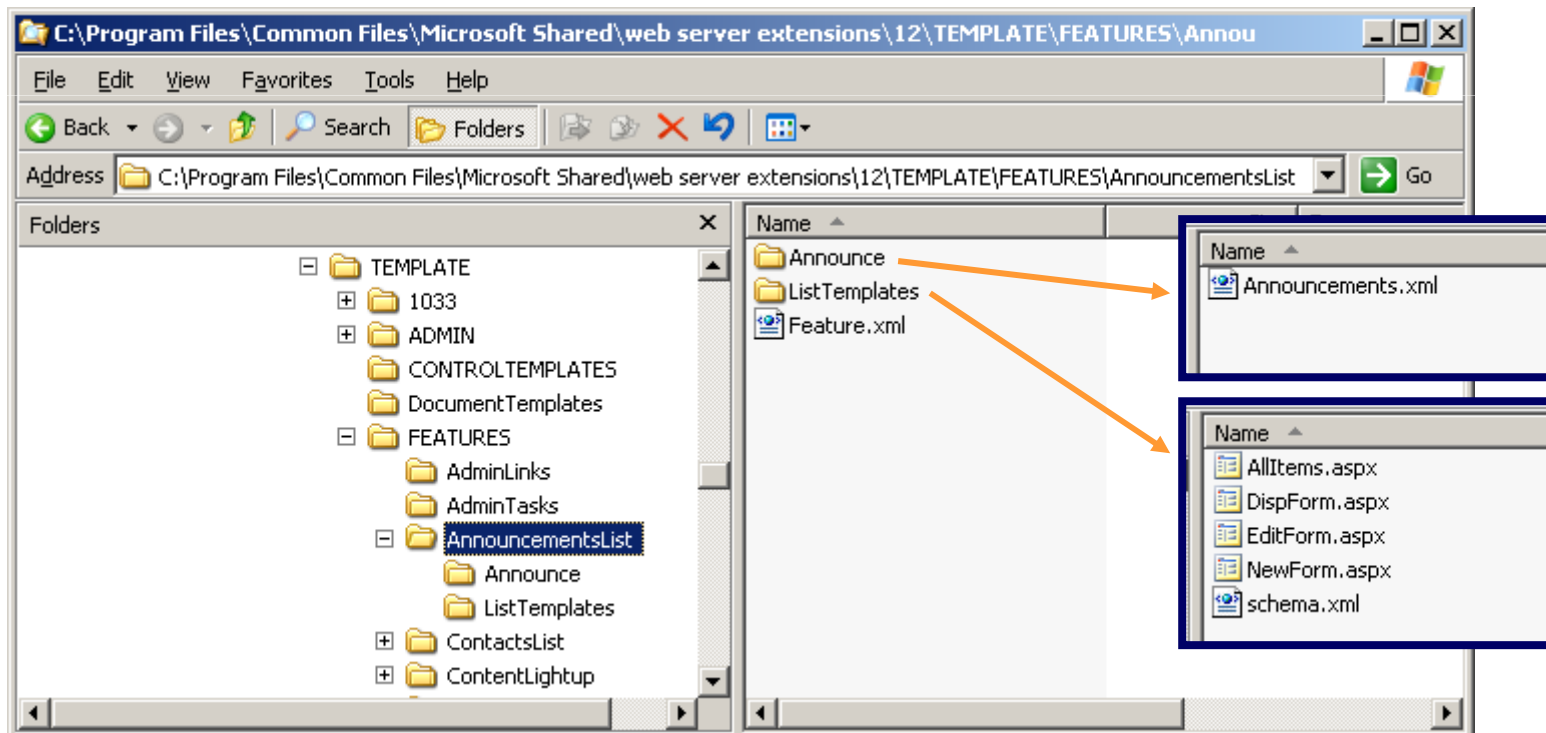


SharePoint System Files



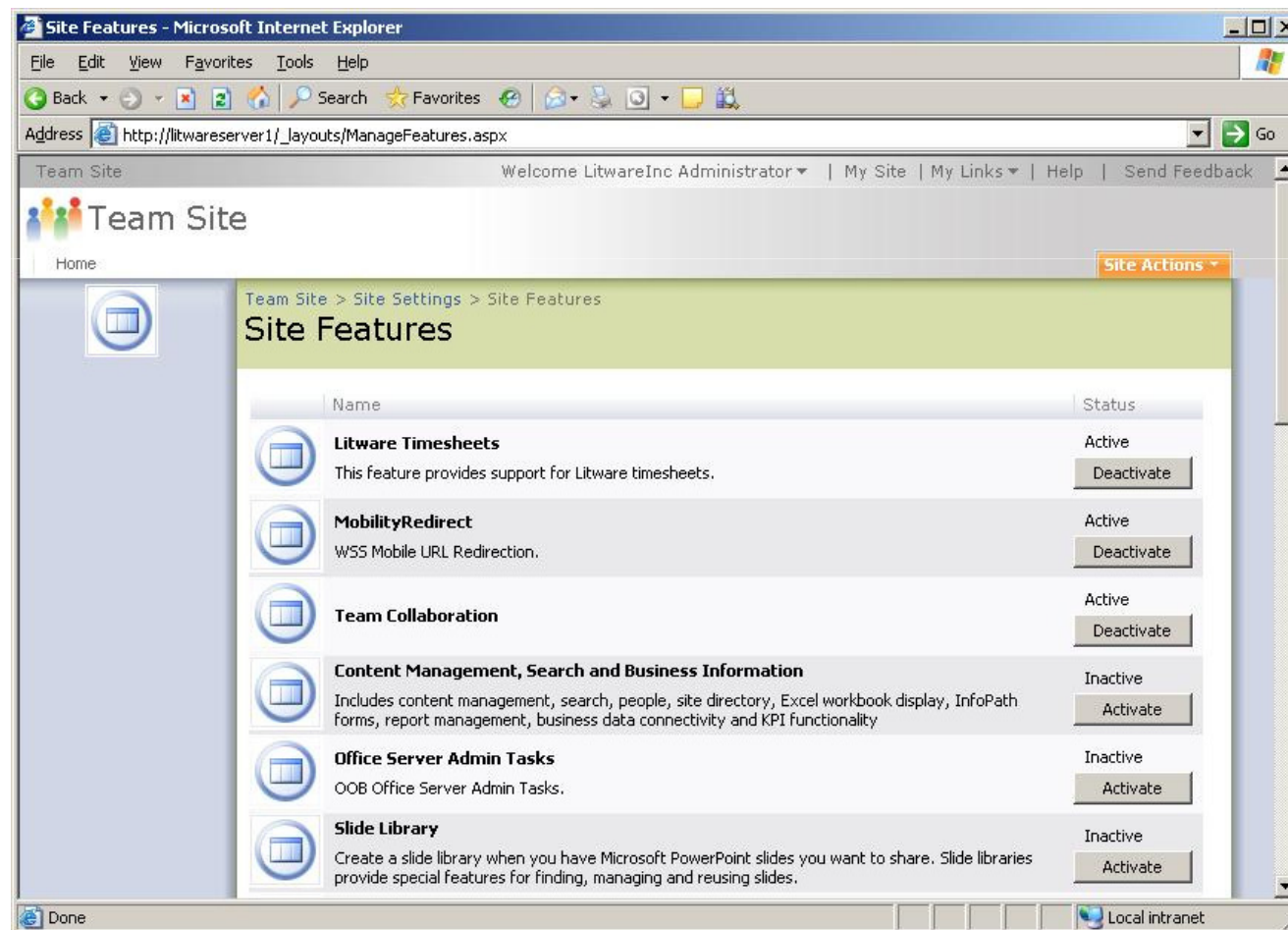
The Features Framework

- Innovation to enhance modular provisioning
 - Group logical elements into scenario-driven “features”
 - Features can be add to and reused across site definitions
 - New features can be activated in existing site to add functionality
 - Features scoped at web, site, web application, farm



User Perspective of Features

- Once installed, user can activate or deactivated feature in context of site or site collection



Elements Available at Feature Scope



Farm Elements

- Item Custom Actions
- Site Settings Links
- Admin Custom Actions



Web Application Elements

- Item Custom Actions
- Site Settings Links
- Admin Custom Actions



Site Elements

- Site Settings Links
- Site Web Part Definitions
- Workflow Definitions
- Site Content Types
- Files [provisioned to root web]
- List Instances [provisioned to root web]



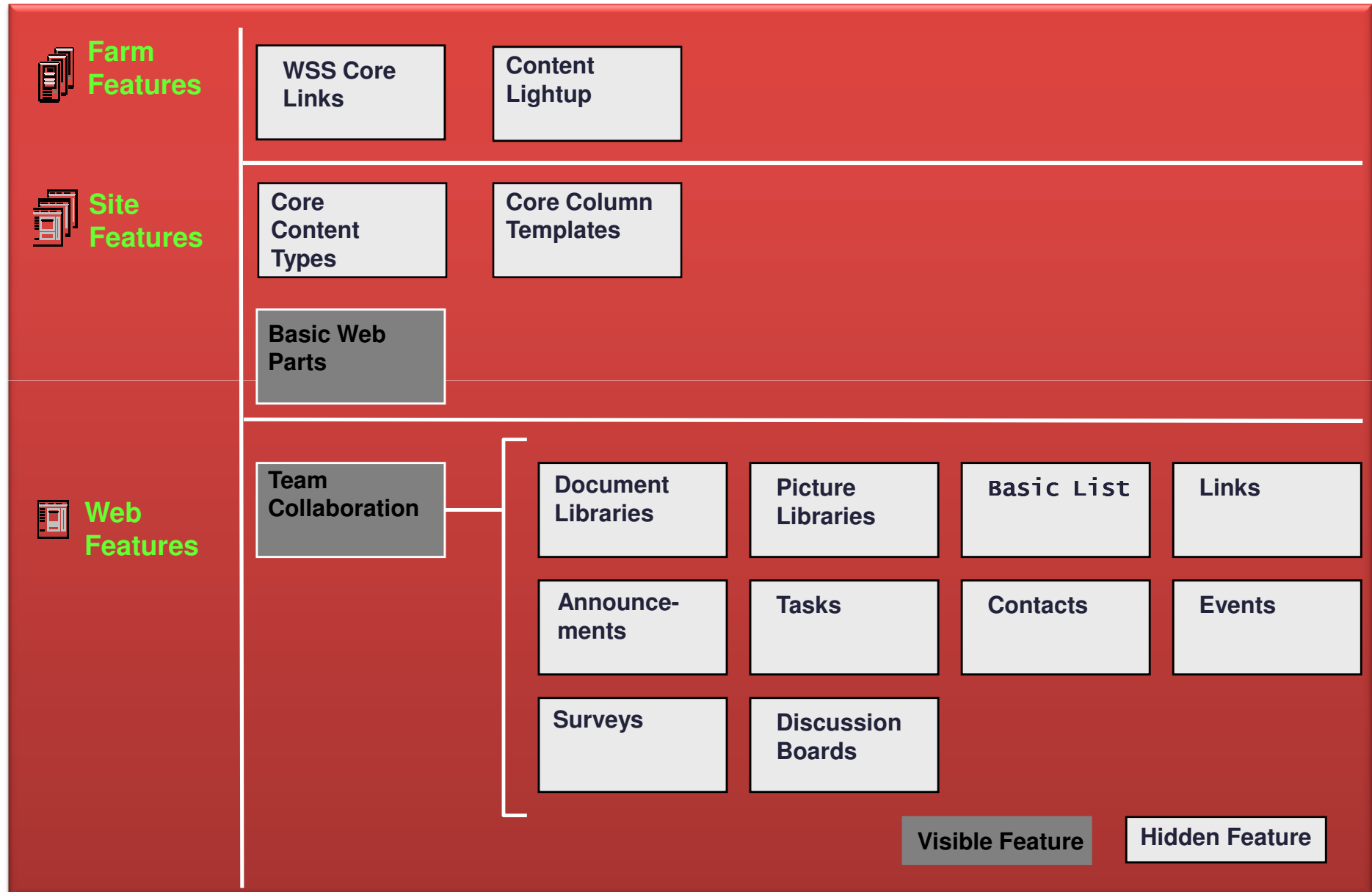
Web Elements

- List Definition + Forms + Views
- List Instances
- List Item Events
- Item Custom Actions
- Web Admin Custom Actions
- Files

Activation Dependencies

- Features can be designed with dependencies
 - Allows one feature to assume another feature is present
- Example: Feature B might depend on Feature A
 - B should be written with activation dependency on A
 - Activating B forces A to be activated as well
 - Deactivating B results in deactivation of A
- Features can be defined as hidden
 - Hides the feature from users in administration activation pages
 - Hidden features still activated by others with dependencies

Out Of The Box Features/Template Architecture



Features And Site Definitions

- Features add modularity to site definitions
 - Site definitions can include feature references

```
<Project Title="My WSS V3 Site Definition" xmlns:ows="Microsoft SharePoint">
  <Configurations>
    <Configuration ID="0" Name="Default">

      <!-- Add in a feature scoped to site collection -->
      <SiteFeatures>
        <Feature ID="5F7FDA92-4C17-4c3e-9E65-5010154FDE77" />
      </SiteFeatures>

      <!-- Add in a feature scoped to site -->
      <WebFeatures>
        <Feature ID="8C6A6980-C3D9-440E-944C-77F93BC65A7E" />
      </WebFeatures>

    </Configuration>
  </Configurations>
</Project>
```

Content Type Item Template

- Used to add custom Content Types to a SharePoint solution.
- All SharePoint List and Library items are implicitly or explicitly bound to a Content Type.
- A Content Type can be added as a project item to the following VSeWSS Project Templates:
 - Team Site Definition
 - Blank Site Definition
 - Empty

Content Type Item Template

- A Content Type is a collection of settings that can be applied to a specific category of items.
- Settings can include:
 - Parent document type.
 - Associated metadata/columns.
 - Attributes of columns such as being a required field.
 - Associated Site Columns.

Content Type - Settings and Files

- The metadata, or properties, you want to assign to this type. Represented by columns added to the list or document library when you add the content type.
- Custom New, Edit, and Display forms to use with this content type.
- Workflows available for items of this content type.

Content Type - Settings and Files

- For document content types, the document template on which to base documents of this type.
- Any information necessary for custom solutions associated with this content type. You can store this information in the content type as one or more XML documents.

Applying Content Types

- Content types created for documents can be applied only to document libraries
- Content types created for list items can be applied only to lists
- Content types created for folders can be applied to both document libraries and lists

Content Type Inheritance

- A Content Type has to be based on another Content Type.
- This inheritance model allows an organisation to specify standard metadata on a base Content Type that will be available in all descendent Content Types.
- All content types ultimately derive from the **System** content type.

Northwind Realty Scenario

- Each branch of Northwind Realty manages a process of negotiating offer documents. While in a state of negotiation, the documents exist in a site set up specifically for collaboration between the two negotiating parties (see the Team Site walkthrough above for further details). When negotiations are completed the parties agree to the final document and price. Northwind Realty then begins the billing cycle and accounting processes.

Northwind Reality Scenario

- The following problems have been identified:
 - Finding the completed legal documents requires access to the negotiation site.
 - There is no central repository for all completed legal documents.
 - Finding aggregate data like total sales or all documents for one law firm is time consuming.

- From these problems it has been suggested that completed contracts should be placed into a central location. A realty agent will be responsible for taking a copy of the completed documents and loading them into a new library and adding required metadata.
- The following requirements have been identified:
 - Contracts should be collected at a per office level.
 - All finalised documents are to be loaded by the Agent making the sale.
 - Metadata is to accompany each Finalised Offer Document containing:
 - PropertyID
 - Settlement price
 - Settlement date
 - Selling Party names
 - Selling Party Legal representation
 - Buying Party names
 - Buying Party Legal representation
 - The document library is to have an identical structure in every office.

