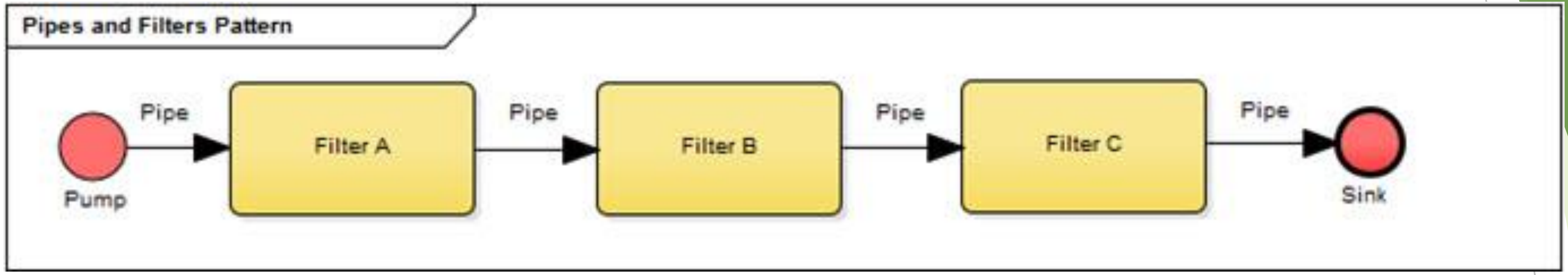
Sitecore 10 Training – Day 4

By Surendra Sharma

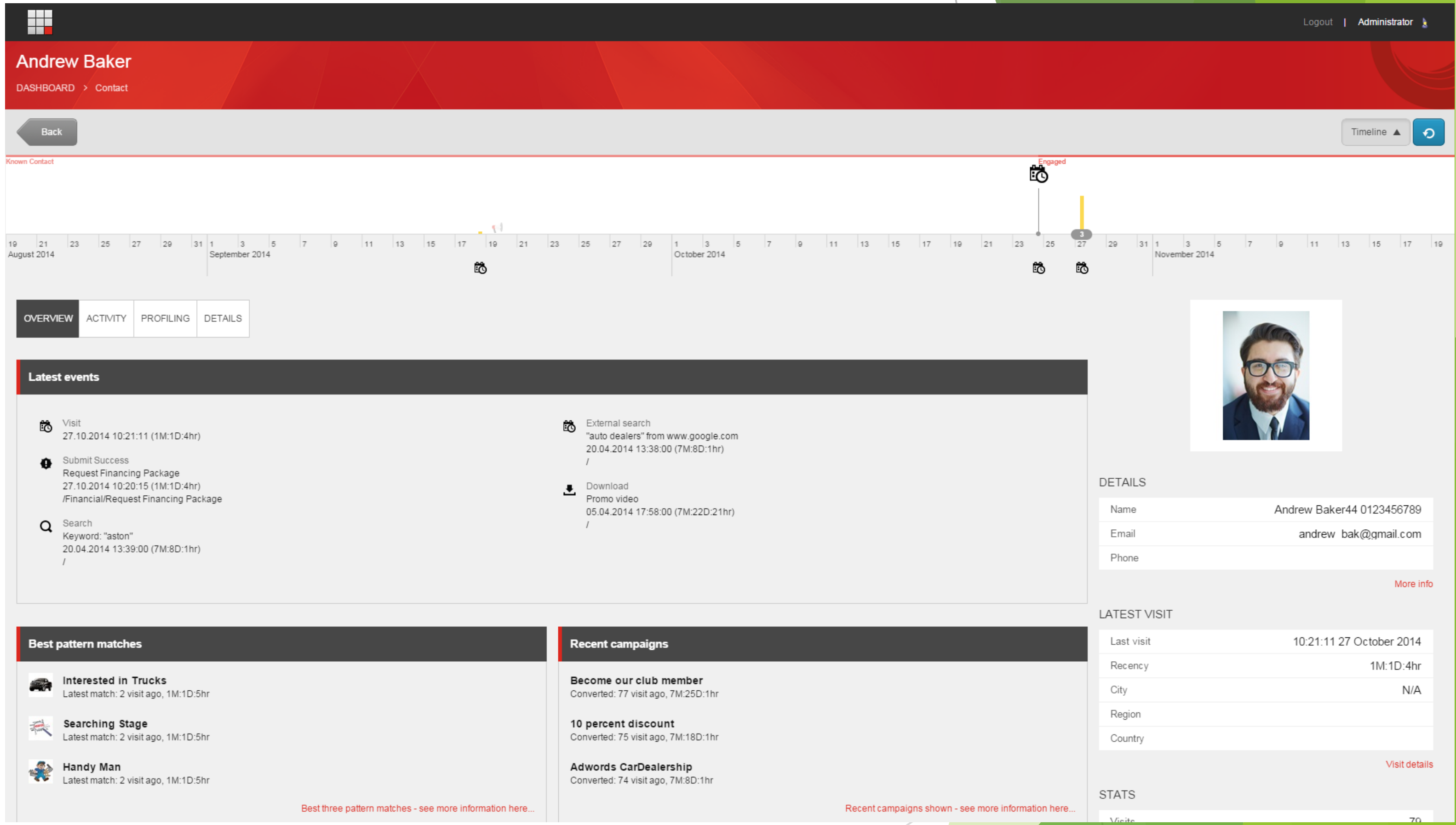
Pipeline and Processor



- Pipelines define a sequence of processors that implement a function, such as defining the Sitecore context for an HTTP request or generating a list of messages in the Content Editor.
- Each processor in a pipeline contains a method named Process() that accepts a single argument and returns void. A processor can abort the pipeline, preventing Sitecore from invoking subsequent processors.
- The argument that is passed to the Process() method must be of a type that is specific to the pipeline or be the default argument — Sitecore.Pipelines.PipelineArgs. To create a pipeline processor, create a class that implements a method named Process() with the same signature as the other processors in the pipeline. This processor can inherit from an existing processor, and you can add, remove, replace, and rearrange processors in the pipelines to suit your requirements.

Experience Profile

- The Experience Profile lets you monitor the behavior of contacts that have interacted with your company or with your website.
- The Experience Profile enables you and your sales teams to monitor the key areas of customer experience and interaction, such as visits, campaigns, goals, profiles, automations, outcomes, and keywords. For example, for each contact, you can see at a glance which events and goals they have triggered as well as how many engagement value points they have accumulated on your website.
- The Home page of the Experience Profile is a dashboard that lets you find specific contacts. When your website has been running for some time, you may have to search through many contacts.



Latest events

Visit
27.10.2014 10:21:11 (1M:1D:4hr)

Submit Success
Request Financing Package
27.10.2014 10:20:15 (1M:1D:4hr)
/Financial/Request Financing Package

Search
Keyword: "aston"
20.04.2014 13:39:00 (7M:8D:1hr)
/

External search
"auto dealers" from www.google.com
20.04.2014 13:38:00 (7M:8D:1hr)
/

Download
Promo video
05.04.2014 17:58:00 (7M:22D:21hr)
/

Best pattern matches

Interested in Trucks
Latest match: 2 visit ago, 1M:1D:5hr

Searching Stage
Latest match: 2 visit ago, 1M:1D:5hr

Handy Man
Latest match: 2 visit ago, 1M:1D:5hr

Best three pattern matches - see more information here...

Recent campaigns

Become our club member
Converted: 77 visit ago, 7M:25D:1hr

10 percent discount
Converted: 75 visit ago, 7M:18D:1hr

Adwords CarDealership
Converted: 74 visit ago, 7M:8D:1hr

Recent campaigns shown - see more information here...

DETAILS

Name	Andrew Baker44 0123456789
Email	andrew bak@gmail.com
Phone	

More info

LATEST VISIT

Last visit	10:21:11 27 October 2014
Recency	1M:1D:4hr
City	N/A
Region	
Country	

Visit details

STATS

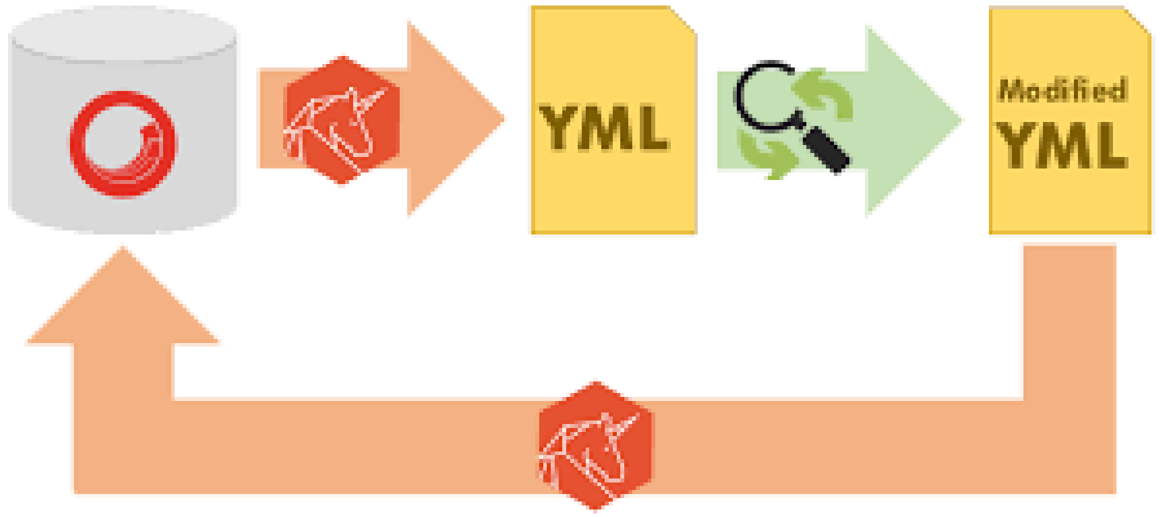
Sitecore xConnect

- xConnect is the service layer that sits in between the xDB Collection database and xDB index, and any trusted client or device that wants to read, write, or search contact and interaction data.
- All clients must use xConnect to read, write, and search xDB data. Access to xConnect is secure by default and all clients must supply a valid certificate thumbprint. You can use the xConnect Client API (C#) or interact with the web API directly.
- Create contact through xConnect console application
- Check the visitor record in Sitecore Experience Profile
- Also check contact details in SQL server table `[xpten_Xdb.Collection.Shard0].[xdb_collection].[ContactFacets]`
- To enable xConnect Anonymous data collection, make below entry "***true***" in "*\YourxConnectApp\App_Data\Config\Sitecore\SearchIndexer\sc.Xdb.Collection.IndexerSettings.xml*" and "*YourxConnectApp \App_Data\jobs\continuous\IndexWorker\App_Data\config\sitecore\SearchIndexer\sc.Xdb.Collection.IndexerSettings.xml*"
  - `<IndexPIISensitiveData>true</IndexPIISensitiveData>`
  - `<IndexAnonymousContactData>true</IndexAnonymousContactData>`
- xConnect Issue : The request was aborted: Could not create SSL/TLS secure channel.
  - Open "***Manage computer certificates***" in a start menu search
  - Find the client certificate (normally in Personal\Certificates) with name "<sitecore>.xconnect". Right click it, choose ***All Tasks***, then ***Manage Private Keys***.
  - Assign "***Everyone***" permission to this.

Check document and Sample Application for more details

Serialization

- Serialization allows you to serialize an entire Sitecore database or a series of items in a database to text files. You can then use these text files to transfer this database or series of items to another database or Sitecore solution.
- The Sitecore serialization functionality is designed to help teams of developers that work on the same Sitecore solution to synchronize database changes between their individual development environments, but is also valuable when a single developer works on a solution.
- Sitecore serialization is typically used in combination with a source control system to make it easy for developers to synchronize database changes between their local databases that they use for development — with the added benefit that the source control system will keep track of database changes and allow you to compare database changes.

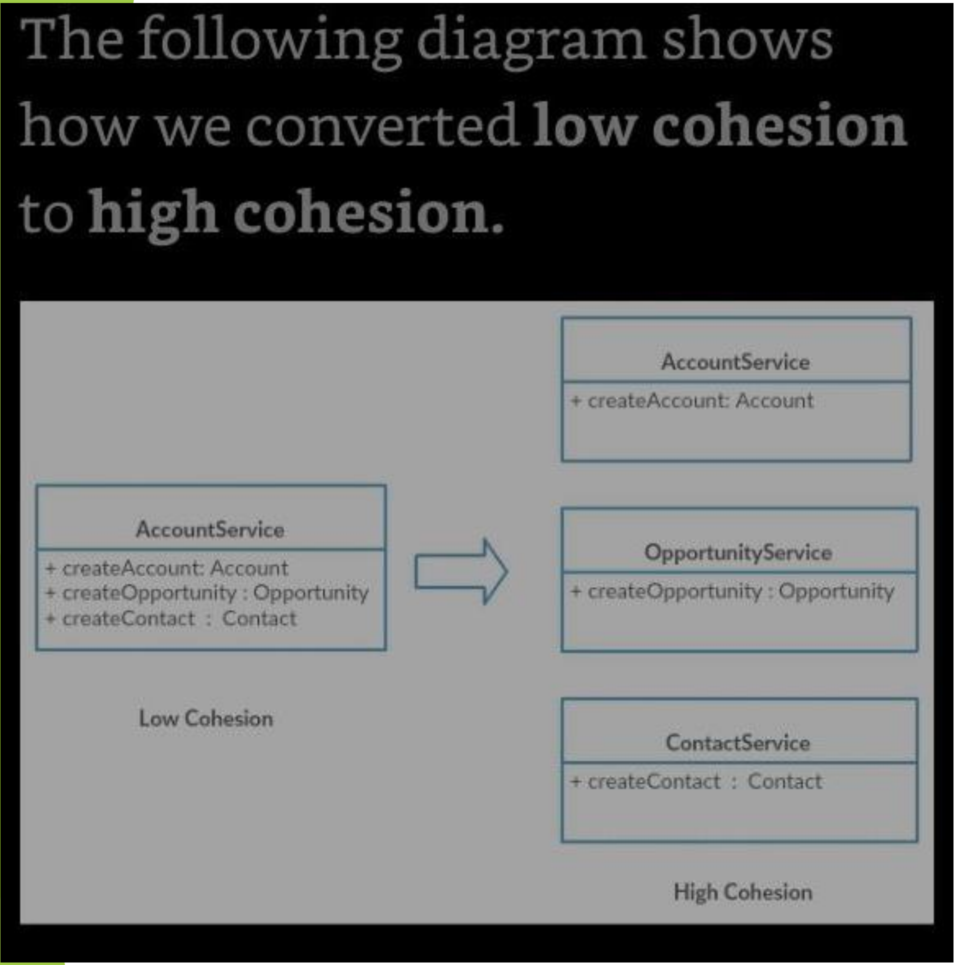


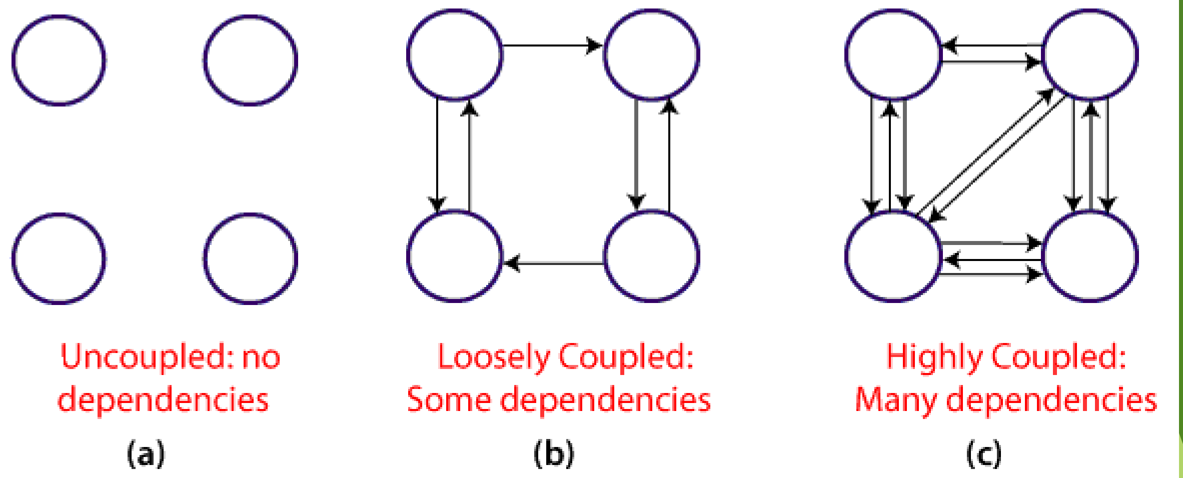
Serialization – Ways to do (Unicorn)

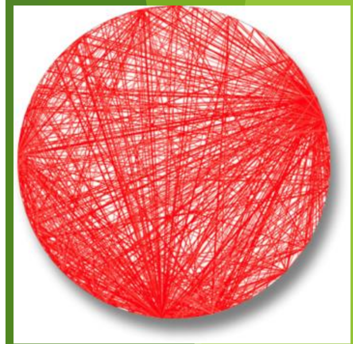
Check document

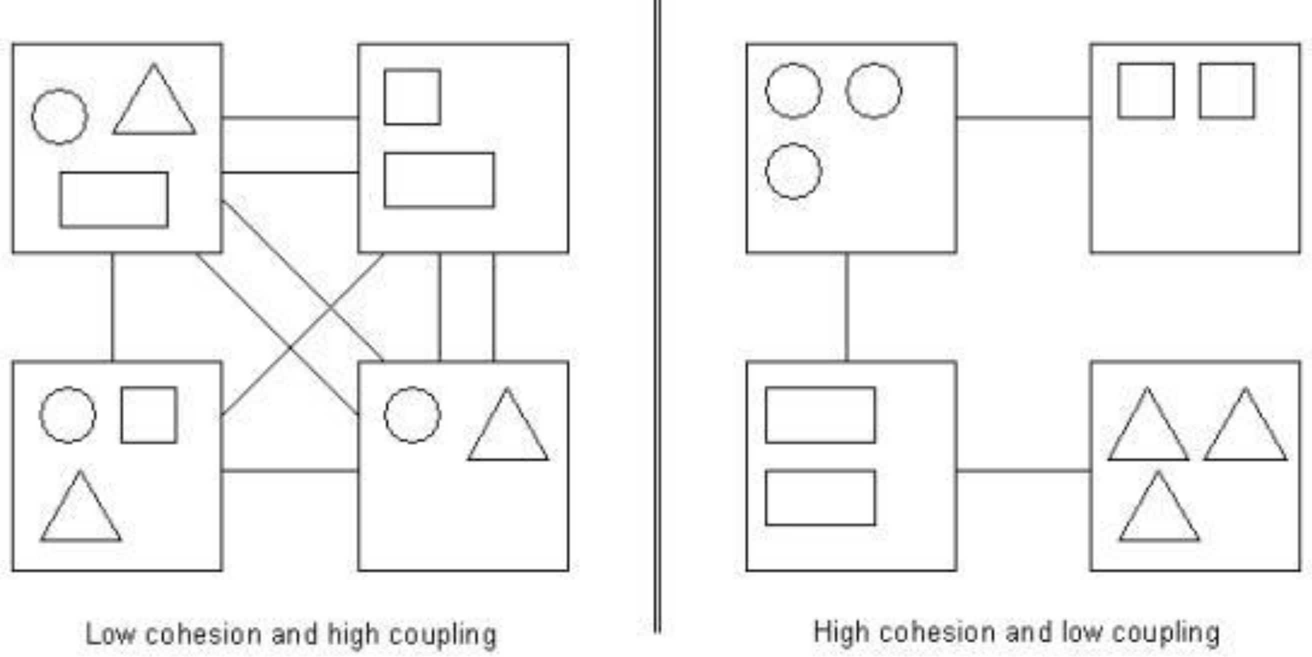
MODULE 6 – Helix, Tools & Best Practices

All good software design will go for high cohesion and low coupling.


The following diagram shows how we converted **low cohesion** to **high cohesion.**

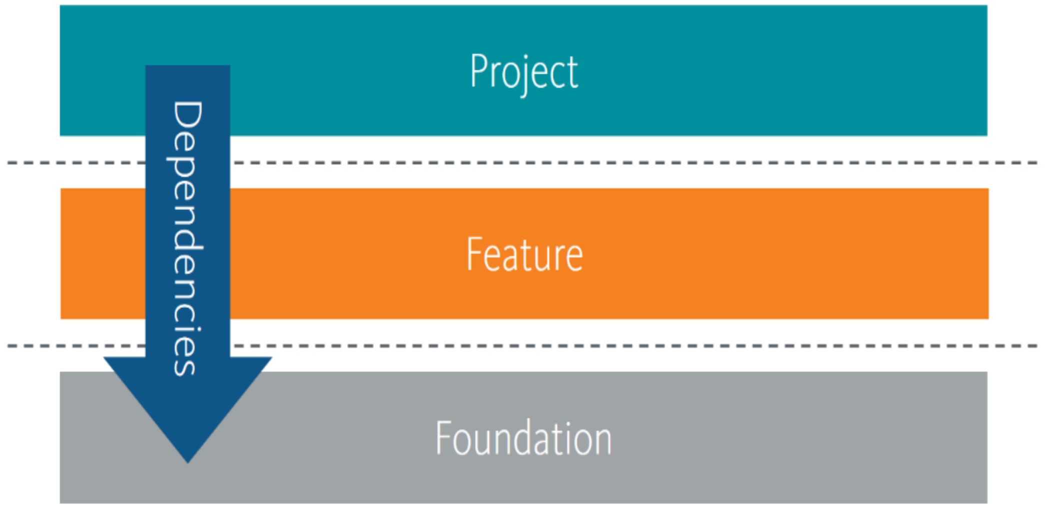

Uncoupled: no dependencies
(a)
Loosely Coupled: Some dependencies
(b)
Highly Coupled: Many dependencies
(c)



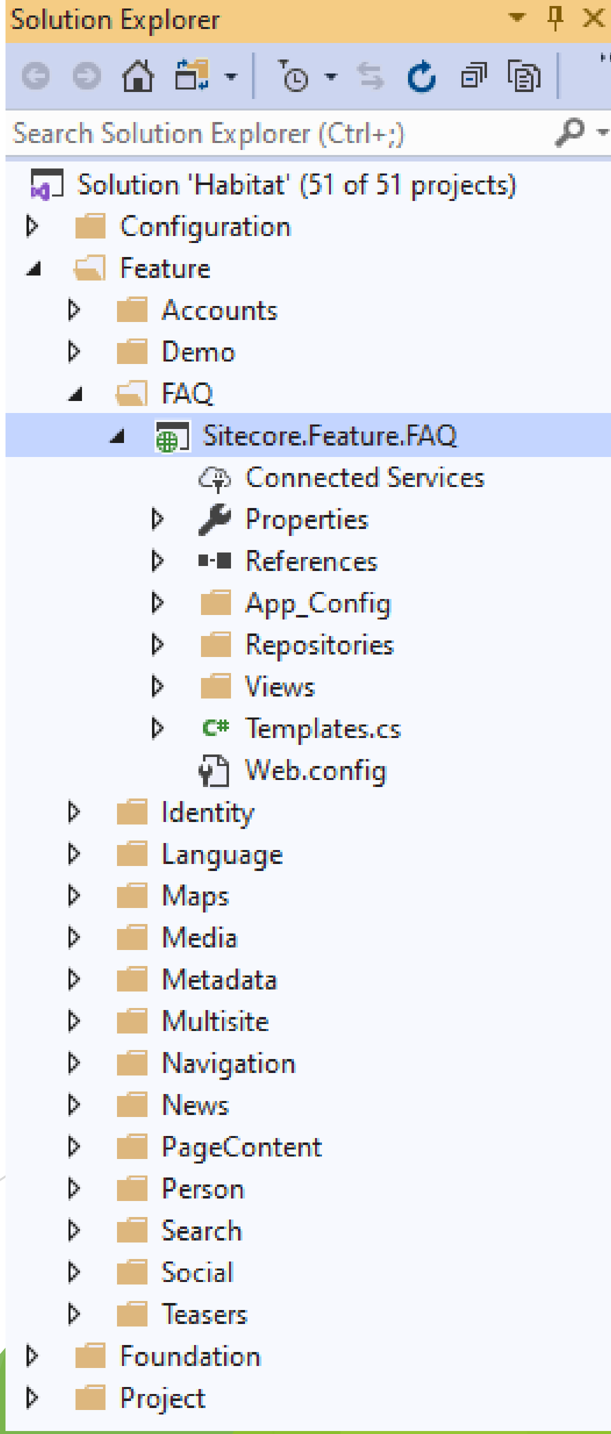

Low cohesion and high coupling
High cohesion and low coupling

What is HELIX

- It’s a Modular Architecture where for each component you will create separate MVC project so that you can reuse and manage component in better way.
- HELIX guidelines uses modular architecture principles by separating components in Foundation, Feature and Project.



- The three layers are defined in Visual Studio as Solution folders, in the file system and directories and in Sitecore as corresponding content type folders.


Solution Explorer

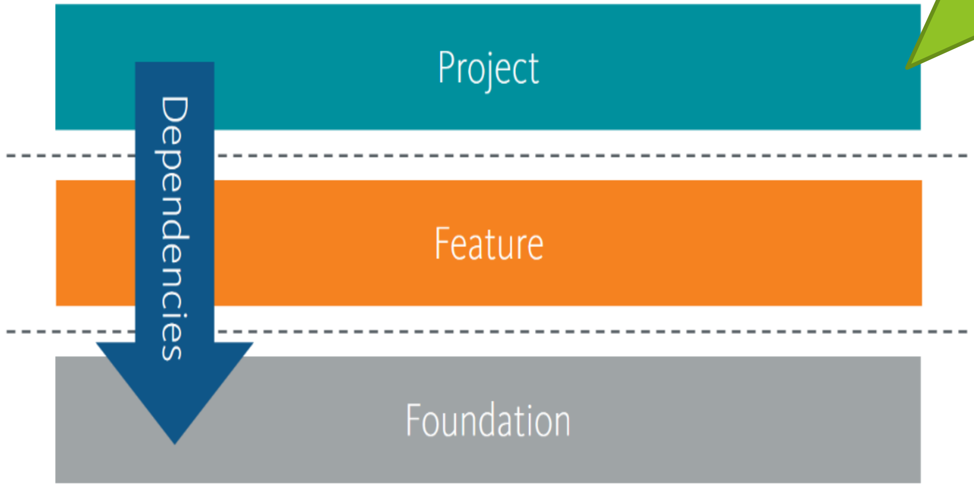
Search Solution Explorer (Ctrl+;)

```
Solution 'Habitat' (51 of 51 projects)
  Configuration
  Feature
    Accounts
    Demo
    FAQ
      Sitecore.Feature.FAQ
        Connected Services
        Properties
        References
        App_Config
        Repositories
        Views
        Templates.cs
        Web.config
    Identity
    Language
    Maps
    Media
    Metadata
    Multisite
    Navigation
    News
    PageContent
    Person
    Search
    Social
    Teasers
  Foundation
  Project
```

Project Layer

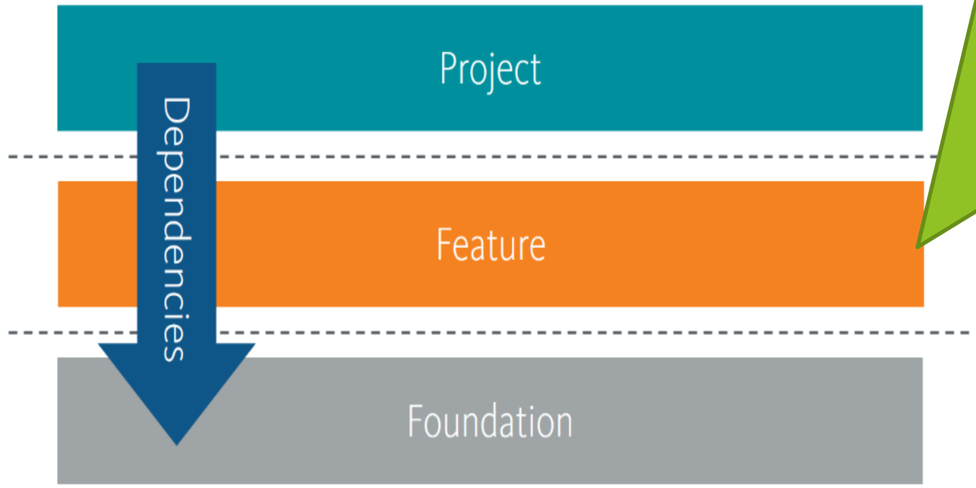
- This is the top layer. This means the actual website output from the implementation, such as, layout file , css, js, fonts, images and graphical design.

- Each time there is a change in a feature, a new one is added or one is removed, then this layer will typically change. That’s why modules in this layer are mostly unstable.



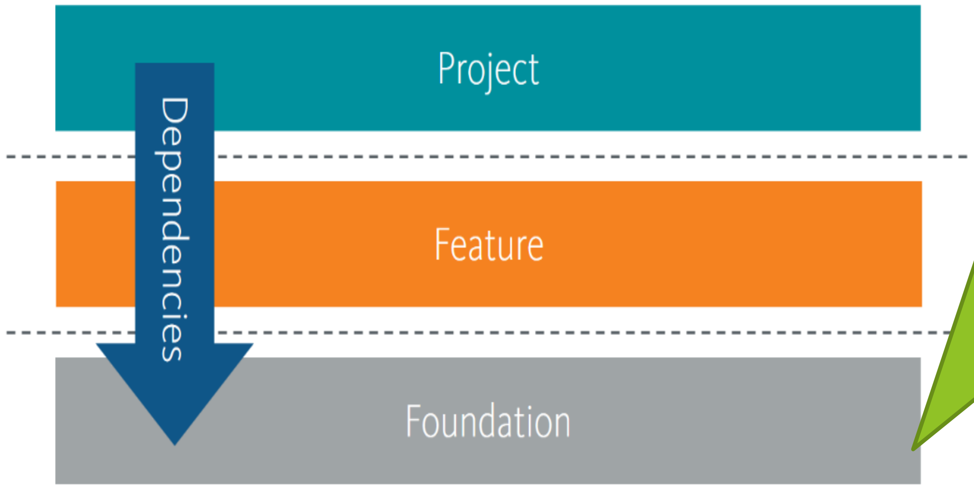
Feature Layer

- The Feature layer contains concrete features of the solution as understood by the business owners and editors of the solution, for example News, Search, Events, Brochures, FAQ, media, carrier, services, help and support, etc.
- The responsibility of a Feature layer module is defined by a business user and not by the underlying technology.
- Each Feature layer module has to strictly conform to the Common Closure Principle.
- This principle ensures that changes in one feature do not cause changes anywhere else, and that features can be added, modified and removed without impacting other features.



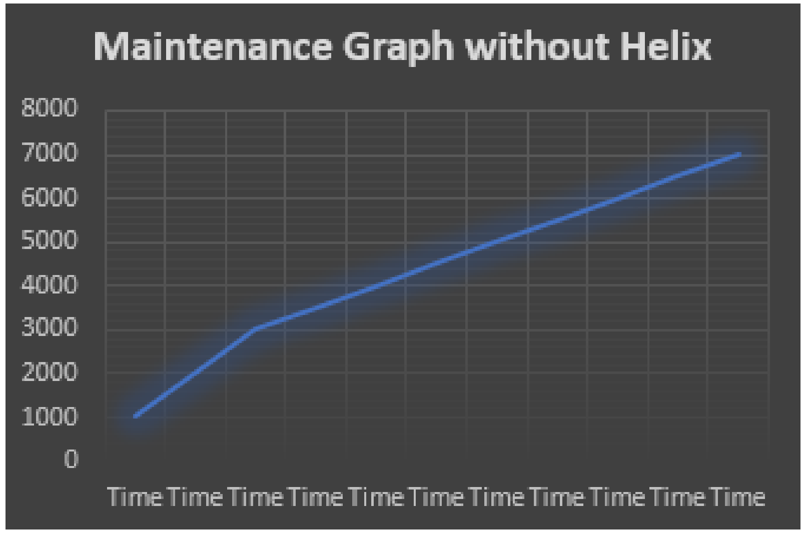
Foundation Layer

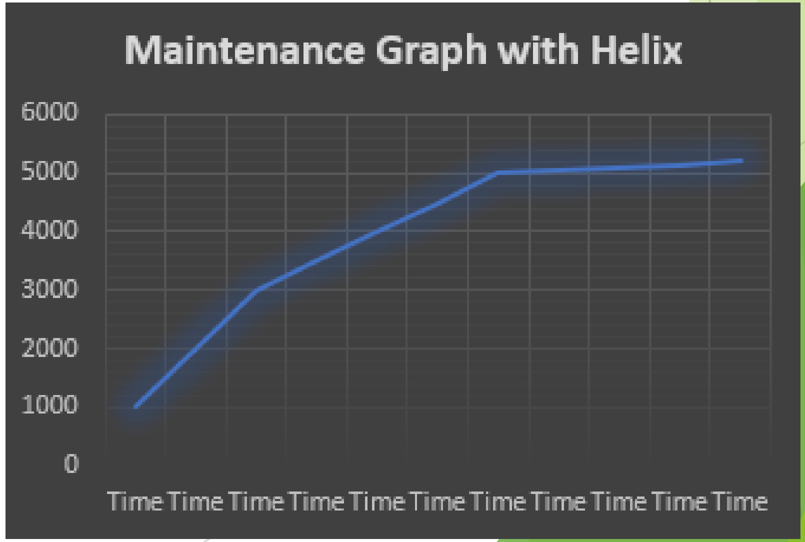
- It forms the foundation of your solution. When a change occurs in one of these modules it can impact many other modules in the solution. This mean that these modules should be the most stable in your solution in term of the Stable Dependencies Principle - Authentication, Triggering Reminder, Sending Mail
- Typically, modules in the Foundation layer are either business-logic specific extensions on the technology frameworks or shared functionality between feature modules.
- One Foundation layer module can depend on another Foundation layer module in the solution – as long as they rely on the Acyclic Dependencies Principle and the Stable Abstractions Principle.



Why HELIX

- A single Sitecore implementation can end up consisting of a large number of modules and this can lead to technical management issues.
- In large scale implementations, it is therefore often helpful to break down your implementation into multiple Visual Studio Solutions – each with a logical grouping, reusable modules.
- For this we have Modular Architecture which suggest Stable Dependencies Principle or SDP. SDP define direction of dependencies. From strong to weak in the direction of base to top.





Habitat

- Habitat is an example website for demonstrating HELIX principles.
- Unicorn is used in Habitat for synching Sitecore items.
- Link is http://habitat.demo.sitecore.net/

Connecting HELIX, Habitat and Visual Studio

Sitecore Project Life Cycle

Development

a. Setting up a development environment
b. Local deployment
c. Version Control

Build and integration

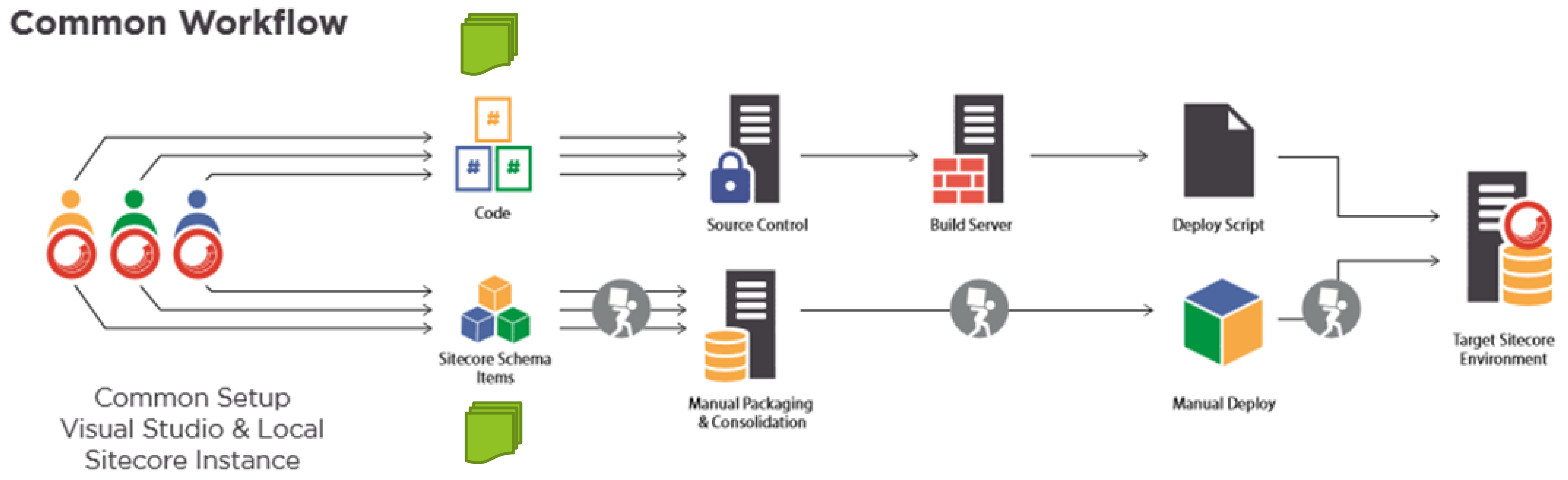
a. Building your solution
b. Integration

Testing

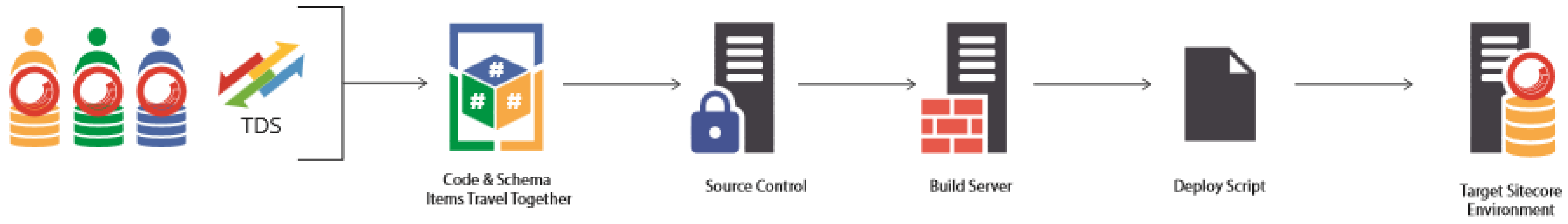
a. Managing Tests
b. Unit tests
c. Integration, Acceptance or other automated testing methods

Deployment

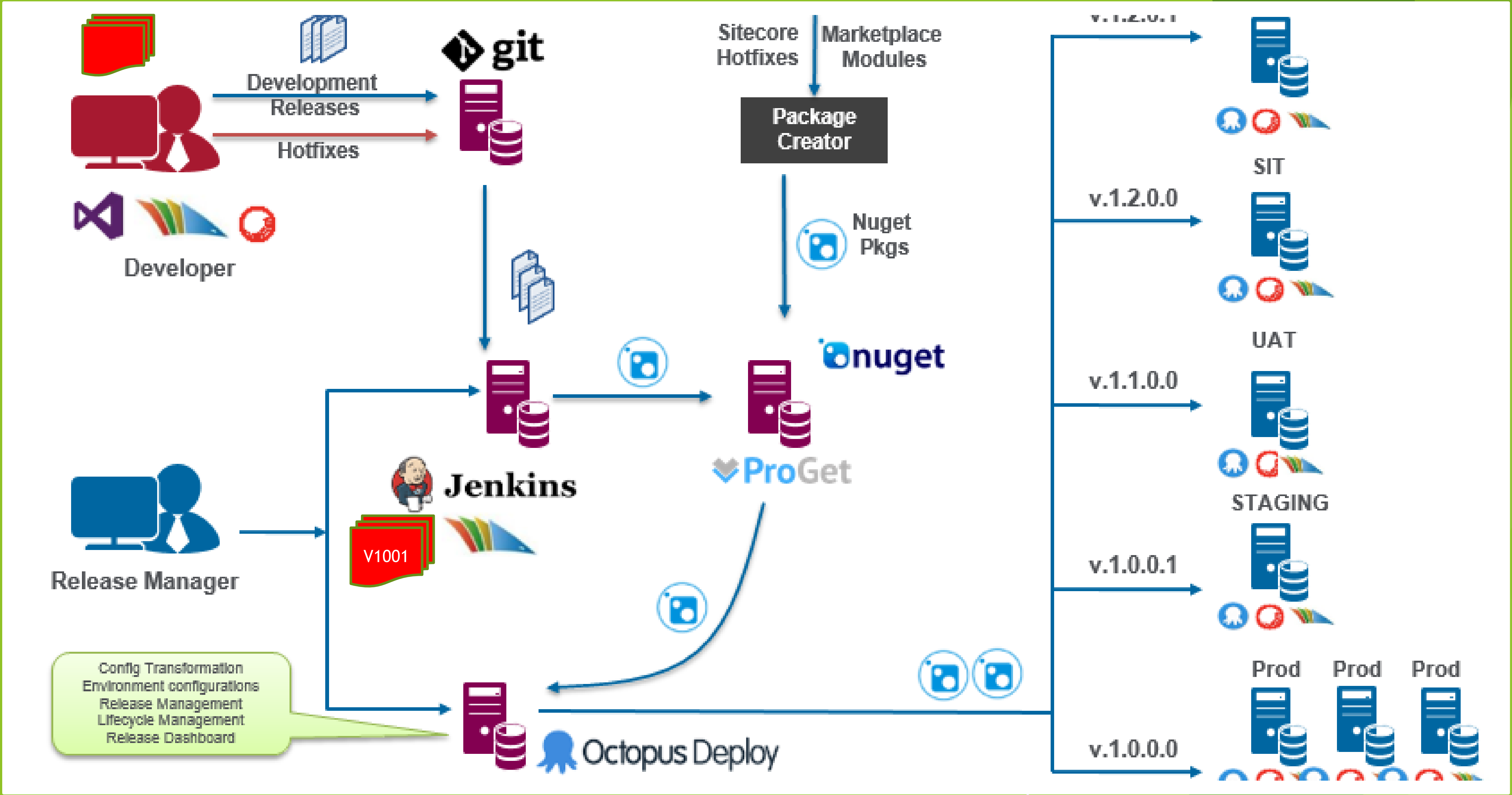
a. Deployment strategy
b. What to deploy and to where


Common Workflow
Code
Source Control
Build Server
Deploy Script
Target Sitecore Environment
Sitecore Schema Items
Manual Packaging & Consolidation
Manual Deploy
Common Setup Visual Studio & Local Sitecore Instance

Workflow With TDS


TDS
Code & Schema Items Travel Together
Source Control
Build Server
Deploy Script
Target Sitecore Environment

Visual Studio, TDS
Local Sitecore Instance
& SQL Database


Development Releases
Hotfixes
git
Developer
Sitecore Hotfixes
Marketplace Modules
Package Creator
Nuget Pkgs
nuget
ProGet
Jenkins
V1001
Release Manager
Config Transformation
Environment configurations
Release Management
Lifecycle Management
Release Dashboard
Octopus Deploy
SIT
v.1.2.0.0
UAT
v.1.1.0.0
STAGING
v.1.0.0.1
Prod
Prod
Prod
v.1.0.0.0

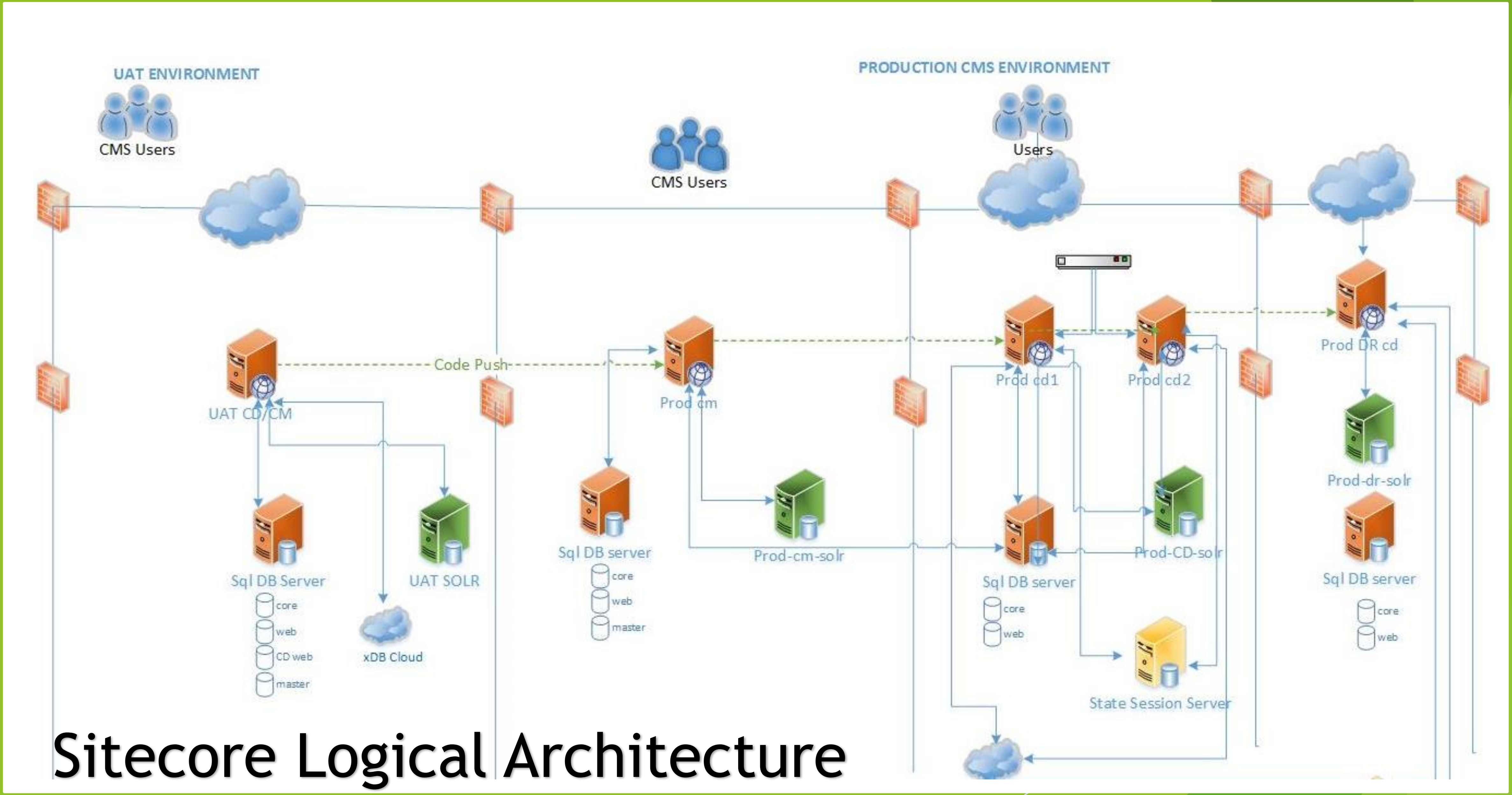
Server Roles

To specify the server role:

- In the web.config file, in the <AppSettings> section, specify the server role value:

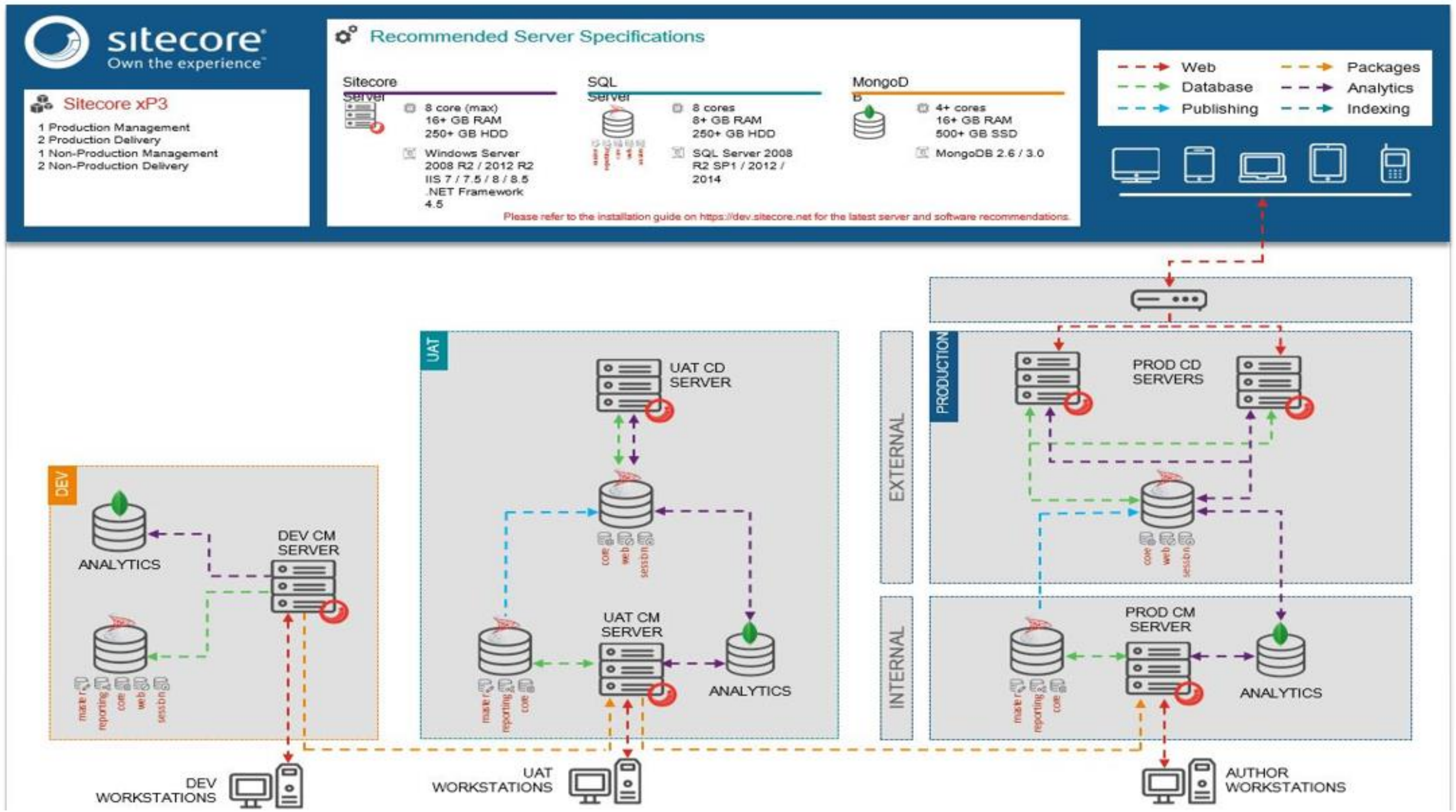
```
 <AppSettings>
  <add key="role:define" value="[server role]"/>
 </AppSettings>
```

- The supported values for the server role are:
  - ContentDelivery
  - ContentManagement
  - Processing
  - Reporting
  - Standalone
- The values are not case sensitive.
- The default value is Standalone, which means that the Sitecore instance performs all Sitecore server roles.



Sitecore Logical Architecture

SiteCore Physical Architecture

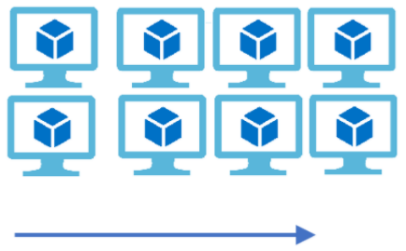


Sitecore topologies and tiers - https://kb.sitecore.net/articles/713688

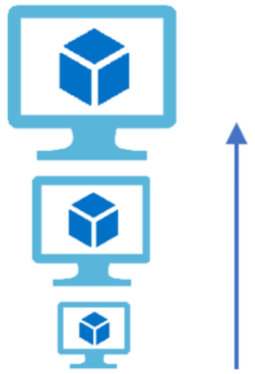
App Service pricing - https://azure.microsoft.com/en-in/pricing/details/app-service/windows/

Scaling

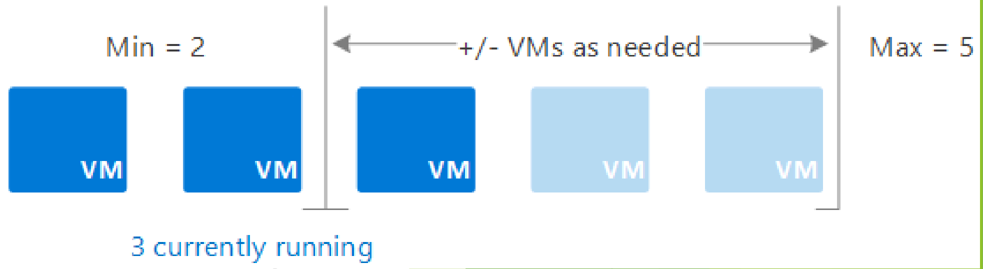
- Horizontal scaling
  - Scale by adding more machines into your pool of resources.



- Vertical scaling
  - Scale by adding more power (CPU, RAM) to an existing machine.



- Auto Scaling
  - Based on parameter like CPU consumption, Memory consumption


Min = 2
+/- VMs as needed
Max = 5
VM
VM
VM
VM
VM
3 currently running