

# ZENworks 7 Planning and Deployment Best Practices

January 31, 2007



**Novell.**<sup>®</sup>

# ZENworks® Desktop Management

## Novell® Methodology

# Novell® Methodology



Business  
Assessment

.....



Technical  
Assessment

.....



Design

.....



Development  
& Testing

.....



Deployment

.....

# Novell® Methodology



## Business Assessment

.....

### Assess:

- Stakeholders' needs and concerns
- Business requirements

# Novell® Methodology



## Business Assessment

---

### Assess:

- Stakeholders' needs and concerns
- Business requirements



## Technical Assessment

---

### Assess:

- Current environment
- Limitations
- Technical requirements

# Novell® Methodology



## Business Assessment

---

### Assess:

- Stakeholders' needs and concerns
- Business requirements



## Technical Assessment

---

### Assess:

- Current environment
- Limitations
- Technical requirements



## Design

---

- Model a technical architecture that works with current the environment
- Plan configurations to meet business requirements

# Novell® Methodology



## Business Assessment

---

### Assess:

- Stakeholders' needs and concerns
- Business requirements



## Technical Assessment

---

### Assess:

- Current environment
- Limitations
- Technical requirements



## Design

---

- Model a technical architecture that works with current the environment
- Plan configurations to meet business requirements

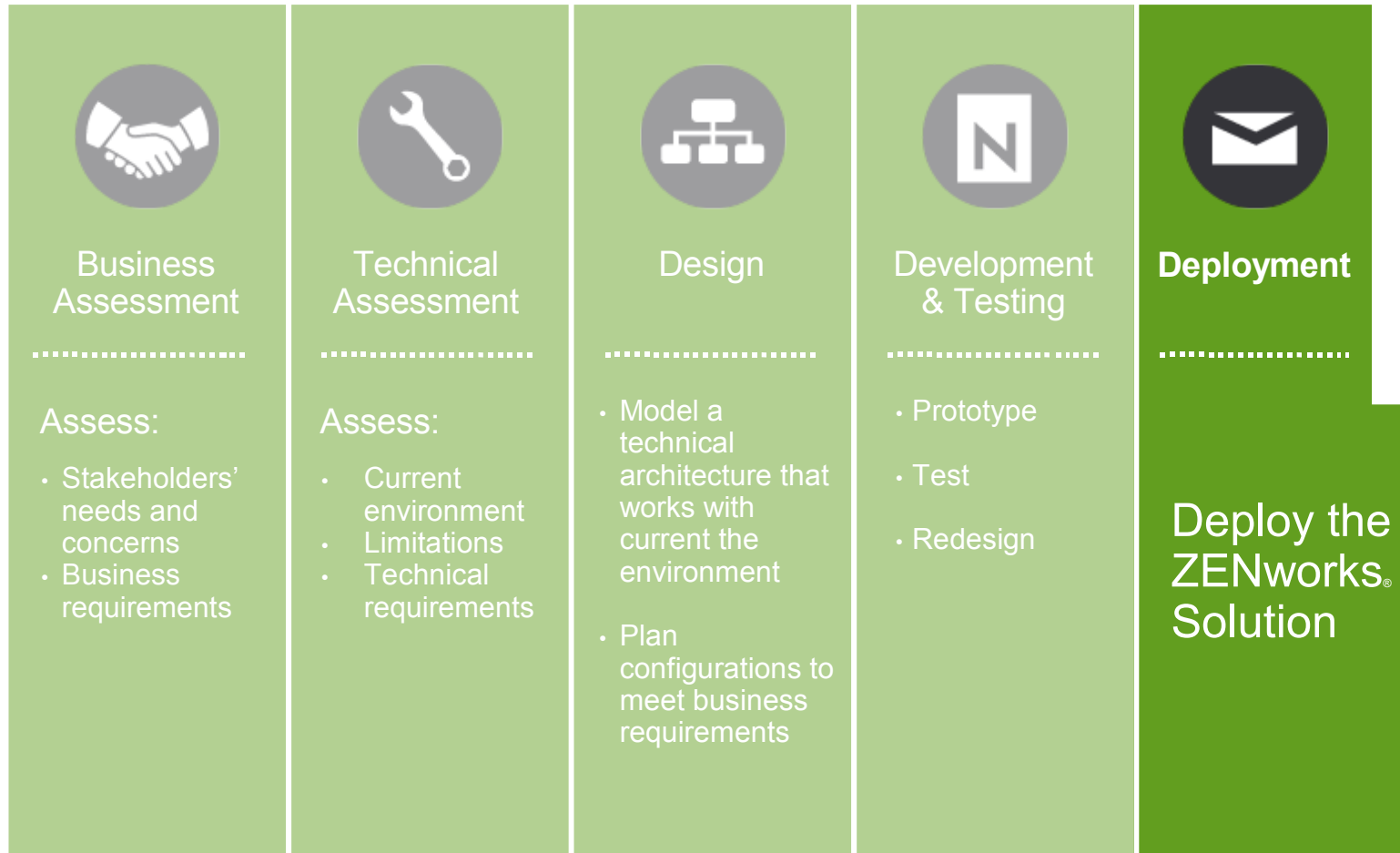


## Development & Testing

---

- Prototype
- Test
- Redesign

# Novell® Methodology



ZENworks® Desktop Management  
Best Practice Design and Architecture

# Design and Architecture

## The Primary Principle

- Approach everything as a formal project, and assign a project manager
  - Upgrade
  - New implementation
  - Migration
- Use this rule-of-thumb and win!!
  - Roughly 80% of your efforts should be spent on designing your solution, and documenting it carefully
  - Roughly 20% of your efforts should be spent on deploying, tweaking, and updating your documentation

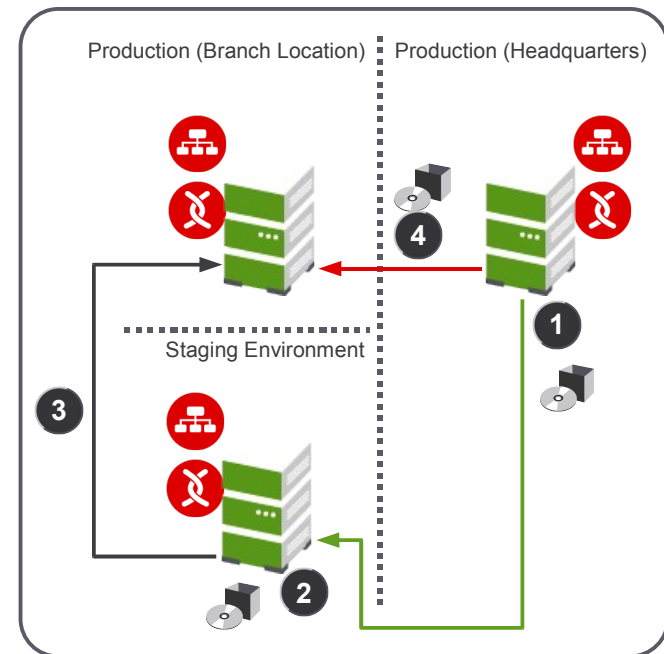
# Design and Architecture

## Principles and Rules

- Structure your directory around the business, locations – but make sure you keep it simple
- Use the ZENMaster Container Structure to manage your applications portfolio
- Keep content close to users
- Choose a naming convention, document it, and stick with it for the long run
- Be consistent where you place your service objects
- Work to associate most applications to containers, workstations, or users
- Avoid the use of “Group” association
- Know your policies, and configure them well
- Centralize what you can... but be realistic

# Design and Architecture Principles and Rules

- Use a staging environment whenever possible
- Use Tiered Electronic Distribution (TED) to synchronize critical application source data, and application objects across the enterprise
- Separate your testing environment from your production environment
- Script and test your production applications outside of the production space
- Create addon images for all core business applications (applications that everyone is licensed to use) – treat all other applications as functional business application
- Organize your corporate applications into either core business applications or line of business applications



**Document everything!!!**

# Design and Architecture

## Roaming Users

- Use the Middle Tier Server to manage your mobile workforce
  - Use a Middle Tier Server or farm of Middle Tier Servers both inside the corporate boundaries, and within the DMZ resolvable from anywhere in the world
  - Front your Middle Tier Server farms with an L4 switch if at all possible
  - Resolve the DNS name of your Middle Tier Server farm to the internal farm or external farm, based on where the user is currently located
  - Remember you need to do very little to tweak the web server – a Middle Tier Server can handle around 2,000 connections

# Design and Architecture

## Single Tree or Multiple Trees?

- Depends on a number of things...
  - Size of your organization
  - Complexity of the solution architecture
  - Your specific requirements
- Benefits
  - Production is completely separate from change and configuration management, testing, and QA reducing the potential for downtime
  - Avoid rigorous change management rules often found in the production file and print environment
- Cons
  - More complex

# Design and Architecture

## Release and Change Management Framework

- The framework consists of...
  - Formal methodologies for managing...
    - > **Application standardization**
    - > **Formal QA/QC processes**
  - Methodologies map over to specific ZENworks® technology sets
  - If you are familiar with ITIL, then you will understand this approach well
- Benefits include...
  - Reduced administrative overhead required to manage the corporate application library and the replication of application objects to downstream sites.
  - A formal QA process.
  - Objects are neatly organized at each site, and containers do not end up cluttered.
  - Administrators can go from one site or divisional container to another and know where all the sites objects will be located.
  - Better security control, and less issues.

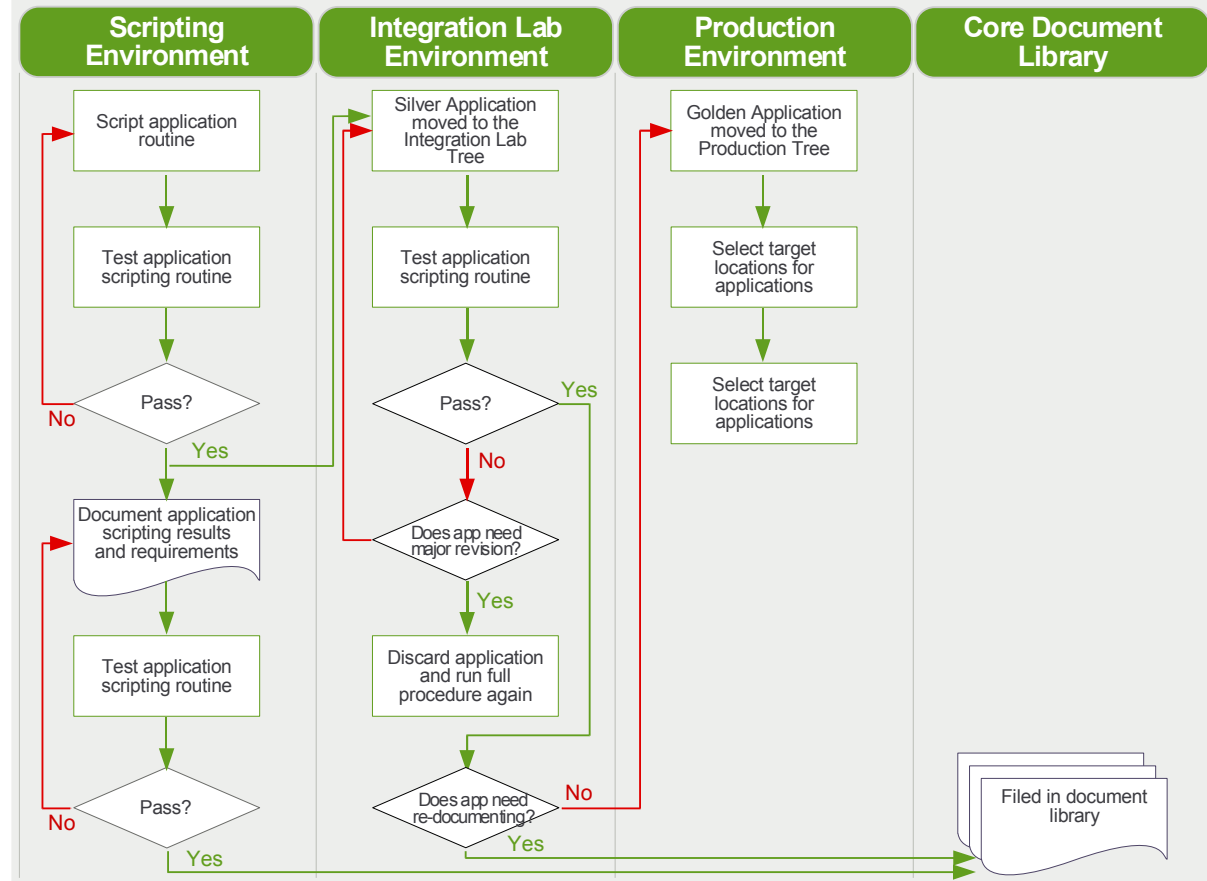
# Design and Architecture

## Release and Change Management Framework

• Identify your QA process, and carefully document the following:

- Scripting environment
- Scripting methodologies
- QA process and flow of applications throughout the environment(s)
- Test cases (L1, L2, etc.)
- Documentation repositories
- Application owners

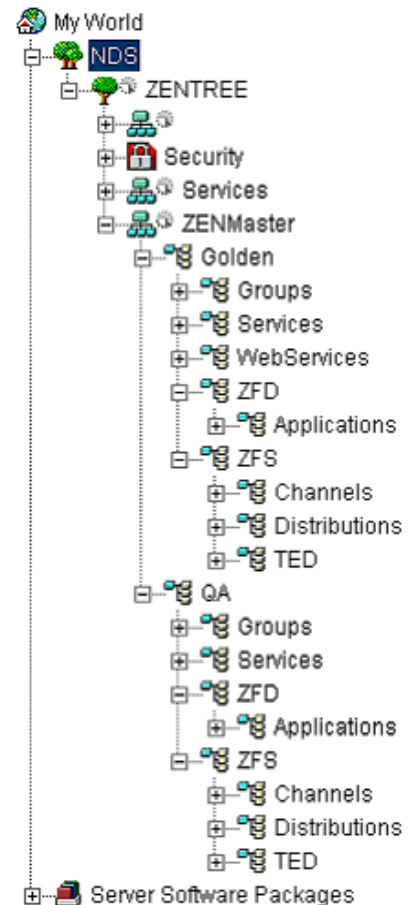
Application Synchronization Flow



# Design and Architecture

## Release and Change Management Framework

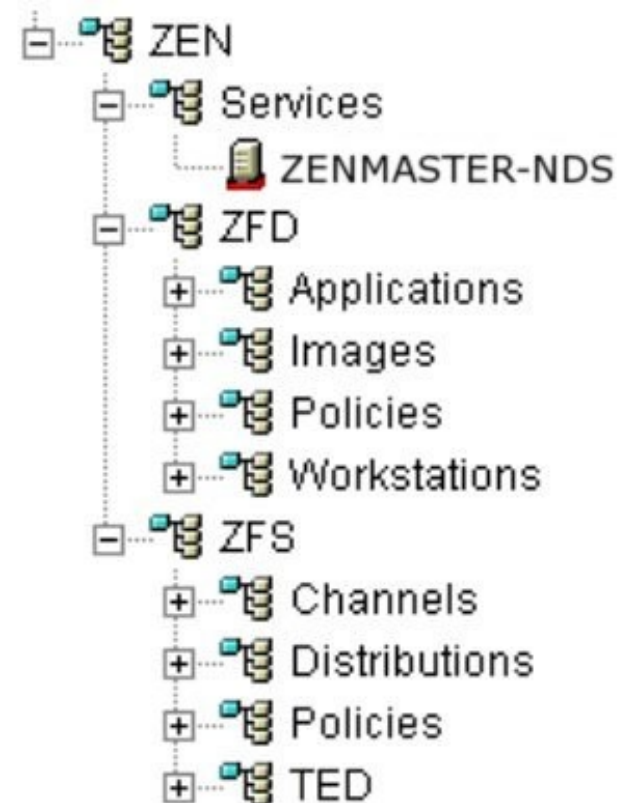
- The Release and Change Management Container structure contains the following characteristics...
  - QA container structure to support testing within the production environment (does not have to be within the production tree)
  - “Golden” or “Mint” container structure for the corporate application portfolio
  - Applications are propagated from here
  - Supports policy based automation



# Design and Architecture

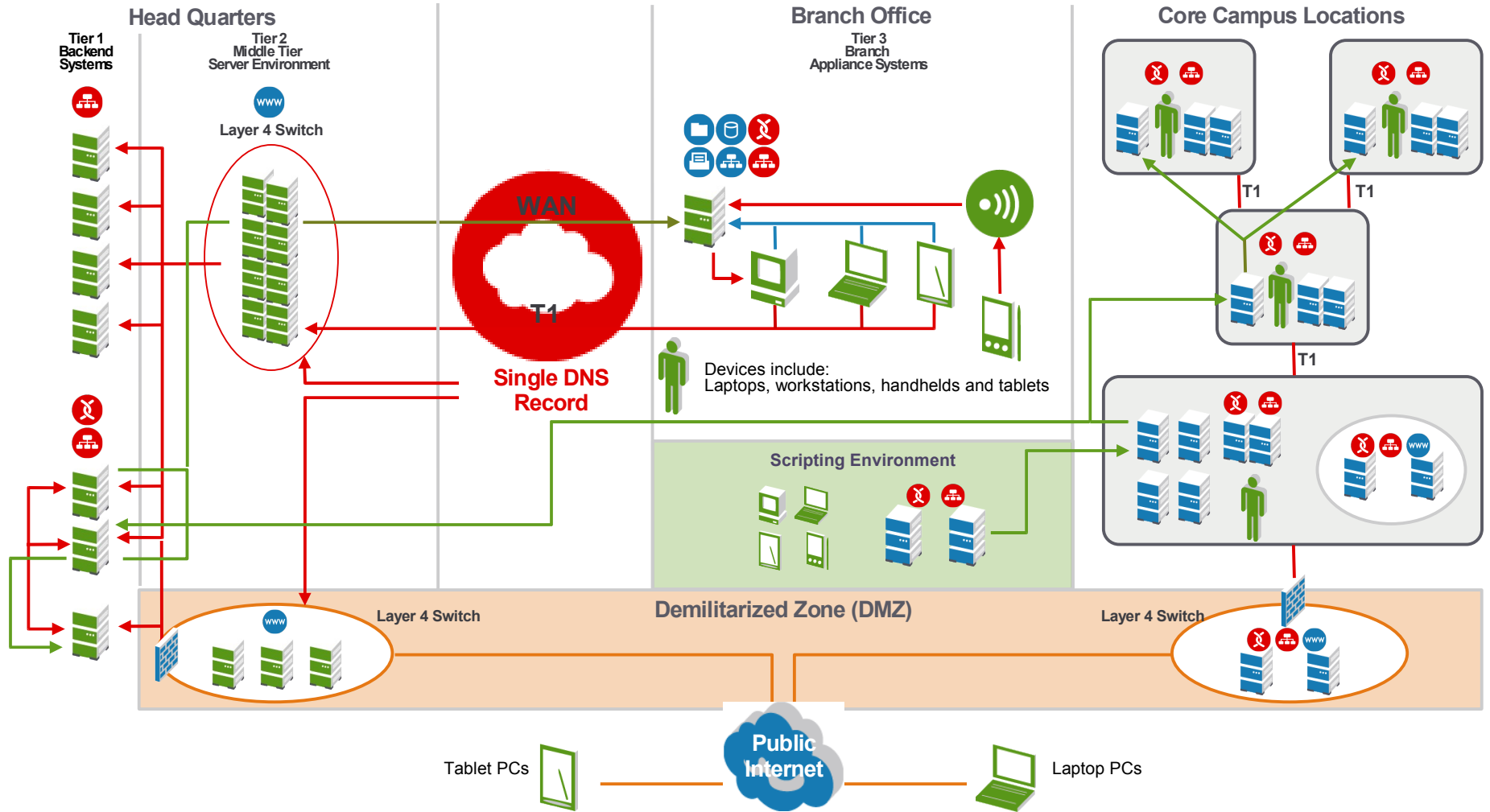
## Release and Change Management Framework

- The Release and Change Management Site Container structure contains the following characteristics...
  - Consistent container structure for site-based applications and other objects...
    - > Policy objects
    - > TED objects
  - The container structure should remain consistent at each location... design, and implement it once
  - Supports policy based automation



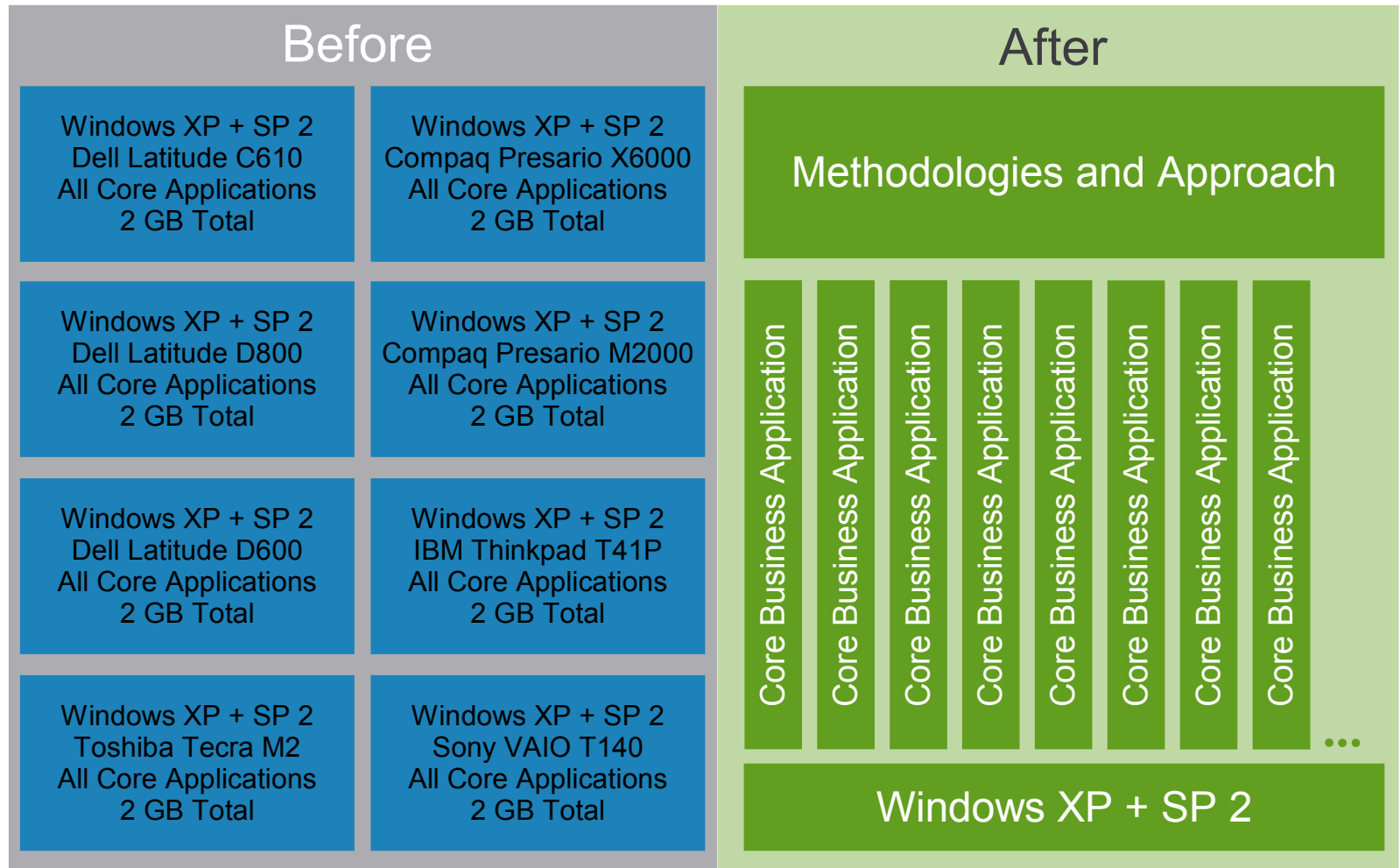
# Sample Architecture

## Centralized or De-Centralized Administration



# Image Management

## Modularize and Simplify



# Upgrading to the Latest ZENworks Simple Upgrade

- Usually means you only have one or two sites to upgrade
  - Use the installer
- Start at the top, and work your way down
- The story stays the same
  - Upgrade schema
  - Upgrade the servers
  - Upgrade your clients

# Upgrading to the Latest ZENworks Complex Upgrade

- Usually means you have a central location, and a number of remote locations
  - Use the installer
  - Use Tiered Electronic Distribution to automate the upgrade
- Start at the top, and work your way down
- The story stays the same (yet again)
  - Upgrade schema
  - Upgrade the servers (possibly automated?)
  - Upgrade your clients

**Novell®**

### **Unpublished Work of Novell, Inc. All Rights Reserved.**

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

### **General Disclaimer**

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

