Cloudhouse Containers White Paper

Version 3.1

A: Ugli Campus, 56 Wood Lane
London, W12 7SB
VAT Number GB 162 2024 53

T: +44 (0)203 51 51 505
E: info@cloudhouse.com
www.cloudhouse.com

Cloudhouse Technologies Ltd.
Registered in England.
Reg. Number 07360219

1 Introduction 3
1.1 Cloudhouse Containers 3
1.2 Support 3
1.3 This Document 3

2 Cloudhouse Containers 4
2.1 Container Overview 4
2.2 Desktop Apps to Windows 10 5
2.3 Desktop Apps to Server 2016 5
2.4 Windows Store for Business and Desktop Bridge (Universal Windows Platform) 6
2.5 Internet Explorer to Internet Explorer 11 6
2.6 N-Tier Server Applications 7
2.7 Evergreen IT and Windows 7
2.8 Improving Security for Business Applications 8

3 Features 9
3.1 Isolation 9
3.2 Redirection 9
3.3 Compatibility 9
3.4 Client-side Integration 9
3.5 Internet Explorer Compatibility 9
3.6 Services and Drivers 10
3.7 Familiar Management and Deployment Experience 10
3.8 Usage Reporting 11
3.9 Technical Specifications 11

4 Auto Packager 12
4.1 Overview 12
4.2 Recommended Platforms for Creating Containers 12
4.3 Supported Operating Systems 12
4.4 Features 13

5 Supported Platforms and 3rd Party Software 14
5.1 Operating Systems 14
5.2 Cloud & Virtualization Platforms 14
5.3 Application Virtualization Solutions 14
5.4 Container Storage 14
5.5 Management and Deployment Solutions 14
5.6 Application and Desktop Publishing Solutions 15
5.7 Layering Solutions 15
5.8 Personalisation Solutions 15
5.9 Assistive Technologies 15

1 Introduction

1.1 Cloudhouse Containers

Enterprises looking to adopt the latest Microsoft and Citrix solutions to transform their business as they move to Windows 10, Office 365 and the cloud find that applications that are core to their business create barriers to adoption.

If the line of business application is incompatible with supported operating systems, IT is forced to continue running outdated operating systems that are vulnerable to attack. Outbreaks like WannaCry and NotPetya are recent examples of how severe the disruption can be and how vulnerable many organisations and public departments are if they remain on legacy operating systems.

Organisations looking to introduce "Evergreen IT", a term that Microsoft *"refers to running services comprised of components that are always up to date. Evergreen IT encompasses not only the services at the user level but all of the underlying infrastructures, whether on-site or outsourced"*. Evergreen IT requires a way to abstract the application from the underlying operating system, so that new updates on the Current Branch, or Current Branch for Business do not affect the behaviour of the application.

IT can improve the security within their organisation by meeting their obligations around security best practices. This includes maintaining their automated patching strategy across all their servers and desktops and avoiding *High* severity failures in security audits (if they can remove unsupported operating systems like Windows XP and Server 2003 from their network).

Cloudhouse's unique redirection and compatibility engine makes it possible for Enterprise customers to migrate their business applications from unsupported, and non-compliant operating systems, to Windows 10, Server 2016 and the cloud. As IT transitions towards "Evergreen IT" they can manage and deploy the applications in Containers to desktops, servers or gold images using their existing deployment tools, for example, Microsoft System Center Configuration Manager (SCCM), ivanti (LANDESK), or build scripts. Applications can then be delivered to a user's Windows device, or published through Citrix XenApp, XenDesktop or Microsoft Remote Desktop Services.

1.2 Support

Customers and Partners with valid support contracts can contact Cloudhouse for support.

1.3 This Document

This document provides a technical overview of Cloudhouse Container, concepts and features. It does not provide specific solution design or implementation advice. Please refer to Cloudhouse Documentation and Knowledge Base for latest details.

2 Cloudhouse Containers

2.1 Container Overview

Containers include all the application files, required runtimes, components, deployment tools, and the embedded redirection and compatibility engine the application requires to run on the new operating system. The Auto Packager creates the Container by packaging the application on the OS that is currently supported by the application e.g. Windows XP. The redirection and compatibility engine, included in the Container, will run in user mode. As the Container is deployed to the target machine, the file type associations are registered, and short cuts created to enable the user to run the application.

Compatibility Containers rely on three key principles:

- Packaging on the operating system on which the application is supported.
- Runtime analysis to detect additional resources the application requires after install.
- The Redirection and Compatibility Engines (AAV) included in the Container.

Figure 1 Cloudhouse Containers

The Auto Packager captures the resources the application's installer/MSI reads from, and writes to, when it is installed on the operating system it is compatible with. At the end of the install capture process, the Auto Packager performs runtime analysis to capture additional application resources that are unknown at install time, for example changes to configuration files or components that are created at run time.

The Redirection and Compatibility Engine intercepts the Windows API calls as the application interacts with the local operating system. The redirected calls abstract the application's files and registry, stored in the Container, from the local resources and the operating system*. This means that, regardless of what modern Windows returns from a system call, the application always gets the resources it expected on the Windows version it was packaged on (e.g. Windows Server 2003 x86, or XP). This light weight approach means that the target application can "see" a combination of the redirected and local file system and registry with negligible performance impact on the application or local machine – the approximate RAM overhead per application is 3-5 MB, with no measurable CPU impact once the application has started.

*The Container does not include the legacy operating system, which means you are not running all, or part of Windows XP on Windows 10.

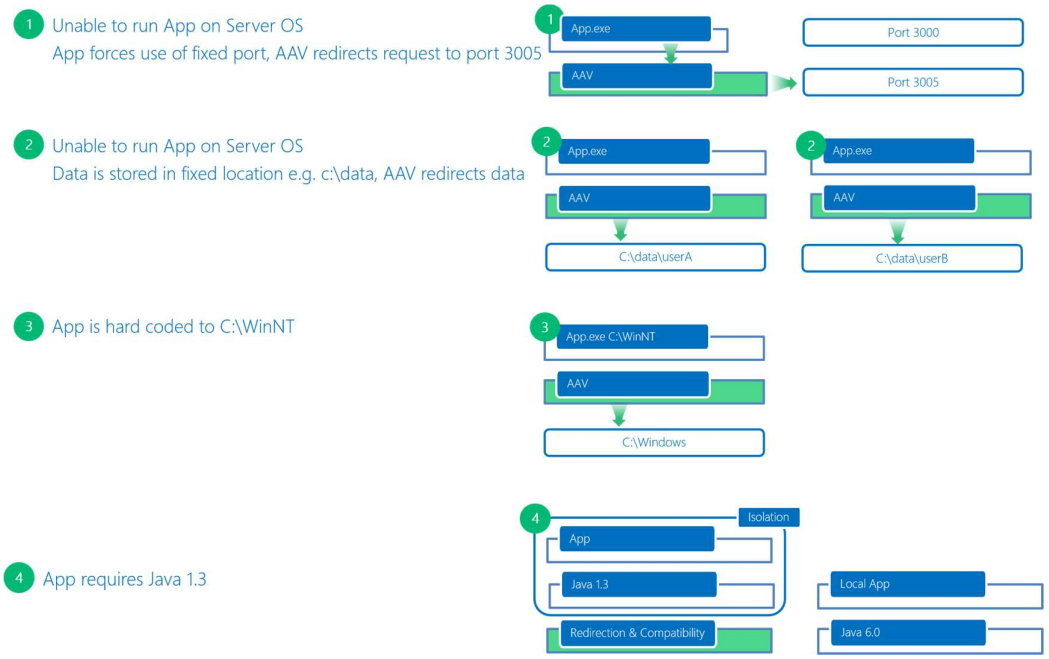

Figure 2 Compatibility and redirection examples

2.2 Desktop Apps to Windows 10

Application and operating system compatibility issues can be overcome enabling Windows XP, Windows 7 32-bit/ 64-bit, and any other desktop applications that are unable to run on Windows 10 64-bit.

Note: Windows 16-bit applications can only be deployed to Windows 10 32-Bit because there is no 16-bit runtime on 64-bit Windows.

2.3 Desktop Apps to Server 2016

Some desktop applications incompatible with a server operating system, are therefore unable to run in a multi-user environment like Remote Desktop Services, or Citrix XenApp. This has forced IT teams to use an expensive VDI solution to solve application incompatibilities issues by running Windows XP or Windows 7 desktops. Organisations looking to eliminate unsupported operating systems like Windows XP can also consolidate to cost-effective XenApp solutions instead.

Containers can solve the following challenges for desktop applications that are unable to run multiple instances simultaneously on a server operating system:

- Applications that bind to a fixed IP address or network port can be redirected to a random port, specified port or IP address.
- Applications that use a global name for their process can be redirected to use a name in the user's local memory name space.
- Applications that write to a shared data location can be redirected to the user's local app data folders.

2.4 Windows Store for Business and Desktop Bridge (Universal Windows Platform)

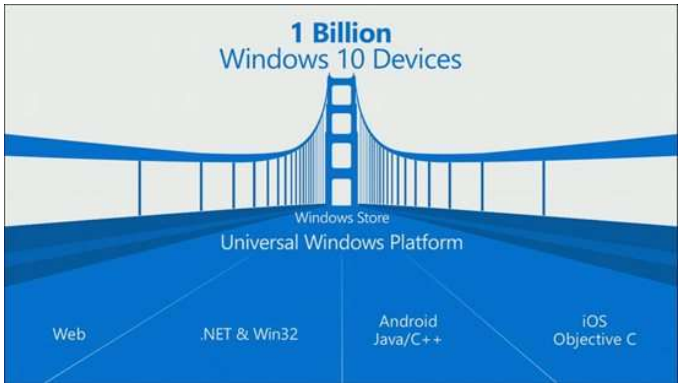

Figure 3 Microsoft's Universal Windows Platform

To deliver applications through Store for Business (a company's private Windows Store), requires the application to be packaged in Microsoft's Universal Windows Platform Appx format. Microsoft provide the Desktop App Converter tool, which can repackage an MSI, or other installer technology, into the required Appx format. However, the application **must** meet all the requirements for UWP otherwise it will be rejected. For companies that are unable to re-write, or modify the source code, packaging in a Cloudhouse Container is an excellent way of overcoming many of the restrictions and rules imposed by UWP. Containers can solve the following types of problems (refer to product documentation for full details):

- Enabling Windows XP, or 7 apps to run on Windows 10
- Legacy runtimes, including .Net 4.6.1 and earlier
- Incompatible install issues, including writes to the local install directory
- Requires Windows services
- Installer requires user interaction

2.5 Internet Explorer to Internet Explorer 11

Browser based applications that require IE 6, 7, 8, 9 or 10 can be run more securely in IE 11, and moved to Windows 10 using Cloudhouse's Containers for IE. The Container can: configure and manage IE Enterprise and Compatibility Modes required by the application; provide browser script compatibility for deprecated APIs; manage multiple and conflicting IE plugins and runtimes like Java and if required, isolate them from others present on the machine. Corporate browser applications typically require security and browsing controls that are specific to the application, and they can be configured and managed so that they apply only to the application without interfering with the general browsing experience, or other IE applications. Administrators can choose to hide the browser controls from the user and restrict browsing capabilities to corporate websites so that the legacy runtimes are not exposed to exploits on the web.

2.6 N-Tier Server Applications

N-tier server applications that run-in data centres on Windows Server 2003, and Windows Server 2008, often prevent organisations moving to the cloud, or completing their data centre consolidation and migration projects, because the application is unable to run on Server 2012 R2 or 2016. The Redirection and Compatibility Engine solves the Server 2003/2008 to Server 2016 migration challenges for applications.

Containers for Server Apps rely on many of the features used in Desktop Apps, additionally it can solve the following specific challenges for server applications that are unable to run on the latest server operating systems

- Applications without media can be packaged using our run time analysis engine to create Containers, the application is monitored to determine what files and registry settings are accessed by the application.
- Overcome dependencies on machine names, IP addresses and ports that were created when the application was installed on Server 2003, can be redirected by the Container so that you can overcome the challenges of packaging applications from running servers without the source media.
- Containers can support Windows services that are common in Oracle Application Server, PeopleSoft, Tomcat, IBM WebSphere or even SQL Server 2005.

Server Apps running in Containers do not change the fundamental architecture, or behaviour of the application

- For example, 3-Tier applications remain 3-Tier applications.
- Scalability and redundancy purposes the same number of servers should be deployed, it is possible that fewer servers may be required because of the performance improvements available in Server 2016.
- High Availability, and Disaster Recover capabilities remain unchanged, if the application provides these then they will be available in Containers.

2.7 Evergreen IT and Windows

The term “Evergreen IT” was coined by PWC in 2009 and included in their Technology Forecast for 2010; it was described as "*at the infrastructure layer, the key concern remains the pooling and virtualization of servers, storage, and networking, so that any dependencies between applications and infrastructure are removed. This removal of dependencies leads to Evergreen IT, in which the various layers of the IT stack can be refreshed independently as needed without constraints from other layers. In essence, Evergreen IT makes IT infrastructure fungible to the point that it can be used for any application, which is fundamental to the extensible enterprise*."

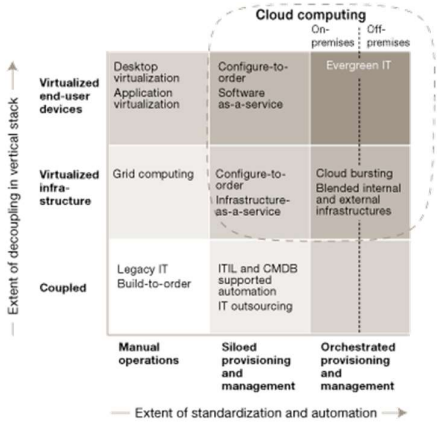

Figure 4 PWC's Evergreen IT

The Container's Redirection and Compatibility Engine abstracts the application from the underlying operating system, so that new updates on the Current Branch, or Current Branch for Business do not affect the behaviour of the application. Containers for applications can be configured so that they are stateless, enabling IT to provision applications on demand adding/removing servers as required.

2.8 Improving Security for Business Applications

Administrators can improve security within their organisation by migrating applications to the latest modern, secure, and supported operating systems so that they can adhere to the latest security best practices including automated patching, and the removal of vulnerable and insecure operating systems on their network.

Cloudhouse Containers provide Isolation and Network Redirections (IP and/or Port) that can complement security products like anti-virus, firewalls and other traditional security products to help protect corporate applications and data.

2.8.1 Create Malware Free Containers

Unlike other application virtualization solutions, Cloudhouse recommend installing anti-virus software on the packaging machine so that the Containers that are created are free* from malware and viruses. The Auto Packager's install capture process can manage differences in the presence of anti-virus software and will retry files that are found to be in use. The vendor's application files are not changed during the packaging processes ensuring vendor signatures remain valid.

*Depends upon anti-virus software and definition files.

2.8.2 Run Malware Free Containers

Containers are compatible with most anti-virus products on the market and because Cloudhouse does not change the application's binaries in any way, packaged files are not viewed as malicious. Creating Containers in the presence of anti-virus software minimises the chances of a virus, or viruses, being included in a Container.

3 Features

3.1 Isolation

3.1.1 Application & Runtime Isolation

Applications with conflicting requirements, or outdated run times, can run safely by isolating them from other applications on the server or desktop. The isolation enables Containers to include runtimes, for example Java 1.3, .Net 2.0 and msxml.dll, that may conflict with other runtimes present on the server, or older versions of Word and Excel that should only be used by the containerized application.

3.1.2 Network Isolation

To help reduce application vulnerabilities, Containers can be configured to complement customer's existing firewall products and provide isolation at the TCP/IP layer by hooking Winsock APIs. The Container can be configured to block all traffic, or only enable communication via specific ports or IPs to the runtimes, or applications.

3.2 Redirection

3.2.1 File and Registry Redirection

By packaging applications in Containers on the operating system it supports, the Auto Packager can create redirections for the registry and file paths including hard coded locations that may not exist on the modern operating system enabling Containers to overcome install and run time problems.

3.2.2 Network Redirections (IP and/or Port)

The Redirection and Compatibility Engine is able to hook Winsock APIs, so that redirection rules can be applied to IP addresses and ports, enabling desktop applications to avoid conflict with other instances of the application when it runs on a server operating system.

3.2.3 Memory and Process Redirection

The Redirection and Compatibility Engine can hook kernel objects, and provide mutex, event and semaphore isolation and redirection to help desktop applications to run on server hosted desktops, for example XenApp and RDS.

3.3 Compatibility

3.4 Client-side Integration

By default, applications behave as though they’re installed natively, so that they can access other applications, printers, and devices. Isolation should be enabled if this is not the desired behaviour.

Traditionally, user mode application virtualization solutions have been tightly sandboxed and unable to go beyond their virtual environment and integrate with local devices. Applications don’t need to be packaged together, or run in the same Container, to work together because the Container’s user mode redirection engine supports integration from the Containerised application to any local device or application by hooking into locally installed processes (either all processes or named processes). This includes full support for inbound integrations from Microsoft Office Plugins, and toolbars because a Cloudhouse plugin can be set to load automatically each time a Microsoft Office application starts.

3.5 Internet Explorer Compatibility

The following features are available for browser based applications that require unsupported versions of Internet Explorer (IE 6 through to 10) to run in IE 11 on Windows 10.

3.5.1 Manage IE Enterprise / Compatibility Modes

Containers can configure and manage the appropriate browsing modes on the end-point so that the application uses the correct rendering engine to display in IE 11.

3.5.2 Browser Script Compatibility

Cloudhouse Containers provide Browser Script Compatibility shims for popular APIs that are no longer supported in IE 11, for example Java script.

3.5.3 Runtime Isolation

Browser applications that require Java, .Net, or other runtimes benefit from the Application & Runtime Isolation provided by Containers.

3.5.4 Manage Configuration and Security Settings

Delivering a single browser experience for multiple IE applications can be a challenge, Containers overcome this by managing the conflicting security and browsing controls required for multiple browser applications to run in different tabs.

3.5.5 Restrict Browsing

URL browsing can be locked down to corporate websites, so that users are unable to follow links to non-corporate websites, which may increase the risk of exploits.

3.5.6 IE as an Application

The user can start the application from a short cut, and browser controls can be locked down and hidden from the user restricting their ability to browse the web and reducing the exposure to exploits.

3.6 Services and Drivers

More common for server-side applications, Containers can support Windows Services, and include MSIs for drivers, which can be deployed at install time.

Note: Drivers are not virtualized and will need to be compatible with the target operating system; drivers are typically deployed by the Container using MSI files as part of the deployment process.

3.7 Familiar Management and Deployment Experience

Cloudhouse Containers can be managed just like any Windows application (MSI, exe), requiring no additional investment in training, infrastructure or changes to established practices. Containers can be deployed with popular management and deployment tools, including AD GPOs, SCCM, ivanti, and custom scripts (refer to Section 5 Supported Platforms and 3rd Party Software). Once deployed, the containerised application can be started by users through short cuts and file type associations or published by Administrators using Citrix or VMware's consoles and wizards.

3.7.1 Instant Deploy and Run

Containers can be deployed to any Windows platform, or a stateless server, without requiring client-side software to be installed first because the required user mode runtime is included in the Container. Applications can be deployed on demand (no reboot required) using the deployment tool included in the Container; which performs the copy across the network, short cut creation and file type association registration. Containers can also run from a central file share without being deployed onto the server or desktop. The time to install across the network is determined by size of the application, speed and bandwidth of the network.

3.7.2 Containers in App-V

For customers who want to standardise on App-V packages, Cloudhouse Containers can be packaged using App-V, so that they can be deployed and managed using the publishing server.

3.8 Usage Reporting

Businesses can understand application usage patterns through Usage Reporting. This includes details on who is using what application, on which OS, when, and how often. Application usage and deployment data is recorded on a central file share in an open format (CSV) so that it can be imported into a customer's existing 3rd party system.

3.9 Technical Specifications

Please check on-line documentation for full details of requirements to install the Auto Packager and run Containers, in summary the

- Redirection and Compatibility Engine is written in C++ and does not require any C runtimes to be installed.
- Cloudhouse Containers and the Auto Packager require .NET Framework 4.0 Client Profile to be installed.
- 4MB client binaries per Container.
- 2-5MB RAM per Container (approx.)
- 2-3% CPU per Container (approx.)

4 Auto Packager

4.1 Overview

The Auto Packager streamlines the creation of Containers for legacy, bespoke and modern Windows applications so that they can run on any of the supported target operating systems. It uses a snapshot and install capture process to create a Container that includes all changes made during the install of the application. Once the Container creation process is complete, the application can be started and run time analysis performed to detect any additional configuration changes that occur when the application runs for the first time. If required, the packaging process can be automated with scripts or driven by tools like Citrix AppDNA.

4.2 Recommended Platforms for Creating Containers

- For Legacy applications use Windows XP SP3 x86 or Server 2003 SP2 x86.
- For Modern applications use Windows 7 x86 if cross platform compatibility is required.

4.3 Supported Operating Systems

- Windows XP SP3 x86
- Windows 7 x86 or x64
- Windows 8.1 x86 or x64
- Windows 10 x86 or x64
- Windows Server 2003 SP2 x86
- Windows Server 2008 x86
- Windows Server 2008 R2 x86 or x64
- Windows Server 2012 x86 or x64
- Windows Server 2016 x86 or x64

4.4 Features

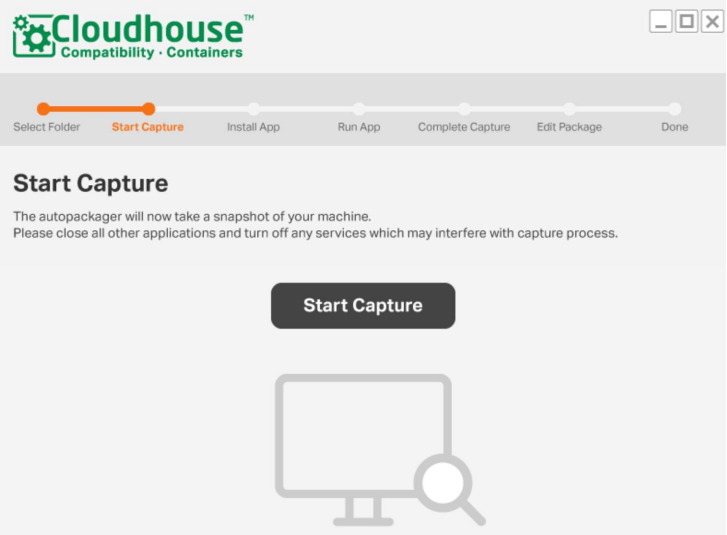

Figure 5 Auto Packager

4.4.1 Install Capture

During packaging a snapshot of the current state of the operating system is taken and a capture process started to record all changes during the install of the application.

4.4.2 Run Time Analysis

Applications typically change the file and registry when they are started and configured for use. For this reason, the Auto Packager offers run time analysis to capture the changes and include them in the Container.

4.4.3 Legacy OS Support

Containers are created on the operating system currently supported by the application e.g. Windows XP. This enables the application to be packaged as the supported operating “sees it”, which differs from the target operating system. In this way applications that include pre-install checks that look for a specific operating system can be handled.

4.4.4 Cross Platform Support

A single Container can be deployed across multiple supported operating systems, if they are packaged on Windows 7 x86 with all versions of .NET Framework installed.

4.4.5 Pre/Post Deployment Scripts

Custom scripts can be included in the Container, and used to deploy MSIs, or execute other tasks that need to be performed as part of the Container deployment process.

4.4.6 Fast Edit, Zero Recompilation

The time and frustration packaging teams face when packaging, testing and updating applications and configurations is reduced with fast editing, which requires no package recompilation. Containers can easily be updated to include application updates or upgraded to include the latest Cloudhouse components.

A Container’s configuration is defined in a series of XML files: Registry writes (AppRegistry.xml), File, Registry and Network redirections (Redirections.xml), File Type Associations (FileAssociations.xml), Shortcuts (Shortcuts.xml), Executable programs (Programs.xml) and Environment Variables (EnvironmentVariables.xml).

5 Supported Platforms and 3rd Party Software

5.1 Operating Systems

- Windows 7 x86 or x64
- Windows 8.1 x86 or x64
- Windows 10 x86 or x64
- Windows Server 2008 R2 x86 or x64
- Windows Server 2012 x86 or x64
- Windows Server 2016 x86 or x64
- Microsoft Azure (Certified for Azure)

5.2 Cloud & Virtualization Platforms

Supported Windows operating systems running on

- Amazon Web Services
- Citrix XenServer
- Microsoft Hyper-V
- Microsoft Azure (Certified for Azure)
- VMware ESX

5.3 Application Virtualization Solutions

Applications running in Containers are compatible with other application virtualization solutions deployed on the server or desktop. Cloudhouse Containers can communicate with applications packaged using these technologies, unless the Container or the 3rd Party application virtualization package has been configured for isolation.

- Microsoft App-V, additionally Containers can be packaged and delivered using App-V.
- Numescent
- VMware ThinApp

5.4 Container Storage

- Local disk
- UNC file shares

5.5 Management and Deployment Solutions

- AD Policies
- Citrix XenMobile

- ivanti (LANDESK)
- Microsoft System Center Configuration Manager
- Microsoft Store for Business
- Windows InTune
- Custom Scripts
- Other 3rd party image management and software deployment tools

5.6 Application and Desktop Publishing Solutions

- Citrix XenApp and XenDesktop (Certified Citrix Ready)
- Microsoft Remote Desktop Services
- VMware Horizon and View

5.7 Layering Solutions

- Citrix App Layers, and App Disks
- FSLogix
- VMware AppVolumes

5.8 Personalisation Solutions

- ivanti (AppSense)
- Roaming Profiles
- Citrix Unidesk

5.9 Assistive Technologies

Containers run in user mode on Windows, and use the local graphics engine of Windows to display the application so AT solutions are all supported, for example

- Dragon
- JAWS
- Microsoft tools in Windows, for example Speech Recognition, Magnifier
- “Read and Write Gold”
- SuperNova