Table of Contents

Why Is App Delivery Important? ................................................................. 4
  What Are the Challenges? ................................................................. 5
How Does VMware Horizon Meet the Challenge? .................................. 5
  About Horizon Editions ................................................................. 7
  About Horizon Platforms .............................................................. 7
Using View as an Application-Delivery Platform ............................. 7
Using VMware Workspace Portal as an Application-Delivery Platform .. 8
Using VMware Mirage as an Application-Delivery Platform ............. 10
Section Summary .................................................................................. 11
Delivering RDS-Hosted Applications ................................................. 12
  About RDS-Hosted Applications .................................................. 12
  Using View to Deliver RDS-Hosted Applications ....................... 12
  Using Workspace Integrated with View to Deliver RDS-Hosted Applications ... 13
Benefits of Using RDS Hosting ........................................................... 14
Use Cases Where RDS Hosting Is the Best Option ........................... 15
RDS Hosting in Action ........................................................................ 16
Section Summary .................................................................................. 18
Delivering ThinApp Packages ............................................................. 19
  About ThinApp ............................................................................... 19
  Using Workspace Integrated with View to Deliver ThinApp Packages ... 19
  Using Workspace to Deliver ThinApp Packages ......................... 21
  Using View to Deliver ThinApp Packages ................................... 22
  Using Mirage to Deliver ThinApp Packages .................................. 23
Benefits of Using ThinApp ................................................................. 23
Use Cases Where ThinApp Is the Best Option ................................ 24
Section Summary .................................................................................. 25
Delivering Citrix XenApp Published Applications ............................ 26
  About Using Workspace with XenApp ......................................... 26
  Delivery Workflow ......................................................................... 27
  Using Workspace to Present XenApp Published Applications .......... 27
  Using View to Deliver XenApp Published Applications ............... 28
Use Cases Where XenApp-Workspace Integration Is the Best Option .. 28
Section Summary .................................................................................. 29
## Delivering SaaS- and Cloud-Based Applications

About SaaS- and Cloud-Based Applications .................................................. 30
Delivery Workflow ......................................................................................... 31
Using Workspace to Present SaaS- and Cloud-Based Web Applications ...... 31
Using View with SaaS- and Cloud-Based Web Applications ..................... 32
Benefits of SaaS- and Cloud-Based Applications ........................................ 32
Use Cases Where Cloud Delivery Is the Best Option .............................. 32
Section Summary ......................................................................................... 33

## Delivering Natively Installed Windows Applications

Conclusion ...................................................................................................... 34
Additional Resources ............................................................................... 35
About the Author and Contributors ........................................................ 36
Why Is App Delivery Important?

Everybody uses applications, from the CEO of an international enterprise to the receptionist in the lobby. Even pizza deliverers use smartphones to take orders and GPS mapping software to deliver. There is an ever-increasing variety of applications and online services that keep the wheels of business turning. And when the applications do not work well, neither does the business. Unfortunately, if the performance of a key application degrades, often the end user notices before IT does. That can translate to loss of revenue, loss of users, or loss of funding.

Consequently, IT no longer focuses on desktops alone when managing user activities. Especially when users work with mobile devices like smartphones and tablets, the applications on the devices are often even more central to the business than desktops are. Therefore, IT administrators increasingly consider software application delivery a top priority.

A number of application-delivery options have been developed in response to this growing need, as well as new terms to talk about those options. For the purposes of this paper, software application delivery—also called app delivery—refers to any method used by IT administrators to make applications available to their end users. For this discussion, app delivery includes methods such as streaming, where the application is not actually placed on the endpoint device at all.
What Are the Challenges?

Application delivery is more vital than ever and also as challenging, with seemingly opposite goals of IT and end users. End users want optimal performance, but as an IT administrator, you need to ensure effective management, appropriate security, and reasonable cost.

• **Demand for a single pane of glass** – End users want easy access from a single pane of glass: one place where they can access all applications and services, from any device, from any location. You need to provide that without compromising cost or ease of management.

• **Demand for self-service** – Instead of sending a request to an overstretched IT department and waiting in the ticket queue for delivery, end users want to download an application themselves from consumer Web sites, and expect to use that application immediately. You need to be able to manage such flexibility while maintaining reasonable security.

• **Demand for better performance** – Although end users want top performance, optimizing app delivery is more than simply improving performance. You need the ability to distinguish among the different types of traffic on the network, facilitate the performance of business-critical applications, and exclude—or at least monitor—irrelevant applications.

• **Increasing variety of applications and devices** – End users want the ability to access their applications on an increasing variety of devices, including PCs, Macs, diskless computers, laptops, netbooks, tablets, smartphones, and more. You need to provide secure access from all of these devices.

• **Supporting both modern and legacy applications** – Despite recent trends toward more smartphone and tablet use, most businesses still maintain legacy applications that were not designed for smartphones, as well as data in those applications. You need to be able to manage both modern and legacy technologies simultaneously.

In the face of these challenges, application users and IT administrators alike have been eager for viable options. VMware® Horizon™ 6 addresses these challenges.

How Does VMware Horizon Meet the Challenge?

To meet the demand, VMware has expanded its software application-delivery options with the release of Horizon 6.0. VMware is already known for providing virtualized applications through VMware ThinApp™, and centralized application and desktop management through View™. Horizon now adds a new option: application remoting based on Microsoft Remote Desktop Services (RDS).

RDS, formerly known as Terminal Services, is a Microsoft technology that enables remote users to share applications installed on servers in the data center, as well as to share session-based desktops.

RDS hosting of applications is also sometimes referred to as **app publishing** or **app remoting**. RDS hosting provides users with access to applications that are installed on a remote RDS host. In Horizon, Remote Desktop Services hosts (RDS hosts) deliver Windows-based applications or desktops. The RDS host is a server containing both Microsoft RDS and View Agent™. By installing an application on an RDS host, you can make a single instance of an application available to thousands of users, who access it remotely.
Horizon has supported many app-delivery options in the past, including the delivery of ThinApp packages, integrated Citrix XenApp published applications, SaaS- and cloud-based applications, and natively installed Windows applications. With the addition of the new ability to deliver RDS-hosted applications, the result is an inclusive delivery infrastructure that can cater to a variety of needs:

- App delivery from RDS hosts in Horizon
- App delivery from ThinApp repositories
- App delivery from integrated Citrix XenApp farms
- App delivery from cloud-based service providers
- App delivery of natively installed Windows applications

Figure 2: Horizon Delivers Any App to Any Device at Any Time
About Horizon Editions
Horizon 6 is available in three editions: VMware Horizon View Standard Edition, VMware Horizon Advanced Edition, and VMware Horizon Enterprise Edition. Horizon View Standard Edition includes View and ThinApp, and supports application delivery through ThinApp. Table 1 summarizes the application-delivery features of the three editions of Horizon:

<table>
<thead>
<tr>
<th>APPLICATION DELIVERY</th>
<th>HORIZON VIEW STANDARD</th>
<th>HORIZON ADVANCED</th>
<th>HORIZON ENTERPRISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Catalog – Citrix XenApp, RDS-Hosted, SaaS, ThinApp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RDS-Hosted Applications</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ThinApp Packages</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: Application-Delivery Features in Each Edition of Horizon 6.0

About Horizon Platforms
Horizon Advanced and Horizon Enterprise editions include three platforms that can be used for app delivery: View, VMware Workspace™ Portal, and VMware Mirage™.

Although View, Workspace, and Mirage have many other capabilities, only their app-delivery functionality is discussed in this paper. Each platform supports different sets of capabilities that extend the delivery infrastructure:

- View
- Workspace
- Mirage

Using View as an Application-Delivery Platform
All three editions of Horizon include View, one of the three platforms for delivering applications to users in Horizon. At its simplest, View displays Windows applications installed on a virtual desktop. You can also set policies in View to control who has access to the applications. View has the ability to remotely display the following types of applications:

- RDS-hosted applications (also called published, remoted, or hosted applications)
- ThinApp packages
- Applications installed in base images

In Horizon View Standard Edition, you can use View to remotely display virtual desktops on endpoint devices, which includes both RDS-hosted desktops and View virtual desktops. You can also display virtualized Windows applications (ThinApp packages) on these desktops, alongside the natively installed Windows applications. RDS-hosted applications are not supported in Horizon View Standard Edition, although multi-user RDS-hosted desktops are.

In the Horizon Advanced and Horizon Enterprise editions, View can display RDS-hosted applications, in addition to the above. You can install an application on an RDS host and display the RDS-hosted application to many endpoints. You can also use View to deliver both multi-user RDS-hosted desktops and View virtual desktops. (RDS-hosted desktops are shared desktops on an RDS host. View desktops are created in VMware vSphere®, are not shared, and are not on an RDS host.)
You can also use View to display XenApp published applications within a virtual desktop, although this option in View contains no special integration with XenApp. See Integrating VMware View and VMware ThinApp with Citrix XenApp for more details.

VMware Horizon Client™ is required on the View desktop. Users log in, and can be authorized using their Windows Active Directory (AD) credentials. In addition, View supports secure HTML access to View desktops without installing client software on the endpoint device. For more information, see the View Administration guide.

View also includes a feature called Unity Touch, which provides a gesture-based option for accessing Windows applications on iPads or other tablets. Because of small screen size, using mobile devices for navigating applications and files can be challenging. Unity Touch makes that easier. Unity Touch allows users to browse, search for, and open virtualized and RDS-hosted Windows applications, and switch between multiple open applications, without the Windows Start menu or task bar. The file manager, task bar, and Start menus are also responsive to touch navigation. For more information, see Introducing Unity Touch.

For more information about View, see the Horizon with View Documentation.

Using VMware Workspace Portal as an Application-Delivery Platform
VMware Workspace Portal, one of the platforms in Horizon for delivering applications to users, is included in the Horizon Advanced and Horizon Enterprise editions, but not in Horizon View Standard Edition. Workspace provides a central, unified portal where applications of all types can be displayed to your end users through an HTML browser.

Workspace can aggregate and display
• RDS-hosted applications
• ThinApp packages
• Citrix XenApp published applications
• SaaS- and cloud-based applications

Workspace works on
• View desktops and RDS-hosted desktops
• Physical and virtual machines
• Both Windows and non-Windows devices (Mac, Linux, iOS, and Android)
Workspace provides a single pane of glass that gives users convenient access to all of their applications and desktops, regardless of type. You can make applications available by default in VMware Workspace User Portal, the end-user interface of Workspace. In the Workspace User Portal, users can see all of the applications that you entitled them to access and arrange the applications as they like, as shown in Figure 4.

Behind the scenes, you control what your users can access through the Workspace application catalog, the repository of all the resources that you can entitle to your users. The availability of particular resource types in your catalog is controlled by which modules are enabled in Workspace. For example, if you enable the Web Applications module, you can entitle your end users to SaaS- and Web-based applications; enabling the Citrix Published Applications module gives you the ability to make XenApp applications available, too. You can entitle users and groups to specific applications and desktops through Workspace, and allow users to self-select from the menu of resources that you have pre-approved.

**Integrating Workspace with View**

Integrating Workspace with View adds single-sign-on (SSO) capability and the Workspace catalog. Workspace aggregates your ThinApp packages with all applications and desktops in your delivery infrastructure, including RDS-hosted applications, Citrix XenApp published applications, SaaS- and cloud-based applications, RDS-hosted desktops and View desktops, and even static Web links.

To integrate View with the Workspace platform, you must configure Workspace with the View information, and configure View to delegate responsibility for authentication to Workspace. Workspace and View must be integrated using the Security Assertion Markup Language (SAML). This establishes mutual trust, which is essential for single-sign-on functionality. With SSO enabled, users who log in to Workspace with Active Directory (AD) credentials can launch applications and desktops from View without having to log in a second time. See [Integrating VMware Horizon Workspace and VMware Horizon View](#).

You create a SAML Authenticator in View to contain the trust and metadata exchange between View and Workspace. After completing the configurations, you add SAML Authenticator information in View, and use VMware Horizon Connector™ to make sure the information is saved and synchronized. After this, any time you make changes in View—such as adding, removing, or modifying applications or desktops—you must sync Workspace with View. The sync process can be done manually, or based on a pre-defined schedule.

After Workspace and View are integrated and synced, your end users can access RDS-hosted applications, View and RDS-hosted desktops, SaaS- and cloud-based applications, and even static Web links through Workspace. To include Citrix XenApp published applications, the Citrix Receiver is required on the endpoint. To
Application-Delivery Options in VMware Horizon 6.0

include ThinApp packages, Horizon Client is required on the endpoint to enable access to View desktops, and Workspace for Windows is required on the endpoint for Workspace entitlement of ThinApp packages on the View desktop. Over HTTP or HTTPS, Workspace can deliver ThinApp packages, and View allows remoting of desktops (via the Blast HTML5 client).

Workspace includes a collection of SaaS- and cloud-based applications that can be included in the Workspace catalog out of the box, such as Google Docs, WebEx, Microsoft Office 365, Salesforce, Axiom, and more. This collection of pre-configured applications is dynamic and is expanded continually. Your end users see only those applications that you enable them to see.

You can also add SaaS- and cloud-based application links that are not included by default. Workspace can integrate Web application links for applications that are either SAML or non-SAML. You can also easily create Web application links that do not require user authentication, and integrate them into the catalog, such as a widely used Web site or Web application that is not already included in the Workspace catalog. You can also use Workspace to add links that resolve to static Web pages. Any subsequent user authentication to these static links can be handled manually. The result is a single application portal for end users that is easy to use and includes every kind of cloud application.

See the VMware Workspace Portal Documentation for more information about Workspace.

Using VMware Mirage as an Application-Delivery Platform

VMware Mirage is one of the platforms in Horizon for delivering applications to users. Mirage is available in the Horizon Advanced and Horizon Enterprise editions only, and not in Horizon View Standard Edition.

One of the features of Mirage is layered images: base layers and application layers. The Mirage base layer includes the operating system and can also include applications you want everyone to have. These applications can be ThinApp packages or natively installed Windows applications. You can also use Mirage to create discrete app layers that contain applications only, and you can deliver these app layers to Mirage-managed devices, independently of the Mirage base layer. These app layers provide customized sets of applications for specific groups of users.

App layers can contain ThinApp packages and natively installed Windows applications. App layers cannot contain RDS-hosted applications, XenApp published applications, or SaaS- or cloud-based applications.

See also Using Mirage to Manage ThinApp Applications.

Figure 5: Mirage App Layers Support Applications That Include Drivers and Other Elements

Mirage app layers provide an alternative app-delivery platform to mitigate many of the difficulties that traditional app-delivery methods are vulnerable to, such as network issues, mismatched configurations on the endpoint, interruption during installation, and so on. Mirage app layers can also include application elements that cannot be virtualized with ThinApp, such as device drivers. See the VMware Mirage Documentation for more information about Mirage app layers.
Section Summary

With the plethora of devices today, you need more diverse delivery options to connect users to the applications they need to do their jobs. Horizon provides that diversity. The challenge is figuring out which delivery option—or combination of options—works best for your environment.

If your enterprise needs an app-remoting mechanism or has legacy applications to maintain, you can focus on RDS hosting. If your enterprise needs an application virtualization mechanism, you can use ThinApp virtualization. If your enterprise already has an investment in Citrix XenApp, you can explore Citrix XenApp integration or migration from XenApp to Horizon. Or if your company subscribes to SaaS Web applications or other cloud-based applications, see SaaS and cloud delivery. However, if your environment actually includes a combination of different application types, a hybrid approach is often the best option.

When choosing an option, you need to consider a number of elements, such as your business goals, what types of applications you need to deliver, what types of devices you need to deliver to, and what kind of infrastructure or platforms you are delivering from. The choice of a delivery mechanism often comes down to how you want to manage each application and its updates. Your business use cases are your best guide. Horizon app-delivery options are described in the remainder of this paper, along with use cases where each option is most advantageous:

- App delivery from RDS hosts in Horizon
- App delivery from ThinApp repositories
- App delivery from integrated Citrix XenApp farms
- App delivery from cloud-based service providers
- App delivery of natively installed Windows applications
- App delivery using a hybrid approach
Delivering RDS-Hosted Applications

To meet the demand for hosted applications, the Horizon Advanced and Horizon Enterprise editions support a fundamental app-remoting option based on Microsoft RDS. You can publish and manage RDS-hosted applications through View, including policy and entitlement setting. You can also integrate Workspace with View to present RDS-hosted applications to your users in Workspace, alongside applications from ThinApp repositories, Citrix XenApp farms, and SaaS and Web application providers. This section describes the following:

- Delivering RDS-hosted applications
- Benefits
- Use cases
- RDS hosting in action

About RDS-Hosted Applications

RDS is a Microsoft Windows component that supports access to and use of remote machines and applications through a network connection. RDS transfers only the user interface to the client system, and then transmits all input from the client system back to the server to be executed there. The RDS-hosting option in Horizon with View provides the essentials for publishing applications. You can install one instance of an application on an RDS host instead of on multiple individual desktops, and make it available to many end users.

Using View to Deliver RDS-Hosted Applications

In a View-only deployment without Workspace, the end user logs in to View. View authenticates the user, and sends the policy and entitlement information to the RDS farm. Unlike RDS farms in Microsoft which are based on Microsoft clustering technology, an RDS farm in Horizon is simply a group of RDS hosts that are managed in View. The RDS host then launches the application.

![Diagram](image)

**Figure 6:** RDS-Hosted Application Delivery Through View

You can use View to apply policies and entitlements, as well as to display RDS-hosted applications. View presents the applications using the PCoIP protocol over the network.

See View Administration for more information about View.
Using Workspace Integrated with View to Deliver RDS-Hosted Applications

To offer RDS-hosted applications in Workspace to your users, you must integrate Workspace and View. This enables your entitled end users to see an aggregation of all their applications in the Workspace User Portal. These applications can include RDS-hosted applications, desktops, ThinApp packages, and SaaS- and cloud-based Web applications.

You can set policies and entitlements for RDS-hosted applications only in View, and synchronize Workspace with View. You can apply View policies to all client sessions, or to specific desktop pools or users, to enable multimedia redirection, USB access, and so on. Workspace reads the policy information from View and aggregates the RDS-hosted applications and RDS-hosted desktops into the Workspace User Portal for the entitled users to access.

![Diagram of RDS-Hosted Application Delivery Through View and Workspace](image)

In an environment where Workspace and View are integrated, the end user logs in to Workspace, and can use single sign-on for applications where SSO is set up. Workspace authenticates the user, and opens a read-only connection with View Connection Server, which communicates the policy and entitlement information to the RDS farm. The application is launched from the RDS host in the farm. The connection between the RDS host and the user is maintained, and the connection between Workspace and the user ends. Nothing is tunneled through Workspace, so if Workspace were to fail, the existing session between the end user and the application on the RDS host would remain connected. The user would not be able to reconnect or re-authenticate, but any existing sessions would stay connected until timed out.

See the [VMware Workspace Portal Administrator’s Guide](#) for more information about Workspace.
Benefits of Using RDS Hosting

The addition of RDS-hosted app delivery to Horizon adds flexibility to your application-delivery options. This provides a number of benefits:

• **Supports many-to-one access** – RDS hosting achieves a higher user density than many other delivery options because you can deploy a single instance of an application on one RDS host instead of on multiple endpoints, and make it available to hundreds of end users. When connected to the RDS server, your users can launch the applications, save files, and use network resources from that remote server—all appearing to the user to be running locally.

• **Reduces infrastructure cost** – You can install a single instance of an application on a remote RDS-enabled server, and provide access to that application to many users. This reduces software costs because you buy one application instead of many. It cuts down on maintenance as well, because you install, upgrade, or troubleshoot the one hosted application, and the next time users log in to the application, they automatically get the latest patched or updated version.

• **Optimizes application performance** – When installed on an RDS host, an application instance is in close proximity to the data center, which often resolves latency issues. Especially with multi-tier architectures, applications often perform better when all components are geographically close. When a user accesses a database remotely, large amounts of data must be transmitted over the WAN, which can impact system performance. However, if the device hosting the application is in close proximity to the database, this traffic is isolated and only the screen updates are sent across the WAN. You can also set up an application tier with more than one RDS host in a farm. If one host has a problem, you can take it down, and the workload is spread across the remaining servers.

• **Supports SPLA licensing** – The RDS-hosting option in Horizon supports Services Provider License Agreements (SPLA), which provides a licensing advantage for RDS-based service providers. There are often restrictions on VDI service providers offering desktops on a pro rata subscription basis, so their customers must have their own licenses. SPLA licensing does not have these restrictions, so RDS-based service providers can offer access to RDS-hosted applications using the SPLA licensing model.

• **Supports both modern and legacy applications** – If you need to manage legacy applications alongside modern ones, RDS hosting can be useful. You can install multiple versions of the same application on different RDS hosts, and make them all available to users. In addition, when you publish ThinApp packages and other Windows applications on RDS hosts, your end users can consume those applications on non-Windows devices such as Mac, iOS, Android, and Linux. Printing to network-based printers is also fully supported on RDS-hosted applications.

• **Provides a single-vendor alternative** – If you need to deliver individual applications as well as virtual desktops, the addition of the RDS-hosting option to VMware Horizon provides the ability to do both with a single solution. For businesses that have an existing virtual desktop infrastructure (VDI), the RDS-hosting option in Horizon extends delivery capabilities to include RDS-hosted applications as well as ThinApp packages. And for businesses that have an existing Citrix XenApp infrastructure and are re-evaluating the need for a major upgrade to that infrastructure, RDS hosting through Horizon opens new opportunities for migration or maintenance.

RDS hosting provides a number of advantages that, together with the other app-delivery options in Horizon, provides you with the tools to mitigate costs and complexities of application delivery and management.
Use Cases Where RDS Hosting Is the Best Option

RDS hosting has been around for a long time and meets a real need. The addition of the RDS-hosting option in Horizon extends your ability to apply the best combination of technologies for each use case:

• **Use Case: Providing greater security** – Medical and financial institutions, and call centers are examples where RDS hosting often makes more sense than providing desktops to individual users, because remoting an application strengthens authentication and security. For example, the Epic System—a software suite for patient care, registration, and scheduling—is widely used by medical personnel. The medical personnel need to be able to see their Epic system, and upload data to the server, but do not typically need to store data on their local machines. IT can install the latest version on an RDS host, update when necessary, and every nursing station has an instantly updated version. When an application is remoted, confidential information is stored in the data center instead of on user devices, and access is fully authenticated.

• **Use Case: Managing frequent upgrades and multiple versions** – Applications with frequent upgrades are good candidates for RDS hosting, because you can move users from one version to another easily. You can also make multiple versions of the same application available to your users, as long as they are installed on separate RDS hosts. For example, when you need to upgrade, you can set up an additional RDS host and install the latest version of the application on the additional host while leaving the old application running. Using parallel environments, you can test the new application, and decommission or upgrade the old application when ready.

• **Use Case: Delivering applications without virtualization** – If you need to deliver applications that require kernel-based device drivers or need operating system integration, virtualizing does not always work. RDS hosting often provides the best option in these cases because the applications are installed directly on the RDS host along with any required drivers, integration, or other elements. The applications and additional elements are all available to users who can access them remotely.

• **Use Case: Supporting BYOD users** – For users on BYOD programs and third-party contractors, you can use RDS hosting to deliver applications to unmanaged machines. There is no need to install the application on unmanaged machines, and you can revoke access to the application quickly if that becomes necessary.

• **Use Case: Securely supporting distributed users** – RDS hosting is often the best option when you need to provide access to remoted applications for distributed users. For mobile users, you can use Workspace to make Windows applications available on non-Windows devices. For remote users, RDS hosting enables you to deliver critical business applications that perform optimally because they are located geographically close to the data center.

• **Use Case: Simplifying management of LOB applications such as SAP** – App delivery grows more complex as the number of applications and endpoints increases within an environment. RDS hosting enables you simplify that by deploying a single instance of a line-of-business (LOB) application to thousands of endpoints, and you can update, patch, and troubleshoot once. LOB applications are typically large applications that are critical to their enterprises, and are integrated with other major systems such as database management systems and systems of record. One good example is SAP, one of the most widely used enterprise software packages for managing business operations and customer relations. You can create application farms that pool the resource load across the farm of RDS hosts, ensuring an LOB application with optimal connectivity. By distributing the load efficiently, you can perform maintenance on specific hosts without having to bring down the application, which improves your ability to meet service-level agreements (SLAs).
• **Use Case: Reducing storage costs** – Similar to using ThinApp, when you leverage RDS hosting, you can reduce the storage overhead for VDI. When you publish an application, you are installing a master instance of that application on an RDS host, and providing many users with a shortcut to that instance. The storage overhead to support the single application instance streamed to many endpoints is a fraction of what it would take to store the application on each individual device.

• **Use Case: Supporting both modern and legacy applications** – If you need to manage both modern and legacy applications side by side, RDS hosting can be the best option because you can maintain multiple versions simultaneously. Printing to network-based printers is also fully supported.

• **Use Case: Delivering Windows applications to non-Windows devices** – You can use the RDS-hosting option to deliver Windows-based applications, including ThinApp packages, to non-Windows devices. You put your Windows applications and ThinApp packages on RDS servers, and then publish those packages to any endpoint device that supports the Horizon Client. Today, that list includes Windows, Mac, iOS, Android, and Linux.

The addition of RDS hosting to Horizon adds another tool to the IT toolbox, and enables you to utilize the best combination of technologies for each user or use case.

**RDS Hosting in Action**

Although detailed instructions are beyond the scope of this white paper, the following summary illustrates how an RDS-hosted application can be set up easily in four basic steps. In this example, you use View to host the Microsoft Remote Desktop Connection application on an RDS host, and entitle users to access it. For complete instructions, see [Setting Up Desktop and Application Pools in View](#).

You can publish an application and entitle users to access it with the following basic steps:

1. **Select the RDS host and application pool.** In the View Administrator in Figure 8, the RDS host is already set up. The host is part of an RDS farm named OFF2K13_1. An RDS farm is a collection of RDS hosts that provide a common set of applications to users—in this case, Microsoft Office 2013 applications. This RDS host already has some Office 2013 applications installed, many of which are also already published in the application pool. In this example, you will add an application.

![Figure 8: Navigating VMware Horizon View Administrator from RDS Host Server to Application Pool](#)
2. **Select the application and publish.** From the list of installed applications, you select the application to be published and provide an ID and display name. Click **Finish**, and the application is installed on the RDS host, ready for entitlement.

![Selecting Application to Publish and Finishing Publishing Procedure](image1.png)

3. **Provide entitlement.** In the View Entitlement wizard, you provide a domain and user name or group names. Click **OK**, and the users are now entitled to access the newly published application.

![Adding Entitlements and Specifying Entitled Users or Groups](image2.png)
4. **Access.** The next time your entitled users launch Horizon Client, this application will be available.

![Workspace User Portal Provides a Consistent Experience](image)

**Figure 11:** Workspace User Portal Provides a Consistent Experience

**Section Summary**

Delivery of RDS-hosted applications provides many advantages. However, not all applications are effective when shared, and most business enterprises do not need to deliver RDS-hosted applications exclusively. For this, Horizon provides multiple app-delivery options:

- App delivery from ThinApp repositories
- App delivery from integrated Citrix XenApp farms
- App delivery from cloud-based service providers
- App delivery of natively installed Windows applications
- App delivery using a hybrid approach
Delivering ThinApp Packages

All editions of Horizon support the ability to deliver applications to physical and virtual desktops from ThinApp repositories. You can create and manage ThinApp packages through View, including policy and entitlement settings. You can also integrate Workspace with View to present ThinApp packages in the Workspace User Portal, alongside applications from RDS farms, Citrix XenApp farms, and SaaS and Web application providers. This section describes the following:

• **ThinApp**
• **Benefits**
• **Use cases**

**About ThinApp**

ThinApp is an application virtualization option that packages Windows applications into portable, virtualized applications that can be streamed from a central file share, or deployed locally on individual desktops as executable files. You can use ThinApp to isolate applications, simplify application customization, deploy applications to different Windows operating systems, and eliminate application conflict.

To virtualize an application, ThinApp scans the computer before and after the installation process, and captures the difference. ThinApp then encapsulates the application files, libraries, framework, and registry settings into a single executable file that you can deploy, manage, and update independently from the underlying operating system. This process decouples the application from the underlying OS. The ThinApp application package looks the same across different machines, providing a consistent end-user experience.

The resulting ThinApp packages can be deployed in two ways. You can deploy a ThinApp package by placing it on a file share and providing the shortcut to your end users. Or you can run a ThinApp package locally by placing the executable file directly on an individual desktop. This is useful if you have end users with poor network connections or no connection at all, such as on an airline flight. Because of the structure, you can isolate the ThinApp package either partially or fully from other applications, and from the operating system. This is useful if you want to deliver several versions of an application that would otherwise conflict.

**Using Workspace Integrated with View to Deliver ThinApp Packages**

In addition to making ThinApp packages available to your end user using View alone or Workspace alone, you can also use an integration of View with Workspace to manage ThinApp packages. Your end users can then access their ThinApp packages from View desktops or RDS-hosted desktops from within Workspace, and you can use Workspace to set entitlements. To use Workspace to manage entitlements, you need to enable Workspace management when you package the applications within ThinApp. Workspace for Windows must also be installed on each View desktop to execute and receive entitlements to the ThinApp packages.

Using View with Workspace provides more benefits than using either View or Workspace alone. View delivers virtual desktops as a managed service, including ThinApp packages on the desktop. However, entitlement through View is per desktop or per desktop pool. Workspace supports entitlement on a per-user or per-group basis, including users and groups imported from Active Directory, as well as Workspace groups. This flexibility provides you with two ways to grant entitlements. You can select a user or group, and then indicate the resources that user or group can access. Conversely, you can select a resource, and then indicate the users or groups who can access that resource.

The appropriate parameters for integration with Workspace are set during the ThinApp capture and build processes. You can then enable the ThinApp Packages module in Workspace, and indicate where ThinApp packages are located. You can determine whether a ThinApp package is streamed from a network share, or copied locally to a virtual machine. You can entitle users to ThinApp packages through Workspace.
1. User launches Workspace from any device, may use SSO when logging in

2. User launches View desktop from Workspace; View desktop displays only the ThinApp packages that user is entitled to use through Workspace

3. View Connection Server brokers access to virtual desktop in the background

4. Virtual desktop is delivered to endpoint via PCoIP or HTML5; ThinApp packages delivered to View desktop via SMB or local deployment

Figure 12: Delivering ThinApp Packages on a View Desktop Through Workspace

If View and Workspace are integrated, your end user logs in to a View desktop or RDS-hosted desktop through Workspace, and looks for the ThinApp package. Behind the scenes, Workspace checks entitlements and displays only those ThinApp packages that the user is entitled to run. The appropriate ThinApp package is deployed either locally to the desktop as an executable file, or streamed to the desktop from a network file share via Server Message Block (SMB), a network file-sharing protocol. The View Connection Server brokers the access to the View desktop or RDS-hosted desktop, which is delivered to the endpoint via PCoIP. Users who are outside the firewall can also access their desktops from a Web browser, supported by the HTML5 protocol. To manage entitlements through Workspace, you must install Workspace for Windows on View desktops and RDS-hosted desktops.

After the user has launched the ThinApp package, the connection between Workspace and the user ends. Nothing is tunneled through Workspace, so if Workspace were to fail, the existing connection between the end user and the application would remain connected. The user would not be able to reconnect or re-authenticate, but any existing connections would remain until timed out.

Using Workspace to Deliver ThinApp Packages

Entitlement capabilities for ThinApp packages are broader using View integrated with Workspace, but it is also possible to entitle users to ThinApp packages using Workspace alone. Workspace provides the flexibility of assigning a ThinApp package to a specific person or group of people. Because you entitle users rather than desktops, a user can run the ThinApp package from any physical or virtual Windows machine. This endpoint must be Windows-based if you use Workspace without View, because ThinApp packages are virtualized Windows applications, and must be used on Windows systems.

![Diagram of ThinApp package delivery through Workspace](image)

**Figure 13: Delivering ThinApp Packages Through Workspace Only**

To access a ThinApp package in a Workspace-only environment, the end user must log in to Workspace from a Windows device. Workspace authenticates the user, and brokers access to the ThinApp package on the file share. The ThinApp package can be deployed from a network file share, or deployed locally as an executable file, and launched directly on the user’s endpoint device. If the endpoint is on a LAN inside a firewall, the ThinApp package is delivered using the SMB protocol. If the endpoint is on the Internet outside of any firewall, the ThinApp package is delivered using HTTPS through Workspace.

After the user has launched the ThinApp package, the connection between Workspace and the user ends. Nothing is tunneled through Workspace, so if Workspace were to fail, the existing connection between the end user and the application would remain connected. The user would not be able to reconnect or re-authenticate, but any existing sessions would stay connected until timed out.

See [VMware ThinApp Documentation](#) for more information about ThinApp.
Using View to Deliver ThinApp Packages

Entitlement capabilities for ThinApp packages are broader using View integrated with Workspace, but it is also possible to entitle users to ThinApp packages using View alone. View delivers virtual desktops as a managed service, including ThinApp packages on the desktop.

Using View, you can assign ThinApp packages from a ThinApp repository to individual desktops or pools of desktops. This enables you to deploy ThinApp packages to many desktops at once. Because View does not enable you to entitle specific users, every user who launches a desktop in the desktop pool has access to all ThinApp packages assigned to that pool. Because the user accesses a Windows desktop instead of the ThinApp package directly, the user can access the virtualized Windows application from a variety of physical endpoint devices, including Windows, Mac, Linux, Android, and iOS. See Managing ThinApp Applications in View Administrator in View Administration.

You also have the option of deploying ThinApp packages on physical or virtual machines without using either View or Workspace.
Using Mirage to Deliver ThinApp Packages

Mirage supports delivery of ThinApp packages and Windows applications using base and application layers. Use of Mirage layers supports both local deployment and network access to applications in a ThinApp repository.

You can create Mirage base and app layers for both physical and virtual desktops. ThinApp packages or Windows applications delivered through Mirage layers cannot be aggregated into Workspace.

See VMware Mirage Documentation to find out how to create and deploy base and app layers using Mirage.

Benefits of Using ThinApp

With ThinApp, you encapsulate an app in a portable package that is placed on a desktop or file share as an executable file, instead of being installed. Because of the structure, you can isolate the ThinApp package either partially or completely from other applications, and from the operating system. ThinApp virtualization provides a number of benefits:

• **Supports one-to-many** – Virtualization eliminates the need to install an application on every desktop. You can place a single ThinApp package on a file share, and many users can stream the application to their endpoints in parallel. This saves you the cost of supporting individual copies of the application on multiple desktops.

• **Simplifies upgrade maintenance** – Upgrading and patching can be done on a single ThinApp package and then distributed to all users. You create an updated version, and the next time end users log in, they receive the new version.

• **Reduces VDI costs** – If you maintain a VDI infrastructure, you can reduce the cost and maintenance of that infrastructure by keeping the size of each gold image small. And because no single image fits every business group, you usually need more than one gold image. The fewer images you have, the lower the corresponding cost. You can use ThinApp to keep the images themselves small, and also to reduce the total number of images. You create a gold image with all the applications and components that HR, Finance, and other departments have in common, such as antivirus, browser, and so on. Then use ThinApp to containerize the applications that are unique to each business group, put them on a central file share, and connect them to the common VDI image via application shortcuts. Now each business group is using the same gold image and desktop pool, allowing you to manage one instead of multiple images. You can use Active Directory group membership or Workspace entitlements to filter the ThinApp packages that each business group can access. This keeps the total number of images down, while providing each business group with exclusive access to the applications unique to their departments.

• **Eliminates application conflicts** – The virtualized application does not write to the registry, OS, or file system, so you can place different versions of an application on the same desktop without conflict, each version encapsulated in its own package. This eliminates dynamic link library (DLL) issues, because the packages run without being installed.

• **Adds portability** – ThinApp packages are portable. The runtime is included as a small number of files inside every captured application, so you can put a ThinApp package on a USB device for offline use.

• **Runs agentless** – You do not need an agent to utilize ThinApp. This differs from other virtualization products which require an agent to be installed on the endpoint.

• **Provides consistent user experience** – The virtual environment presented to the user is a merged view of the underlying physical and virtual resources, and the virtualization layer makes the application look as if it were fully installed. The ThinApp package looks the same regardless of which device is used to access it.

ThinApp virtualization provides a number of advantages that, together with the other application-delivery options in Horizon, provide you with the tools to mitigate costs and complexities of delivery and management.
Use Cases Where ThinApp Is the Best Option

With ThinApp, an application is encapsulated in a portable package that is placed locally as an executable file, or streamed from a network file share, instead of being installed. Because of the structure, you can isolate the ThinApp package either partially or completely from other applications, and from the operating system. Although no one application-delivery option works best in every situation, ThinApp virtualization provides a number of advantages in the following use cases:

- **Use Case: Reducing VDI storage costs** – You can reduce your VDI storage overhead by leveraging ThinApp. When you create a ThinApp package and place it on a file share, you are creating a master instance of an application on a share, and providing many users with a shortcut to that instance. Because it is a shortcut, the storage overhead to support the application streamed to each endpoint is a fraction of what it would take to store the application on each individual device.

- **Use Case: Minimizing failure rates** – Typically any application upgrade or migration involves some rate of failure. But when you virtualize, those failure rates are effectively eliminated because you do not break the application to upgrade it, which can happen when the application is installed directly onto an operating system. ThinApp isolates the application, minimizing the failure rates that typically occur during migrations and regression testing.

- **Use Case: Ensuring high availability of desktop applications** – The ThinApp option can avoid downtime caused by application conflicts. Sharing an application many-to-one can sometimes pose a problem if one user crashes the application, because everyone else using it gets kicked out too. For environments or applications that are prone to application conflicts or crashes, streamed ThinApp applications are often the best option. If a ThinApp package is stored on a file share, users access this one stored package, but bits are streamed to each local device for execution. Alternatively, you can place an executable file on each desktop to give users their own copy of the application. In either scenario, upgrading and patching can be done on a base ThinApp package and then delivered to end users so that the next time they log in, they receive the new version.

- **Use Case: Supporting distributed users with poor network connectivity** – ThinApp provides a deployment alternative for users with a poor connection or no connection at all to the network, such as when traveling, working remotely, or working offline by choice. For these users, you can place ThinApp packages on each desktop, or on a USB device for offline access. The user can launch the local packages and work offline. If application updates occur while offline, the updated versions are accessible after reconnection.

- **Use Case: Mitigating migration issues** – Maintaining legacy software can present a problem that you can often mitigate with either virtualization or remoting. For example, some businesses depend on legacy applications that operate only in a specific browser version. They cannot upgrade the application because of its age, and they cannot replace it because of its importance. One solution is to virtualize the application. The resulting ThinApp package does not write to the registry, OS, or file system, so you can place different versions of the same application on one desktop without conflict, each version encapsulated in its own virtual OS. This eliminates DLL issues, because the packages run without being installed. Because this process decouples the application from the OS, the resulting ThinApp package may also run on an upgraded OS. By maintaining two versions at the same time, you can often avoid recoding, regression testing, and support costs.

- **Use Case: BYOD or consultant support** – When you have consultants or contractors to manage, or employees in a BYOD program, you often need to support applications on unmanaged devices. You may not want to install a management agent and application natively on such a device, especially if the device is not on your domain. You can use Workspace to manage ThinApp packages, which do not pollute the device because the applications are containerized, and deliver applications through a firewall using HTTPS as the delivery mechanism. ThinApp together with Workspace can deliver Windows-based ThinApp packages using HTTPS to non-domain member clients. This includes support for app delivery to PCs outside the firewall. This does require Workspace for Windows to be installed on the end-user device.
• **Use Case: Greater mobility** – ThinApp packages can be placed to execute locally on a local file system or a USB device. This can be more advantageous than app remoting if, for example, the application requires access to local devices such as USB scanners, or is used to create documents to store on the hard disk of the local client. You can deploy, maintain, and update ThinApp packages on USB sticks for greater portability, as well as offline use.

ThinApp virtualization, RDS hosting, and the other app-delivery options available in Horizon provide you with the ability to utilize the best combination of application-delivery technologies for each use case.

**Section Summary**

ThinApp is included in all three Horizon editions, and provides numerous benefits such as the ability to create portable virtual application packages that run without being installed on the desktop. However, not every application is a good candidate for virtualization, and not every business enterprise needs to deliver virtualized applications. For this, Horizon offers alternative options:

• **App delivery from RDS hosts in Horizon**
• **App delivery from integrated Citrix XenApp farms**
• **App delivery from cloud-based service providers**
• **App delivery of natively installed Windows applications**
• **App delivery using a hybrid approach**
Delivering Citrix XenApp Published Applications

The Horizon Advanced and Horizon Enterprise editions support integration of Citrix XenApp farms to deliver XenApp published applications through Workspace. Citrix published applications are created and managed through XenApp, including policy and entitlement setting. Integration of Workspace with Citrix XenApp means that Workspace aggregates XenApp published applications so that they are presented to your users in the Workspace User Portal, alongside applications from ThinApp repositories, RDS farms, and SaaS and Web application providers. You can also use View to display XenApp published applications through remote desktops, but View provides no formal integration. This section describes the following:

• Using Workspace With XenApp
• Delivery process
• Use cases

About Using Workspace with XenApp

Citrix XenApp is an RDS-based virtualization product that extends applications to end users. VMware Workspace can aggregate Citrix XenApp published applications, making it possible for an enterprise with an existing Citrix XenApp farm to deliver their XenApp published applications into the Workspace catalog. This enables your end users to launch XenApp published applications from their Workspace User Portal, using their single sign-on credentials.

(As an unrelated option, XenApp published applications can also be delivered in View desktops. This is not an integration or aggregation through Workspace. See Integrating VMware View and VMware ThinApp with Citrix XenApp for more details.)

You can leverage legacy hardware and software from an existing Citrix infrastructure, and still gain the productivity advantages of Workspace. XenApp entitlement permissions are still managed through XenApp, and there is no VMware code on the XenApp server or the Citrix Receiver. Moreover, there is no dependency on load balancers, and the Workspace-XenApp integration can be deployed without accessing the XenApp administrative environment.
Delivery Workflow
In the XenApp and Workspace integration, end users log in to Workspace using single sign-on if that is set up. Workspace authenticates the users, and opens a read-only connection with the Workspace Integration Broker, which uses built-in functionality to access the policy and entitlement information from the XenApp farm.

Figure 15: Citrix XenApp Integration with Workspace

Workspace uses the Workspace Integration Broker component and the Citrix SDK to handle single sign-on between Workspace, XenApp farms, and XenApp published applications. Workspace does not send any data or make any modifications to your XenApp system or applications.

The application is launched from XenApp via the Citrix High-Definition Experience (HDX) protocol, previously called Independent Computing Architecture (ICA). The connection between XenApp and the user is maintained, and the connection between Workspace and the user ends at that point. Nothing is tunneled through Workspace, so if Workspace were to fail, the existing connection between the end user and the XenApp published application would remain connected. The user would not be able to reconnect or re-authenticate, but any existing sessions would stay connected until timed out.

Using Workspace to Present XenApp Published Applications
Horizon supports integration with Citrix XenApp through Workspace only, and View is not used. To use Workspace to provide access to XenApp published applications, you enable the Workspace Citrix Published Applications module, and provide the location of your XenApp published applications. No scripting is required, and no changes need to be made on your XenApp system. You simply install the Workspace Integration Broker for communication between Workspace and your XenApp farms. See Installing and Configuring VMware Workspace Portal.

All policy and entitlement settings for the published applications are handled in XenApp, not in Workspace. Workspace has a read-only connection; it reads the policy information in XenApp and reflects that information in the Workspace User Portal.

See the VMware Workspace Portal Administrator’s Guide for more information.
Using View to Deliver XenApp Published Applications
Horizon supports integration with Citrix XenApp through Workspace only. Even if you integrate Workspace with both View and XenApp, the XenApp published applications are not displayed in View desktops automatically.

Independent of Workspace, you can deliver XenApp published applications via View desktops (see Integrating VMware View and VMware ThinApp with Citrix XenApp). This is sometimes done when migrating from Citrix to Horizon. To use View to deliver XenApp published applications to the desktop, you install the Citrix Receiver in a View desktop. The delivery of XenApp published applications through View results in additional steps for users to access the application, and therefore is not ideal on an ongoing basis. All policy and entitlement settings are handled in XenApp, not in View. See Avoiding a Costly Upgrade for details.

See Integrating VMware View and VMware ThinApp with Citrix XenApp for more information.

Use Cases Where XenApp-Workspace Integration Is the Best Option
If you already have XenApp or one of its former versions such as MetaFrame or WinFrame, you may have experienced performance or compatibility issues, or frustration with the recent upgrade to a different database structure and its resulting deprecation of some of the familiar features. Delivering Citrix XenApp published applications through Workspace enables you to utilize a legacy XenApp infrastructure. This strategy allows you to maintain a legacy investment, simultaneously broaden application-delivery options, and consolidate multiple investments to drive down costs. Or you may wish to use this integration as a preliminary step toward migrating to Workspace. Following are details on these use cases:

• Use Case: Maintaining legacy Citrix infrastructure – If your business invested in a Citrix XenApp infrastructure, and you want to maintain that investment, you can take advantage of the Workspace-XenApp integration. You can deliver Citrix XenApp published applications through Workspace—either temporarily while you transition to Workspace, or on an ongoing basis—while benefiting from the functionality of the Horizon family of products. Instead of facing major migration and installation time, you can leverage your investment in XenApp and take advantage of the Horizon features without disrupting your user workflow.

• Use Case: Consolidating dual investments – If your business has invested in both Horizon and Citrix XenApp infrastructures, and you want to consolidate, you can take advantage of the Workspace-XenApp integration. For enterprises that also use other VMware virtualization products such as VMware vCenter™ Operations Manager for Horizon™, vSphere, or View, Workspace provides an opportunity to consolidate all investments together into the same product family. You can manage all applications from a central location, and users can access XenApp published applications from the Workspace User Portal. The learning curve is low and the returns high, and you may avoid a costly migration.

• Use Case: Migrating to Horizon – For enterprises faced with a major upgrade to a Citrix XenApp infrastructure, now is a good time to re-evaluate. You may decide to instead migrate from Citrix XenApp to Horizon. You can maintain your legacy XenApp system temporarily without going through the upgrade, and also benefit from the additional application-delivery options available in Workspace. You can deliver XenApp published applications to the Workspace User Portal, or to virtual desktops via View. During a migration to Horizon, you can temporarily deliver XenApp published applications with ease.

The option to deliver Citrix XenApp published applications, along with the other app-delivery options available in Horizon, provides you with the ability to utilize the best combination of technologies for each user or use case.
Section Summary

Delivering Citrix XenApp published applications into Horizon enables you to continue to use a legacy XenApp investment with the additional options that Horizon supports. However, application remoting through XenApp integration does not address every use case. For this, Horizon offers a variety of alternative options:

- App delivery from RDS hosts
- App delivery from ThinApp repositories
- App delivery from cloud-based service providers
- App delivery of natively installed Windows applications
- App delivery using a hybrid approach
Delivering SaaS- and Cloud-Based Applications

The Horizon Advanced and Horizon Enterprise editions support the ability to deliver applications from SaaS- and cloud-based Web application providers through Workspace. Cloud-based applications—also called Web-based or browser-based applications—are delivered to users on the Workspace User Portal. This includes applications that do not require subscriptions, such as Google Docs, and subscribed services known as Software as a Service (SaaS), such as Salesforce. The applications are published and managed by the providers. View cannot deliver cloud-based applications. You present cloud-based applications to your users in the Workspace User Portal, alongside applications from RDS farms, ThinApp repositories, and Citrix XenApp farms. This section describes the following:

• SaaS- and cloud-based applications
• Benefits
• Use cases

About SaaS- and Cloud-Based Applications

Cloud-based applications include both subscribed and unsubscribed applications and services that are hosted in the cloud, such as Dropbox, WebEx, Salesforce, and Axiom. A growing trend is to subscribe to such applications, services, and data from independent software vendors (ISVs) or application service providers (ASPs), instead of installing the applications on premises. In fact, SaaS is sometimes referred to as on-demand software because users typically launch the software only when they need it, and the business pays only for what they use, making licensing customizable per user and more efficient.

In Horizon, Workspace comes with built-in integration of select SaaS- and cloud-based applications in the Workspace catalog. Users can access these default cloud-based applications via single-sign-on authentication. This integration provides a substantial benefit to businesses that want to outsource application maintenance to software service providers. The business subscribes to the service or pays for the use of the application, but does not download it, update it, or maintain servers. Instead, the ASP or ISV hosts the software in the cloud and maintains it there. And Workspace already has a default set of cloud-based applications integrated for you in the Workspace catalog.

In Horizon, integrated Workspace deployments use SAML to provide SSO functionality. SAML is an XML-based standard used to describe and exchange authentication and authorization information between different security domains. It passes information about users between identity providers and service providers in XML documents called SAML assertions.

Many Internet service providers use SAML to contact other online identity providers when authenticating users. Web Services Federation Language (WS-Federation) is used along with other protocols to provide a rich and secure Web-services environment. Workspace gives you the flexibility to integrate all three: SAML, non-SAML, and WS-Federation applications. You can integrate your own non-SAML applications into Workspace, as well as SAML and WS-Federation applications.
Delivery Workflow

With SaaS- and cloud-based application delivery in Workspace, the end user logs in to Workspace, where you can set up single sign-on. Workspace authenticates the user and verifies its data about that user’s entitlement to the cloud-based app. Workspace then opens a read-only connection directly with the software vendor, which provides all policy information. Workspace does not send any data to the vendor.

1. User logs in to Workspace from any device, may use SSO
2. Workspace brokers SSO and user's SAML to authenticate user, and points user to SaaS- or cloud-based app
3. SaaS- or cloud-based app is delivered to endpoint via HTTP or HTTPS

Figure 16: SaaS- and Cloud-Based Web Application Delivery Through Workspace

The application is launched from the vendor, and the connection between Workspace and the user ends. Nothing is tunneled through Workspace, so if Workspace were to fail, the existing connection between the end user and the application would remain connected. The user would not be able to reconnect or re-authenticate, but any existing sessions would stay connected until timed out.

Policies that govern how a SaaS- or cloud-based application behaves are set by the vendor. Workspace reads that information, and provides the ability to entitle or unentitle users to access the application, and to enable single sign-on.

Using Workspace to Present SaaS- and Cloud-Based Web Applications

Workspace ships with a pre-defined application catalog that is populated by default with a variety of the most popular SaaS- and cloud-based Web applications, such as Google Docs, WebEx, Microsoft Office 365, Salesforce, Axiom, and more. This collection of pre-defined applications is dynamic and is expanded continually. Your end users see only those applications that you have enabled them to see.

To enable a pre-defined SaaS- or cloud-based Web application that is in the default catalog, you need a subscribed and licensed account with the specific application vendor. You must also enable the Web Applications module in Workspace. Then with a Web-based wizard, you can enable the specific application and entitle your users. Any entitled user can then access the application through any supported browser.

If you have a new or proprietary SaaS- or cloud-based Web application that is not in the Workspace catalog by default, you can build the integration yourself manually, or request assistance from the VMware Professional Services Organization (PSO). To provide single sign-on, the application must be SAML-enabled or based on WS-Federation.
You can add your own Web links that do not require user authentication, resolving to a static Web page link in the Workspace catalog. Each of these static links behaves like a shortcut to the login page of the application, but does not support single sign-on. You add links using the Workspace administrator console. Single sign-on is not automatically included when adding a static link in this way, but you can either build the authentication yourself manually, or VMware PSO can integrate any application to support single sign-on.

Workspace handles single sign-on and entitlement to cloud-based applications. The vendor sets policies about how the SaaS- or cloud-based application behaves. See the VMware Workspace Portal Administrator’s Guide for more information about Workspace.

Using View with SaaS- and Cloud-Based Web Applications

View does not deliver SaaS- or cloud-based Web applications in a virtual desktop at this time.

Benefits of SaaS- and Cloud-Based Applications

Using Workspace, you can deliver SaaS- and cloud-based Web applications alongside other delivery options. This provides a number of benefits:

- **Cost savings** – Businesses of all sizes can benefit from the lower overhead costs of subscriptions to browser- or Web-based applications, and Software as a Service from ISVs or ASPs, because users access the applications on demand, and you pay only for what your organization uses.

- **Time savings** – The software vendor manages upgrades and maintenance, which frees your IT teams to focus on other important tasks.

- **Mitigating risk of data loss on mobile devices** – When users conduct business using mobile devices, they often store critical corporate data without much IT oversight. One way to mitigate this risk is to use cloud applications, where the data is stored in the cloud instead of on the mobile device.

- **Allowing increased use of SaaS applications** – For many end users, the majority of the applications they need to do their jobs are SaaS-based. Although many companies still maintain legacy applications, users actually access these applications only a few times a week, such as when submitting timesheets or expenses. Workspace consolidates all of that, and provides a single-sign-on capability to access all applications.

- **Delivering applications via self-selection** – Instead of sending a request to an overstretched IT department for an application and waiting in the ticket queue to receive the application, end users today expect to download an application themselves, and want to use it immediately. As the demands for self-service and compatibility are met more and more by cloud and SaaS applications, you also have a growing volume of applications to manage. With Workspace, you are able to provide and securely manage self-service applications without overstressing your own resources.

Delivering SaaS- and cloud-based applications provides a number of advantages that, together with the other app-delivery options in Horizon, provide you with the tools to mitigate costs and complexities of application delivery and management.

Use Cases Where Cloud Delivery Is the Best Option

SaaS-based and cloud-based Web applications are the best option for the following use cases:

- **Use Case: Integrating corporate applications or intranets** – Developing your own corporate cloud-based applications or intranets can provide you with a great deal of control, but often at a high cost. It can often cost less to use existing cloud services instead of developing your own. This can simplify app delivery and maintenance as well. In addition, you can utilize the existing SaaS- and cloud-based applications that are incorporated into Workspace by default, such as Office 365, SharePoint, Outlook 365, and more. This integration simplifies end-user experience by providing single sign-on from Workspace. The Outlook native client Active Profile authentication is also supported through a Workspace-generated password in the Workspace User Portal.
• **Use Case: Delivering non-SAML applications** – You can integrate non-SAML applications into Workspace, as well as SAML and WS-Federation applications. SAML is an XML standard that provides secure Web domains to exchange user authentication and authorization information. Many Internet service providers use SAML to contact other online identity providers when authenticating users. WS-Federation Language is used along with other protocols to provide a rich and secure Web services environment. Workspace gives you the flexibility to integrate all three: SAML, non-SAML, and WS-Federation applications.

• **Use Case: Delivering unintegrated applications** – You can easily create Web application links that do not require user authentication, and integrate them into the Workspace catalog. You can also add links that resolve to static Web pages using the Workspace Admin Console. The result is a single application portal for end users that is easy to use and can display every application they need to get the job done.

• **Use Case: Delivering applications on demand** – Subscribing to applications can lower costs for businesses because they can use the applications on demand, and pay only for what they use. They receive upgrades and maintenance from the outsourced service as well. The use of Salesforce, Dropbox, Axiom, WebEx, Office 365, and other similar Software-as-a-Service applications is increasing. Although most businesses still use some legacy applications, many users increasingly rely on cloud-based applications to do their work. Workspace consolidates such applications on a single console, and provides a single-sign-on capability as well.

The option to deliver SaaS- and cloud-based applications, along with the other app-delivery options available in Horizon, provides you with the ability to utilize the best combination of technologies for each user or use case.

**Section Summary**

The diversity and volume of SaaS- and cloud-based applications are growing rapidly. App-delivery options are also evolving and providing more diverse alternatives. Flexibility is important in meeting future challenges because no one application-delivery option fits every situation. Not all applications can be virtualized, nor can every application be hosted. To provide the flexibility to deliver any application to any device, Horizon supports a combination of other options:

• App delivery from RDS hosts
• App delivery from ThinApp repositories
• App delivery from integrated Citrix XenApp farms
• App delivery of natively installed Windows applications
• App delivery using a hybrid approach
Delivering Natively Installed Windows Applications

Since inception, Horizon with View has provided access to natively installed Windows applications. These applications can be displayed in View desktops.

See also VMware Horizon With View Documentation.

Conclusion

You must consider a wide variety of variables when you choose an app-delivery option. For example, you need to consider the types of applications to be delivered, requirements such as device drivers, whether any applications are restrictively licensed, and whether any are from an external vendor or SaaS provider. You need to consider how your end users work, whether they are stationary or traveling, or working on premises or remotely. You need to consider the endpoint devices, whether your users provide their own in a BYOD program, or use mobile devices in the field. And you need to consider environmental factors such as support for legacy infrastructure, legacy applications, multiple versions of the same application, and frequency of upgrades or license renewals.

If you conclude that you actually need a combination of app-delivery options and not just one, you are not alone. Most enterprises need a combination of options to deliver diverse types of applications to diverse sets of endpoint devices.

Horizon provides that combination. VMware has long provided options for application delivery of virtualized applications through ThinApp, delivery of SaaS- or cloud-based applications from individual software providers, and more recently, delivery of Citrix published applications from XenApp farms. Now, Horizon adds RDS-based app remoting to this mix. In addition, Horizon Advanced and Enterprise editions include Mirage, which provides support for app delivery of ThinApp packages and natively installed Windows applications in app layers.

Horizon Advanced and Enterprise editions also include Workspace, a unified workspace where your end users can log in once from any type of device, and launch any supported application type: ThinApp packages, XenApp published applications, SaaS-based or cloud-based applications, and—now—RDS-hosted applications.
Additional Resources

To find out more about software application-delivery options, take a look at the following resources:

• Introducing Unity Touch
• Setting Up Desktop and Application Pools in View
• VMware Horizon – A Closer Look At Application Remoting
• VMware Horizon FAQs
• VMware Mirage Reviewer’s Guide
• VMware Mirage Documentation
• VMware Horizon (With View) Documentation
• VMware Workspace Portal Documentation
• VMware Professional Services Organization
• VMware Self-Service Knowledge Base
• VMware ThinApp Application Virtualization Made Simple
• VMware ThinApp Documentation
About the Author and Contributors

This white paper was written by Cindy Heyer, Technical Writer in Technical Marketing, End-User Computing at VMware, with many thanks for the contributions of Andrew Johnson, Courtney Burry, Cyndie Zikmund, Heath Doerr, Jason Ruby, Josh Spencer, Kevin Strohmeyer, Mason Uyeda, Peter Bjork, Rasmus Jensen, Rory Clements, Tina de Benedictis, and Warren Ponder.

Cindy Heyer updated this paper to reflect recent terminology changes.

To comment on this paper, contact the VMware End-User Computing Solutions Management and Technical Marketing Team at twitter.com/vmwarehorizon.