

Software2

# Ultimate Guide to Application Virtualization



# Foreword

The ultimate guide to application virtualization, aimed at all levels of knowledge, from those just delving into the world of virtualizing apps all the way to those looking for in-depth knowledge on the different solutions available.

In this guide, we cover the following details of application virtualization:

- The basics; what application virtualization is and how it works
- The reasons organizations choose to virtualize apps and how they can benefit
- An objective discussion of the advantages and disadvantages of certain solutions
- A comparison of how application virtualization compares to other technologies
- Costs associated with virtualizing apps
- Information on the process of implementing app virtualization

# Table of Contents

|   |    |
|---|----|
| An introduction to application virtualization         | 3  |
| Why is application virtualization necessary?          | 12 |
| Pros and cons of application virtualization           | 17 |
| Application virtualization solutions                  | 24 |
| Application virtualization vs. desktop virtualization | 39 |
| The cost of application virtualization                | 44 |
| Managing and deploying application virtualization     | 51 |
| Glossary  | 63 |



# 1. An Introduction To Application Virtualization

## What Is Application Virtualization?



One thing most IT departments have in common is the need to deliver a wide variety of apps to large numbers of users, but the more apps we install, the more likely we are to see conflicts, compatibility issues and other obstacles-to-delivery. Apps may contain overlapping components or require different versions of system files. Over time, new apps may fail to install and we see more errors while in use. Windows performance suffers as more apps are added and removed, with files and registry keys left behind. This is often referred to as system bloat.

App virtualization allows us to package all parts of an application to create a self-contained virtual app. Depending on the technology used, all or parts of a virtual app can be isolated from the system, with the app running in a container or sandbox. Virtual apps are easily deployed to multiple machines and are more likely to work regardless of existing apps or system configuration. With some virtualization technologies, launch times may be accelerated vs the normal time taken to install the app.

When no longer needed, virtual apps can be removed cleanly, reducing system bloat. Some application virtualization solutions also include a licensing component, providing DRM and remote wipe capabilities.

# A Short History Of Application Virtualization

Over the past half-century of computing, software applications have become more capable, more widely adopted and more business critical than almost any other conceivable resource. In conjunction with this, the size and spread of organizations have grown too, necessitating more robust and reliable methods of delivering applications to their end users.

"Users' expectation of IT systems is higher than ever before and users demand greater flexibility in where, and how they work. Choosing the right way you deploy and manage applications with their data is a crucial element in user experience."

**Jurjen van Leeuwen**, Freelance consultant @Leodesk\_IT | MV (Application Virtualization Smackdown, Ver. 5.3)

Legacy solutions, whilst groundbreaking at the time, have been rendered superfluous in the endeavor of delivering software apps and are often seen as overkill; especially when end users may be seeking a small number of specific apps at any one time. What's more, as end-users' expectations of their digital experience have risen, these legacy solutions have become less and less appropriate in certain situations for delivering a seamless and non-complex user experience. As such, application virtualization technologies have become crucial for contemporary IT departments. *So why would you virtualize an application, what problem does it solve?*

There is a number of primary reasons to utilize application virtualization over other delivery methods and technologies. Applications are quickly and easily delivered to an endpoint, with the necessity of creating multiple, different images for different faculties and departments removed. This also results in an easier upgrade process for individual apps as entire images no longer need to be rebuilt and machines no longer need to be reimaged. Another facet of delivering in this fashion is that rollback to previous versions of apps is easier and it becomes possible to execute different versions of the same app simultaneously.

Furthermore, manual install on the side of the user becomes a thing of the past with the chosen virtualizing technology's client/agent handling the groundwork. Alongside this, the inherent system and configuration behind application virtualization enable conflicts to be dealt with more efficiently and in a much more automated fashion.

Despite traditional application virtualization solutions having been subject to some limitations, these solutions were still more effective when it comes to efficient and license/cost/time conservative delivery. Examples of some of the original solutions comprising the application virtualization market landscape are as follows:

## AppStream (Different to Amazon Appstream)

AppStream in both product and company name, this was the first true application virtualization product on the market. AppStream provided the streaming component for Symantec Endpoint Virtualization Suite Professional (SVS Pro, also known as Symantec Workspace Virtualization/SWV) beginning in June 2007 and, [after some delay](#), was [acquired by Symantec](#) circa Q2 2008. [SVS Pro has since also gone end-of-life](#).



## Microsoft App-V

Formerly known as SoftGrid, App-V was developed by Softricity and renamed to App-V following Microsoft's acquisition of Softricity in July 2006. Alongside other Microsoft products such as HyperV and User Environment Virtualization, this marked Microsoft's first foray into the world of application virtualization.



## Turbo.net

Turbo is also known as Turbo.net and originated as Code Systems Corporation's Xenocode. CSC later rebranded to Spoon in 2010. Operating primarily within the space of financial, government, health care, and IT, Turbo is one of the original players in the application virtualization sphere. They have, in recent years, shifted their focus to providing solutions complementary to application delivery alongside their original offering. Examples of this are utilizing containers to streamline the software development process, and using Turbo studio to help package applications into portable applications.



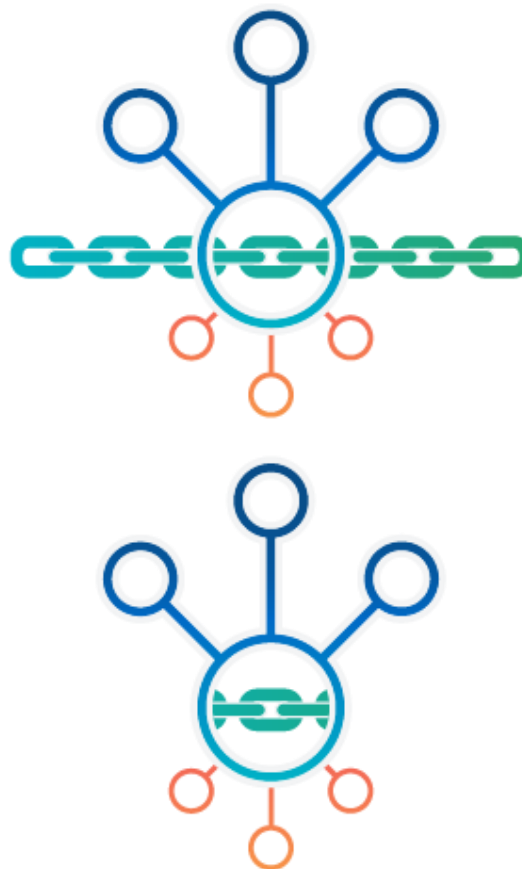


## How Does Application Virtualization Work?

In virtualizing a Windows application, it is wrapped in a 'bubble' or virtual structure. The creation of the virtual structure takes place natively on the end users' device where all of the application's processes are executed but the separation between the application and the operating system itself is retained. These virtual structures are often inaccurately referred to as '*containers*'. Despite appearing to be synonymous, containers are not used to encapsulate applications themselves and are even subject to slightly different definitions on a vendor-by-vendor basis.

The 'Agent' or 'Client' component of the application virtualization product will control and manage the creation of this virtual structure. It will also control access for both the virtualized application and any external processes, such as Windows OS processes, that are running or trying to interact with the application in allowed or appropriate contexts.

Typically, with most application virtualization technologies, the virtual structure that is created is in an 'isolated' state; the virtual structure is isolated from the end user's device operating system. Any external applications or processes running on the user's device will not be able to see or interact with the contents (files etc) within the virtual structure.



However, some application virtualization products such as Numecent Cloudpaging also allow for what is known as an 'integrated' virtual structure. This 'integrated' virtual structure allows native 2-way access to the content/application within the virtual structure. An 'integrated' virtual structure appears and behaves as if the application was natively installed on the user's device.

The virtual structure that is created is a blueprint of the applications traditional installation footprint in that it contains all the files, folders, registry keys, registry values that the application installs or creates. This is defined when the virtual application package is created by the relevant tool, for example, App-V sequencer or Cloudpaging Studio.

The structure will evolve as the application is used in the context of object modifications (such as configuration files, or registry values). These changes are written directly into the virtual structure on the client device, with the exception of any locations that are configured to be excluded. For example, the user's *'My Documents'* folder is excluded from the virtual structure by default. When saving data to an excluded location, this allows for the virtualized application to be physically written to the local disk/file-system and not inside the virtual structure/bubble.

The main reason for this is that if the virtualized application is removed from the user's device, the virtual structure for that application is destroyed and not recoverable making any data inside the structure/bubble lost forever.

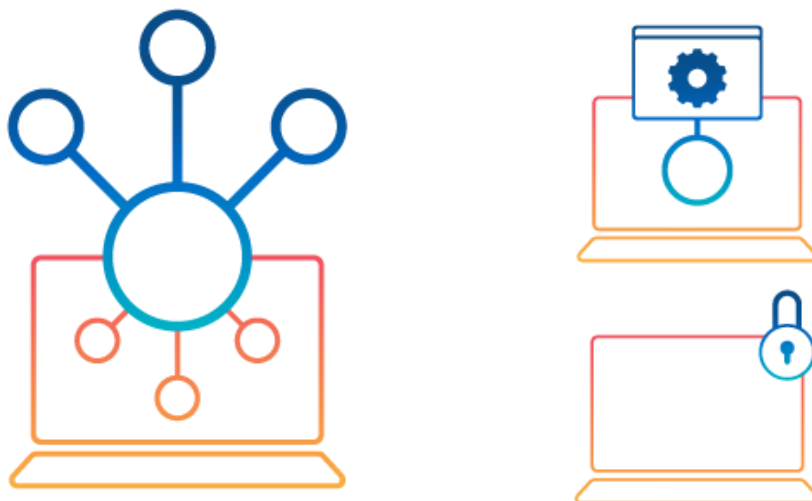


## How Does Application Virtualization Sit Alongside Other Technologies On The Market?

Application virtualization is one of the key technologies available to IT when it comes to application delivery and deployment. As the name suggests, it falls within the general category of virtualization, but with some specific differences to the technology's more-famous sister; desktop virtualization (VDI).

The standard approach to delivering software and applications to end-users is often installation and imaging, which could involve a technology such as Ghost or SCCM. For those not using one of those technologies, delivering software using this approach will be a case of manual installs via CDs or other traditional methods of installation.

Another commonly used approach to deliver software, historically, was RDS (known as Remote Desktop Services or Terminal Services). This approach enables users to access a remote desktop or computer, and all of the end-user services on it, including software applications. Microsoft RemoteApp is an example of an RDS solution, along with more contemporary technologies such as Parallels Remote Application Server (Parallels RAS).



Finally, there's application virtualization, which was born out of the need to have virtual applications without the need for a full virtual desktop infrastructure (as is required for desktop virtualization). This started life as Softricity's SoftGrid before being acquired by Microsoft and reincarnated as App-V. To date there are many more solutions available such as Numecent Cloudpaging, Turbo.net, Cameyo and Symantec Workspace Virtualization (now end of life), to name a few. Application virtualization virtualizes the application itself, without needing to deliver any of the OS to go with it. It enables the access of applications 'on-demand' and the way in which the technology works usually means the application runs on the end-user's device, without the need for a large server infrastructure to deliver the application. This page will explain what it is and how it works in much more detail in later chapters.

So, as an overview, these are typically the types of approaches available to deliver applications:

- Imaging and installations
- Remote desktop solutions
- Desktop virtualization
- Application virtualization

Some useful & related reading...

[Accessing Applications Off-Site with Application Virtualization >>](#)

[Application Virtualization - The Future of The University Desktop >>](#)

[Any App On Demand To Windows And OSX >>](#)



## **2. Why Is Application Virtualization Necessary?**

# The Problem Without Application Virtualization

When it comes to the wider application delivery landscape, there's a multitude of technologies and approaches that organizations can take to deploy software to their end-users. This might sound to many like a 'non-problem' but for institutions with hundreds upon thousands of users, the delivery of applications to each of them in a timely and efficient way that performs exactly as the user expects, when they expect it, is a real challenge. And that's especially true when considering some of the available app delivery approaches on the market for solving that very challenge...

## Approach Number 1 Download, Installation, SCCM, Imaging and CDs

Often regarded as the traditional way of deploying software, installation is now seen as a 'legacy' approach to delivering applications. But in the world of end-user computing, especially within higher education, many IT professionals continue to use this tried and tested route. This is often because tools such as Microsoft SCCM come at no extra cost to universities and colleges and delivering software via imaging processes requires nothing more than a little IT experience and time; lots and lots of time!

And that's where the problem lies with this approach. The more and more end-users you have, and the more software titles that you need to support (and indeed versions of individual software titles) – including updates and patches – the time involved in imaging machines can soon become impossible to manage. Along with handing out physical CDs to install software from, this is where most of our customers once started their application delivery journey and it's this approach that often leads IT to consider a form of virtualization; to centrally deliver software to all end-points in a more efficient way.

Another key problem with this approach is its impact on the user experience. Of all the EUC services an organization provides, when it comes to software delivery end-users have the expectation that what they need will be provided whenever and wherever they need it. They also have the expectation that the software will 'just work' and in the same way as if it were locally installed. Imaging and installation do not necessarily fit this bill; often users will need to request software titles from IT or wait for the application to install after requesting it through SCCM for example. And all of this is before you even consider other types of devices and operating systems, for example Macs and Chromebooks.

Unless the applications are installed onto every device locally, something which is largely unachievable, the software applications deployed using this approach will not be available to end-users on-demand.

[Read more about the problems and challenges associated with traditional imaging for large organizations >>](#)

## Approach Number 2 Desktop Virtualization, VDI, RDS and Terminal Services

After using the first approach (downloads/installations/SCCM), many organizations first look to VDI as the silver bullet to solve their software deployment challenges. Virtual Desktop Infrastructure, often known as Desktop Virtualization, is a solution for delivering all end-user services; desktops and applications. The well-known vendors who operate in this space include Citrix and VMware, each with their own industry-leading VDI solution.

You've probably already considered VDI as a solution to your application delivery problems, with its ability to neatly provide all applications and desktops through a single platform on any device. And it's true, VDI does have the ability to deliver ALL your apps, to both Windows and non-Windows devices. But at what cost do you get these benefits?

Firstly, you've got to consider the actual cost (in money) of a solution like this. Remember, desktop virtualization solutions like these rely on an infrastructure to provide their services. The more and more applications you want to deliver, or the more users you want to deliver to, the more infrastructure you'll need to do so. That means bigger, more powerful hardware and servers are required to carry this heavy load.

Secondly, you've got to consider the user experience of a solution like this. As we've just mentioned, there's a large financial cost to VDI if you want to use its technology to deliver applications. But if you want to do that in a way that offers a great experience, or at least an experience that your users expect, that cost goes up even more. If, for example, you want to deliver GPU-heavy or complex applications to your end-users and you want those applications to perform as if locally installed, the infrastructure required for your VDI solution will be even greater. AutoCAD, MatLab, SOLIDWORKS, NVivo, ArcGIS; these all require huge server power to achieve a great user experience.

## Approach Number 3

# Application Virtualization

The next step in the exploration of application delivery, for a huge number of our customers, was making the step from desktop virtualization through to application virtualization. After struggling through imaging and installation approaches, VDI and desktop virtualization becomes the next clear choice. But when the cost and all other factors are considered, it's often decided that VDI isn't a good fit, especially in higher education. It's at that point that many universities, colleges and organizations look to the market and find application virtualization as an alternative for delivering software.

Naturally, the first port of call in any app virtualization technology search is Microsoft's App-V. It was one of the first technologies of its kind on the market and after Microsoft's acquisition of the technology in 2006 it became the most renowned solution. Unfortunately for many of our customers, their journey with App-V never quite materialized the way they



wanted it to; the technology was unable to virtualize all their Windows applications and they often found themselves back at square one.

However, more and more solutions soon become available on the market, each with an increasing list of benefits and abilities. Next-generation solutions such as [Cloudpaging from Numecent](#), which enables the virtualization of 100% of Windows applications, finally checked the boxes that every Desktop or Endpoint Administrator was looking for:

1. Delivering software in a consistent way to any device
2. Manage software estate and updates from a central location
3. Eliminates the need for a large, server-heavy backend infrastructure
4. No need to virtualize entire desktops just to deliver applications
5. Harness the computing power of the end-user's device to run the application
6. The same performance as if the app were locally installed
7. Delivering of the application 'just in time' rather than 'just in case'
8. Cost-effective licensing and server setup

When considering the other ways in which IT could deliver software to end-users – imaging, SCCM, installations, VDI – it becomes obvious why there's a huge demand for application virtualization... Delivering software in a cost-effective way that offers an IT experience your users want and expect, without the need for expensive backend infrastructure; that's the dream of any end-user computing team at any organization!

Some useful & related reading...

[Moving STEM Apps To The Cloud at Imperial College London >>](#)

[Streamlining App Delivery To Enable Student BYOD With Surrey Software >>](#)

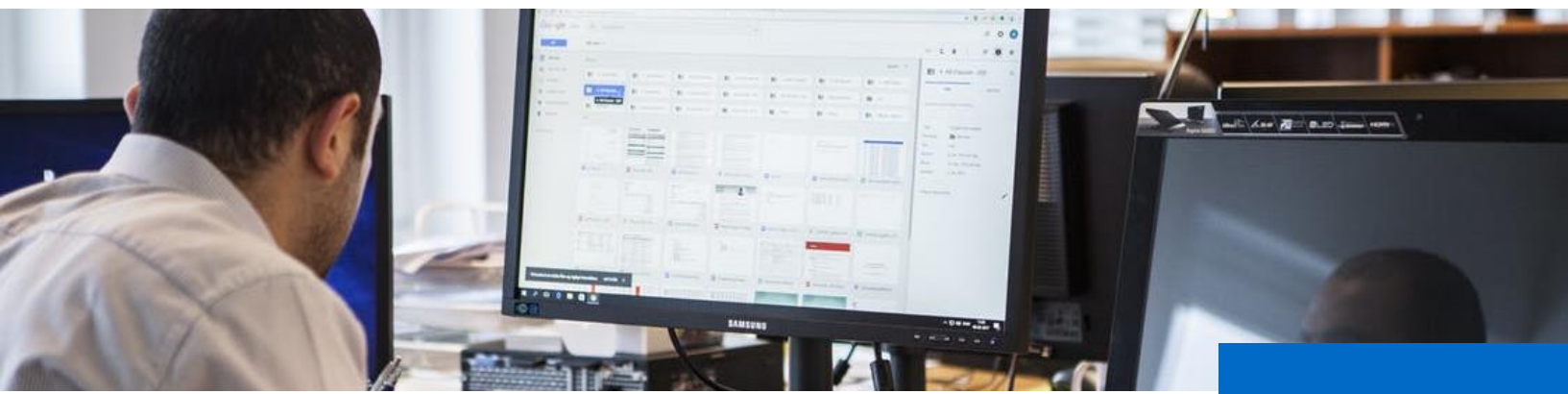
[The Holy Grail of the University Desktop and BYOD in 2018 and beyond >>](#)



# 3. Pros & Cons Of Application Virtualization

# The Benefits And Use Cases Of Application Virtualization

Application virtualization can be generally categorized into three key areas in terms of the problems they solve for various groups of people, those being IT, end-users and those at a business level, such as CIOs/directors/strategists.



## Information Technology

App virtualization, by nature, separates apps from OS and image, which allows for more flexible management and deployment. This also means apps can be updated or otherwise modified without having to redeploy entire images. Naturally, this extends to the addition of entirely new apps to organizations' software suites; they can be virtualized and deployed without having to touch or interact with any existing machines, whether they're managed or non-managed.

Less work to deploy equals a shorter deployment time. Short deployment time equals more time freed up for key IT staff. In large organizations with numbers of users reaching five or even six figures, IT staff availability is essential to keep things running smoothly and provide support for users. A compound benefit is that a [decrease is often seen in the number of support tickets](#) raised once an application virtualization technology has been implemented.

Having an application virtualization technology positioned in a software delivery stack enables in-line patching of apps, as and when the patches are released, resulting in significantly bolstered security and a greatly reduced risk of any cyber attack or breach. The previous three use cases lend to a further use case of flattening the summer workload spike in the effort to prepare classrooms for the start of the first semester. With the workload more widely distributed over the academic year, IT is able to approach recurring responsibilities with a greater individual focus upon each task and more forgiving deadlines. More forgiving deadlines provides a better opportunity to pre-empt and address obstacles that may arise.

In anomalous or unique situations, it may become necessary to award risky admin permissions to otherwise inappropriate organization members. Alongside justified admin permission, if there are too many admins to effectively track, the chance of potentially system-breaking changes being made increases. Application virtualization technologies greatly reduce the number of situations in which this may be necessary and makes it easier to rescind these admin permissions when the time is right.

Apps are separated from OS and image which allows for more flexible and agile deployment options, making it easier to cover all bases and make apps (license-dependent) available literally everywhere. [Reducing image size and bloat](#) is a perpetual endeavor of large organization IT and results in greatly diminished staff workloads and departmental costs to deliver applications. A positive in every aspect!

## End-users

End-users can run apps on-demand without waiting for, or manually executing installation. Less work, less time, streamlined experience.

Apps can be run from any campus computer, as opposed to only in dedicated labs. This creates more availability and flexibility in accessing specific pieces of software to complete work. More versatility in how students are able to work provides a route to success in achieving higher grades.

It becomes possible to run apps on home or non-managed machines. Once again, this flexibility lends greater achievement and success than if users were only able to access software on campus, or on managed machines.

## Business Level

With the potential unlocked to leverage “free” hardware provided by users with the BYOD movement, organizations may begin to consider consolidating their hardware estate. This carries the benefit of reducing both support costs for the hardware, and also necessary recurring costs on new machines to ensure future-proofing and compatibility.

Costs are also slashed by application virtualization when compared with VDI solutions as fewer specialized staff members are required for launch and maintenance. What's more, the skills required to manage an application virtualization estate are more transferrable and universally useful than those for VDI, resulting in a more versatile, adaptable and dextrous team.

Given the amount of time freed up for key IT members by altering the imaging process, key strategic IT goals are easier and quicker to approach for the organization and IT department alike. This facilitates accelerated progress in IT in fields and contexts totally non-related to software delivery! Augmenting the digital offering of the organization adds value to it in the eyes of its end-users and/or students.

With such an optimized user-experience, an organization is providing a better service to its users, who also happen to be its students. With university reputations being all-important in students' selection process, it is imperative to be known for providing high-quality facilities and access. With a better reputation comes higher enrolment retention rates.

## The Limitations Of Application Virtualization

The key limitations of app virtualization are the requirement to package apps, and the success rate of packaging. Whilst all current solutions include semi-automated tools to sequence or capture the installation of an app, this process can be time-consuming and, depending on app complexity, it may not be possible to virtualize an app with some technologies.

*Success rate is generally dependent upon the type of virtualization used; those being sequenced virtualization or configurable virtualization.*

**Phil Morgan, Director of Client Services, Software2**

Industry consultants generally agree that sequencing applications with Microsoft App-V offers a success rate somewhere between 60 and 80%, with a large amount of time needed to reach the latter, and the law of diminishing returns truly coming into effect. Whilst this has improved from initial versions (it is now possible for App-V apps to register a context/pop-up menu, for example) there are still issues with the registration of drivers and services. There are also limitations to the interaction between apps, such that apps that need to work together must be sequenced together.

Solutions using configurable virtualization have a much higher success rate. Numacent Cloudpaging offers the ability to isolate, integrate or install individual files and registry keys. This enables almost any regular Windows application to be packaged. Whilst moving some files and registry keys outside of isolation could reintroduce the possibility for conflicts, this flexibility also allows apps to interact in the normal way. For example, plugins can be packaged and work happily with natively installed software.

Nowadays, nearly all packaging tools have the ability to monitor the installation of an app to capture the files, registry keys and system changes that take place upon installation automatically. Previous 'snapshot' based methods that were typically used for MSI/Windows installer creation have been largely superseded, placing greater importance on the ability of the tool to detect all changes and to effectively filter out background processes during capture.

Larger, more complex applications, particularly those including high numbers of individual files and registry keys, will typically take longer to package and may require significant expertise on the part of the packager; to recognize which system changes captured during installation are actually necessary for correct runtime execution.

This makes the ability to reuse packaging templates important and companies working in the app virtualization space should provide capture instructions for popular or complex applications; particularly where the software titles used are industry specific. Software2 customers have access to an [exclusive and specialized Packaging Service](#) designed to overhaul the efficiency of packaging on demand and balance activity spikes that have traditionally been associated with the summer workload.

As with all packaging-based app deployment, the type of software license activation used may affect the suitability of an app for virtualization.

Generally speaking, DRM free, clone-able license keys or a license server based model will yield the highest benefits, allowing for an app to be packaged once and virtualized on any number of machines. Typically if an installer can be packaged as an MSI it will also be

suitable for virtualization, as the vast majority of software license vendors do support enterprise deployment in some manner.

Apps with single-use license keys can be packaged, but if the license requires activation specific to a single machine or user the benefit of virtualization is reduced.

For enterprise licenses, scripting license registration or activation is generally not a problem as most app virtualization solutions have the ability to execute such actions at runtime. More advanced solutions such as Cloudpaging Studio also allow for the execution of scripts at various points in the app virtualization process, including on removal, such that licenses can also be deactivated and returned to the pool.

Some useful & related reading...

[7 Benefits of Delivering Apps Using Application Virtualization >>](#)

[How to deploy and virtualize any software app to university computer labs in 2017 >>](#)

[Why Don't Universities Need Computer Labs Anymore? >>](#)





# 4. Application Virtualization Solutions

## What Application Virtualization Technologies Exist On The Market?

As with application delivery technologies, no one method of application virtualization is best for all scenarios unless more than one is used in tandem. It's generally expected that only around 60% of applications can be virtualized using one of two methods exclusively due to the way applications interact with the local OS. These methods are the sandbox method and the integrated method. Using the sandbox method, a driver or Windows service can't be sandboxed as it requires system-level access. Used exclusively, the integrated method reduces the benefits of virtualization as conflicts could return.

So let's take a look at the various technologies on the market today...

### Cameyo

Cameyo concentrates upon delivering applications into single executables that can run on any windows device. Use of that application will not modify your system files. Within the last couple of years, Cameyo's offering has extended beyond allowing their applications the ability to be run embedded within the browser, using a mixture of RDP and HTML5. This allows the applications they support within their virtualization stack to run within the browser, thus enabling these applications to be accessed from any HTML5 compatible device. Sources state that Cameyo has hundreds of thousands of users worldwide and the product has been released in 10 different languages.

**CAMEYO**

### **Advantages of Cameyo**

Offers an online portal and ready-to-use, packaged (unlicensed) applications  
Supports HTML5 delivery so any device can use their packaged Windows apps  
Easy and simple costing model

### **Disadvantages of Cameyo**

RDP via HTML5 can be a costly model when scaled exclusively  
Application virtualization engine only supports isolated virtualization  
Apps run in containers so, whilst portable, no interaction with the local OS is possible  
No evidence of heavyweight apps being able to run within this platform

## **Numecent Cloudpaging**

Formerly Endeavours Technologies, Numecent created their application virtualization solution, Cloudpaging with the intention of omni-context (Location, time, device), no install, no download delivery of any Windows app to any Windows device. The application is divided into discrete sections, known as pages, and the principle deployment and launch of the app delivers 5-10% of the total app. The rest is paged and provisioned from the server in an automatic, on-demand basis, which permits the app to launch rapidly following the user's request, with subsequent launches becoming progressively faster due to the app having been cached. This method presents launch times and speeds indistinguishable from those of a native app.



Instead of using traditional push technology such as Microsoft Systems Management (SMS), or clientserver based solutions to install or run entire applications, Cloudpaging responds to user requests for applications and immediately pulls the application to a user's desktop. It does this via a streaming technology called 'Cloudpaging'.

***Spruijt, van Leeuwen and Monaghan, 2016***

Cloudpaging Studio creates streamable and virtualized application sets from an original installer.

Cloudpaging Server controls and delivers those application sets to the clients.

Cloudpaging Player runs on the client machine creating a virtual environment to run the application.

**Advantages of Cloudpaging**

Highest rate of app compatibility of all virtualization products.

The simplest packaging of all products, very little skill required.

Native support for drivers.

Ability to control isolation or integration of individual files and reg keys.

Citrix-ready and SCCM Integrated.

Low cost.

**Disadvantages of Cloudpaging**

From a technology point of view, there are not any disadvantages.

There are no published user stories outside of Software2.

## Microsoft App-V

App-V is the application virtualization offering from Microsoft – often referred to as “Microsoft Application Virtualization” – and is included as standard in most enterprise licenses of Windows. App-V was one of the pioneering technologies in the application virtualization arena, having first being developed by Softricity prior to Microsoft’s acquisition of their technology (known then as Softgrid) back in 2006.

Many application virtualization or packaging experts will recall to this day how Softricity’s SoftGrid was a game-changing technology in the end-user computing space in the early years of 2000s, with a huge promise of how it could totally transform how applications are delivered in enterprise IT. The concept of ‘virtualized applications’ was arguable first coined by Softricity.

Like all the virtualization technologies on the market, App-V makes software available to end-users without the need of physical installs on a per-machine basis. This is achieved through App-V’s ‘application sequencing’ tool, enabling applications to run on the end-user’s device while being completely virtual and ‘self-contained’. Each application delivered using App-V exists in its own isolated environment, removing possible conflicts (in theory!) between applications while maintaining interactivity with the local operating system.

Included within the overall solution is an App-V client, which gives end-users a way to access, launch and use their applications once they’ve been virtualized and published on their device. Through a process known as publishing, this tool also enables IT admins to make applications available to certain users, by creating links, icons and shortcuts in various Windows explorer locations (the Desktop for example). This is also the part of App-V that manages the self-contained virtual environment mentioned above



### Advantages of App-V

Very widely used across enterprise, higher education and public sector organizations  
Relatively large online community and a pool of skilled talent for employers  
Has integration with many 3rd party products including XenApp, Microsoft SCCM and Admin Studio  
Low cost as it's included for many organizations in their Windows license

### Disadvantages of App-V

There's a learning curve even for those with experience in App-V 4.6 due to many changes to the underlying technology  
Product development has been minimal since Microsoft's acquisition in 2006 and as such the technology hasn't evolved like many other application virtualization solutions. Future development has also stalled as the product is not a key priority for Microsoft  
Application Virtualization engine only supports ISOLATED virtualization and isolation options are rigid which makes application compatibility an issue when compared to other products as it can only sequence in the isolated disposition, making the delivery of drivers or Windows Services very problematic if not impossible  
Many users report on the technology achieving a virtualization success rate of around just 65%, much less than the more contemporary solutions on the market

*(eavena et al., 2019) (Wright and Svidergol, 2019)*

## Turbo

Turbo, known as Turbo.net and formerly as Spoon and Xenocode, offers a technology for IT administrators to deliver applications in a more efficient way, by virtualizing the applications and delivering them to Windows devices via the web.



One of the USPs of Turbo is its publicly-available online portal, which gives users the ability to choose, access and stream their applications (Windows apps only) directly to their own device(s). This portal provides a simple user experience through 'one-click access' to any software application. It's for this reason that Turbo is often used by SaaS vendors and software publishers to deliver and deploy their applications to their own customers. Turbo's technology is successful thanks to what's known as the Turbo Virtual Machine (VM) Kernel. Like many virtualization solutions, the Turbo VM Kernel offers a way for users to run applications without going through the time-consuming process of installation and, more importantly, without requiring admin privileges, which many organizations naturally don't want to provide to most of their end-users.

### **Advantages of Turbo**

Packaging Studio creates a self-contained application that can be shared amongst users  
Offers an online portal and ready-packaged (unlicensed) applications ready to use  
More cost-effective than implementing a desktop virtualization solution to deliver applications  
Turbo enables support for legacy versions of Windows

### **Disadvantages of Turbo**

Limitations on packaging technology means that you are limited to the number of applications that can be packaged as it uses the ISOLATED method of virtualization.  
Apps run in containers so whilst portable offer no interaction with the local OS  
No evidence of heavyweight apps being able to run within this platform  
Online community is minimal and very little footprint in Higher Education

*(Spruijt, van Leeuwen and Monaghan, 2016)*

# Cloudpaging Vs. Vmware Thinapp Vs. Microsoft App-v



## Numecent Cloudpaging

Numecent's proprietary streaming logic is called Cloudpaging. Cloudpaging provides a really solid and well-rounded application streaming experience and does an excellent job of optimizing streaming. A central server can be set up for publishing applications to a web portal to allow streaming of the applications over HTTP. The streaming performance is very good over Local Area Networks (LAN), though one of the biggest selling points and where Cloudpaging really shines is the performance over Wide Area Networks (WAN).

One of the most powerful aspects of Cloudpaging is its isolation setting; Some competing products have multiple levels of isolation to choose from but still provide some level of isolation even in the least isolated mode. With Cloudpaging you can select to stream the application and run it with, or without isolation. This means that Cloudpaging can help you to virtualize 100% of Windows apps and is also much more dynamic than delivering all applications as layers.



Even without setting an application to run completely without isolation, Cloudpaging can handle drivers when running isolated. The fact isolation can be turned off on a per-app basis gives Cloudpaging a big advantage.

From a packaging perspective, most applications can be packaged from running a setup capture. This means less scripting, less MSI/Windows Installer packaging and less developing packages.

## VMware ThinApp

VMWare ThinApp is an application virtualization tool that allows you to perform a simple setup capture with the resulted output being written mainly to text files. The main project file allows you to change settings such as project name, application dependencies etc. as a Project.ini file; a simple text file. It can be opened, modified and recompiled within a text editor app. Making changes to modify the registry is straight-forward as well and mirrors the process of modifying settings. The registry text file can be opened, modified and recompiled within a text editor app. The level of isolation on a particular directory can, once again, be changed by opening an attribute text file, making any desired modifications and then recompiling.

Although the text files make it simple to get started, their primitive nature can be frustrating at times. Making a small cosmetic change to an application results in needing to recompile the entire app. If it's a large or heavyweight app then this process could be very time-consuming. As ThinApp is a portable application virtualization tool, ThinApp applications can be run from almost anywhere because users don't need to install software or device drivers. In addition, ThinApp applications don't need admin rights to access applications from remote locations, such as off-campus coffee shops, train stations, etc. This also comes at a price, though. Packaging tools such as Numecent's Studio and Microsoft's App-V Sequencer provide a graphical user interface (GUI) to set many advanced settings that may be confusing to figure out in ThinApp, or worse yet, not available as features in ThinApp at all.

It has been known for packagers to experience problems with ThinApp when virtualizing Office Add-ins on top of a locally installed Microsoft Office. Linking multiple large dependencies together can be challenging and not without issue. Some administrators say VMware ThinApp is more cumbersome to deploy. It's also trickier to manage applications because it doesn't come with a centralized management platform.

## Microsoft App-V

Microsoft App-V 4.x and 5.x are actually both in the top three application virtualization solutions in terms of user-base. App-V is by far the dominant solution in enterprise for various reasons. It was affordable for Software Assurance (SA) customers in the past, but is now free for all SA customers and comes as part of the latest version of Windows.

Whilst Microsoft App-V has a GUI editor and a sequencing capture tool, it does not have the simplicity of Cloudpaging's packaging, lacking organization in the GUI and lacking power in setup capture tools. With App-V the packager needs to know how to use Open Software Description (OSD) files, deployment config files and security descriptors. The learning curve for a packager is high with App-V as once you have sequenced or captured an application, a lot of technical knowledge is required to make the App-V package virtualize correctly.

App-V is quite rigid with its isolation and only provides a limited amount of control over the isolation of your apps. App-V 5.x introduced App-V sub-systems that have helped increase application compatibility, though to truly provide a simple and effective solution, App-V needs to provide the ability to optionally reduce the level of isolation. App-V provides centralized management, so admins can limit users' access to certain apps. The key to the success of all application virtualization products including App-V is the ability to package all applications, not just the majority. One of the issues with Microsoft App-V is that it can struggle to deploy 100% of all Windows apps.

## Cloudpaging Features

Cloudpaging is a foundational technology framework to transform native software delivery, deployment and provisioning from the Cloud, both public and private, and on-premise.

The Cloudpaging methodology relies on two key technology components: The VMMU and its Cloudpaging container. The VMMU (virtual memory management unit) works by paging a pre-virtualized image and virtualized instruction set of the application, and then directly paging those instructions into the Cloudpaging container residing on the local or virtual machine. Cloudpaging Studio reformats the software application into those pre-virtualized instructions, encrypts it, and divides them into code fragments called 'pages'. Those pages are then placed in a 'cloudified' format so that it can be consumed by the VMMU.



The 'cloudified' application is then fetched by the user's computer, by the VMMU and paged into a container. When the VMMU requests to start an application, it knows what needs to be pulled down to get the application running, which is typically 1/20th of the size of the application in the beginning. As the application executes, the VMMU is smart enough to pipeline additional instructions, which are required to run before the CPU is required to execute them. This VMMU is part of the embedded Cloudpaging Player (under 3MB) running in the kernel of the user's computer and incorporates the Cloudpaging container.

The VMMU and the Cloudpaging container provides two main benefits. First, it reduces the digital delivery time of any native application by 20 to 100 times by virtualizing the asset to be delivered in the Cloud (read further below). Unlike traditional remoting solutions, Cloudpaging does not transmit pixels from the cloud like RSH and RDSH, nor does it execute the application on a server. Cloudpaging transmits pre-virtualized software instructions from the Cloud (a page at a time and on-demand) which are then executed on the user's machine transiently. Cloudpaging's protocol and technology allow running applications over bad, high-latency Internet and WAN connections such as mobile and wireless connections a practical reality. Secondly, Cloudpaging technology can be used to deliver many types of digital assets, including Windows applications, Android applications, Operating Systems, and more while providing the highest level of compatibility between all applications and the operating system.

One of the benefits of Cloudpaging's VMMU is the enhanced application performance that end users will experience. One of the reasons for the accelerated speed at which an application can launch with Cloudpaging is that only the page fragments required to run an application are delivered from the Server to the Player immediately at startup. Once the app is running, the remaining page fragments are sent to the Player as needed. The result is a faster experience for the application user. In fact, independent software vendors are often interested in Cloudpaging technology to enhance their end users' experiences with their applications, while enterprises and IT departments enjoy the benefits of increased productivity and decreased support tickets.

Only five to ten percent of a software app is required to launch. With Cloudpaging's patented technology, applications can launch up to 95% faster and lighter. The reason for this is that Cloudpaging only delivers the page fragments required to launch an application on the Cloudpaging Player. In fact, the application is never installed on the end user's device. What this means for your bottom line is that fewer resources and less storage are required for an organization's end users to launch and run their "Cloudified" software.

Cloudpaging technology is unique because it combines the benefits of layers and containers, without any of the challenges and conflicts. Its distinct combination of dispositions allows your apps to run on any virtual or physical desktops with a Windows OS. During the initial packaging process, application files and registry keys are assigned to any of three dispositions. The first disposition is an isolated "layer" wherein assets are paged into a "container" and only visible to the application itself. This method of isolation is similar to the traditional application virtualization technology as seen in App-V and ThinApp. The advantage of the isolated disposition is the low risk of conflicts between the app and the OS. The second disposition is an integrated "layer" wherein the assets are paged into a container, yet visible to the local system and other traditionally installed or paged apps. This patented disposition is unique to Cloudpaging, allowing apps to behave as though they are natively installed, yet easy to lift and shift. The third disposition is an installed "layer" which is also unique to Cloudpaging. This physical disposition actually pages assets from the app onto the OS and restores original content upon deactivation, eliminating the need for cumbersome workarounds and scripts. The result of these dispositions is superior application compatibility.

## Product Spotlight: Cloudpaging

Numecent Cloudpaging is a traditional application virtualization and application streaming solution. However, unlike other application virtualization technologies, Cloudpaging uses flexible virtualization layering which allows a mixture of 'isolated' and 'integrated' virtualization modes to be used within a single virtual structure/bubble at a granular file, folder, registry key level.

The key differentiator between Cloudpaging and other technologies is the flexible layering, allowing for a much higher success rate when virtualizing applications. In particular, the 'integrated' virtualization mode which allows for the virtual structure/bubble created to have native 2-way access. This means that external applications or Windows OS processes can interact with the application inside the virtual structure. It also allows for the application within the virtual structure/bubble to interoperate with existing applications or Windows processes on the client device. This is key as most Windows applications are designed and expect to interoperate with other applications along with the running OS. Restricting applications in an 'isolated' virtual structure only can cause the application to behave incorrectly or in many cases cause a loss of functionality.

However, Cloudpaging also supports and allows the more traditional 'isolated' virtual layer to be used. The unique point, however, is that both 'isolated' and 'integrated' layers can be used within a single virtual structure at a granular level. This allows for complete control at file, folder and registry key level. Having the options of the different virtualization modes allows for the situations/use cases where files, components or pre-requisites required by the application need to be isolated.

A typical example is legacy applications that require an older version of the Java Runtime. These components or runtimes can be 'isolated', so they do not interact or cause any interaction with other applications or versions of the component/runtime on the user's device. However, the virtualized application may require or interoperate with other applications or Windows OS components. In this case, the virtualized application its self would be configured to use the 'integrated' virtualization mode. This then allows the application to interoperate

required with other components/applications. Both the application and components/runtimes are contained within the same virtual structure, so the virtualized application can access these 'isolated' components/runtimes. The application its self is using the 'integrated' virtual layer, so it is able to seamlessly interoperate with external applications and the operating system.

Cloudpaging also uses next-generation application streaming technologies (labeled 'Cloudpaging'). This is an evolution of the traditional block-based streaming which allows for a smaller block size to streamline the delivery of content. It also improves the performance of the virtualized application by allowing fast and more targeted segmentation of the virtualized application.

Following a user's app request, validation of the requesting user and validation of an app's license, the delivery of the application is initiated via virtualization. The app is served from distribution points that deliver 'virtual structures/bubbles' containing the app, which is isolated from OS. A constant connection is maintained during this step to deliver the minimum essential components of the app first, followed by non-essential components upon user request. The user requests these non-essential components simply by using the areas of a piece of software they are tied to, for example, context menus, GUI features, etc. The packaged apps must be consistently stored somewhere as an origin for their delivery; these are referred to as distribution points and also come in the form of servers.

An example of this process for AppsAnywhere used with Cloudpaging is as follows:

End-user connects to AppsAnywhere portal and, following validation of their context and credentials, requests an application.

Admin License Servers otherwise referred to as Cloudpaging servers, provide licenses the AppsAnywhere portal.

The app request reaches what's known as the Paging Servers or distribution point, which then serves the app to the end-users' devices

Some useful & related reading...

[RFP template for application delivery, virtualization and VDI solutions >>](#)

[Alternatives to discontinued Dell vWorkspace and flexiLabs solution \(formerly Quest\) >>](#)

[Alternatives to Symantec's end-of-life Workspace Virtualization and Streaming product >>](#)



# 5. Application Virtualization Vs. Desktop Virtualization



# Differences Between Application Virtualization And Desktop Virtualization

The most obvious difference between application and desktop virtualization is that application virtualization focuses and only contains the application content. With desktop virtualization, you are virtualizing the user's entire desktop OS. The virtual desktop will normally be running in a datacentre on server hardware. This could be a public, private hybrid cloud or on-premise.

Using application virtualization with desktop virtualization allows you to virtualize a user's entire desktop environment. It also allows for reduced virtual desktop image size as the applications do not need to be physically installed on the virtual desktop.

Both application and desktop virtualization have their place in today's organizations. Desktop virtualization has many use cases that suit such a technology better than application virtualization alone, such as in cases where an application needs to be next to its data; a client/server application for example. In this case, the application will connect to a database or service which it requires to function. Having the Client component of the application running locally on a user's device may lead to poor performance depending on the distance and connectivity to the backend database or service infrastructure. Having the application (Client) and database/service instance (Server) in the same datacentre allows for fast connectivity and thus better performance.

The other common use case is to support non-Windows operating systems. At the moment all application virtualization solutions that are available only support Windows applications and Windows operating systems. With a desktop virtualization solution, a Windows operating system environment can be provided and accessed from a non-Windows device. In this situation, it is also still common for application virtualization technologies to be used as well to deliver applications into the Windows virtual desktop but obviously, they cannot deliver the applications to the end user's non-Windows device.

## Will Just One Technology Suffice, Or Are Multiple Required?

Application virtualization and desktop virtualization often get banded as the same, with the defining line between the two technologies often feeling blurred or unclear.

Desktop virtualization, or VDI, is an entire virtualized desktop. This means that it is running on a server, rather than the end device. Example products in this area are Citrix Virtual Desktops (formerly XenDesktop), VMware Horizon 7 or Parallels RAS (Remote Application Server).

Application Virtualization is the virtualization of an individual application, or perhaps a suite of apps. The application still then needs to execute somewhere (you're not virtualizing anything more than an application). Typically it will run locally, utilizing the processing power of the end device. It is, however, important to highlight that you can also deliver a virtual application to a virtual desktop. Example products in this space are AppV from Microsoft, Cloudpaging from Numecent, VMware's solution (Which is now rolled into their Horizon suite and was formerly ThinApp) or Turbo.

As to whether one is better than the other, both technologies have their place in a desktop estate. It depends on the use case as to which one fits, if not both. Typically we would see the use case for VDI being a subset of users, or for a specific purpose, or application.

The generally overarching use case for desktop virtualization would be consistency or compliance; With VDI, every desktop will be the same as they are all running in the same place. Administrators are able to spin up new desktops on-demand with the knowledge that they are always the same, regardless of the device the end-user is accessing from. This is also a use case for application virtualization, where the end device may be able to run the application. It could be a windows application and the user be on say a Chromebook, or tablet, so the application needs to run elsewhere.



Compliance is the other major factor. Perhaps the user is in an industry with strict data rules or accessing sensitive content that an organization doesn't want to be saved locally. Running a virtual desktop minimizes the risk of rules being broken and helps maintain compliance. The consistency and compliance of virtual desktops will bring some benefits to an IT department which is why many IT administrators are proponents of virtual desktops.

By utilizing application virtualization you strip a desktop down to its bare bones, perhaps limited to Windows10, Microsoft Office and commonly used items, such as a pdf reader or web browser. The advantage of this is that your imaging process is simplified and often reduced to a single image. You can then use application virtualization to deliver software on-demand, where and when it's needed. Application Virtualization has perhaps got a bad reputation because of limitations with the more commonly known tools. With Cloudpaging from Numecent, you can virtualize 100% of Windows applications, even those legacy apps, and those with device drivers, plug-ins or services. One good use of application virtualization can be for applications that require an old version of Java; you can virtualize the old Java and only let it work with the given application and still have a more recent version installed on the end device for any other applications that require it, helping mitigate security issues.

As for a comparison between the tools, perhaps it's fair to think of application virtualization as the engine in the car; the delivery/deployment tool and virtual desktop as the body of the car, the piece that the engine fits into, and the element that users see and interact with.

As for which is right for you it's more about understanding your users. What are they trying to do, what apps are being accessed, where are they located, the end device and perhaps their role in the organization. Analyzing some of those will help you determine the scenarios where either of the technologies is the better fit.

# Comparing Costs

|                                     |  <b>Application virtualization</b> |  <b>Desktop virtualization</b> |
|-------------------------------------|---|---|
| Virtualization license cost         | \$\$  | \$\$\$  |
| Server hardware cost                | \$  | \$\$\$  |
| Application packaging cost          | \$\$\$  | \$\$  |
| Staffing cost                       | \$  | \$\$\$  |
| End-user hardware cost              | \$\$  | \$  |
| Additional Microsoft license costs  |   | \$\$\$  |
| <b>Overall cost to deliver apps</b> | <b>\$</b>   | <b>\$\$\$</b>   |

Some useful & related reading...

[What's after VDI? Replacing virtual desktops with application virtualization at PSU >>](#)

[The Slow Demise of VDI \(and The Rise of The University App Store\) >>](#)

[The Slow Demise of VDI \(and The Rise of The University App Store\) >>](#)



# 6. The Cost Of Application Virtualization

## How Are Application Virtualization Technologies Priced?

With a wide range of technologies available from many different virtualization vendors, each application virtualization tool is priced differently. Generally speaking however, these technologies are all priced through an annual license model (link to hidden costs in app delivery post). Depending on the technology and your requirements, you can usually pay for a device license – sometimes known as a user license, which covers only a set number of devices – or a site license, which will cover every user or employee and give you the flexibility to increase this without needing to change your billing model with the vendor. Often vendors will stipulate that a site license is required to deploy virtual applications off-site or to BYO devices.

Annual license models involve an annual fee, as the name suggests, which is essentially a subscription to the technology. The benefit of technologies that are licensed and priced through this model is that all maintenance is usually included as part of the contract, including all upgrades and support. Some application virtualization technologies may follow this model but bill their customers on a monthly basis, to help their customers tie in the costs with budget cycles etc.

This model is very different to the ‘Perpetual Licensing’ of desktop virtualization (VDI) technologies where, on many occasions, the customer pays upfront for a particular version of the technology but must then also pay an annual maintenance charge for upgrades and ongoing support. In nearly all cases, VDI solutions also require additional Microsoft licenses to enable the delivery of application to BYO devices. Those are called VDA licenses (Virtual Desktop Access) or for other solutions, CALs (Client Access Licenses). Application virtualization technologies almost never need either of those licenses. That’s because the end-users aren’t accessing their applications through a virtual or remote desktop, the applications are being ‘streamed’ to them and are running locally on their device!

That summarizes how application virtualization technologies are priced from a *licensing perspective* but it's also worth noting the other costs involved and how they stack up against comparable technologies (such as VDI) on the market. One key thing to consider is the hardware cost or CAPEX (capital expenditure) required for application virtualization; almost non-existent!

As a solution used to deploy applications, the infrastructure and upfront investment required for these technologies is minimal in comparison with what VDI customers must pay on day 1. Application virtualization technologies usually only require a few lightweight servers, namely streaming servers and a licensing server, to support tens of thousands of users. Conversely, when using a desktop virtualization solution to deliver applications, to support 10,000+ users requires an eye-watering amount of physical technology to be invested in. Of course, these days this can be cloud-based, but at an even greater cost! The reason for the minimal infrastructure with application virtualization is down to how the technology itself works; essentially streaming applications where they are run locally on the end-user's device, rather than running on a centralized server somewhere else on-site.

The other costs involved, as with any technology solution, are that of staffing or resource costs. To maintain an application virtualization environment, you'll need at least one expert who has experience in packaging applications to get them in a state ready to deploy using virtualization. This skillset is common in enterprise and higher education IT, where administrators have been packaging applications in one way or another for decades, for example with App-V or other similar technologies. Once the applications are packaged, there is very little additional expertise required to deliver the applications. The skills that this does require are usually readily available within any IT, end-user computing (EUC) or desktop team.

[Read more about the costs of delivering software within higher education >>](#)

## What Do They Cost Per User?

One of the huge benefits of application virtualization versus desktop virtualization is the ability to take advantage of endpoint computing resources. Instead of having to build a huge backend to process and compute all of those apps, with application virtualization, IT departments can push apps down to the endpoint hardware where they belong. Not only this, but if the institution supports BYOD, now they have FREE hardware to run apps and can start to right-size their lab spaces thus saving even more money

With minimal hardware costs for application virtualization solutions, the only cost that you need to consider is the cost of licensing the technology solution itself. Except for Microsoft App-V (see below for information around App-V pricing), the majority of application virtualization technologies are priced per user or device. Certain vendors such as Cameyo and Turbo.net offer fully-hosted versions of their virtualization solution, in which the price per user is greater but the vendor offers the full service, without you needing to host anything yourself onsite.

The per user cost of application virtualization solutions varies massively from solution to solution. However, all of the technologies available on the market do offer heavily-discounted pricing for education institutions such as universities and colleges.

So, why is App-V free? Largely, Microsoft focuses on the enterprise when developing its licensing strategies around its many product offerings. To help customers migrate from Windows 7/8 to Windows 10, licensing for App-V was rolled into Enterprise Licensing for Windows 10. This signifies the understanding from Microsoft that not all line-of-business applications will run on Windows 10 in a native format, and that an application virtualization solution might be needed. Most enterprise and higher ed customers will maintain a campus-wide or volume licensing agreement with Microsoft and thus will now have access to App-V licensing free of charge. App-V is included as standard within Microsoft's Desktop Optimization Pack (MDOP).



For up-to-date pricing of the other application virtualization technologies, see the vendors' websites for more information, including details of hosted vs self-hosted and education vs enterprise pricing.



## What Are The Ongoing Costs Of App Virtualization?

Let's review licensing costs of these technologies first, and the ongoing costs for those. There are two models prevalent in the application virtualization space. The first is a more traditional model usually made up of a perpetual license plus ongoing annual maintenance cost. The annual maintenance entitles the customer to upgrades and technical support, usually provided in various tiers of coverage, response time, and cost. The second is the ever-increasing annual subscription model. Here, both license and maintenance fees are rolled into one and billed annually or sometimes monthly. The advantage to a subscription model is avoiding the need for a huge cash outlay for perpetual licenses and often adds a more predictable set of figures for ongoing budget forecasts.

When it comes to the longevity of application virtualization within your organization, as a technology it is much more adaptable to change as you grow. This flexibility and scalability is thanks to the technology harnessing some of the power of the end-user's device, rather than relying on a backend infrastructure. So, as you add more users to your app virtualization estate, you don't need to increase the hardware requirements to support them like you would with a VDI solution.

You can continue to add more and more end-users with application virtualization, and at most all you will need is some additional lightweight stream servers and perhaps licensing servers, too. And as you grow you'll be unlikely to need additional packagers or IT administration staff, because you'll have a similar number of applications that need to be deployed regardless of how many end-users you have accessing them.

As you'd expect with any software solution, as you increase the number of 'seats' or end-users that you're licensing, the overall price you pay for the application virtualization tool will increase. That said, many vendors will negotiate a lower per-user cost when increasing the total number of users within your contract.

Some useful & related reading...

[Exploring the hidden license costs in application delivery and VDI solutions](#)

[Examining the costs of delivering software apps in Higher Education](#)

[Deliver apps to any device at a fraction of the cost of traditional VDI solutions](#)



# 7. Managing & Deploying Application Virtualization

## What Hardware Is Needed For Application Virtualization Solutions?

*An indicative example of using application virtualization technology at a typical university*

To support an end-user computing environment of around 15,000 – a typical EUC environment at many higher education institutions across North America and Europe – with Cloudpaging from Numecent, you'll need approximately 4 'paging' servers and 2 admin licensing servers.

Many Software2 customers can deliver all their software titles to any device, including enabling BYOD and delivering 'heavyweight' applications, with the addition of 3 AppsAnywhere servers on top of that. So, with fewer than 10 servers (plus databases) you can support 15,000 students in a typical application virtualization estate. Without exception, our customers have been able to implement this environment onto their existing virtual infrastructure, without the need for additional hardware.



## What Staff Resources Are Required To Manage An Application Virtualization Estate?

A key area in which application virtualization reduces costs for organizations is the staffing resources required. The workload necessary to deploy and manage an app virtualization estate is significantly lower than the viable alternatives of either desktop virtualization or traditional imaging, both in terms of quantity of staff and breadth of specialized skills required.

“ *With app virtualization, there is quite literally less to do and fewer skills required to do it.* ”

For deployment, application virtualization relies upon packaged applications; apps separated from the underlying operating system by a virtual structure. To achieve this, packaging experts are needed to properly and consistently ‘place’ applications into these virtual structures. Packaging is a comparatively transferrable skill in the field of IT and the sub-field of software delivery. From a skillset perspective, many organizations’ packagers are not limited to packaging; they’re able to take an active part in the rest of the delivery process; technical packaging knowledge is required, though it can be learned by experienced individuals with relative ease. The general process of packaging has common themes between different application virtualization technologies and can even be required for certain desktop virtualization and imaging methodologies and approaches. In Software2’s experience, it is not uncommon to see two, or even one packager manage app virtualization estates of up to 400 apps. This is helped further by the seasonal nature of packaging and the elimination of the need to reimagine and redeploy everything in order to patch, update or add single software titles.

“ *It only takes a handful of experienced packagers to manage a huge application virtualization estate.* ”

Many service-oriented application virtualization vendors are also experts in packaging applications, with teams of designated packagers to help them advance their product and to share knowledge and best practice with their customers. There are three common ways in which vendors may help their customers with the task of packaging, those being packaging training, pre-packaged apps/recipes or packaging subscription services.

## Packaging Training

For organizations wishing to keep their services and packaging entirely in house, especially those intending to build a department around delivering software via application virtualization estates, packaging training may be the way to go. Some vendors provide internally certified/accredited training with specific products. The merit of this approach is self-sufficiency with the detriments being the lack of guarantee that in-house resources will be able to package a specific app or app set and that any changes to a virtualization product may necessitate further packaging training.

[Read more about Software2's packaging training and certification >>](#)

## Pre-packaged Apps/Recipes

Vendors may opt to sell pre-packaged applications or recipes for packaging specific applications on a per-app basis. One-off, flat costs would deliver one of two things to purchasing organization. Purchasing pre-packaged apps would likely be delivered in the form of a file, or a set of files which would require a small amount of configuration (dependent upon the specific app package) to slot into an application virtualization estate and become entirely deployable. That configuration is likely to be some slight amends to file paths and references to make the package compatible with specific server architecture, replacement of any organization-specific or user-specific details, and the input of registry keys or registry key systems. The benefit of this approach is that the packaging is handled almost entirely out-of-house and so staff workload and time required is reduced. Many application virtualization vendors provide free recipes detailing step-by-step instructions on how to package specific apps; particularly for open source apps.

## Packaging-as-a-Service (PaaS?!)

With the 'as a service' business model in full swing, many application virtualization vendors now offer packaging as a service. The simplest form this comes in is subcontracting the time of one or more of a vendor's packaging experts, effectively totally outsourcing the packaging process. The greatest benefit of this is that an organization is guaranteed to be delivered a fully-functional and ready-to-go packaged app specific to that organization and their unique details and license keys. A negative to this is that contracting third-party packagers' time is often the least cost-effective approach.

New models for packaging services are beginning to surface in the application delivery space; models that allow vendors to serve more than just one customer at a time and therefore reduce the cost of outsourcing packaging for customers. Subscription-based services providing different levels of access with optional extras are growing in popularity in other areas of business and software delivery is no exception. Giving organizations access to a large library of pre-packaged apps for a regular fee allows packagers to cease the

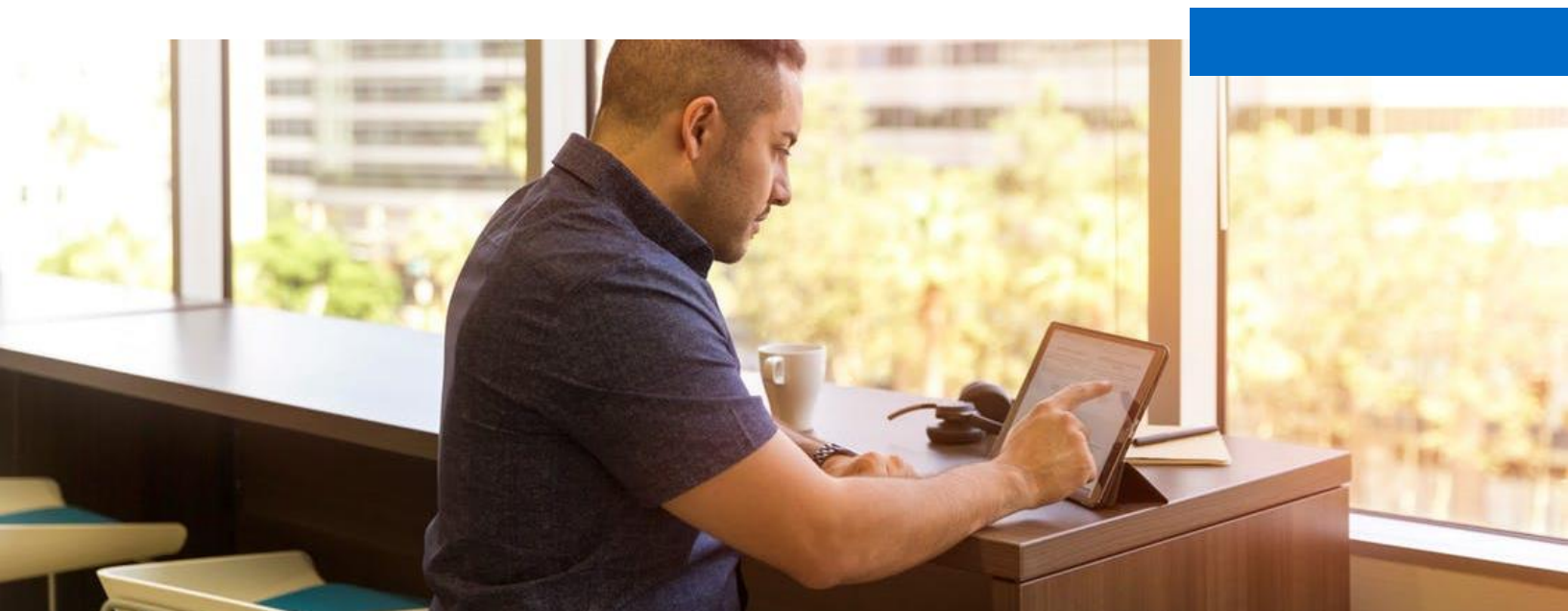


duplication of work to deliver more quickly, and to pass cost savings on to the customer. Software2's new Packaging Subscription Service even offers seasonal discount for those organizations who are able to preempt their needs and confirm packaging requirements in advance. This is instrumental in meeting the restrictive timescales of needing everything virtualized and ready to deploy for the start of each semester.

The benefits of packaging as a service are numerous; All of the benefits of the previous approaches apply alongside a quicker turnover time, more availability in the context of booking packaging time and by far the most cost-effective method of outsourcing packaging.

[Read more about Software2's revolutionary Packaging Subscription Service model here >>](#)

Following the packaging of apps, it's just a case of publishing the packages to a preferred deployment solution, such as Microsoft SCCM or AppsAnywhere. Deploying a single image through SCCM is a small task and given the simplicity of deploying packaged apps through AppsAnywhere, deployment becomes a much less laborious and complex task; one which is more of an admin task and would be safe in the hands of a junior member of IT, or even a student employee. This frees up more experienced and higher skilled members of IT to address more urgent or key strategic matters.



Virtualized apps are available on-demand and packages function as if the app were locally installed. Virtualizing and deploying applications is a more friendly and robust process for end-users, minimizing the chance of any obstacles and, by extension, support tickets. This enables support staff to work on other, more urgent tickets, higher priority projects or reduces the number of support staff required altogether. It is worth still considering a single, principal image with most-used apps that must be omnipresent within a campus, such as Microsoft Office, web browsers or PDF readers. Even using virtualization and a basic software image in tandem, there is still only one image to manage and it is common to ALL managed machines. This removes the need to rebuild images and reimage countless cross-sections of lab machines to perform updates, patches or additions.

To put this into perspective, all of the previously mentioned staff resources required for application virtualization are also needed for desktop virtualization. Alongside this, VDI also calls for several highly-experienced system admins and system architects. Packaging is still a challenge that must be overcome (particularly in cross-platform delivery) and support team requirements are the highest for any of the viable software delivery approaches. Not only is more support staff essential, unless a very significant investment has been made in high-end hardware, many insurmountable support tickets may be logged due to application performance issues.

## Deploying Applications With Application Virtualization

Before an application can be deployed using virtualization the original app installer must be converted to the relevant format. This process is generally known as packaging, or more specifically Sequencing (App-V) or Cloudifying (Cloudpaging). The procedure is broadly similar to that used for legacy MSI creation, however the tools available for app virtualization are generally easier to use and provide faster results. This is because virtualization allows for the isolation of the application files from the rest of the system. Once packaged a virtualized app can be widely deployed with little to no risk of conflicts (with files or keys that already exist on a target system).

Let's consider the process and benefits in more detail. The general approach of most packaging tools is to record the execution of the original installer in a 'vanilla' OS environment. That process is generally known as 'capturing'. The goal is to capture only the intent of the original installer, excluding all background processes. Most tools will pick up some background noise – to a greater or lesser extent – which then requires the packager to review the results and to manually exclude unrelated files from the package. With MSI deployment tools such as AdminStudio, there is only one option – that is to delete files and keys that are not needed. If a required file is accidentally removed, the app will not work when it's deployed.

Virtualization solves this by providing the ability to isolate all or parts of an application. Solutions such as App-V sequence the app into a completely isolated package that will not conflict or even interact with other applications on a target machine. This removes all conflicts but comes at the expense of compatibility and interoperability. Apps that contain drivers or services often do not work in isolation, so the overall effectiveness of the solution is reduced. Industry experts generally expect a 60% success rate with App-V sequencing, rising to 80% given extended efforts by expert packagers.

Numecent Cloudpaging solves this interoperability issue in a unique way known as configurable virtualization. Any part of the app can be completely isolated from the target

system (as with App-V), written to disk such as with a driver or service, or exist in a mixed mode known as the integrated layer. These settings can be applied generally at the time of capture, or at an individual file and registry key level. Multiple captures at different layers allow for the quick and easy isolation of pre-requisites such as Java and C++ redistributables, whilst allowing the main application to operate as if natively installed. Cloudpaging Studio is also widely regarded as one of the best packaging tools for producing clean captures due to a comprehensive pre-filter, so there is less clean-up to do after the initial capture of the app installer. In contrast with legacy packaging tools, if the packager is unsure whether a file or registry key was captured as part of the original installer, that file can simply be moved to the isolated later where it remains available to the app, but will not conflict with any target system.

Older snapshot-based tools, which compare the system state before and after the app installation, have long been surpassed by the above approaches and are not worth considering. The high level of noise that they capture makes them impractical due to the time required to clean a package.

Once we have our virtualized app it should be possible to deploy to any machine. Tools such as ThinApp can provide a single file which may be widely copied and executed without license control. However, in most cases, it's desirable to be able to manage the deployed apps from a single admin interface, such that access can be revoked if necessary, and the app updated when required.

These days, most modern app virtualization solutions provide a web-based portal where end-users can login to install or launch their apps. This approach is rapidly growing in popularity with VMware, Citrix, Microsoft and Numecent all providing such portals. Often, the ability to create legacy desktop or Start Menu shortcuts for virtualized apps is also supported, however, these must be managed by Group Policy or SCCM.

In all of the above cases, access can be restricted to AD users, groups or machines. However, this simplistic domain-joined view of app deployment is often limiting in the modern workplace, where remote working and BYOD are now commonplace. AppsAnywhere from

Software2 aims to address this by integrating with nearly all of the leading app deployment solutions. AppsAnywhere provides a single app portal with additional license controls such as device ownership, network connection, device type and geolocation – allowing users to roam with their apps whilst also adhering to software license terms.

Of particular note is the ability in Cloudpaging to make apps available offline, whilst still retaining license control. For laptop users this is a must, providing for native, work anywhere experience. A permitted offline period can be set as well as an expiration date for each app. Once reached, access is revoked and the app can be removed automatically, even if the user is offline and the app has previously been virtualized to their machine.



## How Do You Deploy Applications

Traditionally applications are deployed based upon the end-user's device. In most organizations, an image is created and assigned to a machine. The image may contain a base level of applications. Some additional applications can then be installed on demand by the user. In addition in some environments, this can lead to extremely large and complex images. Sometimes into the 100GB's. It is possible to reduce the size of these images by using faculty/department-specific images. An example of this would be to only image machines in an engineering department with heavyweight apps such as AutoDesk Suite, MATLAB and Solidworks. However, multiple images can be just as difficult and time-consuming to maintain than a single, large image. Managing these, deploying these and eventually updating these all becomes cumbersome and time-consuming work. This model lacks flexibility and does not account for the dynamics of the application layer. Updates and changes are common but having to re-image a machine is not a trivial task.

With application virtualization, the whole paradigm of application delivery can change from machine-based to user-based. Application virtualization enables true separation of the application layer from the Operating System. This means you can deploy one single image across an entire estate and then layer down applications on-demand to any user on any machine. And of course, the actual imaging process is vastly improved. Reduce the image to the base OS with some core service apps and it becomes easy to manage, deploy and change.

When you realize the potential of separating applications from the OS, hidden benefits become all too apparent. Use different versions of the same application on the same device, isolate an application (or parts of) from the underlying operating system and conflicts can be removed, update applications with zero touches of the endpoints and of course provide applications to unmanaged devices (BYOD).

With this approach, you can achieve ultimate flexibility. Any application can be delivered to any user on any device conflict-free.

## Updates, Access, Control

In the interest of providing a consistently high-quality user experience, and minimizing IT support required by users, access to applications should always be on-demand and through an application store. Users should not be expected to be experts in different Operating Systems and experience should be the same regardless of such. Once an application is deployed from an AppStore using application virtualization, it is much easier to maintain and update. Any update or upgrade can be pushed out on-demand, whether rolled it out universally and organization-wide or by rolling out to groups of users in a specific order. Application updates become much simpler when IT only has to handle the update and not worry about the other applications, conflicts, and the OS.

Furthermore, most application virtualization solutions utilize a client to control the application. This provides an additional layer of license control; It means you can revoke access at any given time, expire applications for users, etc.

Finally, some of the solutions available monitor the usage of the applications you have offering a greater degree of business intelligence to ensure you are getting the most value from your software assets and number of licenses.

Some useful & related reading...

[Combining Cloudpaging and App-V to virtualize 99.9% of apps with AppsAnywhere >>](#)

[Delivering Apps on Demand at University of Michigan – Webinar >>](#)

[How to deliver Windows apps to non-Windows devices >>](#)



# 8. Glossary



**Application layering:** Application layering is the technology and principal behind application virtualization; the process of dividing applications into discrete, base components, separate from the operating system they're to be executed on in a way that allows them to still communicate with that operating system.

**Application packaging:** Application packaging is the process assembling a collection of files that make up the structure of an application and bundling them into a package/appset, targeted for automated deployment. These packages are tailored to meet the installation requirements of specific environments and corporate standards. This is done using a number of methods including, but not limited to:

- Configurable app events (which are set to trigger at various points of virtualization)
- Virtual isolation and the use of environmental variables within the registry to replace any hardcoded paths.

**Application streaming:** Application streaming is the method of delivery for virtualized applications. Only essential portions of applications are initially streamed to the end-user device with non-essentials, such as some context menus and functions not required for start-up being streamed on-demand.

**Containers and containerization:** TBA

**Image:** A snapshot or exact replica of the contents of a digital storage device, whether that be a storage device, an install package or an entire machine.

**Image size:** The overall size of a disk image in Megabytes or Gigabytes that represent the complete contents of the image including the Operating System, Drivers, Software, and Preferences.

**Integrated virtualization:** Any type of data, in this case, packaged application files, that is separated from the OS it is to be executed on by a virtual structure, but is still visible to that OS as if it were locally installed.

**Isolated virtualization:** Any type of data, in this case, packaged application files, that is separated from the OS it is to be executed on by a virtual structure and is only visible within the virtualization application. Also referred to as 'sandboxed'.

**Unified application delivery:** Unified Application Delivery is the ability to integrate multiple application delivery technologies and bring them together into once concise and easy to use service, delivered in a customizable, centralized Appstore.

## Postamble

With great thanks and appreciation to the following authors, contributors, publications.

Tony Austwick | Peter Cooke | Nick Johnson | Umair Kahn | Phil Morgan | Scott Redfern | Matt Rodgers | Phil Spitze | Leon Walsh

Spruijt, R., van Leeuwen, J. and Monaghan, R. (2016). Application Virtualization Smackdown

Wright, B. and Svidergol, B. (2019). Virtualizing Desktops and Apps with Windows Server

eavena, U., Mariano Gorzelany, A., nickbrower, U. and Lich, B. (2019). Overview of Application Virtualization

## Bibliography

eavena, U., Mariano Gorzelany, A., nickbrower, U. and Lich, B. (2019). Overview of Application Virtualization. [online] Docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/microsoft-desktop-optimization-pack/appv-v4/overview-of-application-virtualization> [Accessed 27 Jun. 2019].

Wright, B. and Svidergol, B. (2019). Virtualizing Desktops and Apps with Windows Server 2012 R2 Inside Out: Planning and Implementing App-V | Microsoft Press Store. [online] Microsoftpressstore.com. Available at: <https://www.microsoftpressstore.com/articles/article.aspx?p=2346348> [Accessed 27 Jun. 2019].

Spruijt, R., van Leeuwen, J. and Monaghan, R. (2016). Application Virtualization Smackdown. 5th ed. [ebook] PQR, pp.30-51. Available at: [https://www.software2.com/downloads/16/Whitepaper\\_Application\\_Virtualization\\_smackdown.pdf](https://www.software2.com/downloads/16/Whitepaper_Application_Virtualization_smackdown.pdf) [Accessed 27 Jun. 2019].

# Next steps to improving the way you deliver applications...

Get a personal walkthrough of how AppsAnywhere works from one of our product experts

Arrange a demo

Experience AppsAnywhere for yourself in our free, hands-on trial environment

Try it now

# Software2

software2.com