

Case Study: Gateley Legal – Moving to a Virtualized Application Model with App-V and SCCM

A business and operational change project to replace a thick OS image and local MSI packages with App-V virtualization, delivered through SCCM.

Overview

Gateley Legal relied on a traditional desktop build: a thick Windows image with business applications baked in, supplemented by locally installed MSI packages. This approach worked, but it made the estate harder to maintain, slower to change and more complex to manage as application demands grew.

I delivered a major business and operational change project to move Gateley Legal to a virtualized application model, using Microsoft App-V integrated with System Center Configuration Manager (SCCM). The result was a cleaner, thinner operating system image and a more flexible way of delivering applications to users.

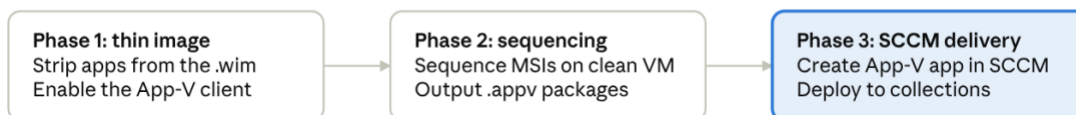
The Challenge

Thick images and locally installed MSIs create a tight coupling between the operating system and the applications on it. Every application update risks touching the base image, and every image change needs careful testing against everything it contains. The goal of this project was to break that dependency, so that the operating system and the applications could be managed, updated and delivered independently.

The Approach

The project was delivered in three phases, each building on the last.

Three phases take apps from MSI to SCCM-delivered bubbles



delivery phases · 3 steps

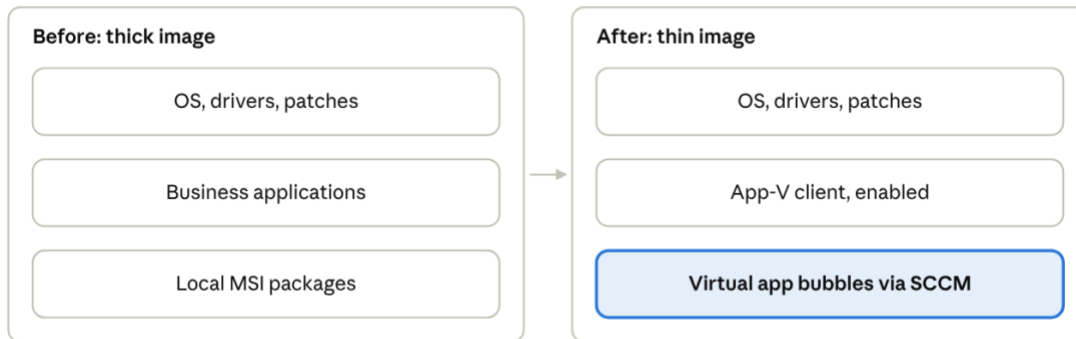
The work moved from a lean OS image, through sequencing, to SCCM delivery.

Phase 1: Clean Up the Windows OS Base Image

Moving to application virtualization meant the Operating System Deployment (OSD) task sequence needed to follow a thin image strategy.

- **Stripped the base image:** All business applications were removed from the primary Windows golden image (.wim).
- **Kept it clean:** The image was limited to the Windows OS, required hardware drivers, security patches and baseline configuration settings.
- **Enabled the App-V client:** From Windows 10 (1607) onwards, and in Windows 11 Enterprise, the App-V client is built into the OS. No agent installation was required. The client only needed to be enabled via PowerShell (Enable-Appv) or Group Policy.

A thin image keeps applications out of the OS build



thick vs thin image · before and after

Applications left the golden image and now arrive as App-V bubbles.

Phase 2: Convert and Sequence Applications

Rather than installing traditional MSIs, applications now run inside an isolated virtual “bubble”. This meant repackaging existing MSIs as .appv virtual packages.

- **Built a clean sequencing VM:** A dedicated, non-domain-joined virtual machine matching the production OS bitness, with the App-V Sequencer (part of the Windows ADK) installed. A clean snapshot was taken before every build to keep each capture consistent.
- **Sequenced the applications:** The Sequencer was run, the traditional MSI installer launched, default application settings configured, and the capture stopped.
- **Produced the package outputs:** Each sequence generated a .appv package, along with optional _DeploymentConfig.xml and _UserConfig.xml files for tailored scripting and shortcut delivery.

Phase 3: Integrate and Deploy with SCCM

No separate standalone App-V server infrastructure was needed. SCCM acted natively as the Electronic Software Distribution (ESD) system, controlling content routing and virtualization rules.

1. In the SCCM console, navigate to **Software Library > Application Management > Applications**.
2. Create a new Application and set the type to **Microsoft Application Virtualization (App-V)**.
3. Point to the .appv file on the network share. SCCM reads the application data and configures default detection methods automatically.

4. Where applications needed to interact, for example an Excel add-in working with a virtualized Excel, configure an **App-V Virtual Environment** (Connection Group) in SCCM so the virtual bubbles can communicate.
5. Distribute content to Distribution Points and deploy to User or Device collections.

Benefits Delivered

- **A thin, clean golden image** containing only the OS, drivers, patches and baseline configuration, so it is simpler to build, test and maintain.
- **Independent application lifecycle**, with applications updated and delivered without rebuilding the OS image.
- **Isolated applications** running in their own virtual bubbles, reducing the risk of conflicts between applications.
- **Reduced infrastructure overhead**, as SCCM handled delivery natively with no standalone App-V server estate.
- **Flexible targeting**, with applications deployed to users or devices through SCCM collections.

Looking Ahead: MSIX

App-V is on a fixed extended support lifecycle ending in 2029. For any organization starting this transition today, it is advisable to evaluate MSIX alongside, or instead of, App-V. A direct move to MSIX using tools such as the Microsoft MSIX Packaging Tool avoids a second major packaging migration in a few years' time.

Summary

By separating applications from the operating system, this project gave Gateley Legal a leaner base image and a more manageable, flexible approach to application delivery. Combined with SCCM's native App-V support, it delivered meaningful operational change without adding infrastructure complexity.