info@apriorit.com
www.apriorit.com

Case study

Enhancing the Security and Performance of a Virtual Application Delivery Platform

Our client is a US-based company providing a Virtual Application Delivery (VAD) platform for remote access to digital workspaces. They wanted to enhance VAD platform security and performance by adding new capabilities to their platform and needed quality support for existing features.

The Apriorit team implemented several custom features to provide a better user experience, improve VAD security, and make the client's product more efficient.

The client

Cameyo is a US-based VAD service provider that operates worldwide. The company's VAD platform provides customers with secure access to business-critical apps on any device, from anywhere, without the need for VPNs.

The challenge

Our client wanted to know how to secure a virtual application delivery platform and ensure efficient and timely support for their product while introducing new features. In this way, they wanted to address the requests of existing customers and attract new customers. They were looking for a team of experienced developers who could adjust to fast-changing priorities and project goals while maintaining the overall quality of delivered features.

Our approach

After analyzing the project and discussing possible improvements with the client, the Apriorit team focused its efforts on achieving key five goals:

We split these technical tasks into two categories:

1. **Supporting existing functionality** by fixing bugs discovered by our quality assurance (QA) specialists as well as by addressing bugs and errors reported by the platform's end users

2. **Research and implementing new functionality** to increase the product's value for both existing and potential new users

To address our client's needs and improve the product's viability, we formed a team that consisted of a project manager (PM), a QA specialist, and expert Windows software developers with keen knowledge of C/C++, Go, and Remote Desktop Protocol (RDP).

PROJECT DETAILS

TEAM

- PM
- QA
- Backend developers

TECHNOLOGY STACK

Languages:

- C/C++/C#
- Go
- Java J2EE
- Remote Desktop Services
- Active Directory

Virtualization technologies:

- Guacamole
- FreeRDP

Platforms:

- Windows
- Linux
- Cygwin
- ChromeOS

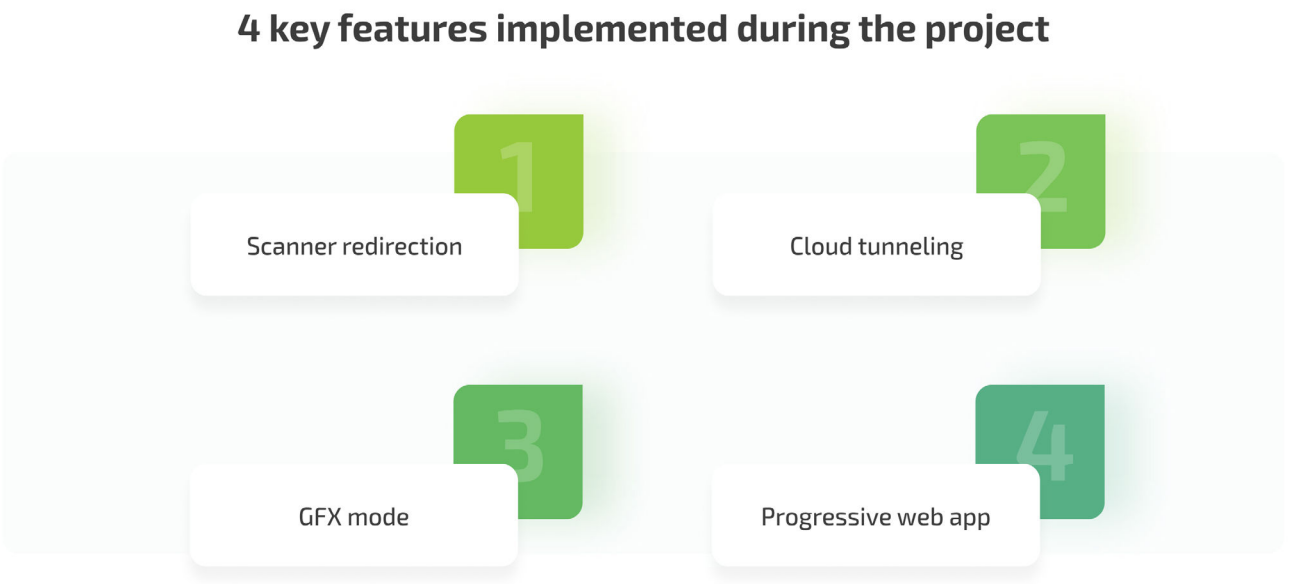
Results

The Apriorit team implemented several features that improved Cameyo's virtual application delivery platform security, performance, and usability. We also introduced new project management tools and approaches to reduce the project's administrative overhead. Additionally, our specialists implemented on-the-go updates to project documentation, which ensured better trackability of all product changes and made it easier to implement further product improvements.

How we did it

Over the years of our cooperation, the Apriorit team performed various tasks aimed at helping our client achieve their goals, from researching competitors and market trends to building new features and testing the security of the VAD platform.

Here are four main features our team implemented for the Cameyo platform:



Let's take a detailed look at each of these features.

1. Scanner redirection

Apriorit developers implemented a scanner redirection feature and added it to the native Cameyo client written in .NET. This feature forwards a real scanner to a user's RDP session, enabling a user who runs an application on the remote server to work with a selected scanner as if it were a local device.

One of the key challenges for us was to address the limitations in rights that the Cameyo client had because it didn't get installed in the end user's system. To overcome these limitations, within the Cameyo native client, we built a custom scanner client responsible for working with scanners. The workflow of this client looks like this:

1. On the endpoint side, our custom scanner client connects to a remote endpoint to enable the end user to work with remote scanners. This client can perform operations such as forwarding a list of available scanners to the user, confirming the selection of an active scanner, and transmitting supported operating modes.

2. On the server side, we implemented a custom **TWAIN**-compatible provider which also serves as a server for our custom scanner client. Using this provider, we pass scanner-controlling commands from an RDP user session to the scanner client and send scanned data back to the user.

We built a custom TWAIN provider based on an open-source solution written in C, adding support for session isolation to the original solution. As the native Cameyo client already provides an active RDP session for client–server communication, in the TWAIN provider, we used virtual Windows Terminal Server (WTS) channels.

We also improved data caching in our custom scanner client to increase the speed of transitioning scanned images and implemented an extra adapter process to make the client compatible with x86 applications.

Finally, since the native client and our scanner client are written in different languages, we needed to implement a cross-language RPC. To do this, we used **Apache Thrift** and created an additional transport layer protocol that uses virtual RDP channels.

2. Cloud tunneling

To enable secure remote connections to the Cameyo server, the Apriorit team implemented a cloud tunneling feature. This feature works as a cloud node connecting the platform's end users to the Cameyo server and offers a secure alternative to setting a direct connection between a user and a server.

Working on this feature, we used two programming languages — Go for a tunneling service and C for Guacamole modifications.

Since Cameyo's browser module uses Guacamole to launch browser sessions, we implemented logic that turns the Guacamole server into a client. This allowed us to establish a secure connection even with the firewall enabled. A traditional configuration won't work in this scenario.

We also implemented a custom tunneling service that acts as an intermediary between the two connecting clients. In this service, we implemented two servers that:

- Establish a connection between our RPD application and a browser
- Perform a handshake to exchange settings

Thanks to this feature, users can securely operate sessions on on-premises servers with no need to use inbound firewall ports or VPN solutions.

3. GFX mode

The web client of the Cameyo platform relies on a Guacamole module. While this module ensures platform availability for different web browsers and operating systems, the version of Guacamole used in the Cameyo web client was rather slow. This harmed the platform's performance when end users worked with dynamic graphical content.

To address this challenge, the Apriorit team implemented GFX mode, which improves the performance of Cameyo's web client when it comes to graphics applications.

We cooperated closely with Cameyo's internal team when researching possible solutions and implementing GFX mode. Cameyo wanted their web client to be able to accurately display video content in high quality, with at least 30 frames per second (FPS) and no artifacts, thus enabling end users to comfortably work with modern video editing applications.

To make this happen, we wanted to leverage available RDP features supporting HD screen transfer codecs such as avc444 or avc420. However, the Guacamole protocol didn't allow us to achieve the results we wanted.

Looking for a possible solution, we evaluated several options and settled for a simple yet seemingly efficient idea — to redirect the stream of encoded graphical data from FreeRDP to a browser and have it decoded there.

To do this, we needed to extend the Guacamole protocol, enabling it to transfer content encoded with the H.264 codec. On the browser side, we decided to use WebCodecs — a built-in Chrome API that allows for encoding and decoding audio and video data.

We started with researching possible ways for getting data from **FreeRDP** and looking for codecs with the least possible CPU/GPU load. After a series of experiments with different codecs, we settled on libx264 — a CPU-based H.264 encoder that allowed us to work with HD video data without spending too much CPU time on it.

4. Progressive web app

Another feature we implemented was a progressive web app (PWA) solution for ChromeOS. This feature allows end users on ChromeOS to download the Cameyo web application to their device for easier and faster launching of remote applications.

To introduce this feature, we wrote a JavaScript script that solves two tasks:

1. Implements a mechanism for accessing the portal by opening the desktop application
2. Caches the content to speed up access to the portal website

As a result, end users can conveniently work with the Cameyo web application, even without needing to open their web browser.

Challenges and solutions

When working on this project, the Apriorit team successfully addressed several challenges:

Clarifying objectives — This project was rich in experimentation and adjustments, as the client had many ideas on how to improve their product. To ensure that the development process was easy to plan and evaluate, we helped our client turn these ideas into clear, measurable specifications. In particular, we hosted retrospective meetings with our client where we evaluated our progress and clarified objectives for the feature under work.

Ongoing maintenance of project documentation — We needed to write and update a lot of technical documentation in parallel with developing new product functionality. By following Apriorit's internal standard procedures, we were able to keep all project documentation up to date, making it easier to track all changes made to the product and maintain high quality in our end results.

Complex support procedures — Due to the platform's high customizability, it was challenging for us to recreate and fix bugs detected in a particular customized environment. To improve the efficiency of our support services, we designed a standard questionnaire to be filled out by Cameyo specialists when they receive a support request from end users. This questionnaire allowed us to organize information needed for debugging and increase both the speed and efficiency of support.

Project management overhead — Initially, all tasks for the Aprioit team were assigned in Trello, which made the project management process time-consuming and unclear. To address these issues, we moved all project-related activities to Jira. After configuring key processes, building transparent roadmaps, and establishing an iterative project workflow, we were able to make the development process even more efficient and predictable for both parties.

The impact

Working with Apriorit enabled Cameyo to provide their end users with new features and improve product performance, addressing the needs of both existing and potential end users. The robust development expertise of Apriorit specialists helped our team effectively tackle challenges of varying complexities, from delivering a simple PWA solution to implementing complex GFX mode functionality.

Following our time-proven internal development standards, we established mature development processes for the project, efficiently cooperated with the client's in-house development team, and reduced the project's administrative overhead.

Currently, we are working on new product functionalities and providing support and maintenance for already implemented features of the Cameyo platform.

Want to improve the performance and security of your legacy product?
Trust Apriorit developers to bring your product to new heights.