



Case study

Cross-Platform Data Backup Solution Development: Windows, Android, macOS, iOS

Our client is a US-based provider of data backup and recovery solutions for mobile devices and computers. They initially developed software for Windows but then decided to make it cross-platform and develop versions for macOS, iOS, and Android.

Since Apriorit was directly involved in creating and supporting the initial Windows solution, our client chose us to further develop and improve the product. As a result, we built a cross-platform data backup solution from scratch and significantly enhanced the existing Windows version of the software.

The client

Our client's company provides data backup and recovery solutions for smartphones, tablets, and computers. Their philosophy is to keep products highly efficient, secure, and intuitive, ensuring that even users with little experience interacting with modern devices can successfully protect their personal data from accidental loss.

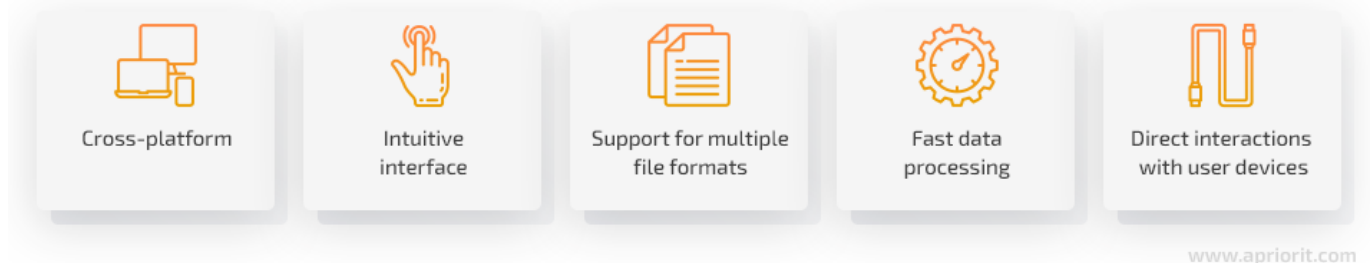
The challenge

Marketing studies show that a large part of our client's target audience uses devices that run macOS. Additionally, smartphones are continuously increasing in popularity and replacing

computers and laptops for some users. These devices are especially popular among regular computer users because of their intuitive interfaces.

This is why our client decided to focus on implementing a data backup solution that would support three more platforms — macOS, iOS, and Android — in addition to Windows. The major requirements of this solution were:

Client's requirements for data backup software:

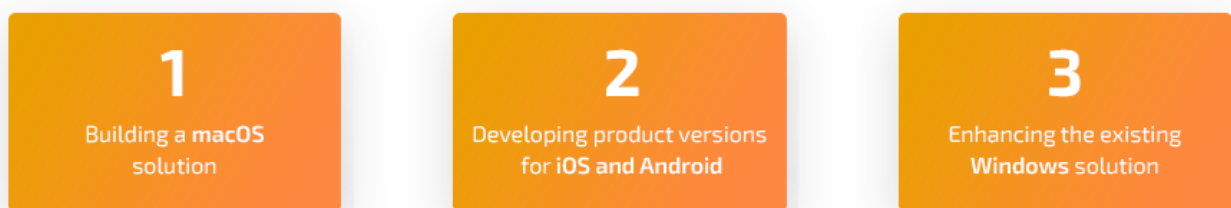


The Apriorit team developed and supported an initial Windows solution that allowed for backing up valuable user data on external storage. When the project entered a new development stage, our client chose us as a trusted and proven contractor. Having worked with Apriorit for almost 18 years, our client was sure of our expertise and professionalism going into the project.

Our solution

Before beginning work, our specialists conducted in-depth research to estimate possible risks and establish clear specifications for the products to be developed. We then created macOS, iOS, and Android versions of our clients' cross-platform backup software and upgraded the existing software for Windows according to our client's requirements.

Three major parts of the project



We also used Firebase analytics and Firebase Crashlytics to gather logs and ensure better analytics.

The dedicated Apriorit team for this project consists of developers, researchers, QA specialists, a development manager, and a project manager.

The result

The Apriorit development team successfully created product versions for iOS, macOS, and Android and enhanced the current Windows solution.

We released the solutions for each of the three new platforms as soon as they were ready. Thus, we managed to speed up the time to market, help our client receive revenue at early project stages, and gather valuable feedback to improve those solutions that were still under development.

We continue working with our client and keep supporting the developed solutions, adding new features, improving existing ones, and preparing new releases for all four supported operating systems.

General requirements

Our customer approached us with an idea to enhance the current version of their product for Windows and build three more solutions for other operating systems. Their basic set of requirements included the following:

- **Provide a cross-platform USB solution.** While the applications for iOS, Android, macOS, and Windows would be different, the hardware component (a USB stick) provided along with the software solution had to work with any of these operating systems. To make this happen, we needed to choose the file system wisely.
- **Design an easy-to-use interface.** The user interface (UI) of the solution had to be developed in a way that made it easy to use for the target audience. Besides, it had to look attractive and be friendly to users with special needs, which included implementing large buttons, clear texts, and contrasting colors.
- **Support different file formats and large amounts of data.** The solution needed to have minimal limitations on the types and amounts of data it can work with.
- **Process data quickly.** The speed of the applications was essential, especially when interacting with smartphones and tablets. End users can't spend lots of time waiting for data to be backed up and processed. Moreover, the longer a backup takes, the greater the risk of its being interrupted. Therefore, devices needed to save data at a speed of no less than 1 GB per minute.
- **Direct interactions with devices.** The hardware part of the solution had to be designed in a way that allowed it to directly interact with iOS and Android devices. The biggest challenge was to make solutions for iOS devices interact with custom hardware using custom protocols.

Our approach

Having an initial vision of the new solution and general requirements, we proposed the following approach:

Our approach



First, our development team conducted **initial research**, estimating the complexity of the solution and exploring possible pitfalls and available features. Since initially our client had only a general idea and a basic set of requirements, this research was necessary to specify some critical details.

Then our business analysts prepared a **specification**, outlining the scope of work, the UI of the final solution, and general functional and non-functional requirements. Once that was done, the specification was approved by the customer to make sure all stakeholders were on the same page.

Next, the team started working on **demos**. This allowed us to get early feedback from the client and make changes to the developed functionality.

Once an application for a particular operating system was finished, we **released** it. Thus, we were able to gather feedback from end users and incorporate it when developing new versions of the solution or making improvements to released versions. This also allowed our customer to gain revenue at the early project stages.

Scope of work

Since project development included creating three different solutions for different operating systems, we broke our work process into five stages. The duration of each stage depended on the complexity of the tasks; on average, each took two to four months. We released all the final products on time.

Building a data recovery and backup solution



Stage 1. Develop the Android version

Building solutions for Android and iOS was the most challenging part for our team because these applications had to be able to back up data to external storage without using Windows or macOS as a mediator. The apps had to be intuitive and allow end users to achieve the desired results in a few clicks. They also had to work without any additional user adjustments like installing third-party software or altering the device settings.

We decided to start with building the Android application because Android users represent a large share of the target market. Also, the Android app allowed us to validate our solution for working with smartphones and tablets. The interaction between devices and USB sticks implemented during this phase helped us improve the initial development strategy and understand how the solution should work for end users.

We started working on the Android solution by developing a proof of concept (PoC) that included the following features:

- **Detecting data to be backed up.** As we couldn't directly ask users to select the path to data in the file system, the solution had to be able to communicate with users in understandable terms (using images, videos, etc.) and detect data to be protected.
- **Receiving permission for accessing the required folders.** Once data to be backed up was located, the solution had to be able to copy it. But before performing any data manipulations, it needed to request and receive specific permissions from the end user. This process had to be very transparent to ensure that end users granted permissions.
- **Detecting the Device ID of the flash drive.** Our client's solution consists of two parts: software and hardware. The software part had to be designed in a way that allows it to work only with valid licensed hardware.
- **Creating data storage on the USB stick and copying data to it.** Backing up data is a crucial part of the software's work. Files have to be placed in a certain location on the USB device according to business rules. These rules are added to a special folder on the USB stick, and data goes in specific subfolders depending on its type. We had to avoid data duplication and arrange information restoration in the most convenient way.

Once we had developed the PoC and it was approved by the customer, we researched third-party obfuscation solutions available for Android applications of this class and suggested using DexProtector. We also implemented licensing and made it possible for the app to run in the background.

Then we designed the final application, finished development, and checked usability. We released the developed solution to Google Play and started receiving feedback from end users.

Stage 2. Develop the macOS version

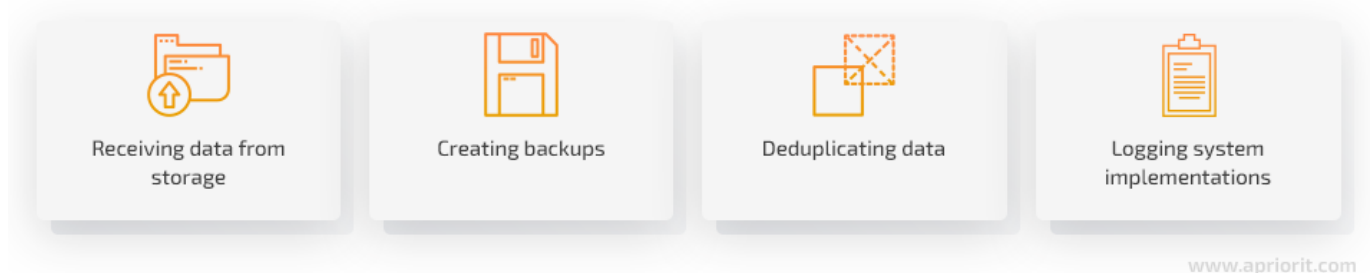
We started building a macOS version of the app before the iOS one so we could gain the necessary experience working with USB technologies for Apple products before moving to more challenging iOS device interactions. Another reason was to give our customer time to analyze how the Android solution was accepted and decide if any changes were needed to the iOS solution.

Our client requested to make the macOS version work and look the same as the Windows version. The most challenging task was to create a similar interface for macOS to facilitate the transition from Windows to Mac (and vice versa) and minimize the number of marketing materials needed. Also, moving from Windows and Android users to a completely new segment of Apple users required additional analysis of the market and user expectations.

With all the requirements elicited, the specifications approved, and the initial simple design created, our team proceeded to development.

Major features to be implemented in the macOS version of the product included:

Features for macOS data backup and recovery solution



- **Receiving backup data from both local and external storage.** The solution needed to not only back up data from macOS directories but also work with both iOS and Android devices connected to it.
- **Creating backups.** Just like the Android and Windows applications, the macOS solution needed to place backed up data in a certain location for storage and restoration. The macOS application also had to provide users with the means of accessing backed up data.
- **Deduplicating data.** Since the solution is meant for everyday use, we had to enable mechanisms that ensure data isn't duplicated in storage and at the same time is safely collected.
- **Implementing a logging system.** To provide the best customer support, we had to build an advanced logging system and an easy mechanism for providing logs. In case of any issues, end users should be able to send logs to support engineers. These logs must contain enough information to help engineers locate and solve the problem.

Stage 3. Develop the iOS version

The list of features for the iOS application had to completely match those of the Android app. However, unlike on Android, iOS interactions between software and hardware can be done only via an MFi license.

The MFi licensing program allows developers to get the hardware components, tools, documentation, technical support, and certification logos needed to create electronic accessories that connect to iPod, iPhone, and iPad.



There were two possible solutions:

- Find a third-party subcontractor that already has an MFi license and hire them to prepare hardware parts.
- Receive our own MFi license from Apple (which is an extremely time-consuming process) and prepare hardware parts ourselves.

Since our client wanted a quick release and had limited time, we selected the first option.

The subcontractor our client chose provided representatives with poor English skills. As there were no other languages both of our teams could use to understand each other, we had to struggle with linguistic barriers.

Another issue was the final result of the subcontractor's work. We expected to receive a fully developed SDK with unit tests ready for integration. Instead, we got an SDK that required a lot of additional work from the Apriorit team to adjust it for the final product.

Luckily, we predicted such risks and warned our client about them when we began working on the iOS app. Therefore, our project manager was ready to adjust the development process accordingly without altering the initial estimates.

After we prepared the hardware solution and its protocol, our team started working on implementing the UI and application logic. Just like at the Android development stage, this

process included detecting data to back up, creating backups and logs, licensing, and allowing data to be backed up when the app is running in the background.

We released the iOS solution to the App Store as soon as it was ready.

Stage 4. Improve the Windows version

Unlike more adaptable Windows products, macOS products have stricter UI requirements. And since our client requested to make the macOS and Windows versions of the app very similar, we had to slightly change the Windows product so it would match the UI of the macOS version.

After analyzing the Android, iOS, and macOS versions of the product along with feedback from end users, it was obvious that the Windows version also required significant changes to better meet users' needs.

We started with creating a roadmap for these improvements, discussed it with our client, and prioritized the most essential updates.

Our team has already started adding new features to the Windows application according to the agreed plan. The work is still in progress and is being done in iterations. After each iteration, we release an enhanced version of the product and review and update the roadmap.

Since the Windows solution has the largest number of users, it requires the most attention. Therefore, our team decided to implement the most valuable features in the Windows product first, then continue improving the other solutions.

Stage 5. Expand the project

It was decided to create a completely new product that's based on the existing one but provides more capabilities or backing up and protecting data. This was our client's preferred strategy since market analysis showed that it was more efficient to build a new solution with various options rather than developing multiple editions of the same product.

This stage of the project has just started.

The biggest challenges

Working on this project and meeting all the requirements presented two main challenges for our team:

1. Designing a unified interface for different operating systems

According to our client's requirements, the interfaces for all product versions had to be very similar to ensure that end users can easily switch from one version to another without

additional stress. This was a challenge because different operating systems have different rules and requirements for applications.

We had to start with developing the macOS solution before proceeding to improving the existing Windows product, as the macOS platform has more restrictions on the application's UI. Then we improved the Windows product accordingly to match the UI of the macOS version.

2. Developing the iOS version of the application

The iOS app had to use a custom protocol to connect to iOS devices with custom hardware (a USB storage device). This hardware needed to meet MFi certification requirements and receive a license.

Our team researched possibilities for interacting with MFi devices, and it appeared to be the largest challenge for Apriorit, as there was no open documentation on this process. Information can only be received directly from Apple while applying for an MFi license or from third-party companies that already have this expertise.

Therefore, we had to work with a third-party contractor, which was difficult due to language barriers. Another issue was the incomplete SDK provided by the subcontractor. Luckily, we predicted this drawback and were ready to fix the SDK ourselves.

Thanks to our rich experience working with Apple review and App Store Connect, our team managed to make the iOS solution pass through the review and released it on time.

The impact

Our client's data backup and recovery solution has already been released and has been well accepted by end users. Our client continues to work with the Apriorit team on both improving the existing product and creating the new one.

Our team plans to add some new features to our client's existing solutions to make them even more helpful and convenient. Since we released all new versions of these data backup and recovery software solutions, the total number of users has doubled and keeps growing.

Apriorit teams provide high-quality mobile app, web product, kernel and driver, and other [development services](#). Contact us to start building your dream project right away.