



# Transition to an Integrated Financial Desktop: Key Decisions

A paper published in partnership between DataArt and OpenFin

# Transition to an Integrated Financial Desktop: Key Decisions

This paper identifies some key considerations and decisions that product owners and application stakeholders need to make in the process of transition to the integrated desktop user experience where multiple in-house and third-party vendor applications can interoperate seamlessly. DataArt offers a simple, proven methodology for assessing the initial scope of work required to effect this transition, as well as some high level recommendations based on our practical experience with our customers.

## Introduction

An integrated financial desktop, where multiple applications can be effortlessly stitched together to support complex workflows, is an appealing vision for the industry. By leveraging the ability to pass context and data between apps behind the scenes, desktop integration platforms enable an intuitive, seamless “mobile app-like” user experience. It is a welcome change from the legacy world of disconnected desktop systems, which force professionals to act as the “human integration layer”, performing tedious manual copying and pasting information between applications (provided by different vendors and/or different teams at different times) throughout their day.

Inspired by this vision, technology teams in many asset management firms, banks, and software product companies are drafting roadmaps to enable better interoperability and user experience for their products and applications. However, performing this transition efficiently, with minimum investment and fast time to market is a challenge for many product managers, as well as for technical stakeholders.

Some of the questions product or technical teams are often faced with at the beginning of their journey include:

- Where do I start?
- How much change will my product require?
- What are the key considerations for the long-term viability of our future state implementation?
- How do I choose the JavaScript framework or library that is the best fit for my needs?
- What implementation timelines should I expect?

Organizations need a model, or a high level framework to help them make key product and technology decisions, correctly identify the scope of changes, and be able to plan and execute a feasible transformation roadmap.

In this brief paper, we offer a proven framework that can help product and application teams craft a path forward on their journey to the integrated desktop. When we work with our customers to help them achieve the benefits that integrated desktop platforms have to offer, we take them through the following steps:

**01**

Align on the desired business outcomes to be achieved by the transition

**02**

Understand your existing application types and their specific advantages and limitations

**03**

Identify target execution environments and distribution channels in scope for your application or product (web, desktop, mobile, etc.)

**04**

Make key decisions and perform the initial scope of work assessment

Once these steps are completed and socialized with all major stakeholder groups, you will be in a position to craft your product roadmap, technical architecture design and other standard technology implementation steps.

## 1. Align on Goals and Objectives for Transitioning to Integrated Desktops

If you are reading this whitepaper, you are probably already well aware of the benefits that transitioning to an integrated desktop can bring to your organization, and of the pain points it is designed to alleviate.

Pain Points	Integrated Desktop Benefits
Lack of integration between multiple applications on the desktop forces users to transfer data manually between them, resulting in low productivity and substandard user experience	Interoperability via front-end integration does away with manual repetitive tasks and delivers a seamless data flow between applications, drastically improving productivity and user experience
Modernizing legacy applications (re-implementing them as modern web-based solutions) is slow and costly	Desktop integration platforms accelerate the migration by offering a desktop environment that enables stitching together legacy applications with the emerging new codebase, while ...

## Pain Points

## Integrated Desktop Benefits

*Continued...*

at the same time delivering a native desktop experience users are familiar with (e.g. window management). This way, new functionality can be delivered incrementally, without compromising the stability of the existing applications.

Lack of application interoperability makes it difficult to create seamless workflows and creates application and configuration management overhead for users

Desktop integration platforms provide unified configuration and launcher capabilities (via application manifest), as well as the ability to compose custom workflows by linking applications, components and widgets (the “composable desktop” concept)

Legacy 3rd party desktop applications must go through costly and lengthy deployment/onboarding process in enterprise environments

Desktop-side integration provides for a native desktop experience without the onboarding and deployment hurdles traditionally faced by legacy desktop applications – in other words, it combines the ease of deployment of web apps with the native experience of installed apps. This enables frequent updates and enhancements and shortens time to market for new functionality.

Pain Points	Integrated Desktop Benefits
Slow rate of rollouts of releases, new features, slow time to market (legacy applications)	Conversion to modern web application patterns in conjunction with desktop integration indirectly enforces modularity and micro-frontend architecture, enabling multiple parts (or modules) to be iterated on simultaneously and thus accelerating time to market.
3rd party software products require costly customization and integration efforts to enable complex user flows on the financial desktop	Interoperability via a common standard (FDC3) provides for front-end enabled data flow between applications from different vendors (and teams) via common data definitions and an application message bus

Addressing these pain points and realizing these benefits can provide the means for companies to achieve a variety of business outcomes, such as:

- Expanding your customer base by making your company data and functionality available on financial desktops
- Exposing and monetizing your enterprise data quickly via easy to consume front end applications
- Evolving your product more quickly, protecting revenue and fostering customer loyalty through new value-added functionality and seamless user experience
- Achieving faster time-to-market and better return on development budgets for enterprise in-house development organizations

- Enabling faster product innovation and better business agility by re-packaging existing monolithic products into more granular, focused offerings
- Making it easier to integrate more automation, augment human capabilities with ML, etc. by re-imagining monolithic systems as modular, composable services, thus driving further value to end users

Whatever your organization's specific business goals and objectives are, it is critical that, at the beginning of your transformation journey, you reach clear alignment between various groups of stakeholders (both business and technology) on the desired benefits and the business outcomes they will drive for your customers and your business.

This alignment and agreed upon set of objectives will help guide and inform your technology transformation by providing a clear framework for making important decisions about your product or applications.

## 2. Understand Existing Product/Application Technology Advantages and Limitations

If you are building a greenfield product, you have the freedom to choose any technology that is supported by your in-house skills and can deliver on the business requirements. For existing software products that need to be transitioned to the integrated desktop, you are looking at the following scenarios:

**Web Applications** are typically implemented with HTML or JavaScript technology. They are run in web browsers and typically can be run on desktops or on mobile devices, with some dependency on the complexity of their functionality (some highly complex UIs may not translate perfectly to mobile with a great user experience, based on controls and screen sizes).

Since desktops integration platforms fully support running web based applications on a desktop, existing web applications have a great advantage: they may require little to no change to be enabled on the integrated desktop. If your web application is capable of being executed in the Chromium browser, it can be ported to the financial desktop with very little work.

**Desktop Applications** are typically written in Java or .NET and cannot run in a web browser. Desktop integration platforms support various integration adapters for applications implemented with these technologies. However, they still need to be installed on desktops in a standard way, and distribution via web (the way consumers access web or mobile applications) is not available to them. Most enterprises have comprehensive security and technical due diligence onboarding processes in place for desktop apps. This deployment overhead makes it hard for product companies targeting enterprise desktops to quickly deliver new versions of existing products or provide even trial access for new applications.

### 3. Define Target Execution Environments for Your Product/Application

This step is crucial to understanding the future scope of work and to making some key decisions with regards to the future state product. Most products strive to support both web and desktop environments, and to provide responsive application behavior on mobile devices. If you need to support more than one distribution channel (or target execution environment) for your application, you have to make decisions regarding user experience for each channel. Desktop integration platforms enable application interoperability, as well as layout management capabilities and streamlined deployment. Desktop platforms also allow for reuse of the same application views and components in both a web browser and on mobile (since most of



them are built on top of the Chromium browser), so you do not have to re-implement the same application in .NET/Java or JavaScript. However, they do not provide an execution container for mobile devices, so certain desktop UX features (e.g. layout management, window docking/stacking) will not be available on mobile. Therefore, if you want to make your application available via multiple channels, your technical team has to come up with a solution that enables reuse of the existing application components and business logic, and implements optimal channel-specific UX.

## **4. Conduct an Initial Assessment to Identify a Possible Scope of Work and Make Key Decisions**

Now that you have completed steps 1 through 3, you are in the position to perform the initial assessment of the future scope of work through answering the key questions below. We are also including some recommendations based on our experience and best practices.

### **Who are your target users?**

Integrated desktops have proven themselves to be extremely beneficial for certain specific industries and user roles that have to rely on multiple disparate apps and are operating on crowded desktops. If your target user or industry is not on this list, we would recommend returning to the earlier step to re-validate the benefits and desired business outcomes. You want to make sure that the benefits you envision are addressing known pain points for your target users.

*Industry sectors and user roles who benefit the most from integrated desktops:*

- Banks
- Brokers
- Asset managers
- Hedge funds
- Wealth managers
- Insurers
- Financial services vendors
- Trading
- Sales
- Research and analytics
- Portfolio management
- Compliance
- Risk management
- Operations
- Support and Call Centers

## **Do you need integration with third-party systems / applications?**

Integrated desktop platforms are implemented on the basis of industry standard protocols like FDC3, which was founded by OpenFin, to enable interoperability. However, certain vendors and products may not have FDC3 integration on their roadmap. If your application relies heavily on such integration, you need to define specific scope of integration, i.e. message types and functionality that need to be supported in near- and long-term (data context sharing, support for messages and alerts).

If your target customers do not use your application as part of a multi-step workflow where it needs to be integrated with other systems, you can most likely exclude interoperability/integration from the future scope of changes.

## **How will users discover and launch your application?**

Desktop integration platforms provide a centralized application menu. Be aware that your scope of work should include registering your application in the Application Catalog and providing required metadata, such as application description, etc. Another thing to keep in mind is that your application should expose its functionality via FDC3 API to other applications and be automatically discoverable. For instance, if you decide to give your Charts, News, or Quotes applications the ability to start automatically when a user needs to display news or view quotes, without the need to launch them manually via a menu, this work should also be included in the scope.

## **How would your users be authenticated and authorized?**

Single Sign-On (SSO), i.e. centralized, one-time user authentication is the ideal user experience to shoot for. If your application contains its own authentication mechanism, consider including SSO authentication in your roadmap. Most desktop integration platforms provide services or mechanisms for implementing SSO. If you want SSO capabilities, but also want to retain the ability to provide your product as a standalone web application to be used in the browser, plan for supporting potentially different, or unified authentication mechanisms.

## **What desktop-native UX elements do you want to support?**

If you are transitioning from a traditional web application to desktop, consider the following desktop-specific functionality for your scope of work:

**Multi-monitor and multi-window support:** Most browser-based applications are designed to run as a single instance in the web browser. On the desktop, users can typically launch multiple windows with the same view. Make sure that your application can support this from the user experience, functional and performance perspectives. If launching another instance of the same view consumes too much memory or CPU resources, or it takes too long to start, user experience and ultimately even adoption can suffer. This could mean that you need to plan some refactoring for the purposes of improving performance, or consider breaking existing views down into several smaller widgets.

**Stacking, docking and resizing:** Integrated desktops give users a great degree of flexibility in creating new application views and workspaces for their convenience by resizing windows and stacking and docking them together. If your existing application does not behave responsively – in other words, if it does not adapt well to re-sizing, shows scroll bars, or even hides content – then you may need to plan re-working your application to enable adequate responsive behavior.

## **Does your application have custom pop-ups and notifications?**

If not, your users may benefit from leveraging the centralized notification, messaging and alerting functionality provided by the integrated desktop platform like the one recently launched by OpenFin. If it does, you may consider retiring your custom implementation of alerts and notifications and integrating with the [OpenFin's Notification Center](#).

## **Does your application have a toolbar and other common functionality (User Profile, Centralized menu, Settings, etc.)?**

If the answer is yes, then when your application is run within a desktop window, it may overlap with menus and functionality provided by other applications already installed on the desktop. Additionally, duplicating toolbars and other common functionality in every window is not aligned to UX best practices. In this case, arriving at optimal application navigation usually requires some thoughtful refactoring to align with the common desktop behavior.

## **Look and Feel**

The bar of expectations for user experience has been elevated substantially in recent years. Since UX is a crucial factor in competitiveness in adoption, desktop platforms go to great lengths to make integrated applications attractive and easy to use from the UI perspective. Your future state application's look and feel will need to match these high user expectations and be in line with the current UX trends (for instance, preference for dark styles and themes on desktops). If your existing look and feel is significantly different from other applications on user desktops, consider including the work to bridge those UI gaps in the scope of transition.

## **Manage transition risk by moving incrementally**

As you are working through the steps and questions outlined above and making those important product decisions, please keep in mind that the easiest and fastest way to achieve your business goals is to move in controlled, incremental steps. In considering your alternatives, our recommendation is to keep the scope of transformation to a minimum, and to favor decisions that enable a quick and light-weight transition. Many customers are surprised how

quickly they can achieve Proof Of Value implementations, and how inexpensive it may be to get important quick wins. This kind of rapid progress is possible with the help of a specialized partner who can also provide expertise and consultancy on the technology, software delivery and implementation aspects. Partnering with a specialist firm can help you clarify and streamline your decision making process, boost your in-house expertise, and accelerate time-to-market.

## **Whenever possible, go with the Lift & Shift transition approach**

Lightweight changes that we are calling Lift & Shift enable the fastest time to value:

- Existing web applications require minimum changes and the least amount of effort to integrate. At the same time, certain changes related to multi-monitor support, control resizing, and dark theme availability can be mandatory.
- If no interoperability is required (i.e. data context sharing), then your existing application can be integrated with OpenFin with zero changes required. Any JavaScript framework (e.g. Angular, React, Vue, Polymer, Svelte) that works in Chrome and Chromium browsers will work in OpenFin (which is built on Chromium).
- Desktop applications are typically not a fit for lift & shift strategy, and should be re-engineered incrementally as interoperable web applications.

## Why OpenFin

OpenFin is the financial industry's operating system, enabling rapid and secure deployment, native experience and desktop interoperability through its leading integrated desktop platform. OpenFin currently powers over 250,000 desktops for 3,500 firms in more than 60 countries around the globe, with over 2,400 applications in the rapidly growing ecosystem.

Clients work with OpenFin in different ways: some to speed up their transition to web technology, others to make their apps more intuitive and easier to use. In all cases, OpenFin streamlines processes and saves time and resources both internally and on the client side.

OpenFin now serves as an established leader in the field working closely with buy-side and sell-side firms, asset managers, hedge funds, banks and technology vendors to help them realise their digital transformation and desktop technology strategies.

OpenFin has helped clients achieve an unprecedented level of digital transformation. [Barclays leveraged OpenFin](#) to introduce a new RESTful microservices framework, create a modularized development pipeline and deliver the final BARX One Connect App with speed and efficiency. This enabled all Barclays equity traders to more easily access real-time and historical analytics.

[Liquidnet leveraged OpenFin](#) to integrate several different acquisitions with one another, including a research platform and two analytics platforms. Using OpenFin's platform API, the firm was able to keep the independent portals from each acquisition and bring that disparate content into a single Liquidnet window frame, retaining rich functionality while giving users access to a single intuitive platform.

These are just two of many success stories. OpenFin's roster of users speaks for itself. OpenFin now counts 20 of the top 25 global banks, largest asset managers and hedge funds in the world as clients. These important strategic financial institutions are complemented by a powerful ecosystem of financial services vendors who choose to build and deploy their products on OpenFin, including, Tier1, Cloud9, IPC, Factset, BidFX, Qontigo, Trumid, Flextrade, Luminex among other firms.

## Contact OpenFin

Email

**info@openfin.co**

Offices

### **New York City**

80 Broad, 35th Floor  
New York 10004

### **London**

14 New Street  
London EC2M 4HE

## Contact DataArt

**openfin@dataart.com**

Phone

**New York +1 212 378 4108**

**London +44 20 7099 9464**