



Delivering TeX Functionality: Integrations, Built-In Apps, & Linking

HqO's platform architecture is structured to provide, support, and configure utility across the functional verticals that comprise tenant experience (e.g., access, ordering, booking, transportation, etc). Functional verticals may have multiple modes of implementation.

THESE INCLUDE:

- Deep Integrations to external service providers.
- HqO Built-In applications.
- Web Linking either deeply or as lightweight solutions.

Deep Integrations

Deep integrations are accomplished via direct partnership and development between HqO and 3rd party service providers via APIs (Application Programming Interfaces) or SDKs (Software Development Kits).

API APPROACH

In this form of integration, a service provider presents an API that enables access to their essential functionality to which HqO will integrate. These APIs define how HqO and 3rd party software applications communicate and engage, request services, and access and transfer data.

- The user experience is entirely within the HqO app, which is normalized across all service providers for a given functional vertical. Integration to their APIs is performed within our back-end server layer.
- API integrations not only provide superior user experience and optionality for buildings and portfolios, but they also enable the richest collection of tenant experience data for analytics.

- Current development time varies by functional vertical and the availability, quality, and alignment of the provider's APIs.
- This is our preferred approach and we leverage APIs whenever possible.
- We envision and are helping to drive a future where the APIs are standardized within many of the functional verticals, thus facilitating rapid onboarding of service provider options for building and asset managers.

SDK APPROACH

Variation of deep integration whereby a service provider delivers a software library or SDK to HqO and HqO integrates this SDK directly into the HqO app. The SDK is responsible for communicating with the provider's server layer as needed to perform the functionality.

- HqO endeavors to retain a normalized user experience across service providers to the extent possible, depending on the SDK and the functionality in question.
- We avoid these types of integrations when we can as they increase the size of the app.
- However, there are verticals in which this is necessary due to interactions with device or external hardware. The most common example is digital access.

Delivering TeX Functionality: Integrations, Built-In Apps, & Linking

HqO Built-in Apps

- HqO Built-In applications are similar to the API approach described above in that we optimize the user experience, normalize it to other options within the functional vertical, and collect rich tenant experience data for analytics.
- The difference is that no 3rd party service provider is involved. The functionality is implemented entirely by HqO.

Web Linking

Web linking is essentially a navigation from the HqO app out to a 3rd party asset. This asset may be a web page displayed within a web-view (i.e., essentially a mini-browser within the HqO app) or another app entirely on the user's mobile device.

- Not the HqO preferred approach because the user experience is entirely under 3rd party control, it is difficult to obtain meaningful analytics data once the user is within that external experience, and additional user sign-on may be required depending on the functionality in question.
- Web linking is very easy to implement requiring little and in some cases no development. Thus be a fast path to functionality while a deeper integration is being performed.

- There may be exception cases where a service provider does achieve a sufficiently rich user experience in a vertical that does not require an additional sign-on, but this would and should be tightly scrutinized if deeper integration options are viable.

DEEP LINKING

- Deep linking is a variant of web linking in which the link from the HqO app sends the user to a very specific point in the 3rd party web page or mobile app.
- This method at least provides a reduced friction experience for the user, although it may require the 3rd party to provide specific navigation targets.

LIGHTWEIGHT WEB LINKS

- Lightweight links are the easiest to implement, but the least robust form of web linking.
- Essentially simple navigational links to an external website or app without any specific internal target therein. Often no additional support is required from the 3rd party.
- Lightweight web links may be used for prototyping, though are not recommended for rich user experiences.
- We do not generally consider them to qualify as an "integration" given their purely navigational nature.