S-Docs
Simple. Smart. Secure.

TECHNICAL WHITEPAPER

# A Guide to Evaluating Salesforce AppExchange Apps

There are thousands of apps available today on the Salesforce AppExchange. This paper will help you assess security, speed and scalability differences between different app architectures. It will help you identify which apps share your Salesforce data as they interoperate with Salesforce, and provide you with some key questions to ask vendors during your evaluation process.

## Overview

Since 2005, the AppExchange has provided a store for Salesforce customers to find, install, use and share experiences on thousands of apps that can add real value to your business. While the benefits of an app are touted, the risks associated with implementation are often omitted and left unclear. Most listings on the AppExchange give little technical detail on the app construct or how it interoperates with the Salesforce platform and your customer data. These details are critical to understanding performance, security, adoption, compatibility and total cost.

“App listings use phrases like “native integration” or “without ever leaving Salesforce,” which can mislead you on how these apps use and share your Salesforce data.”

Vendors are not making your research efforts any easier. They are constantly updating their listings, adding terms like “native,” “on-platform,” or “intelligent” to convey a sense of security, ease of use, or leveraging the latest technologies. The vagueness of some technical terms lends them to being exploited. For example, some AppExchange listings use phrases like “native integration” or “without ever leaving Salesforce,” which can mislead you to believe your Salesforce data is not being shared. In actuality, the latter description is likely referring to the user experience of “not leaving” the Salesforce user interface, while in the meantime your data is being shared with their external servers for processing.

This whitepaper will examine how many apps are constructed, the tradeoffs of differing architectures, and key questions to ask vendors during your evaluation. It is intended as a guide to help you understand the benefits and risks of various app architectures.

## First Mover Disadvantage

Today, we universally assume speed to market as a benefit, where laggards continue to be at a disadvantage. This is not the case with Salesforce apps.

In 2005, Salesforce introduced the AppExchange as a way to broaden and expand their capabilities and interconnectivity. At that time Salesforce did not have the broad feature set they boast today. The Force.com platform would not be available for another four years. This meant that apps originally listed on the AppExchange during the early years were primarily used to either link Salesforce to legacy systems or fill a particular feature which Salesforce lacked. Those apps used Salesforce APIs to communicate with Salesforce, and some continue to use antiquated  protocols like SOAP more than a decade later.

In 2009, Force.com was introduced as Salesforce’s Platform as a Service (PaaS) and this dramatically changed the ecosystem. Developers could now build a native app, meaning an app built directly on the Salesforce cloud. For existing vendors who already had a product built with existing customers and many year's worth on their existing stack, the cost to convert to Force.com was not advisable. Most of these existing vendors continued on their current stack. They quickly found they were using yesterday's interface and technology as Salesforce continued to rapidly evolve. Despite this, many of these first movers continued to thrive based on marketing and brand, not technology.

"While native apps are not a panacea, you should look to native apps first to meet your requirements before you investigate non-native apps."

## Summary of Native Benefits and Disadvantages

For solutions entering the market after 2009, native meant a new way to deliver functionality within Salesforce by leveraging the customers' existing investment in Salesforce. A native solution provides these key benefits:

- Security: Data stays within Salesforce
- Speed: No latency due to network traffic between clouds
- Scalability: Platform elasticity matches Salesforce
- Ease of use: Uses Salesforce UX/UI and existing admin tools
- Cost: Infrastructure already paid for with Salesforce licenses
- Uptime/Reliability: Availability matches Salesforce
- Data Residency: Meets stringent citizen data laws
- Integration: Relieves API usage, simplifies authentication, reporting integration
- Future Proof: New Salesforce features must work with existing stack

While building on the Salesforce platform certainly has its advantages, it does not work for all cases. There are use cases when off-platform solutions are often more suitable:

- Working with greater than 30MB or containing video
- App functionality requires file system, or OS-level access
- Apps that require streaming data
- Heavy use of external (non-Salesforce) data
- Computation intensive applications
- Portability: Interfacing App with other CRM systems

While native apps are not a panacea, you should look to native apps first to meet your requirements before you investigate non-native apps.

## Native Salesforce App Defined

Today, the term "on Salesforce" can be confusing because it no longer refers to a unified platform but is now used by acquired technologies that operate under the Salesforce corporate umbrella. Products like Sales Cloud, Service Cloud, Force.com and Communities all share the same infrastructure or stack. This is referred to as the core stack. That core stack contains all of the components needed to run an instance of Salesforce including the database, application servers, load balancers, cache servers, and search servers. This is what we should consider "on platform." However, Salesforce acquired technologies like ExactTarget and Heroku that are not on the core stack. Nonetheless, these are often described as "on Salesforce," which can be misleading. Some technologies like Salesforce CPQ are marketed as native, but still use off-platform services for processing calculations or document generation.

"A native app is one that resides entirely within the confines of an existing Salesforce core stack."

A native Salesforce app is one that resides entirely within the confines of the existing Salesforce core stack. It is built in the same Salesforce development environment and uses the same commercially available development tools that are provided to all Salesforce clients for customizing their own instances. It does not rely on any external systems, third-party web services or browser plug-ins even if those services are owned by Salesforce. All web pages must be served from the Force.com domain and all code must be compiled and executed on Salesforce servers for an app to be considered native.

A native app consists of a code base called a "managed package" that is made available on the Salesforce AppExchange and installs into your existing Salesforce instance. Once installed, a native app can be considered part of Salesforce in the same way your own customizations become part of Salesforce or how Chatter is part of Salesforce.

S-Docs is 100% native to Salesforce.

## Non-Native Salesforce App Defined

A non-native Salesforce app does not live 100% within the Salesforce platform. Elements of the application may be within Salesforce and could be considered a "hybrid" app; however, the app itself may be hosted elsewhere (such as Amazon) and data may be transferred between platforms and stored on an outside, third party server. Be mindful that some vendors state that "your data is never stored externally." This claim is made because your data is processed in-memory and then flushed from that memory cache. This does not change the fact that your customer data has changed possession.

## Details of Native App Benefits

### EASE OF USE

- Since native apps are exclusively written for Salesforce on Salesforce, they better leverage the existing Salesforce user experience. For example, S-Docs uses the same styles, picklists, grids, icons, labels and taxonomy as your existing Salesforce UI. This makes S-Docs more intuitive to Salesforce users rather than requiring them to use a generic interface that was designed for a variety of CRM vendors.

### SECURITY

- All client data stays within Salesforce (unless you download, export or email it). It is not sent nor stored on external servers nor does data enter from external servers, which may be the case with a non-native application.

- A native Salesforce application conforms with the same security settings and sharing rules that you have already created within Salesforce. With a non-native app, these rules and settings may not be applied, and as a result, data could be vulnerable. Hybrid apps may well be secure, however they may not necessarily be using Salesforce security and sharing settings.

- Integrations with a native app are more secure. For example, API interactions with any native app services require authentication with Salesforce. This ensures the highest security and minimizes the need for specially built integrations. For example, a single-sign-on (SSO) integration requires no additional consideration to work with a native app.

- All apps on the Salesforce AppExchange must pass the Force.com ISV Security Review prior to being published on the AppExchange. https://developer.salesforce.com/page/Security_Review

### DATA RESIDENCY

- Since all data continues to be exclusively stored on existing Salesforce servers, native apps inherently meet the strictest government data residency requirements.

### PERFORMANCE / SCALABILITY

- In general, native apps will exhibit better execution performance because all processing occurs on the same platform. Performance is accelerated by the elimination of having multiple calls to send and receive data from external services.

- All native apps must adhere to Salesforce's execution and governor limits. These limits apply to code created on the Salesforce platform and help enforce rules that prevent runaway processes, use of unsupported methods, or executions that monopolize resources. These governors ensure that S-Docs code is efficient and follows best practices.https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm

- Native apps do not count against your Salesforce org's daily API limits. By contrast, non-native apps may require multiple API calls per invocation. These limits need to be considered especially when other apps are also sharing in those limits.

- Native apps exhibit the same benefits of scalability as the Salesforce platform. So, adding thousands of users to your system, or generating thousands of documents, is all handled by the existing Force.com infrastructure.

## RELIABILITY

- Since native apps reside 100% within Salesforce, they will always be up and running when Salesforce is running. Non-native and hybrid apps depend on outside servers and network connectivity which may or may not have the same enterprise-class infrastructure as Salesforce.

- As a native app, S-Docs contains all test classes and test code coverage as required by Salesforce. Comprehensive testing ensures that the app runs as designed and continues to run uninterrupted. https://developer.salesforce.com/blogs/developer-relations/2012/11/how-code-coverage-works.html

## ADMINISTRATION

- Since a native app is part of Salesforce, it is much easier to implement and maintain. S-Docs exclusively uses existing Salesforce admin setup menus for administration.

- User provisioning is handled through standard Salesforce setup screens. Native apps do not require a separate procedure, login or service to manage user accounts. Native apps do not require setup of account tokens.

- Access within the native app is also managed through standard Salesforce permissions and record sharing rules.

## CUSTOMIZATION

- Native apps can be customized to your needs by adding new fields, workflow rules, validation rules, triggers etc. Your customizations are not affected by subsequent upgrades of Salesforce or S-Docs.

## INTEGRATION

- All Salesforce objects inherit APIs including REST APIs. This allows for an easy integration with external systems.

- Native apps are even easier to integrate with your existing Apex code. Calls to global services are available that make custom integration straightforward.

## COST

- A native app leverages the Salesforce platform which is already paid for with your existing Salesforce licenses. Therefore you do not pay twice for infrastructure.

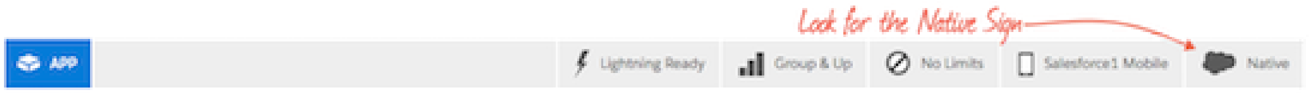
- Non-native vendors usually charge an additional amount for workflow or API/Batch Processing capabilities because it taxes their infrastructure. Since S-Docs is part of the Salesforce infrastructure, which is already paid for by your existing user licenses, there is no added cost for these capabilities with S-Docs.

## FUTURE COMPATIBILITY

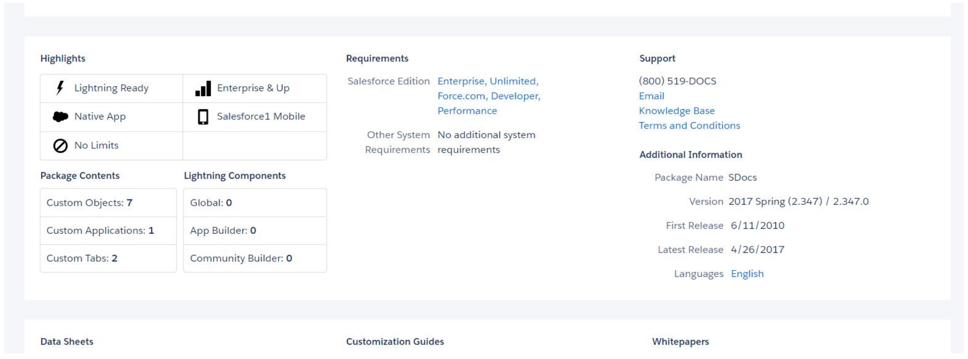
- As Salesforce continues to innovate and release new features and functionality it is critically important to Salesforce that customizations previously made by their customers not only continue to work, but can easily take advantage of these new capabilities. Native apps are uniquely positioned alongside your existing Salesforce customizations because they are made using the same technologies. Non-native apps are constantly retrofitting their product with each Salesforce release to prevent breakage when features are added or decommissioned.

## How to Identify a Native Salesforce App

All native applications on the Salesforce AppExchange are listed with a graphic on the page that denotes their native status.



Likewise, in the AppExchange listing you should see a “Native App” listing in the Highlights table. Keep in mind, this is not 100% accurate.



Highlights

| | |
|---|---|
| Lightning Ready | Enterprise & Up |
| Native App | Salesforce1 Mobile |
| No Limits | |

| Package Contents | Lightning Components |
|---|---|
| Custom Objects: 7 | Global: 0 |
| Custom Applications: 1 | App Builder: 0 |
| Custom Tabs: 2 | Community Builder: 0 |

Requirements

| | |
|---|---|
| Salesforce Edition | Enterprise, Unlimited, Force.com, Developer, Performance |
| Other System Requirements | No additional system requirements |

Support

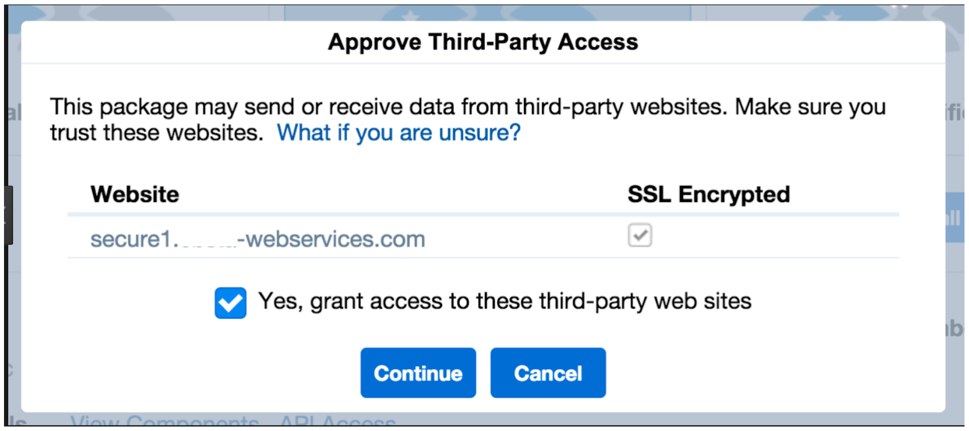
(800) 519-DOCS
Email
Knowledge Base
Terms and Conditions

Additional Information

| | |
|---|---|
| Package Name | SDocs |
| Version | 2017 Spring (2.347) / 2.347.0 |
| First Release | 6/11/2010 |
| Latest Release | 4/26/2017 |
| Languages | English |

Data Sheets
Customization Guides
Whitepapers

The true test happens during the installation of your app. If you see a message similar to the one below, it indicates that the app you are installing is NOT native. This is specifically requesting that you allow Salesforce to communicate with an external service.



**Approve Third-Party Access**

This package may send or receive data from third-party websites. Make sure you trust these websites. What if you are unsure?

| Website | SSL Encrypted |
|---|---|
| secure1.[redacted]-webservices.com | ✓ |

✓ Yes, grant access to these third-party web sites

S-Docs
Simple. Smart. Secure.

## Key Questions to Ask During Your Evaluation of a Non-Native Application:

1) Does the app require you to whitelist any IP addresses in Salesforce to use any feature of the app?
2) Does your company use any web services that are required to use any feature of the app?
3) Where are those services hosted? Are they regional and meet data residency requirements?
4) How are services encrypted?
5) How are networks separated between corp domain and services?
6) What are the SLAs for your service and performance requirements for uptime, bandwidth and latency?
7) What is the communication process for downtime and breaches? Can you provide a history of those for the previous 24 months?
8) When was the last time you implemented your emergency management procedures?
9) What audits and certifications do you have?

**COMPANY OVERVIEW**

Founded in 2010, the S-Docs team is led by a former Director and Technical Architect from Salesforce and is comprised of experts in Salesforce and document solutions. S-Docs is now used by thousands of global subscribers from all industries with a multitude of use cases, but they all share one key theme: The need for a powerful yet easy-to-use native solution. We are proud that they have made us the #1 native document generator and put us in the top 1% of all apps on the Salesforce AppExchange.