

# DreamFactory Customer Privacy and Security Whitepaper

## Delivering Secure Applications on Salesforce.com

By Bill Appleton, CTO, DreamFactory Software  
[billappleton@dreamfactory.com](mailto:billappleton@dreamfactory.com)

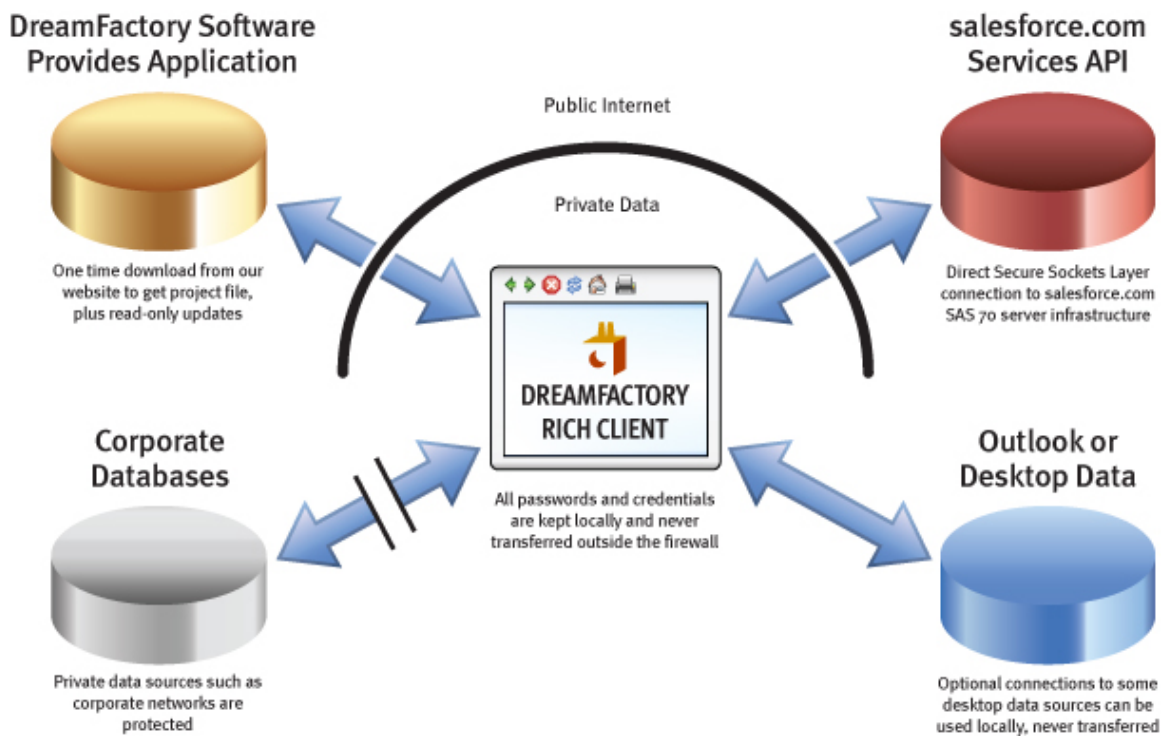
## Introduction

DreamFactory Software is the world's leading publisher of rich applications for next generation web service platforms. Our suite of business applications available on salesforce.com's AppExchange use the DreamFactory runtime client, including DreamTeam, Carousel, FormFactory, OrgView, SnapShot, DoX, and Web Meeting Mashup, as well as a host of additional applications from other partners. Our client technology does a better job of protecting your privacy and security than any other method of consuming web services. This white paper discusses the advantages of our unique architecture and explains the security and privacy polices put in place by DreamFactory Software, Inc.

## Server-less Architecture

DreamFactory Software's runtime client and rich applications can be downloaded from our website at [www.dreamfactory.com](http://www.dreamfactory.com). Embedding a DreamFactory application is similar to embedding an animation or a movie file on a web page (i.e. there is a client player and a rich document that contains the content). All of our different applications take advantage of the same runtime player. The client player implements a security sandbox that protects local files, other documents, and private network assets. The player is carefully written to minimize and abstract contact with the local machine, but to also allow access to certain external web based assets or local desktop files under user control.

Once the rich application is downloaded into the secure DreamFactory player, the current salesforce.com session ID is supplied through the Custom Link or S-control as a URL parameter. The embedded client application uses the session ID to communicate directly with salesforce.com using pure XML web services. No other servers are involved. Once you download the rich application from our website, the process is complete and in turn, DreamFactory does not host any of your data. Your private data, username, password, and session ID are not transferred anywhere other than back and forth to salesforce.com. Your session ID is used for communicating directly with Salesforce while our application is running. Your private salesforce data is not transmitted, duplicated, or cached in any other server or database. All communication is conducted directly through an encrypted Secure Sockets Layer (SSL) pipeline with the salesforce.com SAS 70 certified servers that store your data.



**Figure 1: DreamFactory Security Architecture**

The DreamFactory client operates in much the same manner as a web browser, but instead of communicating with HTML, we communicate with XML directly to the service architecture. One advantage of this is the communication speed is greatly enhanced due to the direct connection, but another key advantage is that your data travels back and forth to Salesforce directly and is not read by any other server. New versions of our embedded applications become available at the DreamFactory web site from time to time and are brought down to replace the existing version. Customers that want to embed their own private versions of these applications can do so, although they will have to manage their own application updates.

The DreamFactory client can also enable direct connections to Outlook or desktop data sources if the end user requests this. A client security certificate is presented when appropriate to insure that this is what the end user intended. The DreamFactory engine is not capable of accessing these private resources without client permission. Nor is the engine capable of accessing any foreign URL with data, such as a corporate database or other network device behind the firewall. The only network assets that DreamFactory is allowed to access are public web services like salesforce.com. The security settings of the DreamFactory client can also be managed by a corporate IT department if desired.

## Salesforce Native Operation

DreamFactory applications use salesforce.com directly, and so they are completely limited by the current account settings and visibility limitations as defined by the salesforce.com administrator. When a user logs into Salesforce and navigates to one of our rich applications, all the settings for that user are transferred through the session ID to the data accessed by that application. There is no way for a DreamFactory application to “see” things that the user cannot otherwise access through the salesforce.com HTML interface. All DreamFactory applications must be purposefully embedded by an administrator in the salesforce.com HTML interface with a Custom Link or S-control for deployment.

DreamFactory applications “sense” the current user settings for read, write, create, and delete of the various objects used by our applications. For example, a user may be prompted that they do not have the ability to delete a certain object, and in other cases restricted objects simply do not show up at all for some users. Even if the user was somehow able to go forward and actually attempt to delete some protected object this would simply generate a runtime error message generated by the salesforce.com service architecture.

## The Native Data Advantage

DreamFactory's applications store the data that they need in your salesforce.com account using appropriately named custom objects, similar to any other custom and standard objects in the Salesforce system. There are many powerful advantages to storing this mission critical data directly in your salesforce.com account and not on someone else's server.

First, you don't have to depend on a potentially unreliable third party to store your data. Second, our applications take full advantage of all the existing data and other customizations that you have added to your salesforce.com account, including custom fields, integration links and custom objects. Lastly, all of the familiar salesforce.com services and interfaces work with the custom objects that our applications use to store their data, including reporting, dashboards, editing, linking, workflow, mobile access, offline access, and all other native capabilities of the salesforce.com platform.

## DreamFactory Software Security Policy

The DreamFactory client enables your personal computer to communicate directly with your salesforce.com account. All transactions take place directly with salesforce.com and are conducted using the SSL protocol. All transactions conform to the salesforce.com API security policy, and the policy that you have established for the users in your account. None of your private data or even your session ID is transmitted to us or to any other database. We never store your username or password. Our servers are only used for the initial download of the runtime player. The [Security White Paper](#) listed below has additional technical information about DreamFactory Runtime security.

## DreamFactory Software Privacy Policy

The DreamFactory client enables your personal computer to communicate directly with your salesforce.com account. All of the data used by our rich applications is stored in your account. None of your personal data is sent to us or stored on some other server. Any personal data that you explicitly send to us, for example to request product information, will be used solely for tracking product downloads, building customer relationships, and providing technical support. Your personal information will never be transferred to a third party or become linked to any database external to DreamFactory Software.

## Appendix: Common Security Issues

What follows is a series of security issues that you should consider, and the unique approach DreamFactory takes in resolving these issues.

### Issue 1: Protecting Private Data

Many AppExchange applications use JSP, ASP, Flex, Java, or some other server based technology to render HTML pages in the salesforce interface. In order to run the application, they must ship the current session ID back to their data center and access your account data from there. While salesforce has a secure SAS 70 certified data center to deliver their service, most other partners do not. Where is their data center? What is their security policy? Who has access to your data? Is your data cached on their server? Which records are they reading with your session ID? What other databases do they have? What other servers are they connected to?

DreamFactory doesn't have a server. We aren't "hosting" anything. You download the rich application from our web site and then you are done with us. Your private data is never shipped anywhere, except back and forth to salesforce.com through the encrypted SSL pipeline. We use the exact same HTTPS protocol that the browser does, except we read and write XML instead of HTML. Your private data, username, password, and session ID stay in your salesforce account. We can't loose or steal your data. Our company doesn't have your data in the first place.

### Issue 2: Protecting Credentials

When you log into salesforce.com a session ID is generated, and this ID is used by the service architecture to access your data. Shipping private data across the Internet is one thing, but server based solutions must also transfer your session ID to their data center. If your session ID is lost or stolen then any data visible to the current user can be read, updated, or even deleted. There is no way to know what account data has been accessed by a third party once they have your session ID, because these transactions take place behind their firewall. The potential for lost or stolen account data increases with every new server that has access to your session ID.

DreamFactory only uses your session ID inside your personal computer to access your salesforce account. The session ID is used for communicating directly with salesforce.com, and then it is destroyed the moment you navigate away from our application. The session ID is only used to access the data you request and then see moments later on your computer screen. Neither your private data nor your session ID is used for any other purpose. Please feel free to verify all of these claims and the secure behavior of our applications with a network monitor.

## Issue 3: Using JavaScript

At first glance, one might think that the standard browser security model enforces the secure behavior of S-controls written in JavaScript. The JavaScript is stored in your salesforce account, and because of the originating host security model, it should only be able to access the data in your account, right? Unfortunately, this is not true. Any JavaScript with the ability to read data using a session ID is also able to log into ANOTHER salesforce account and move data between the accounts.

In this situation the data would be transferred under the cover of the SSL connection, so the use of two simultaneous accounts is not easy to detect. A malicious person could use a free, anonymous Developer account as a waypoint to store stolen data while the JavaScript S-control appeared to be operating normally.

DreamFactory Software's patented security model eliminates the ability for an application to steal data in this manner. The use of the salesforce transaction URL is allowed, but the use of the salesforce login URL is protected. Any application that attempted to log in to a secondary salesforce account would trigger a security alert dialog for the end user.

There are some other troubling characteristics of JavaScript based systems, including Ajax, Flash, and Flex. An important vulnerability is the "prototype hijacking" exploit. This basically allows any running JavaScript application to gain control of the web services pipeline. Stefano Di Paola and Giorgio Fedon present a detailed discussion of this problem in their paper [Subverting Ajax](#). A related security problem with JavaScript is the [JSON Cross Domain Attack](#). In either case, any use of JavaScript with secure or private data should be carefully evaluated by the enterprise customer.

The scripting engine in the DreamFactory client was specifically designed to prevent the prototype hijacking exploit. System routines can be sub-classed, but this does not effect the operation of other project files. The JSON cross domain attack is irrelevant because our security sandbox applies the originating host security model to all assets, including text and even graphics.

In conclusion, we have designed the DreamFactory client to deliver a safe, secure, private, and rich experience to our customers. Your security and privacy are of the utmost concern to us. Please feel free to contact us directly with questions, comments, or problems.

### **Additional Information:**

[Security Whitepaper](#)

[Enterprise Installation Guide](#)

[Verisign Certificate](#)

### **JavaScript Security Links:**

[Subverting Ajax](#)

[JSON Cross Domain Attack](#)

[Top 10 Web 2.0 Attack Vectors](#)