



Get More Out of Salesforce Tap into the Power of *Force.com*

If your sales and marketing team has hit the ceiling with a standard Salesforce CRM, you need to learn about *Force.com*, the Salesforce development platform. With *Force.com*, software developers build powerful custom applications that expand what your team can do in the Salesforce environment. Organizations that don't have a Salesforce CRM can also harness the power of *Force.com* by developing standalone applications tailored to meet their specific needs. This white paper provides an introduction to the *Force.com* development platform and explains why you need to understand the difference between a sales consultant and a Salesforce developer.

Introduction

Is your sales and marketing team ready to do more with Salesforce? If so, and you're the one who has to make it happen, you might be wondering where to start.

Did you visit *salesforce.com*? If you did and you're not yet fluent in *forcespeak*, you might have found the jargon a little disorienting. Certainly, the finer points of *Sales Cloud*, *Service Cloud*, *Custom Cloud*, and *Collaboration Cloud* take some effort to master. And sorting through all the licensing options available for each cloud is a whole separate adventure.

Yet probably the most frustrating part of this brain-numbing exercise is the sneaking suspicion that it would be really easy to turbocharge your CRM if only you knew a little more about how Salesforce worked—the real fundamentals. That would definitely give you a better idea of how to go about setting up the right Salesforce system for your sales and marketing team.

Here's some good news: The inner workings of Salesforce.com CRM applications are not shrouded in mystery. In fact, they're all built on a single development platform, *Force.com*, that's readily available to third-party software developers. But what exactly does that mean?

Salesforce was designed to be customized.

It means you can view the impressive array of features in a standard Salesforce package as simply a generic toolkit that gets you started. Sure, some organizations do just fine with the standard collection of features, but many other organizations—like yours—eventually decide to reorganize the toolkit, add to it, or even simplify it by tossing out tools they don't need. Thanks to *Force.com*, you can easily do any of these.

With *Force.com*, developers can build software applications within the Salesforce environment. So, if you already have a Salesforce CRM, your custom apps will blend right in with your standard feature set. Plus, even organizations that don't have a Salesforce CRM can benefit from powerful standalone software applications built in the Salesforce environment.

Developers can build powerful, custom Salesforce apps on the *Force.com* platform.

This white paper presents an overview of *Force.com*, the Salesforce development platform. The paper briefly describes some of the standard *Force.com* development tools, as well as the system limitations that developers must take into account as they build custom applications. Finally, the paper contrasts the role of a sales consultant with that of a Salesforce developer and explains why it's important to distinguish between the two.



What is *Force.com* anyway?

Salesforce.com offers a diverse menu of sales and marketing applications, each with multiple licensing options, yet every one of their applications is built on a single, underlying technology: the *Force.com* development platform.

Software developers use *Force.com* to build custom applications with the same tools that Salesforce.com used to build their *Sales Cloud*, *Service Cloud*, *Custom Cloud*, and *Collaboration Cloud*. So when we say we're "developing an app on the *Force.com* platform", it means we're using *Force.com*'s programming language (Apex) and standard development tools to create new Salesforce features that plug into the same data as your standard features.

If your organization already has Salesforce, you automatically have access to *Force.com* development tools.

If you don't have a Salesforce license but you want to use custom Salesforce apps, you can buy *Force.com* licenses for your users, at lower cost than Salesforce licenses.

Custom Development with *Force.com*

Salesforce is an application that runs on an Oracle database, which makes it immensely scalable and able to support high-volume transactions.

Because Salesforce must be customizable to meet the unique demands of individual client organizations, Salesforce.com rolled out a series of development tools based on a Java-like coding language called Apex.

Development Tools

Here are thumbnail sketches of some *Force.com* tools that developers use to build custom Salesforce applications:

- **Apex programming language**
Developers use Apex to add business logic to your Salesforce apps. They also use Apex to define and constrain other tools, like classes, triggers, and controllers.
- **Classes**
These are the blueprints from which specific objects are generated. Classes can be scheduled, run manually, or called by a trigger or Visualforce page.
- **Triggers**
Very powerful tools, triggers can be programmed to do just about anything, including update related records, call classes, trigger e-mails, and send data to external APIs.
- **Visualforce (VF) pages**
The most powerful development tool available, a VF page allows the developer to create a very flexible user interface with Salesforce as the backend. VF pages support HTML, Java script, AJAX, and other Web



languages, and they can be rendered many ways—for example, in a Salesforce tab, popup window, button, or page layout component. VF pages can contain application code (in Apex controllers), call classes, and query data. They also allow users to interact with, update, and create records, as well as update and relate multiple objects in a single insertion.

For more information about *Force.com* development tools, check out the [Force.com Workbook](#) and the [Visualforce Developer's Guide](#).

Salesforce Limitations

Despite its very powerful development tools—or perhaps because of them—the *Force.com* platform has some built-in limitations. You see, all Salesforce applications *must* function within certain limits to prevent any single client from hogging transaction cycles at Salesforce.com's shared data centres.

Experienced Salesforce developers understand these limits and know how to work within them—and sometimes around them—to build robust applications on the *Force.com* platform.

Unfortunately, there are a lot of inexperienced developers who are unaware that constraints even exist. Because of this, they simply write code until they encounter a restriction and then they deal with it—or they wash their hands of the matter and let their client handle the problem.

Since most restrictions are related to data volume, errors can usually be avoided—if you know what you're doing.

There are actually three constraints that developers must address when building a custom Salesforce application: data storage limits, query limits, and SQL limits. Let's have a look at each of them.

Data Storage Limits

Salesforce data is expensive, so beware—especially if you have marketing automation applications. They're the greatest data users.

To keep costs down, we always recommend limiting record creation. Most Salesforce accounts start with 1GB of data storage, or about 500,000 records. When you exceed this limit, you'll need to pay for storage, which can range from \$800–\$1,500/GB (500,000 records).

E-mail marketing is one activity that can rapidly exceed data storage limits. For example, an organization that sends e-mail to 70,000 contacts could generate 100,000 records per mailing. The organization would exceed their storage capacity after only five mailings unless they scrubbed unnecessary records.

Query Limits

These are, by far, the hardest limits to deal with. As soon as your application works with even a few thousand records, it will likely run up against query limits.



For instance, there are limits on the number of records that can be called, updated, or edited. There are also limits on the number of records you can keep in memory.

The only way to get around these limits is to *bulkify* your queries, meaning you must design them to process multiple records at once. However, this isn't always possible so you might have to manipulate your application to somehow work within the query limits.

SQL Limits

Salesforce uses a query language called SOQL that mimics SQL calls but does not support all SQL read/write call types. The limitations can be cryptic and they often frustrate new developers.

On the bright side, Salesforce.com is slowly bringing in new features to support standard SQL read/write calls.

Consultants Are Not Developers

Sales consultants are not application developers, yet they often represent themselves as development experts.

In particular, be wary of consultants who recommend a Salesforce solution and then try to up-sell the implementation. Does their organization have the resources, expertise, and experience to competently engineer software for you? Do they know what they're doing when it comes to requirements analysis? Can they build a backlog and implement it?

We've seen the trail of destruction that a jack-of-all-trades sales consultant—a so-called “expert”—can leave behind and it's not pretty. Several times, we've had to pick up the pieces afterward and build a proper software application that actually worked. Obviously, it's better to get it done right the first time.

When should you use a consultant?

Sales consultants are typically early lifecycle implementation partners. They're often qualified to set up and tune your Salesforce application because the architecture is pre-packaged. You should never hire a sales consultant to build a software application.

When do you need a developer?

You'll need a software developer if you plan to build custom Salesforce functionality, such as inventory management with SAP integration, workflow, real-time inventory chart updates, or any standalone application. If you plan to *develop* new apps, you'll need a qualified *developer*.





Architech Solutions is a Toronto-based technology consulting and software development firm. We design and build powerful, user-centred systems that work. We're agile, disciplined, and passionate about delivering for our clients.

Architech Solutions
70 Bond Street
Suite 400
Toronto, Ontario
M5B 1X3

Phone: (416) 607-5618
Fax: (416) 352-1768

To book a free on-site Discovery Workshop led by our team of consultants, e-mail info@architech.ca or visit us at www.architech.ca

© 2011 Architech Solutions Consulting Services Inc. All rights reserved.

