



 [Print Article](#)  [Close Window](#)

From: www.cio.com

Floating Customer Master Data into the Clouds

– David Taber, CIO

June 14, 2012

In [18th century opera](#), stories sometimes became so convoluted that libretto authors had to resort to divine intervention to make the plots resolve. God came down from the clouds to solve impossibly-thorny problems in the third act.

In distributed business systems, we have the messiest of plot complications—there's no source of truth for who all our customers are. Accounting has one version, distribution may have another and CRM may have a third. This problem multiplies with international divisions, global customers, and business units that were the result of mergers and acquisitions.

The more distributed your systems are, the more likely you have this problem. I have one client that is endlessly creating hundreds of dupe account records every day because it has more than 75 systems, each of which thinks it owns the definition of accounts.

If you are lucky enough to already have an [ESB](#) and customer master database, and everyone's already using them, congratulations. The other 99.9 percent of you should read on.

Deus Ex Machina: A Loosely Coupled Solution to Loosely Coupled Data

The problem comes from loosely coupled systems. However, the solution strategy can come from the ultimate loosely coupled system—cloud computing. Let me explain.

Analysis: [CRM Systems: Unite or Die?](#)

If business rules (and politics) support it, you could create a customer master database as a cloud service. The process of setting up that service isn't that much different from the good old days of database; after all, you're just using a more ubiquitous set of protocols. Remember, traditional customer masters typically become [boil-the-ocean projects](#) for political and data quality reasons, not because of technical issues.

A different approach is to choose a cloud application, rather than a centralized database, to be responsible for the customer master. The knee-jerk choice for that central (cloud) application is accounting. While that system will typically have very good data quality, let's not jump to a conclusion just yet.

For an application to be a suitable customer master repository, its infrastructure must:

- Be accessible, calling in and out from everywhere in the enterprise with [SOAP and RESTful APIs](#).
- Be extensible, handling extra fields that act as foreign keys for every other applications' use, as well as extra text fields to handle multiple versions of the account name, such as A&P vs. The

Great Atlantic and Pacific Tea Company, as well as other key parameters.

- Handle different categories of accounts, including direct, partner, prospective and inactive.
- Handle foreign character sets and multiple address formats.
- Scale to the number of accounts and peak load update rates at reasonable cost.
- Provide secure access rights and update privileges for each class of account and user, with the ability to encrypt sensitive data as needed.
- Host code to operate on account data in support of transformation, workflow, data hygiene and other functions.
- Be reliable and available at least 24 hours a day, six days a week, with auto-retries when other parts of your infrastructure are not responding.

Cloud CRM a Prime Spot for Customer Master Data

I'm going to argue that the natural system for hosting customer master data is the CRM. Because of its station in the business process, it already has all the accounts in it. The CRM system already has pointers to the important records that are children of accounts—deals, contracts, contacts and (potentially) customer support cases and shipping information. CRM systems are already close to sources of new account information, including partners, [ecommerce systems](#) and portals. (Our firm has deployed both [NetSuite](#) and [Salesforce](#) in this use case, as they satisfy the bulleted criteria above, though it's worth noting that other cloud CRM offerings may be suitable as well.)

Analysis: [Cloud CRM: The Politics of Data Ownership and Control](#)

Our argument for this approach is the KISS principle. We want to make the vision of a customer master a practical reality in a fairly short time. Putting customer master data in a cloud CRM also facilitates application of agile projects, as a new system can be brought into the customer master fold with each new scrum cycle. Beyond avoiding some of the political realities of traditional customer master projects, there's also the IT advantage of not having to procure, learn, deploy and administer another piece of technology.

Of course, this strategy cannot diminish the big challenge of reconciling all the different "account lists" across your enterprise. You'll still need to apply the standard approaches for account naming conventions, deduplication, record ownership and account/divisional hierarchies. God may descend from the clouds in this little opera, but miracles can only go so far.

David Taber is the author of the new Prentice Hall book, "[Salesforce.com Secrets of Success](#)" and is the CEO of [SalesLogistix](#), a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel and India. Taber has more than 25 years of experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter [@CIOonline](#), on [Facebook](#), and on [Google +](#).

© 2012 CXO Media Inc.