I had two enterprise customers this week who both had similar security requirements, and both sets of requirements were solved with a similar pattern that I thought you might find useful.

Imagine your company organizes itself into different groups, and the work of each group needs to remain segregated from the work of the other groups.

So you configure your CRM organization with three business units — let’s called them *Red*, *Blue* and *Green* — each under the root parent business unit. And you configure the security roles with business unit-level privileges. So far, so good.

But now there is a requirement that some users in the *Red* business unit need to be able to work with records in the *Green* business unit, but not the *Blue* business unit. And some users in the *Blue* business unit need to be able to work with records in the *Red* business unit, but not the *Green* business unit. But no one is allowed to work with records in all three business units. What now?

Teams for Owning and Sharing Records

This is where teams come in.

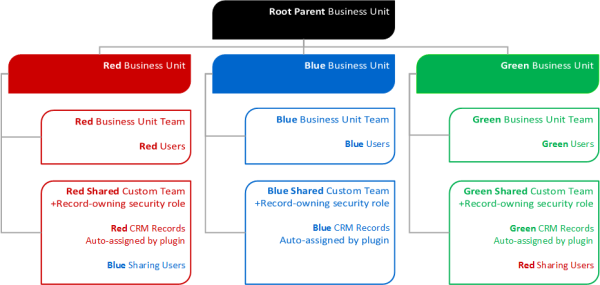
CRM creates and manages a team for each business unit. These business unit teams have the same name as the business unit. Members of the business teams is managed by the CRM platform when we add or remove users from the business unit, so there’s not much we can do with the business unit teams.

To solve our challenge, we’re going to create another team for each business unit own the business unit’s records share them with selected users from other teams.

So we now have six teams:

* The default teams for each business unit: *Red*, *Blue* and *Green*.
* Custom teams to own and share records: *Red Shared*, *Blue Shared* and *Green Shared*. (Optionally, you can create one set of teams to own the records and another set of teams to share the records, according to your security requirements).

**Note:** The naming conventions for these teams are important, but I haven’t found the perfect naming convention for lots of teams with a similar purpose, so your suggestions are welcome!

[](https://msdnshared.blob.core.windows.net/media/MSDNBlogsFS/prod.evol.blogs.msdn.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/55/55/3187.Fig%201.png)

Managing Record Ownership with Teams

Instead of records being owned by users, in this design we’re going to have records owned by custom teams instead.

Team-ownership of records has several benefits:

1. Teams don’t leave the company and have their accounts disabled.
2. Teams don’t move from one business unit to another causing mayhem when all their records move with them.
3. We can easily share records with lots of users from other business units by adding them to a team.

We’ll need to assign a security role to the record-owning teams so that they have privileges to own the appropriate records. Use the base security role you would assign to typical users. (If you have separate sets of teams for owning and sharing records, you could give the record-owning teams with a security role with Read, Assign, Append and Append to privileges at the business unit level).

I recommend using a plugin to reassign all your sensitive records to the appropriate record-owning team. You could use a workflow, but I prefer using a plugin. The plugin fires as soon as the record is created and uses some record-based logic to determine which record-owing team to assign the record to. For example, if *Red* represents a sales region, then opportunities would be reassigned based on the sales region of the potential customer. It’s a good idea to use configuration data rather than hard-code the teams in the plug-in; for example by adding a team lookup to the sales region records.

This plugin is critical. We’re going to have *Red* users working with *Green* records. We need to use the appropriate business logic to ensure that if, for example, a *Red* user creates a quote on a *Green*opportunity that the quote is owned by the *Green Shared* team and not the *Red* user, otherwise *Green* users won’t be able to see the quote.

We’re also going to reassign all the existing user-owned records and assign them to the record-owning teams instead. You could achieve this programmatically if you have a large number of existing records to reassign.

Optionally, you can modify any views that are based on users owning records such as the *My Accounts*, *My Contacts*, *My Cases* and *My Opportunities* system views. For example, you can modify the views to filter on the *Created By* user, or use a custom user lookup on each record instead of the *Owner* field.

Step one is complete. All our sensitive records are now owned by a custom, record-owning team in the appropriate business unit.

Manage User Access by Using Teams

The next step in our design involves adding users to our record-sharing teams. Users in the *Red* business unit don’t need to be added to the *Red Sharing* team because the users and the team are already in the same business unit and their security role gives them privileges to work with records owned by users or teams in their business unit. So all we need to do is add the appropriate users from the *Red* business unit to the *Green Sharing* team, and some users from the *Blue* business unit to the *Red Sharing* team. Hey presto! And we’re done.

This design has two big advantages over the more common design pattern of simply sharing records with another user or team:

1. Our design won’t create any records in the *PrincpleObjectAccess* (POA) table which has been known to lead to [performance issues](http://blogs.msdn.com/b/crminthefield/archive/2011/06/09/principalobjectaccess-performance-recommendations.aspx).
2. We’re not using the POA table. That’s so handy, it’s worth stating twice!

If you’ve created separate teams for owning and sharing records, you’ll need to assign a security role to our custom, record-sharing teams but usually one your existing security roles will be appropriate. For example, if most of the users you want to share records with are sales people, then you can assign the *Sales Person* security role to the record-sharing teams (or whatever security role you normally assign to sales people).