Costco Global Teams & Team Performance

Problem Statement

At Costco, the organization structure and hence the employees are managed in a very granular way. The managers want to look at only their reportee's tasks, metrics, etc in the Team Performance dashboard.

So, we were looking to create many granular teams (about ~2000), a manager team for each (another 2000), and each team containing a 1:1 BPM group/role to manage the members. This number will increase.
*PS: We recently created LDAP security groups for some of the granular groups too in an effort to fix 856 doc authorization slowness (with case license)
These teams would then live in a toolkit and be shared by many process apps.

As an example of the granularity that Costco needs within their buying staff teams, here are some examples of existing roles that have been created - this is how the task assignments and Team Performance visibility needs to be broken out:

COMMON_(Role)_(Dept)_(Buying Country)

So for a single department (Department 20) that covers one buying country (USA), for example, here are the roles that are needed:

COMMON_BuyingStaff_20_USA
COMMON_Buyer_20_USA
COMMON_BuyerManager_20_USA
COMMON_GMM_20_USA

All of the above would roll up to the parent:

COMMON_BuyingStaffManager_20_USA

During analysis, it was found that, as expected, Team records are created for each toolkit snapshot in the target deployment environment. However, when a previously installed snapshot has a dependency on TeamTK_1.0, for example, and another snapshot (whether the same process app or not) is installed which also has a dependency on TeamTK_1.0, then duplicate records are inserted into the LSW_Participant and LSW_Participant_Group tables.

We understand this was designed keeping snapshot-level team management via admin console in mind. But this many records could get unmanageable when we have 20+ process apps with monthly releases sharing the same “Shared Team” toolkit.

Previous Discussion

Costco previously had a lab advocate meeting on July 24th 2015 on this topic, here are the minutes of meeting:

*Don W and Mano P presented this doc to IBM***:** ***Teams/Groups Discussion***
Dieter Koenig and Gabriel Dermler from IBM attended.

Questions/Discussion Points:

1. *“Global teams” are still discussed within IBM to be provided in the ibm bpm tool*
2. ***Create the 4000+ teams in toolkits to be used across apps?*** *IBM agreed that DB will be bloated since so many teams will be created in db every time new snapshots are installed. But mentioned that it will not affect process, task and other event transactions in BPM.*
3. *How many apps will be installed and active in a given time in Prod? - about 20 + in future*
4. *IBM mentioned that some level of performance testing were done using many teams and using team performance dashboard, but not at this level and with many active apps*
5. *Suggestion made to start using BPM admin cleanup commands to purge instances, snapshots etc*
6. *Suggestions made to wait for more developments on "Global Teams", but could take long*
7. ***Create the 4000+ teams in a Process app to be used across apps?***
   use of tw.system.org.findTeam(teamId, processAppId) api to find teams from a Team processapp might work, but this sounds hacky and probably not supported by IBM
8. *How do other companies use/configure large number of business group as Teams?*

Outcome:

1. *IBM was not confident on creating so many granular teams and suggested us to start using the admin cleanup commands.*
2. *Suggestions were made to Wait for "Global Teams", though there were no dates mentioned.*
3. *Log an RFC with the problem statement to the tool development team (which we have not done yet, should we?)*

April 2016 - Latest CoC Questions to IBM before the advocate call (with feedback from Dana)

1. Requesting an update on “Global teams” concept mentioned by Gabriel during the July 2015 advocate call. Any timeline or more details about implementation available?

2. We currently have the Teams setup at a higher level, with more granularity at the group/role level. But, when we perform task assignments using Team filter services, we encounter two issues:
    a. We are unable to set the Team’s ‘Manager Team’ to a dynamically filtered list of the more granular managers.
    b. Dynamic teams created by Team Filter Services do not show up in the Team Performance dashboard. Instead, tasks assigned to a dynamically created Team from a Team Filter Service show up in the Team Performance dashboard under the team specified in the ‘Team:’ mapping on the Assignment tab. This isn’t acceptable as it doesn’t show a narrow enough list of tasks to the managers - it shows all tasks for all members of the higher level Team before the filtering is applied. (Details of testing around this concept can be found here: https://docs.google.com/presentation/d/1oiUJ4bkDtC6r85ORmiHSAelYTb-QK6GCbp5EfV7SUnU/edit#slide=id.p)

    Is there a way to set a dynamic manager team for the dynamic team filtered team, also making sure the Team Performance dashboard will work when the member of this dynamic manager team uses the dashboard?

    Can we submit a RFC?

3. If we do create the granular teams, how can we reuse these granular teams in a federated environment (assuming the LDAP teams are not very granular)?

4. Are there any specific wsadmin commands or APIs to cleanup Teams and group/role records in product XREF tables associated with inactive snapshots without affecting the snapshot instances, document authorization, or orphaning the tasks (think Team assignment), etc?

5. Why are duplicate Team asset records inserted into the database upon snapshot deployment when the same snapshot of a given toolkit already exists in the target environment due to a dependency on another (previously installed) process app?

6. Another option explored was #7 from our previous discussion - this involved creating all of the Teams in a process app instead of a toolkit, which would greatly reduce the number of records in the product database, and still provide a single, common Team across all other process apps. The steps we took to test this were:

a. Create a new process app, ‘Enterprise Teams’.
b. Add several Team assets to the process app.
c. Create a system service that returns a String teamId based on a String teamName.
d. Create a Web Service and implement an Operation named ‘fetchTeamIdByTeamName’ using the service created above.
e. Now, from a different process app (other than the ‘Enterprise Teams’ process app), create a new BPD with a system task, a script block task, and a human task.
f. Add a variable called ‘team’ of type Team, and another named ‘teamId’ of type String.
g. In the system task, implement a service that will call the ‘fetchTeamIdByTeamName’ operation from the ‘Enterprise Teams’ web service and pass in the name of one of the Team assets created in step b above. This service should output a teamId, returned from the web service.
h. The system script block task can now use the following to obtain the team:

```
var processAppId = tw.system.model.findProcessAppByName("Enterprise Teams").id;
tw.local.team = tw.system.org.findTeam(tw.local.teamId, processAppId);
```

i. Finally, in the Assignments tab of the human service, change the ‘Team:’ mapping to use a variable, and choose ‘tw.local.team’.
j. Now run the BPD...

Unfortunately, what happens in the above scenario is that the BPD fails on the task assignment because the ‘Team:’ mapping in the Assignment tab expects the NAME (String) of a team only - you can’t provide an actual Team object. Then, when only a name is provided - such as ‘tw.local.team.name’ - then the process app can’t resolve the Team asset from the other process app and it throws a message saying that Team doesn’t exist.

It seems like there are improvements that could be made here which would allow for the functionality that Costco needs - namely if the ‘Team:’ mapping on the Assignments tab of a human activity could accept a Team object or could resolve any team from any process app.