Response Summary

We understand the feedback from NetApp team and see that there is definitely an issue with the solution. However, this is NOT the NetApp FUEL plugin implementation limitation, but FUEL limitation itself. Namely, there is no functionality in FUEL to manage backends for Cinder, hence FUEL plugin developers have to use their own way to set multi-backend functionality for Cinder.

Please see the possible next steps options below in the document.

Feedback and Comments from NetApp

Feedback from NetApp: The NetApp plugin and I believe all plugins fail to work in a multi-backend multi-plugin environment. The NetApp plugin is a bit better in the sense that it fixes LVM or Ceph and then makes itself work in a 2 backend environment, but when you add in another plugin… whomever is the last in the alphabet wins the only seat in the config.
Mirantis Comment: Yes, this is correct. All plugins for Cinder (including Cinder-NetApp) set backends so that do not expect any other changes in cinder.conf. It happens due to plugin developers can not guess and predict the additional changes that other plugins may make in the same files. Plus, there are no standard instructions on how plugin developer should safely manage cinder backends during deployment by means of FUEL.

Feedback from NetApp: #1 and the most egregious is that FUEL does not put LVM or Ceph in their own section within the Cinder.conf.
Mirantis Comment: Yes, this is correct. FUEL by default doesn’t use multibackend and puts all settings for particular backend in common section [DEFAULT]. This is a standard MOS behaviour. We can raise the feature request to MOS product team and try and release it in on of the next MOS versions.

Feedback from NetApp: #2 is that there no ‘standard’ method for the any plugin to build a proper list for the enabled_backends parameter.

Mirantis Comment: Yes, this is correct. FUEL does not provide such functionality to collect information about enabled backends for Cinder. The only way to get this information is to read cinder.conf, parse it and analyze it.

Feedback from NetApp: The NetApp plugin checks the config to see which backend was chosen by the use in the initial config, uses that answer to configure either LVM or Ceph in a section then, then sets enabled_backend to that option plus netapp. The only 2 options get out of it are NetApp and Ceph or NetApp and LVM.
The cinder_netapp class takes single backend_name (this could take more, but read on) as an argument and builds an enabled_backend list. The supporting classes for the Netapp-plugin mainly the params class lookup the backends defined by FUEL and expect one of 2 things (lvm or ceph) and pass one (and only one) of those to the cinder_netapp class.
Mirantis Comment: Yes, this is how the plugin works. The plugin uses only one of the default backends (LVM or Ceph) as FUEL can not set both of the backends active at the same time.

Feedback from NetApp: Thus if SolidFire defined a backend in our plugin (which we would do) the NetApp class if it were to run after the SolidFire (or Pure, or anyone else) class, would remove the SolidFire backend definition.
Mirantis Comment: It was described in comments to the first feedback item. Could you also let us know what happens when to run SolidFire plugin after NetApp one? Does the plugin rewrites all settings that NetApp plugin made?

Proposed Action Plan: options

#1 - Immediate workaround

As the very quick and fast workaround, we would suggest to execute SolidFire plugin at the end of deployment phase and let it to write desired settings into cinder.conf. Since FUEL has no functionality to manage Cinder backends at the moment, SolidFire plugin developers will need to find their own way to avoid rewriting existing parameters. However, there is a risk that e.g. syntax error could prevent the cloud from working correctly, so SolidFire should be very careful to ensure other backends are not impacted.

#2 - Mirantis professional services engagement to extend NetApp FUEL plugin

We can change MOS 8.0 (and the later versions) Cinder Netapp plugin to make it compatible with any future Cinder plugin. It will take time to develop the feature and to test with another Cinder plugin, at the end Cinder Netapp plugin should be validated by partner-integration team. This will have to be done as the separate services engagement. **Note:** this engagement is needed only if NetApp would like to make sure that NetApp FUEL plugin is capable of handling the already installed Cinder backends (i.e. NetApp FUEL plugin is added to the MOS solution post-deployment).

#3 - Add the proper Cinder backend feature support to MOS/FUEL roadmap
We can raise the feature request to MOS product team and try and release it in on of the next MOS versions. This feature should collect information about Cinder backends before deployment and allows to get this information during deployment.

#4 - NetApp to develop the multi backend feature in FUEL
FUEL is open-source solution and is under the OpenStack big-tent program, hence everyone can work on its development and extensions.

Mirantis recommendation

- Use approach #1 to resolve the immediate issue with Solidfire.
- NetApp product management team to work with Mirantis product management team to agree on the roadmap of the FUEL extension to handle multi-backend functionality in Cinder properly (approach #3).
- NetApp to extend the NetApp FUEL plugin documentation to make NetApp customers aware of the FUEL limitation with regard to Cinder multibackend and the way to address it.