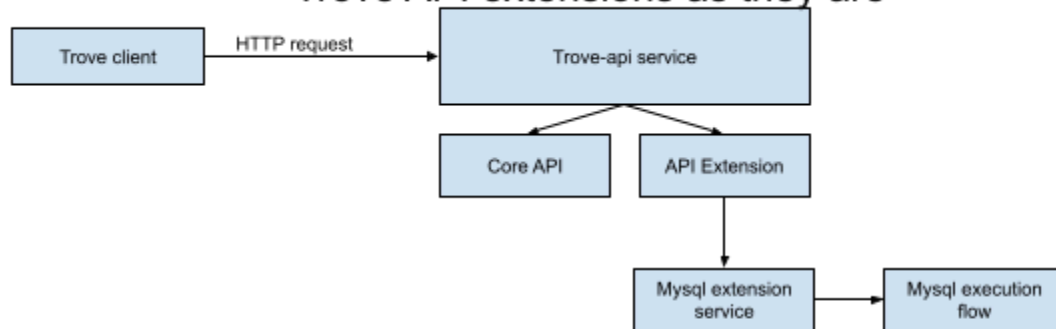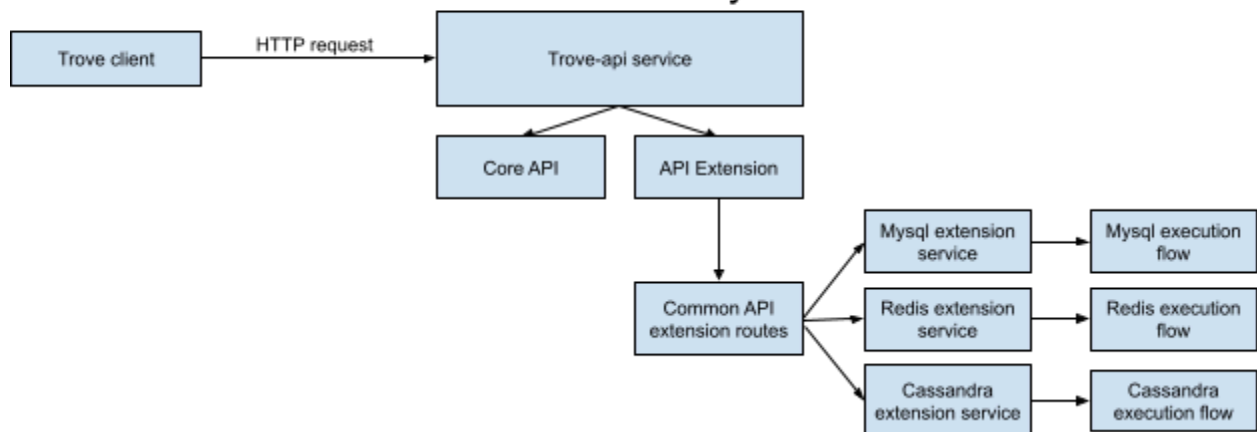## Trove API extensions as they are

Trove API extensions as they should be

At this moment if contributors will decide that one of the integrated db engines would support users/databases CRUD operations they will come into the issue of non-pluggable extensions for described operations. Here [1] you can see that users/databases CRUD operations will go through routes defined for mysql database integration.

I would like to suggest more flexible mechanism for such API extensions:
1. Extensions should be implemented per datastore manager, such as mysql, redis, cassandra, mongodb.
2. ReST service routes for API extensions should be common for all of datastores. It means that implemention should be imported/used according to uniq attribute, best option for such attribute – datastore manager.
3. Even if datastore doesn't supports users ACL or storage distinction (databases in terms of mysql, keyspaces in terms of cassandra) API extension should be implemented, each method should raise NotImplmented exception with meaningful message "Not supported".

As you can see at [2], mechanism implemented according to rules given above. So, I would like to hear all your thoughts about that.

[1] https://github.com/openstack/trove/blob/master/trove/extensions/routes/mysql.py
[2] https://review.openstack.org/70742