

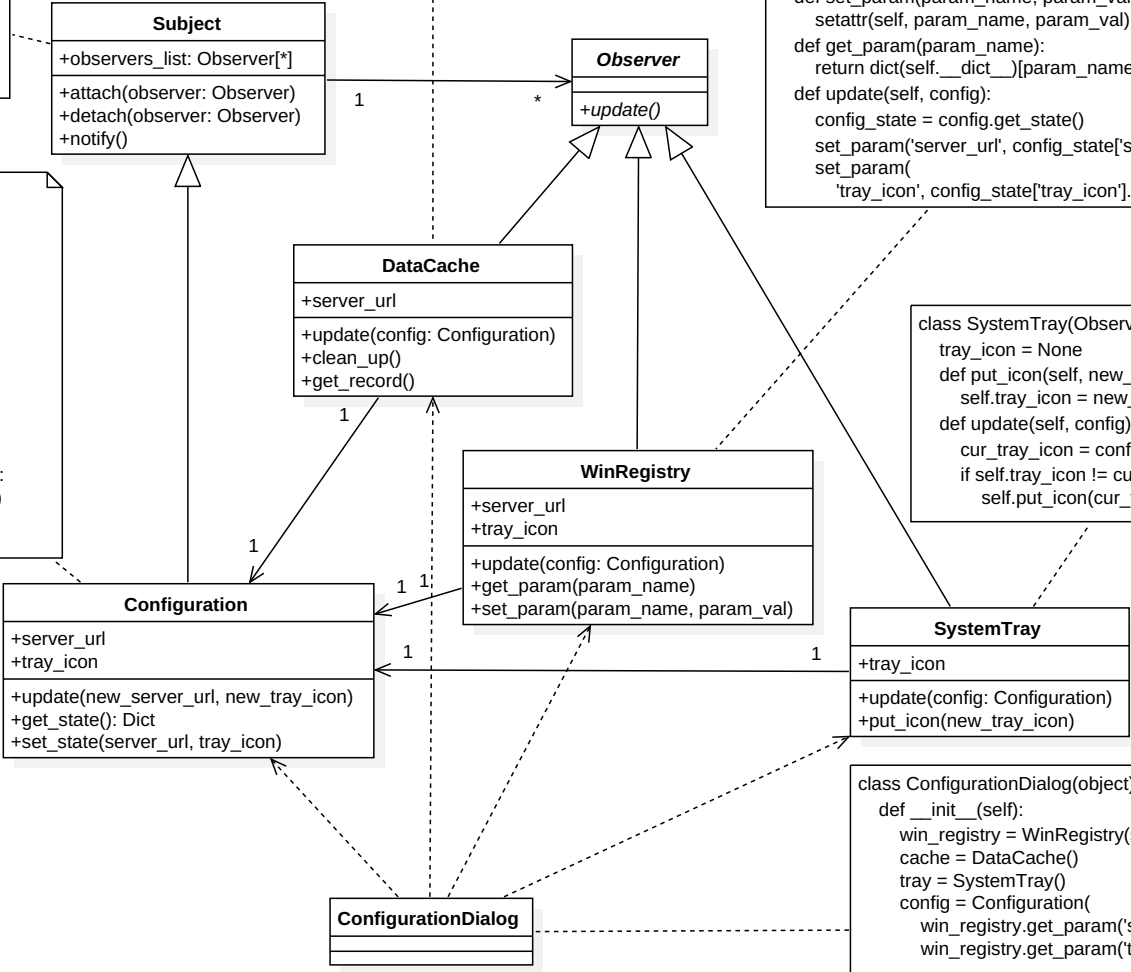
```
class Subject(object):
    observers_list = []
    def attach(self, observer):
        self.observers_list.append(observer)
    def detach(self, observer):
        self.observers_list.remove(observer)
    def notify(self):
        for observer in self.observers_list:
            observer.update(self)
```

```
class DataCache(Observer):
    server_url = None
    def clean_up(self):
        pass
    def get_record(self):
        pass
    def update(self, config):
        cur_server_url = config.get_state()['server_url']
        if self.server_url != cur_server_url:
            self.clean_up()
            self.server_url = cur_server_url
```

```
class WinRegistry(Observer):
    def __init__(server_url, tray_icon):
        self.server_url = server_url
        self.tray_icon = tray_icon
    def set_param(param_name, param_val):
        setattr(self, param_name, param_val)
    def get_param(param_name):
        return dict(self.__dict__)[param_name]
    def update(self, config):
        config_state = config.get_state()
        set_param('server_url', config_state['server_url'])
        set_param('tray_icon', config_state['tray_icon'].to_binary_data())
```

```
class Configuration(Subject):
    def __init__(server_url, tray_icon):
        self.server_url = server_url
        self.tray_icon = tray_icon
    def get_state(self):
        """
        Return all attributes of object in dictionary
        """
        return dict(self.__dict__)
    def set_state(self, server_url, tray_icon):
        self.server_url = server_url
        self.tray_icon = tray_icon
    def update(self, new_server_url, new_tray_icon):
        self.set_state(new_server_url, new_tray_icon)
        self.notify()
```

```
class SystemTray(Observer):
    tray_icon = None
    def put_icon(self, new_tray_icon):
        self.tray_icon = new_tray_icon
    def update(self, config):
        cur_tray_icon = config.get_state()['tray_icon']
        if self.tray_icon != cur_tray_icon:
            self.put_icon(cur_tray_icon)
```



```
class ConfigurationDialog(object):
    def __init__(self):
        win_registry = WinRegistry(some_ulr, some_icon)
        cache = DataCache()
        tray = SystemTray()
        config = Configuration(
            win_registry.get_param('server_url'),
            win_registry.get_param('tray_icon'))

        config.attach(tray)
        config.attach(win_registry)
        config.attach(cache)
        config.update(new_url, new_icon)
```