

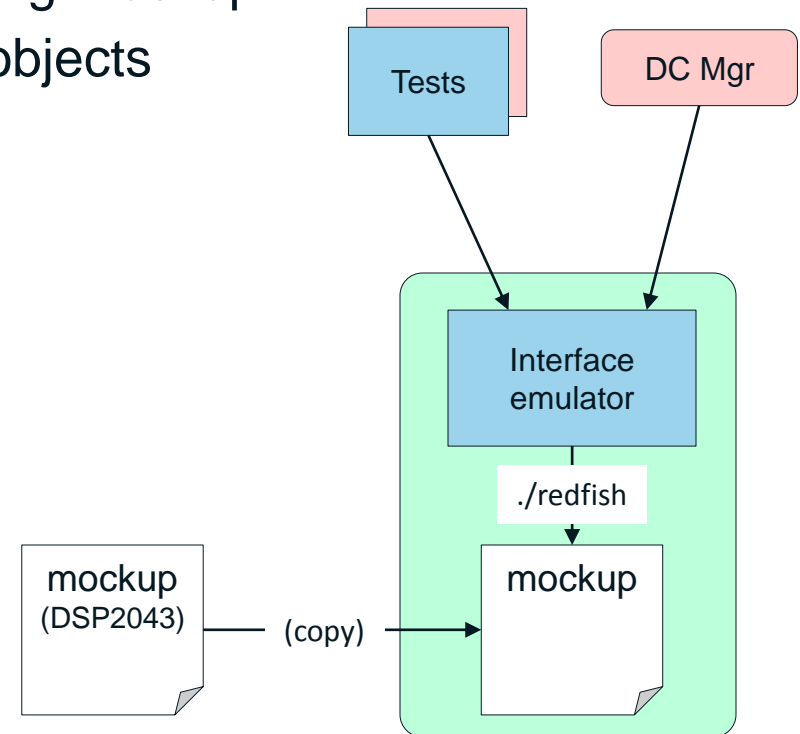
Redfish Interface Emulator

John Leung
Intel Corporation



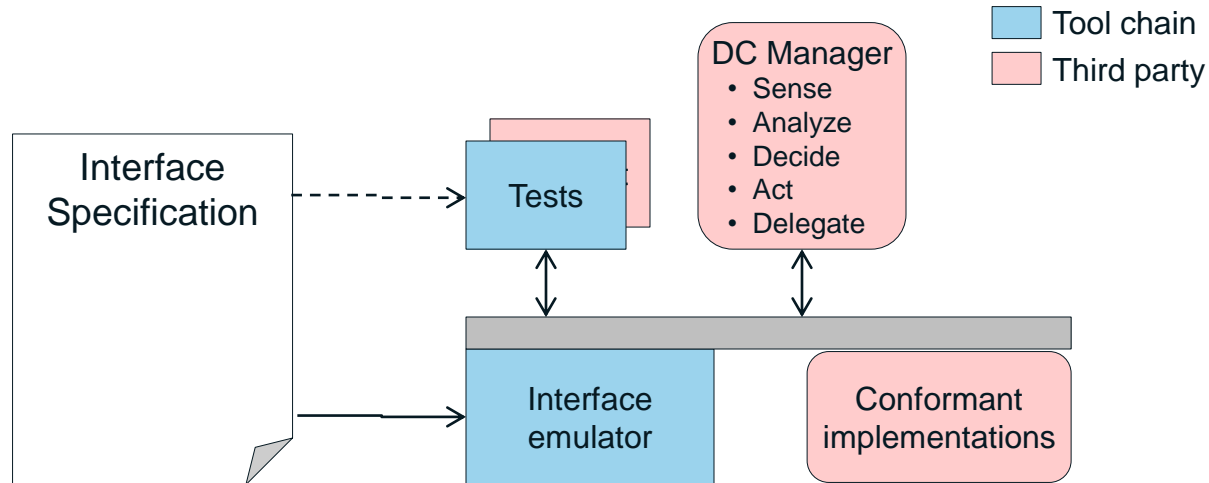
Redfish Interface Emulator

- Emulator which can quickly emulate Redfish extension and models
- Rapid instantiation
 - Static emulation only requires copying mockup
 - Dynamic emulation – code Python objects



Value of Interface Emulators

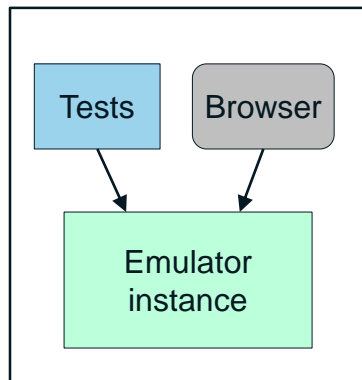
- Assist in designing interface - instantiate modeling concepts
- Allow development of unit and validation tests prior to availability of implementations (HW, SW, FW)
- Ability to replicate/scale – multiple instances can be hosted in a Foundry



Design Goals: Execution Environments

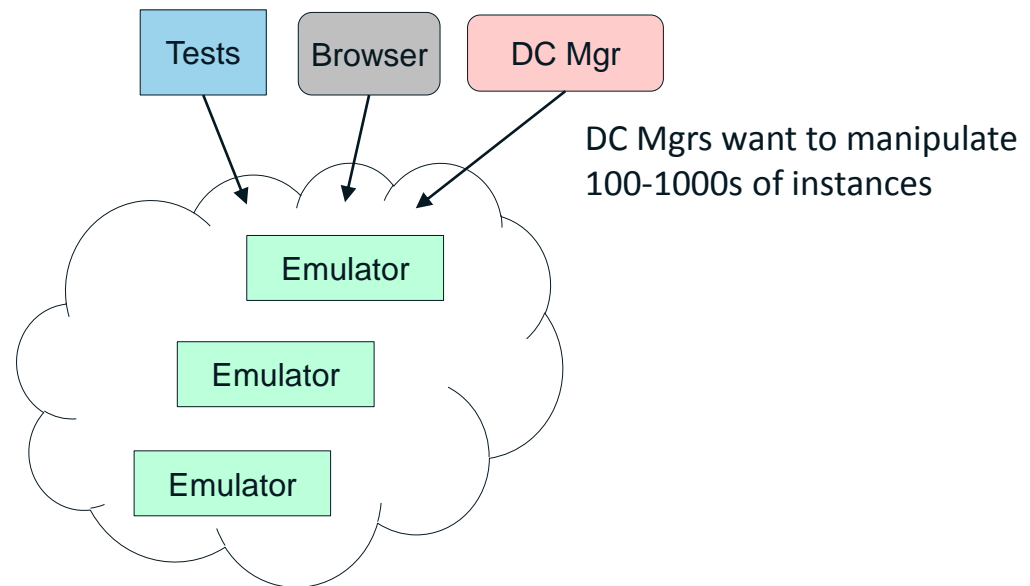
- Instances and tests can be hosted standalone or in a cloud foundry

Standalone



```
emulator-config.json
{
  "SPEC": "Redfish",
  "MODE": "Local"
}
```

Cloud Foundry



```
emulator-config.json
{
  "SPEC": "Redfish",
  "MODE": "Cloud"
}
```

Development Tree

Emulator

- **emulator.py**
- ./api_emulator
- ./templates

Unit Test

- **unittest.py**
- ./unittest_data

Local instance

- install.py
- ./dependencies

Cloud Foundry

- Requirements.txt¹
- Procfile

¹List of Python packages to auto-loaded by Foundry

-
- **emulator_resource.py**
 - **exceptions.py**
 - **static_loader.py**
 - **utils.py**
 - ./redfish/static
 - ./redfish/templates

Instantiate in Cloud Foundry

```
> cf push redfish
urls: redfish.cloudfoundry.com created
```

Access

```
GET http://redfish.cloudfoundry.com/redfish/v1
```

To validate, run unit tests

```
$ python unittests.py Redfish "redfish.cloudfoundry.com"
```

Inspect "test-redfish-emulator.log"

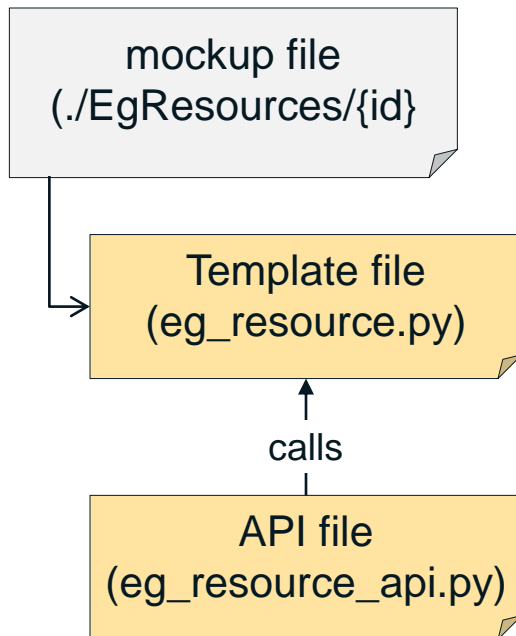
Legend

red python source

blue location for mockup

Creating Dynamic Resources

- **Create a Template file and API file**
- **Add APIs to accessible URIs**



```
class ChassisAPI(Resource):
    def get(self, ident):
        try:
            for cfg in members:
                if (ident == cfg["Id"]):
                    break
            config = cfg
            resp = config, 200
        except Exception:
            traceback.print_exc()
            resp = INTERNAL_ERROR
        return resp

    def post(self, ident)...

    def delete(self, ident)...

    def patch(self, ident)...

class ChassisCollectionAPI(Resource):
    def get(self)...

    def post(self)...
```

Emulator coding capabilities

- Randomize value that are unique per instance (e.g. serial number)
 - Can specify format for unique value (regex-like pattern)
- Ability to HTTP POST an Action (e.g. reset)
 - Fork a process to handle action while acknowledging request
- Ability to change property value
 - Combined with above to enable **Action=ApplySettings**
- Eventing (subscription/publish)
 - **POST `http://<URL>/redfish/v1/EventService/?Action=Subscribe`**
 - **{ "Destination": "(dest-url)", "Types": ["StatusChange"], "Context", "SystemAdmin" }**
- Support HTTP and HTTPS

Backup

Api_emulator Directory

Files & Directories	Description of Contents
dependencies	Required Python packages
exception.py	Exceptions thrown in Emulator
resource_manager.py	Loads and constructs the resources to be emulated. Loads both the static and dynamic resources. Handles creation, deletion and utility of resources
static_loader.py	Called by resource_manager.py to load the static resources from ./redfish/static
utils.py	Supporting functions (3)
./redfish	Directory of static and dynamic resources
./redfish/static	Static resources – mockup directory tree
./redfish/template	template files for instantiating dynamic resources

Python

- Python 2.7
- Python packages (./dependencies)
 - Flask
 - Flask-RESTful
 - aniso8601
 - itsdangerous
 - Jinja2
 - MarkupSafe
 - pytz
 - requests
 - six
 - Werkzeug