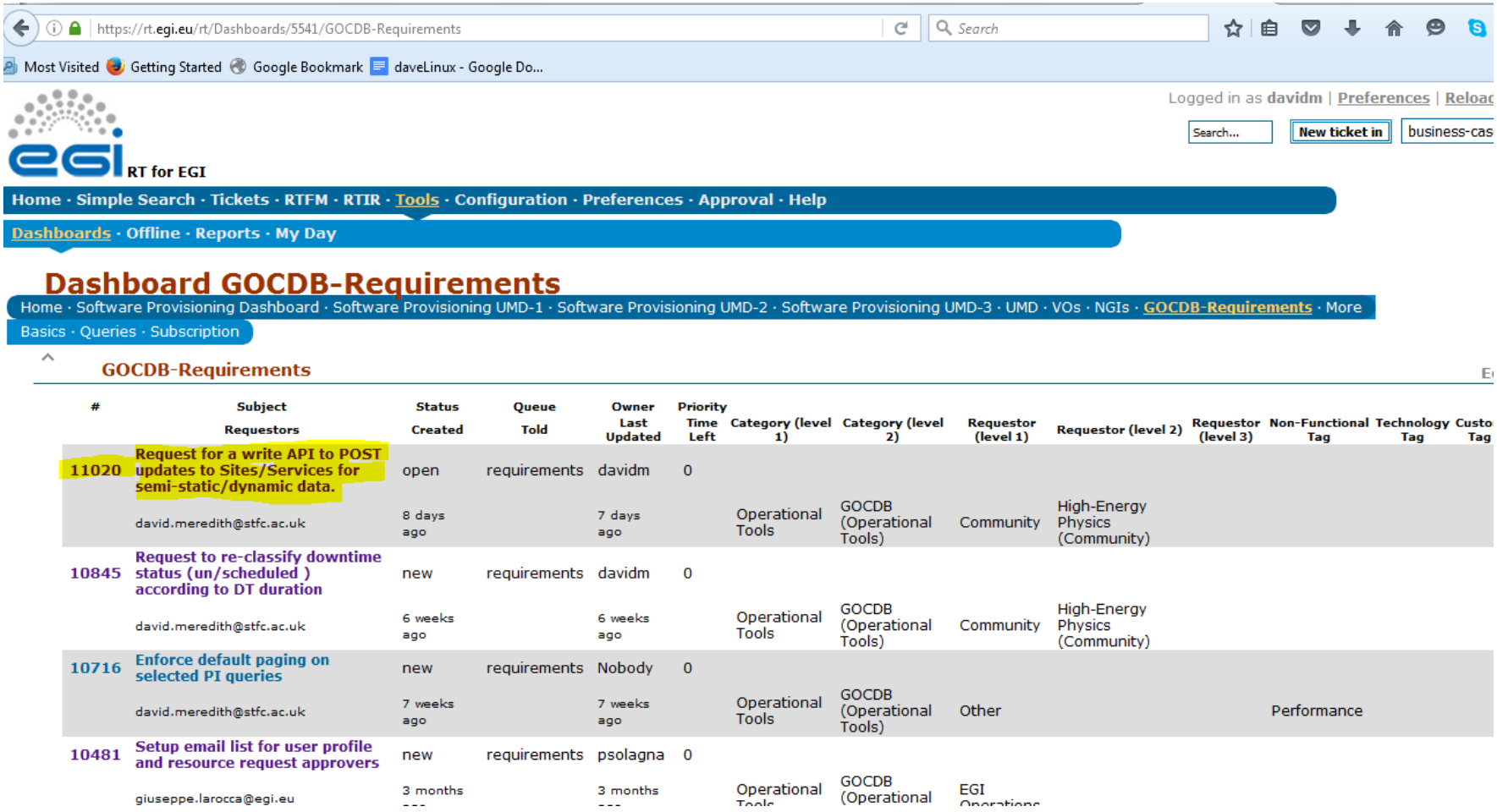
A Write API for GOCDB

david.meredith@stfc.ac.uk

12/05/2016



Ticket 11020 added to the GOCDB requirements tracker – Request for a write API to POST updates to Sites/Services for semi-static/dynamic data.

Requirements for Write API:

- CRUD? CreateReadUpdateDelete (suggest we start with just Update, we can review requirement for Create and Delete later)
- Which GOCDB entities need to be updated? (Site, Service, ServiceEndpoint?)
- Which attributes need to be updated?

The following set of attributes are considered to be sufficient to cover experiment use cases:

Attribute generic name	GLUE 1	GLUE 2	Example
CE endpoint	Included in GlueCEUniqueID	GLUE2EndpointID	ce503.cern.ch:9619
Queue name	Included in GlueCEUniqueID	GLUE2ComputingShareMappingQueue	grid_2nh_atlas
CE type	GlueCEImplementationName	GLUE2EndpointImplementationName	HTCondorCE
CE LRMS type	GlueCEInfoLRMSType	GLUE2ManagerProductName	Condor
Max CPU time	GlueCEPolicyMaxCPUTime	GLUE2ComputingShareMaxCPUTime	2880
Max Wallclock time	GlueCEPolicyMaxWallClockTime	GLUE2ComputingShareMaxWallTime	2880

The IS TF will investigate how to move this information from the BDII to GOCDB/OIM taking the following actions:

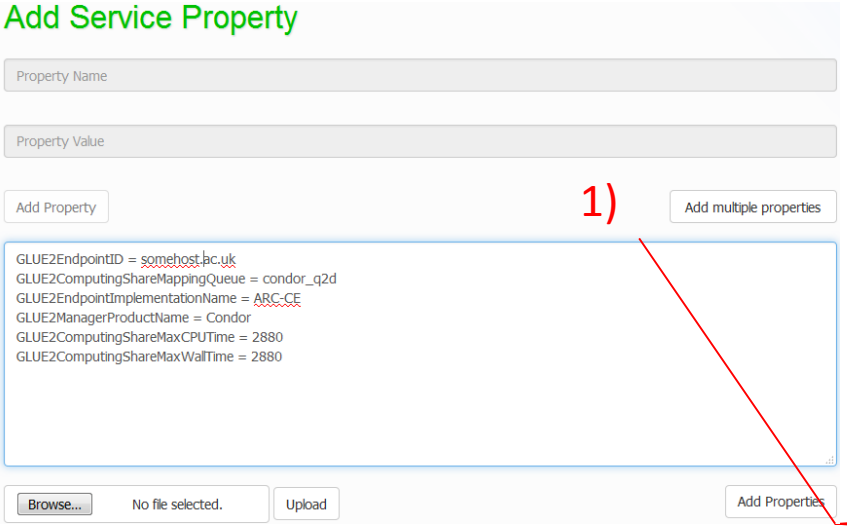
- Agree on a naming convention for the attributes (Generic name? GLUE1? GLUE2?)
- Agree on the possible values for type names (CE type and LRMS type. Shall we rely on GLUE Enumerated types?)
- Identify a set of volunteer sites to define the new attributes in GOCDB and OIM
- Using the web UI, use extension properties to define key/value pairs matching the list of attributes
    - GOCDB: a writeable API has been requested and is under discussion
- Ask AGIS to consume this information from GOCDB/OIM instead of the BDII

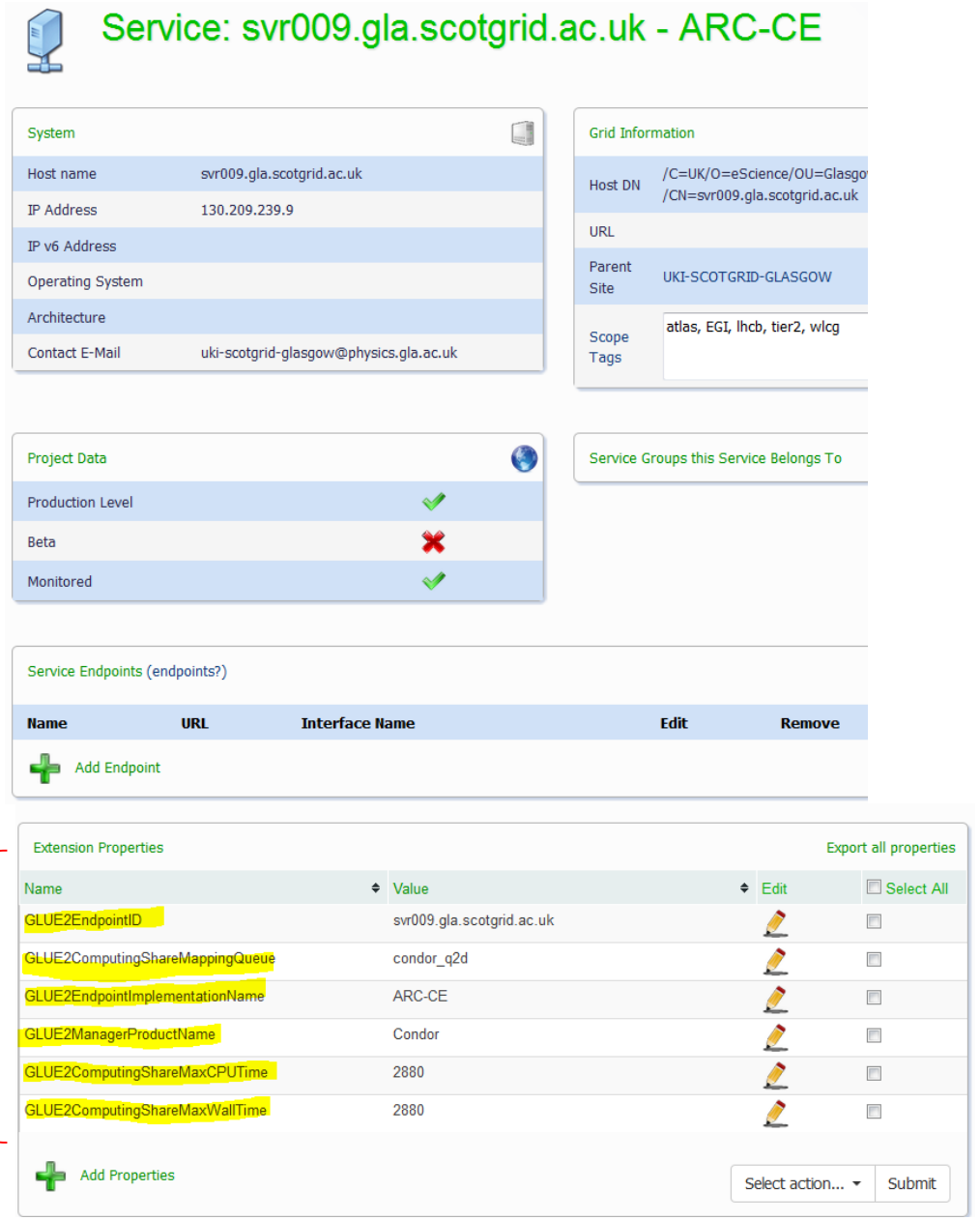
After this exercise is carried out, IS TF will be in a position to evaluate whether it's a feasible choice and plan for a wide deployment if necessary.

https://twiki.cern.ch/twiki/bin/view/EGEE/StopBDII#Computing

Propose we use Custom Properties to Prototype/Explore the Requirement First

Paste/upload multiple-property template file to service:





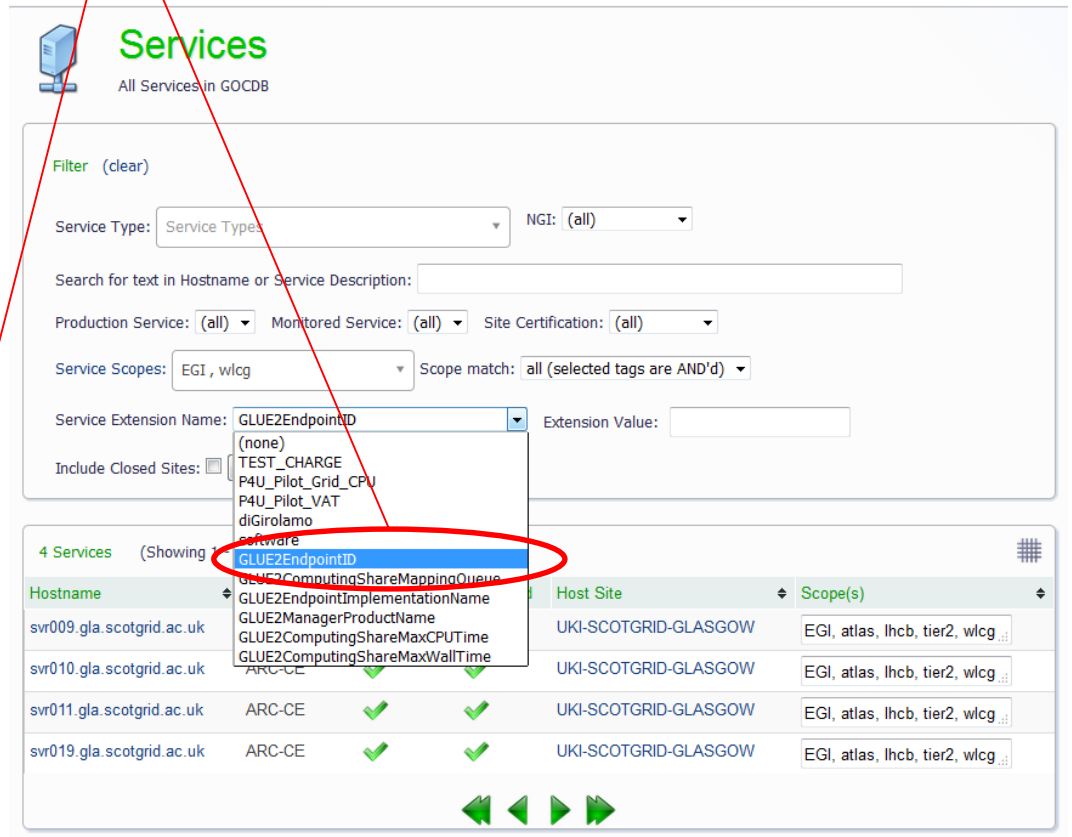
- WLCG could share the template file for all WLCG sites/services

view-source:https://goc.egi.eu/gocdbpi/public/?method=get_service_endpoint&sitename=UKI-SCOTGRID-GLASGOW

isited Getting Started Google Bookmark daveLinux - Google Do...

```
</SERVICE_ENDPOINT>
<SERVICE_ENDPOINT PRIMARY_KEY="2712G0">
  <PRIMARY_KEY>2712G0</PRIMARY_KEY>
  <HOSTNAME>svr009.gla.scotgrid.ac.uk</HOSTNAME>
  <GOCDB_PORTAL_URL>https://goc.egi.eu/portal/index.php?Page_Type=Service&amp;id=271
  <HOSTDN>/C=UK/O=eScience/OU=Glasgow/L=Compserv/CN=svr009.gla.scotgrid.ac.uk</HOSTD
  <BETA>N</BETA>
  <SERVICE_TYPE>ARC-CE</SERVICE_TYPE>
  <HOST_IP>130.209.239.9</HOST_IP>
  <CORE></CORE>
  <IN_PRODUCTION>Y</IN_PRODUCTION>
  <NODE_MONITORED>Y</NODE_MONITORED>
  <SITENAME>UKI-SCOTGRID-GLASGOW</SITENAME>
  <COUNTRY_NAME>United Kingdom</COUNTRY_NAME>
  <COUNTRY_CODE>GB</COUNTRY_CODE>
  <ROC_NAME>NGI_UK</ROC_NAME>
  <URL></URL>
  <ENDPOINTS/>
  <SCOPES>
    <SCOPE>EGI</SCOPE>
    <SCOPE>wlcg</SCOPE>
    <SCOPE>tier2</SCOPE>
    <SCOPE>atlas</SCOPE>
    <SCOPE>lhcb</SCOPE>
  </SCOPES>
  <EXTENSIONS>
    <EXTENSION>
      <LOCAL_ID>222</LOCAL_ID>
      <KEY>GLUE2EndpointID</KEY>
      <VALUE>svr009.gla.scotgrid.ac.uk</VALUE>
    </EXTENSION>
    <EXTENSION>
      <LOCAL_ID>223</LOCAL_ID>
      <KEY>GLUE2ComputingShareMappingQueue</KEY>
      <VALUE>condor_q2d</VALUE>
    </EXTENSION>
    <EXTENSION>
      <LOCAL_ID>224</LOCAL_ID>
      <KEY>GLUE2EndpointImplementationName</KEY>
      <VALUE>ARC-CE</VALUE>
    </EXTENSION>
    <EXTENSION>
      <LOCAL_ID>225</LOCAL_ID>
      <KEY>GLUE2ManagerProductName</KEY>
      <VALUE>Condor</VALUE>
    </EXTENSION>
    <EXTENSION>
      <LOCAL_ID>226</LOCAL_ID>
      <KEY>GLUE2ComputingShareMaxCPUTime</KEY>
      <VALUE>2880</VALUE>
    </EXTENSION>
```

Custom properties are then available for those services via the API and UI

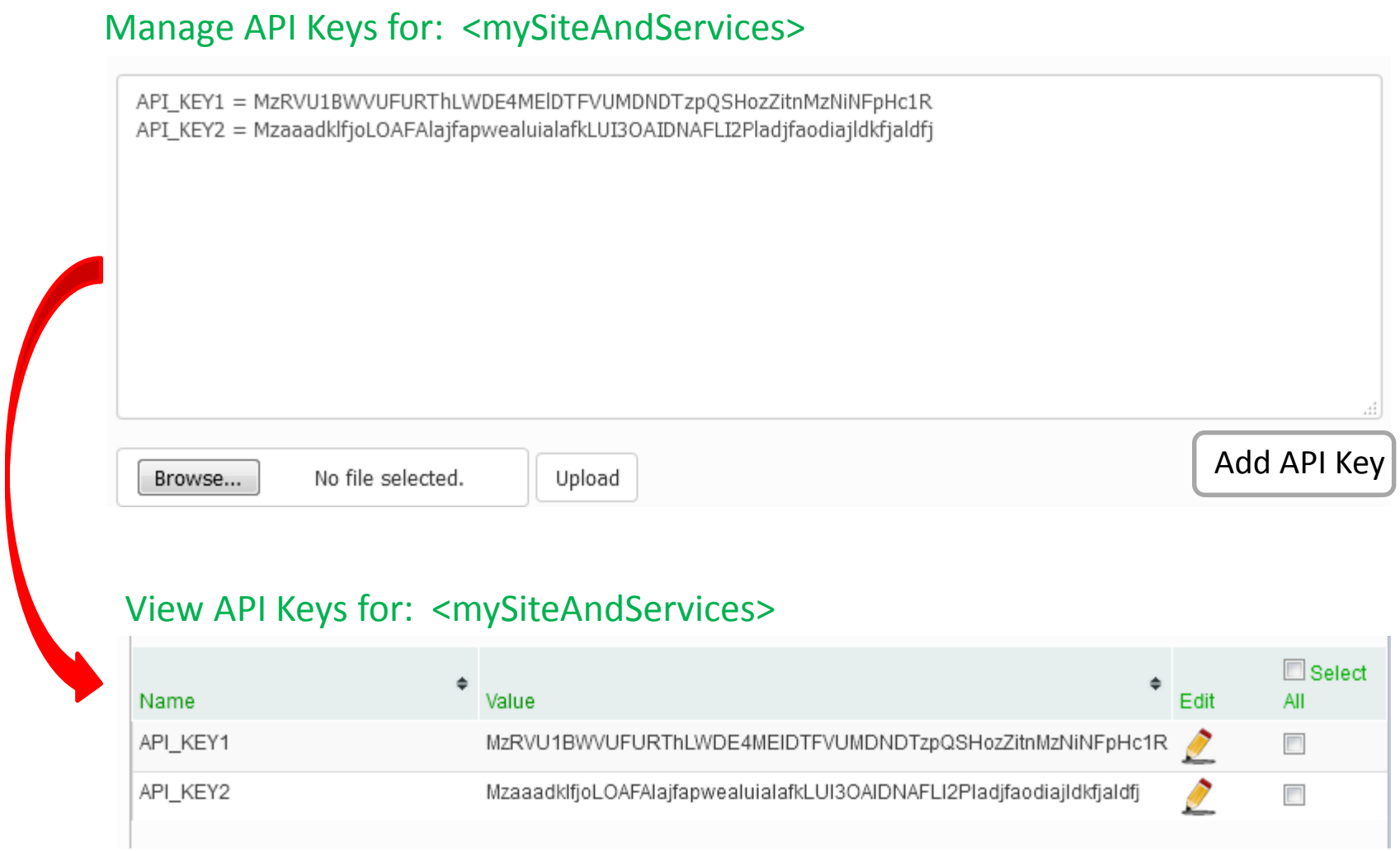


Write API to Update/Add Custom Properties

- Proposal 1: Add a write API methods to POST properties to a specified Site, Service, ServiceEndpoint, e.g.
  - '/gocdbpi/site/<siteId>/updateCustomProps'
  - '/gocdbpi/service/<serviceId>/updateCustomProps'
  - '/gocdbpi/serviceEndpoint/<servEndpId>/updateCustomProps'

- Proposal 2: To secure the methods, allow SiteAdmins to create/manage their own API Keys via the portal UI
  - Existing SiteAdmins create keys in portal UI.
  - API keys can be viewed/revoked in portal UI.

SiteAdmin creates API Keys for their resources (Site/Services):



Only users with permissions will be able to mange/view the API Keys in the UI

Sample Curl Client Access (not implemented yet!)

```
curl -X POST
https://goc.egi.eu/gocdbpi/service/1234/updateCustomProps
-H "Content-Type: application/json"
-d '{"GLUE2EndpointID":"some.endpoint.ac.uk",
"GLUE2ManagerProductName":"Condor"}'
-H "Authorization: Basic \
MzRVU1BWVUFURThLWDE4MElDTFVUMDNDTzpQSHozZitnMzNiNFpHc1R"
```

POST props

API Key passed as HTTP BASIC header

Server checks the API Key against the keys registered to that Site/Service.

Considerations

- Examples shown just for custom properties
- Could extend for actual Site/Service/Endpoint attributes later on (if required, will need to add new attribute types to Site/Service/Endp instead of relying on custom props)
- Increased server load – may need to scale up (to monitor/test)
- Timescales ~ 3mths for 1st MVP - Requires new developer to get up to speed (DM stepping down and providing oversight role only)