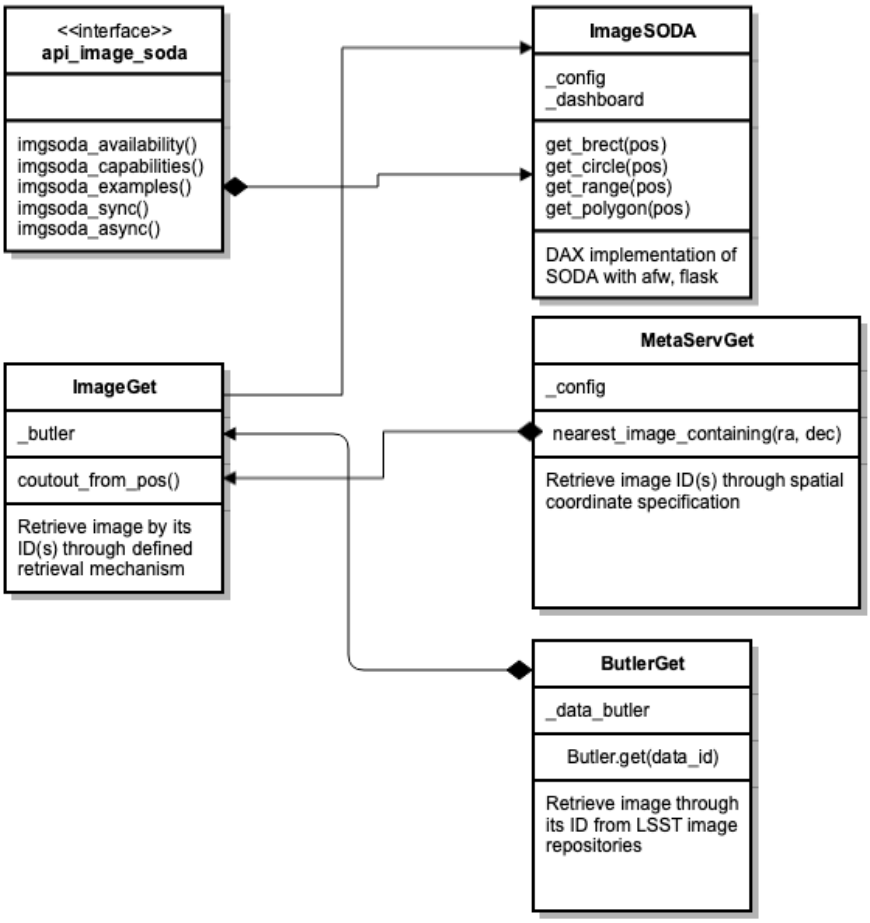
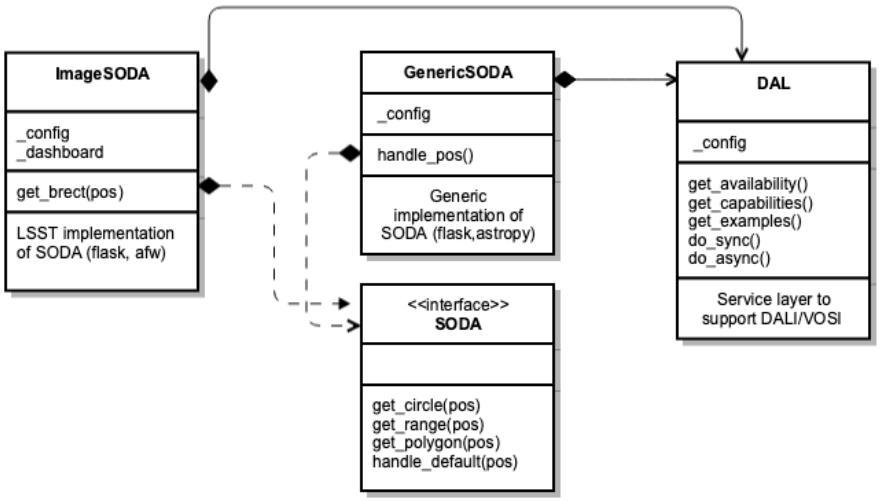
ImageServ API - SODA [Current]

The DAX implementation for SODA v1.1 has been deployed at lsst-lstp-int and lsst-lsp-stable.

ImageServ Class Diagrams

The following two diagrams describe the main classes and their interactions with each other in the ImageServ SODA design.





SODA Usage Examples

The coordinate system has to be ICRS with coordinates in degrees, per SODA v1.1 specification.

Note: for now ID is used as the image dataset identifier, but really with it should be the unique image ID, or the data_id per LSST parlance.

Example 1-3 for the standard SODA shapes (Note: URLs accessible within LSST-LSP cluster or through NCSA VPN ONLY):

Example 1: *CIRCLE <longitude> <latitude> <radius>*

https://lsst-lsp-stable.ncsa.illinois.edu/api/image/soda/sync?ID=DC_W13_Stripe82.calexp.r&POS=CIRCLE+37.644598+0.104625+100

Example 2: *RANGE <longitude1> <longitude2> <latitude1> <latitude2>*

https://lsst-lsp-stable.ncsa.illinois.edu/api/image/soda/sync?ID=DC_W13_Stripe82.calexp.r&POS=RANGE+37.616820222+37.67235778+0.07684722222+0.132402777

Example 3: *POLYGON <longitude1> <latitude1> ... (at least 3 pairs)*

https://lsst-lsp-stable.ncsa.illinois.edu/api/image/soda/sync?ID=DC_W13_Stripe82.calexp.r&POS=POLYGON+37.6580803+0.0897081+37.6580803+0.1217858+37.6186104+0.1006648

The following example is defined by LSST DAX only:

Example 4: *BRECT <longitude> <latitude> <width> <height> <size_unit>, size_unit = pixel | arcsec*

https://lsst-lsp-stable.ncsa.illinois.edu/api/image/soda/sync?ID=DC_W13_Stripe82.calexp.r&POS=BRECT+37.644598+0.104625+100+100+pixel

Generic Implementation of SODA

Historically VO services have been implemented in Java, in conjunction with XML as the linga franca, from the beginning for almost 20 years since.

Given the popularity of astropy and affiliated packages in the astronomy community, there is strong value proposition for Python based service implementation,

which can take advantage of popular and proven Python packages, such as pyvo, astroquery, astrocut, etc.

Hence ImgServ can implement this generic layer with the changes to the following modules:

a. ImageGet

The part in ImageGet should be replaced is usage of the DM Data Butler for retrieving desired image via its data ID, a.k.a. image ID. Specifically, the get() of the Butler API for image retrieval needs to be implemented per storage access of the underlying image dataset.

b. MetaServGet

MetaServGet should be replaced with querying protocol such as SIA or TAP/ADQL based on the TAP_Schema of the metadata , a.k.a ObsTAP, of the image dataset.

b. ImageGetter

ImageGetter defines a programming interface for performing image operations such as cutouts. It tied together ImageGet, MetaServGet, and the Butler. The geometric or image operations used in cutouts should be replaced with affiliated packages in astropy, such as Cutout2D, or emerging GitHub project astrocut by the Space Telescope Science Institute(STScI).