

Discussion on whether IVOA should take a role in defining a way to answer questions like:

Find where I can run <code> on <dataset>.

Involves description of systems from base OS machines up to application libraries. E.g. from a bare VM with only the OS available to an iPython notebook with predefined library modules.

When the description is in place a system based on a UWS machine plus a protocol specification can bring the code to the data, e.g. specifying a language to run the code.

An analogy with TAP services may look like:

TAP uses UWS asking ADQL jobs.
A TBD-code-protocol uses UWS asking for <some language (e.g. python)> coded jobs.

and the analogy may be pushed up to (like):

A TBD-protocol uses UWS letting Docker descriptions be loaded and returning an ssh endpoint.

Progress on this topic requires:

- gathering/surveying informations on the languages and technologies to be included as a minimal first step towards a model for resource description
- defining a registrable resource extension to let discovery be possible
- define a protocol (more protocols) to access those resources

Resource's type in an "Everything as a Service" scenario may go from a VM, a container, or other bare machine, through middleware and ready to use environments up to libraries and applications ready to launch.

A check on existing (if any) description models of such resources, useful for discovery, should also be done while defining the resource metadata.

IVOA WGs that may be interested in this are: GWS (mainly), DAL, Apps, Registry.