WP4 CEVO / WP5 ESAP

ExecutionPlanner data model

D.Morris
Institute for Astronomy,
Edinburgh University

ESCAPE WP4 TechForum
March 2022

The problem

Different types of executable task

Jupyter notebook

Docker container

Kubernetes deployment

Workflow task

Different types of execution platform

Jupyter hub

Binder hub

Rosetta

Portainer

Openstack

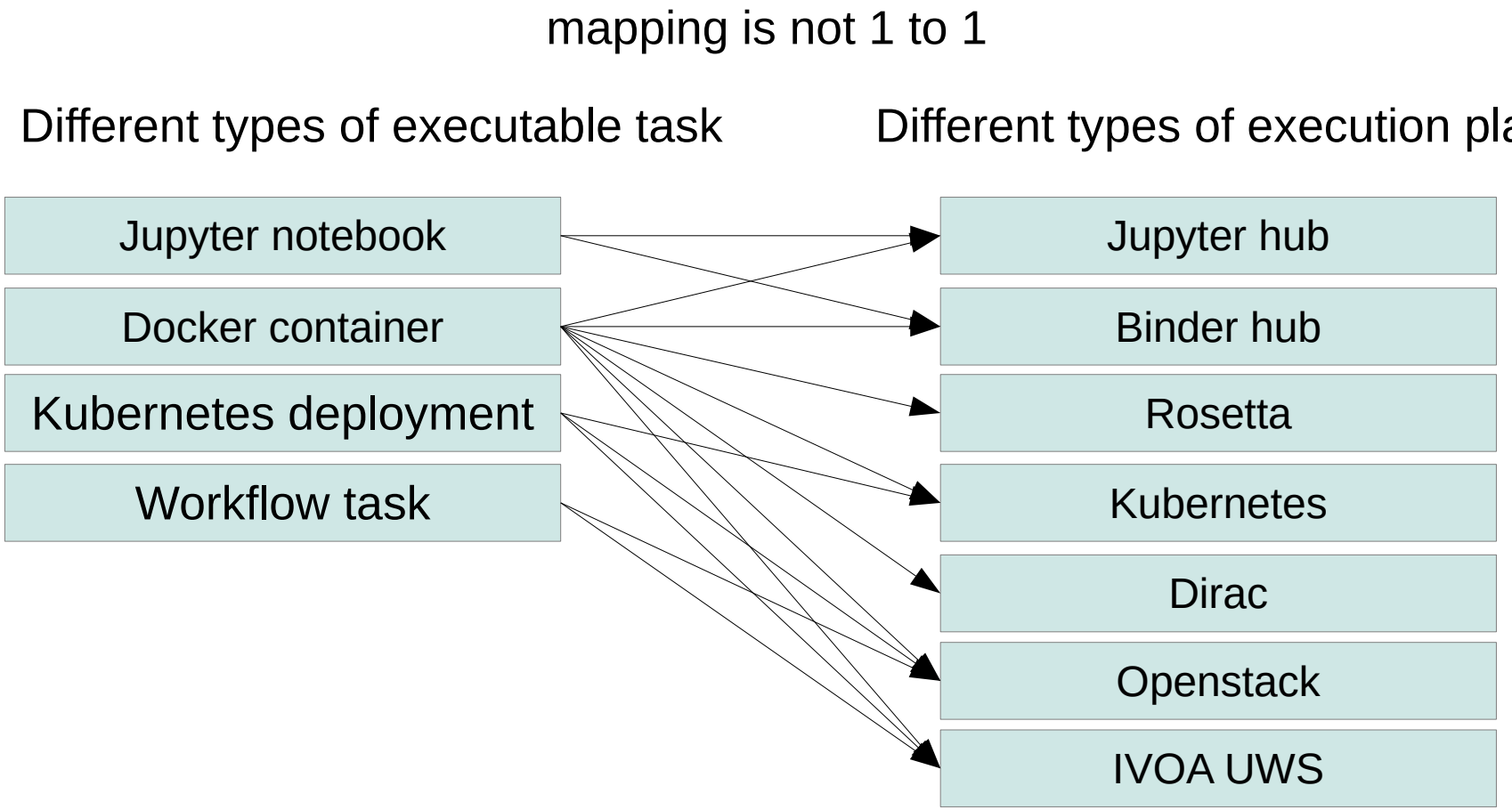
Dirac

IVOA UWS

D.Morris
Institute for Astronomy,
Edinburgh University

The problem

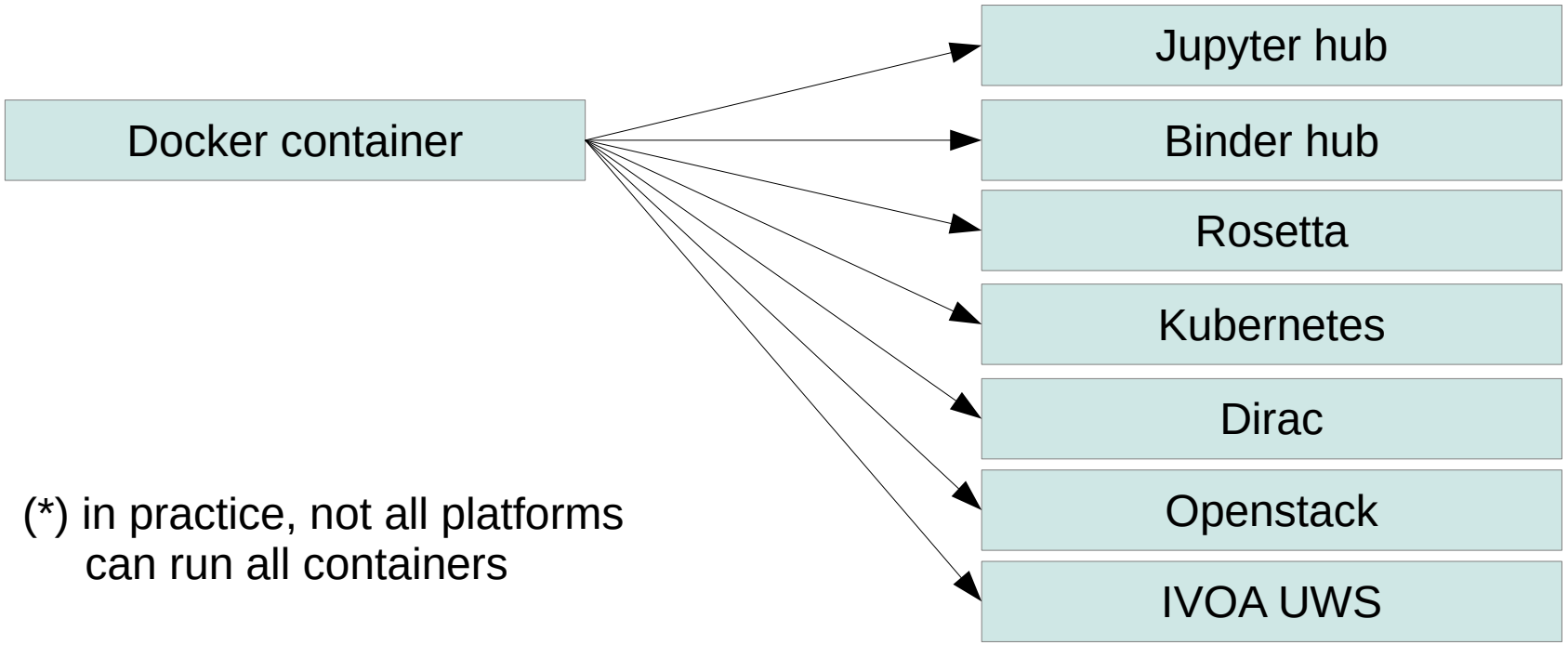
mapping is not 1 to 1



D.Morris
Institute for Astronomy,
Edinburgh University

The problem

In theory you can run a Docker container on all of these *

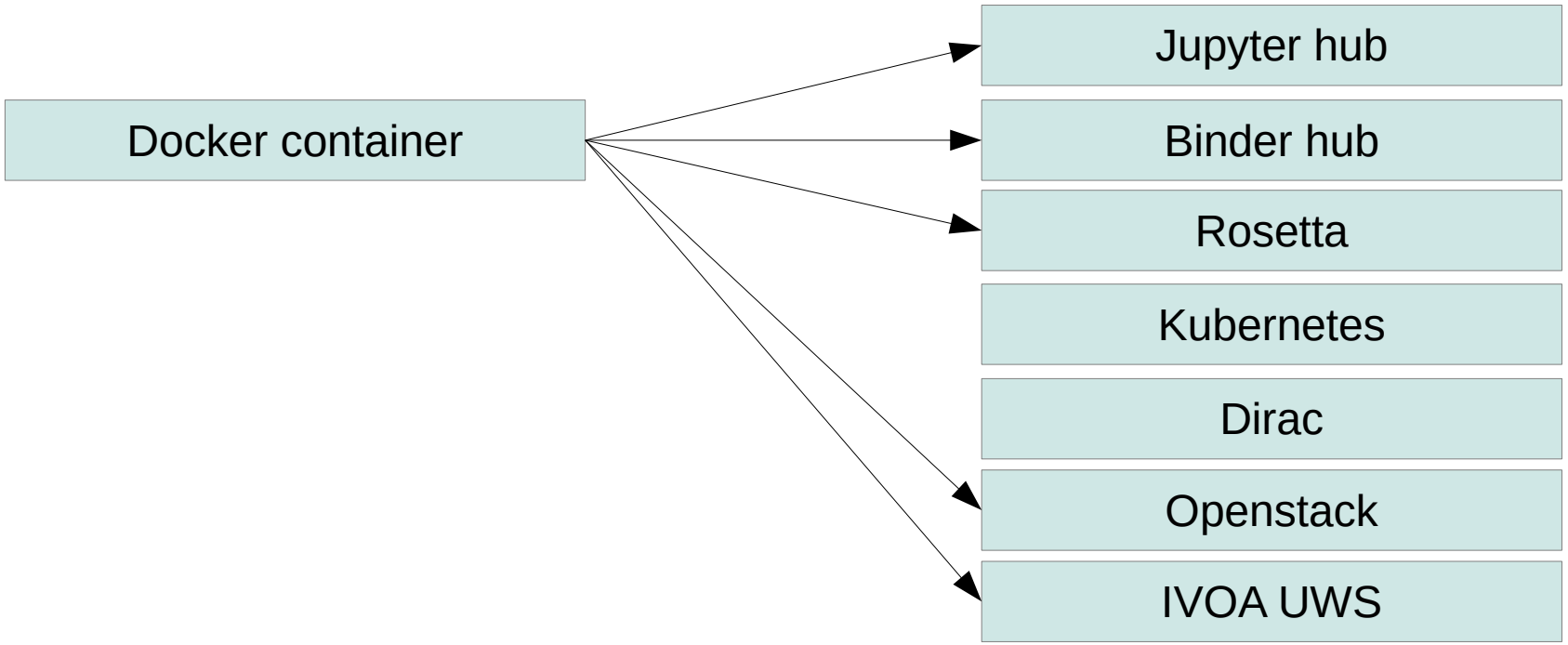


(*) in practice, not all platforms can run all containers

D.Morris
Institute for Astronomy,
Edinburgh University

The problem

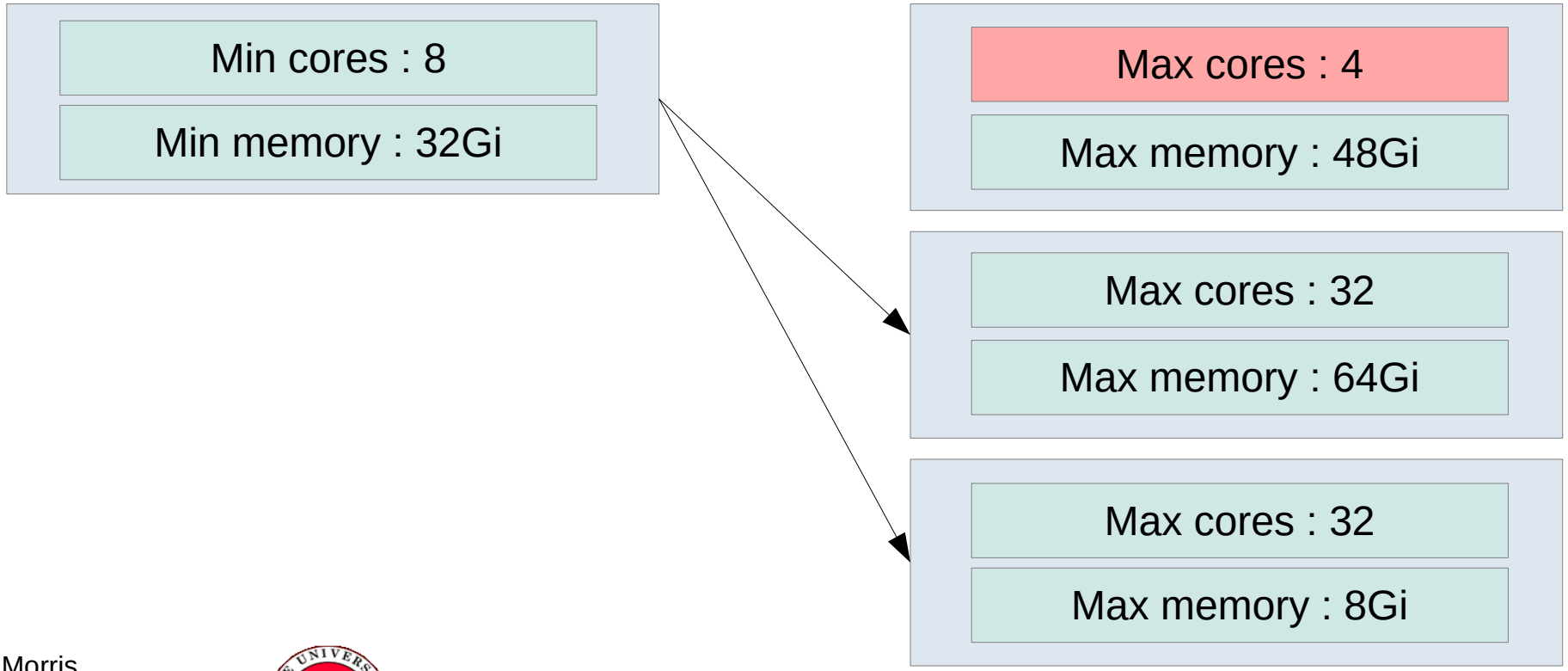
A specific container may only run on some platforms



D.Morris
Institute for Astronomy,
Edinburgh University

The problem

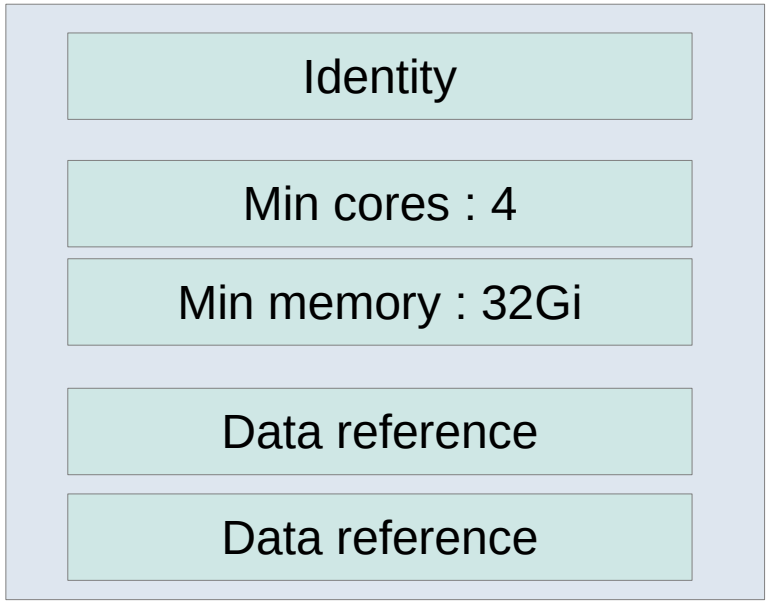
It gets more complicated if we add compute requirements

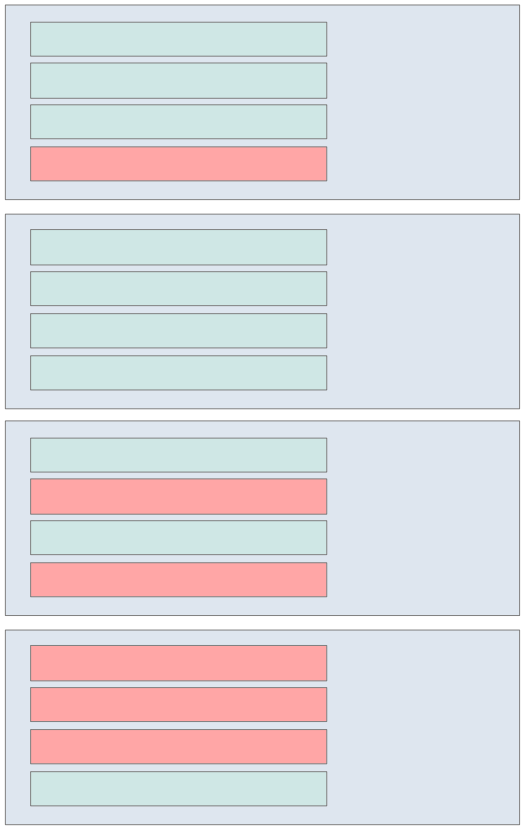


D.Morris
Institute for Astronomy,
Edinburgh University

The problem

It gets more complicated if we add data access and user identity





D.Morris
Institute for Astronomy,
Edinburgh University

Local solution

ESAP portal understands the capabilities of each type of platform

ESAP portal maintains a database of metadata about each platform

ESAP portal decides which platforms can be used to run a task

This works up to a point, but becomes complex to maintain as we add more requirements to the task description.

D.Morris
Institute for Astronomy,
Edinburgh University

Distributed solution

Each platform knows it's own capabilities
Each platform implements a simple webservice API

canIDoThis <task description>

ESAP portal queries each platform to see if it can run the task.

Implementing the decision is delegated to the platform providers.

Work in progress

IVOA ExecutionPlanner webservice specification

Join the discussion at the IVOA interop meeting in April

D.Morris
Institute for Astronomy,
Edinburgh University

Plugin solution

Each platform is handled by a plugin

Each plugin implements a simple Python API

canIDoThis <task description>

ESAP portal queries each plugin to see if it can run the task.

A simple plugin could use local metadata about the platform to make the decision.

A plugin for a more complex service could call the webservice API to ask the service it accepts the task. Delegating the decision to the service.

In both cases, the data model for the request and response are the same.

D.Morris
Institute for Astronomy,
Edinburgh University

Plugin benefits

Simple platforms have simple plugins

Simplest plugin just checks the task type and says yes or no.

canIDoThis <task description>

Plugin for a complex platform would use the web service interface to query the platform itself.

canIDoThis <task description>

In both cases, the Python interface and data model are the same, simplifying the code in the ESAP portal.

ESAP can start using simple plugins now

Evolve to using more complex plugins as they become available

This decouples ESAP development from the slower IVOA standards process.

D.Morris
Institute for Astronomy,
Edinburgh University

The data model enables a conversation between the user and the platform.

The process starts with metadata included in an OSSR record.

This metadata extends the existing code meta metadata to describe the executable components associated with an OSSR entry.

For example, if we have an OSSR record for a Python image processing library.

The OSSR record would contain the source code and documentation for the library.

Typically the OSSR record might also contain a couple of notebooks demonstrating how to use the library, and a Docker container with the library already installed.

These are the things that we want to run on a compute platform.

D.Morris
Institute for Astronomy,
Edinburgh University

Top of the data model is a list of executable components associated with an OSSR record.

```
executables : [

    {
    name: "Example 1"
    type: "purl:jupyter-notebook"
    description: "A Jupyter notebook demonstrating how to use the library for image processing"
    content: {
        -- notebook specific stuff --
        }
    },

    {
    name: "Demo container"
    type: "purl:docker-container"
    description: "A Docker container with the library installed and ready to use"
    content: {
        -- container specific stuff --
        }
    }
]
```

D.Morris
Institute for Astronomy,
Edinburgh University

Each item in the list gives us a human readable name and description, and a machine readable type.

```
{
name: "Demo container"
type: "purl:docker-container"
description: "A Docker container with the library installed and ready to use"
content: {
    -- container specific stuff --
    }
}
```

Enough to enable the platform to display a list of things that the user can run.

Example 1	A Jupyter notebook demonstrating …	[run]
Demo container	A Docker container with the library …	[run]

D.Morris
Institute for Astronomy,
Edinburgh University

The metadata for each item includes three blocks of requirements for data, storage and compute resources.

```
{
name: "Demo container"
type: "purl:docker-container"
description: "A Docker container with the library installed and ready to use"
data-resources: [
    ....
    ]
storage-resources: [
    ....
    ]
compute-resources: [
    ....
    ]
content: {
    -- container specific stuff --
    }
}
```

D.Morris
Institute for Astronomy,
Edinburgh University

The compute-resources section contains a list of metadata describing the compute resources needed to run the task.

```
{
name: "Demo container"
type: "purl:docker-container"
....
compute-resources: [
        {
        mincores: 4,
        minmemory: 8Gi,
        minstorage: 20Gi
        }
    ]
}
```

The list would normally contain a single entry with the minimum criteria needed to run the task.

The user would normally never need to see this information.

This information is used to select compute platforms that are able to meet the minimum requirements for the task.

D.Morris
Institute for Astronomy,
Edinburgh University

The data-resources section contains a list of the data resources needed to run the task.

```
{
name: "Demo container"
type: "purl:docker-container"
....
data-resources: [
        {
        name: “images”,
        description: “A collection of images to process”,
        cardinality: “multiple”,
        content-type: “purl:image”,
        required: true,
        location: “”
        }
    ]
}
```

This entry describes a collection of images needed to run the task, but it does not specify where to get the images from.

The combination of required=true and a null location indicates that the user needs to supply this information before they can run the task.

D.Morris
Institute for Astronomy,
Edinburgh University

The ESAP portal can use the data-resources to create a shopping list for the user, showing them what data they need to find before they can run the task.

```
{
name: "Demo container"
type: "purl:docker-container"
....
data-resources: [
        {
        name: “images”,
        description: “A collection of images to process”,
        cardinality: “multiple”,
        content-type: “purl:image”,
        required: true,
        location: “”
        }
    ]
}
```

images	A collection of images to process	[select]

D.Morris
Institute for Astronomy,
Edinburgh University

The user can use the ESAP portal to identify the images they want to use and add their location to the data resource.

```
{
name: "Demo container"
type: "purl:docker-container"
....
data-resources: [
        {
        name: “images”,
        description: “A collection of images to process”,
        cardinality: “multiple”,
        content-type: “purl:image”,
        required: true,
        location: “esap-basket:ca240519-1802-43e3-881d-cae6f47dc890”
        }
    ]
}
```

The user would not need to edit the raw JSON.

The data selection would be done using the ESAP portal interface.

D.Morris
Institute for Astronomy,
Edinburgh University

The storage-resources section contains a list of data storage needed by the task.

```
{
name: "Demo container"
type: "purl:docker-container"
....
storage-resources: [
        {
        name: “results”,
        description: “The processed images”,
        minsize: “20Gi”,
        minlifetime: “PT4H”,
        location: “”
        }
    ]
}
```

This entry describes a request for 20Gi byte of storage for the results that will persist for 4 hours after the task has completed.

The compute platform should allocate the storage and update the location to point to the results when the task has completed.

D.Morris
Institute for Astronomy,
Edinburgh University

The previous section should have given you an overview of the structure of the data model for an executable task.

Starting with the metadata template for the executable component associated with with an OSSR record.

Then conversation with the user to select the executable they want, display a shopping list of data they need to find, select the data they want to use and link it to the task.

At this point we have a completed data structure describing the executable task along with the compute, storage and data resources it requires.

The ESAP portal can now pass this data structure to each compute platform, asking if it is able to fulfill the requirements and run the task.

canIDoThis <task description>

D.Morris
Institute for Astronomy,
Edinburgh University

Each platform would respond with a yes/no boolean indicating if they are able to execute the task, and with an updated version of data model.

The simple form would just display a list of platforms that are able to execute the task and prompt the user to select one of them.

Platform a	Compute platform in Paris	[run]
Platform b	Compute platform in Munich	[run]

We can also use the updated version of the data model to make a more informed decision about which platform to select.

Platform a	Compute platform in Paris	[run]	[details]
Platform b	Compute platform in Munich	[run]	[details]

D.Morris
Institute for Astronomy,
Edinburgh University

For each set of minimum requirements in the request, the updated response from the canIdoThis() query may set the corresponding maximum values to indicate what the platform is able to offer.

```
{
name: "Demo container"
type: "purl:docker-container"
....
compute-resources: [
        {
        mincores: 4,
        maxcores: 8,
        minmemory: 8Gi,
        maxmemory: 16Gi,
        minstorage: 20Gi
        maxstorage: 80Gi
        }
    ]
}
```

In this example, the is offering to provide twice the requested cpu cores, memory and storage space.

D.Morris
Institute for Astronomy,
Edinburgh University

This kind of over provisioning can happen on something like an Openstack platform where the machines are only available in fixed sizes.

The user interface may use this additional information to prioritise platforms that are able to offer us more resources.

Platform a	Compute platform in Paris	[run]	8 cpu,16Gi mem
Platform b	Compute platform in Munich	[run]	4 cpu,8Gi mem

D.Morris
Institute for Astronomy,
Edinburgh University

Why have we spent 20+ slides describing a data model for ESAP WP5 in a CEVO WP4 meeting ?

This talk was supposed to be about the IVOA ExecutionPlanner webservice developed as part of CEVO WP4.

The reason is we are using the use cases set by the ESCAPE participants to develop a common data model for all three applications.

If we can use the same data model for the metadata in the OSSR records, the ESAP plugins and the CEVO ExecutionPlanner interface, then they will all be interoperable.

D.Morris
Institute for Astronomy,
Edinburgh University

If the platform plugins in ESAP and the IVOA ExecutionPlanner service use the same data model, then ESAP will be able to use both local and remote compute platforms interchangeably.

If the OSSR metadata and the IVOA ExecutionPlanner service use the same data model, then it should be possible to select an executable from the OSSR and run it on a compute platform in Canada.

If the data model is the same, then it should be possible to take a Docker container developed for a Japanese instrument and run it on a compute platform hosted by an ESCAPE partner in France.

D.Morris
Institute for Astronomy,
Edinburgh University