ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

IVOA ExecutionPlanner updates

IVOA interop, May 2023

Dave Morris, Edinburgh University

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

The problem – different science platforms use different technologies

We end up having to understand all of them.

jupyter
binder
Apache Zeppelin
DIRAC
THE INTERWARE
EUROPEAN OPEN SCIENCE CLOUD
docker
openstack.
kubernetes

Which becomes more complex as the questions get more detailed.

Simple interface that we can all implement

CanIDoThis?

```
POST
  task = { …. }
```

Response

```
YES
    details = { …. }


NO
    details = { …. }
```

Learn from TAP/ADQL and separate the API from the content

CanIDoThis?

```
POST
    lang = IVOA-TDL-1.0
    task = { …. }
```

If the service doesn’t understand the task description language

It can just say NO.

This allows early prototypes without having to agree the full language

(*) It may never be needed, but flexibility is good
(*) I don’t know of any TAP services that use alternative languages

D.Morris
Institute for Astronomy,
Edinburgh University

Part of the task description can include the resources needed.

```
{
  ....
  "data-resources": [
      ....
      ]
  "compute-resources": [
      ....
     ]
  "storage-resources": [
      ....
     ]
 }
```

Everything has a minimum and maximum value

The client specifies the minimum needed to get the task done

```
"compute-resources": [
          ....
          mincores: 4
          minmemory: 80G

         ]
```

The client specifies the minimum needed to get the task done

The service responds with the maximum it can offer

"compute-resources": [

....

mincores: 4

maxcores: 8

minmemory: 80G

maxmemory: 120G

]

Part of the task description can include date/time ranges.

```
minduration: 2hr

daterange: {
    minstartdate: 09 May 2023
    maxenddate: 11 May 2023
    }

timerange: {
    minstarttime: 9:00
    maxendtime:  17:00
    }
```

Service responds with one or more offers.

Each offer includes a specific time within the requested ranges.

```
offer: {

    expires: 9 May 15:30

    startdate: 11 May 2023
    starttime: 10:00

    minduration: 2hr
    maxduration: 4hr

    }
```

A service may return several offers in the same response.

It is up to the client to select the best one.

```
10 May 10:00 => 4 hours, 4 cores, 80G memory

10 May 16:00 => 1 hours, 12 cores, 128G memory
```

If you extend the ranges you may be offered a larger allocation.

```
13 May 02:00 => 24 hours, 56 cores, 512G memory
```

Add a new API to run the task- ExecRunner

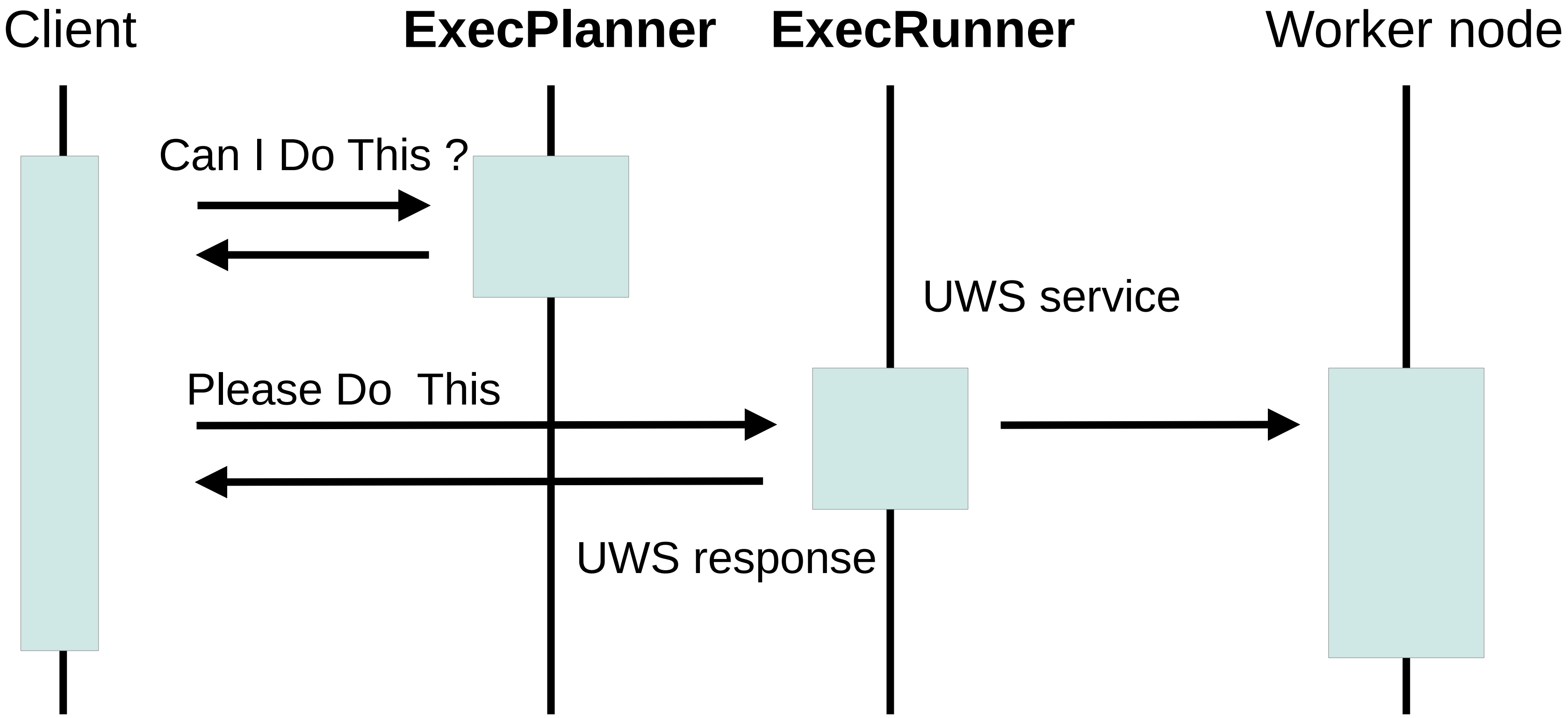
The ExecRunner can be a simple extension of UWS.

The ExecPlanner offer includes the ExecRunner endpoint.

Pass an offer from ExecPlanner to ExecRunner and ask it to run it.

```
PleaseRunThis <task>
```

The ExecRunner API hides the implementation details of the worker.



Client

ExecPlanner

ExecRunner

Worker node

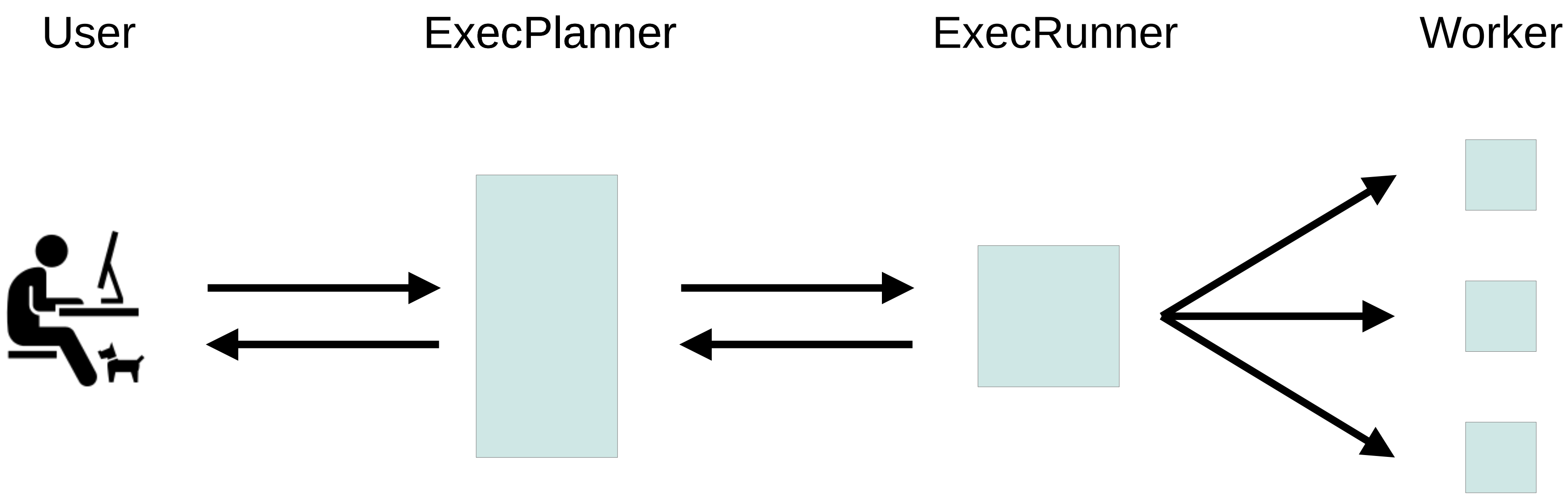
Can I Do This ?

UWS service

Please Do This

UWS response

Some projects may choose to have a single ExecPlanner and a single ExecRunner service.



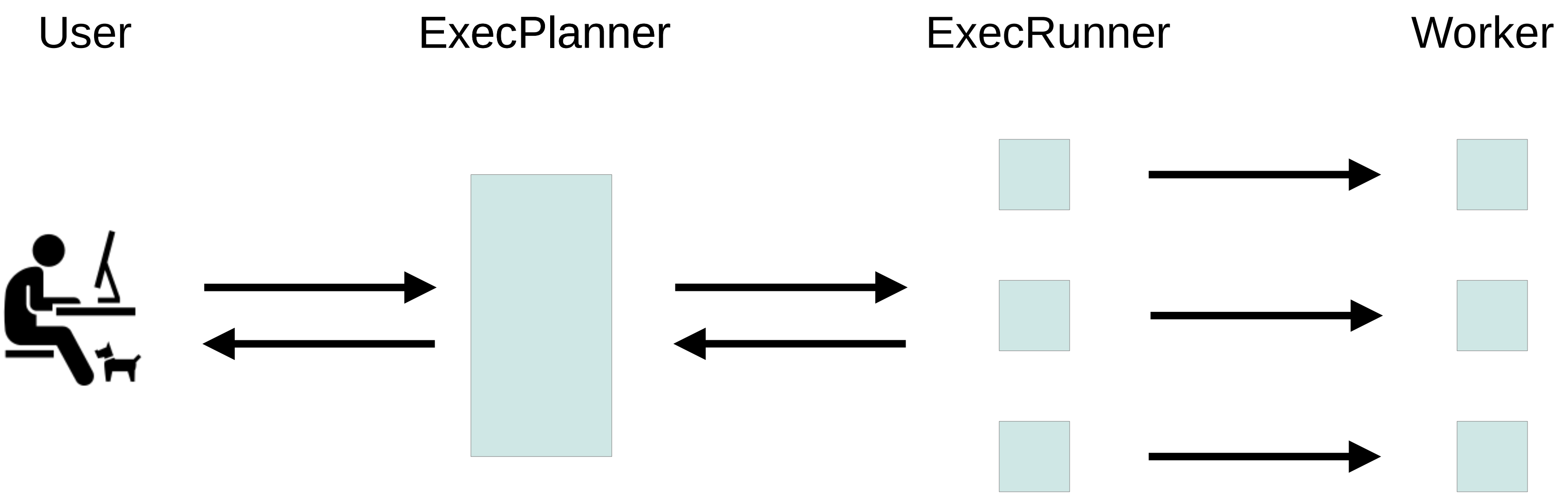
User

ExecPlanner

ExecRunner

Worker

Some projects may choose to have a single ExecPlanner and multiple ExecRunner services.



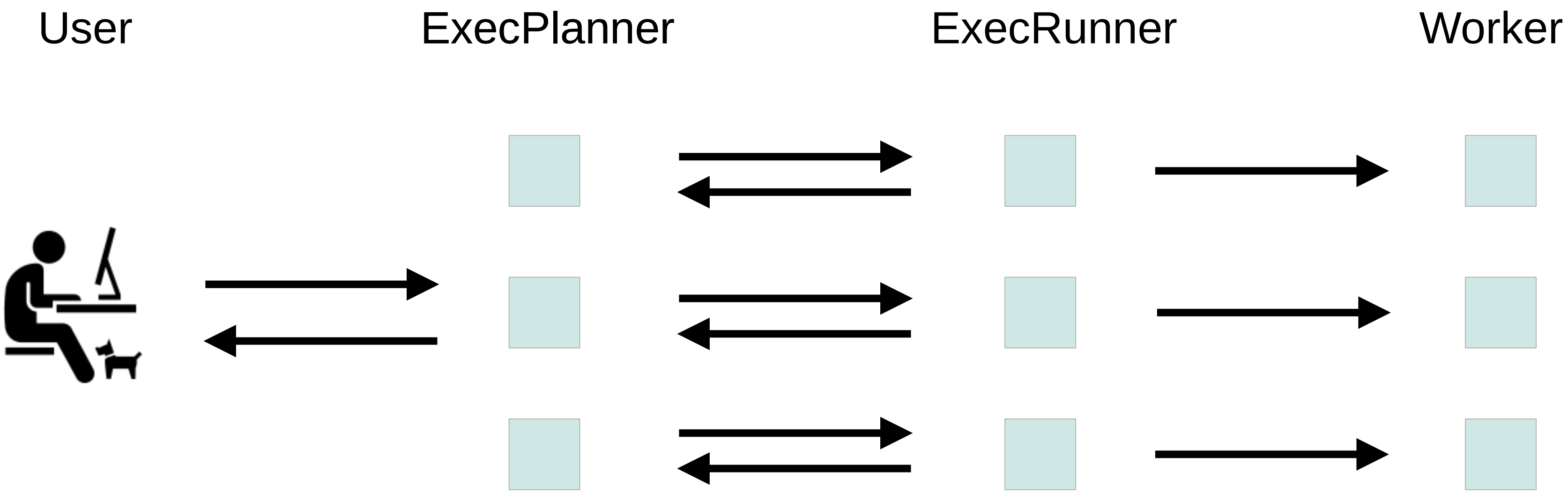
User

ExecPlanner

ExecRunner

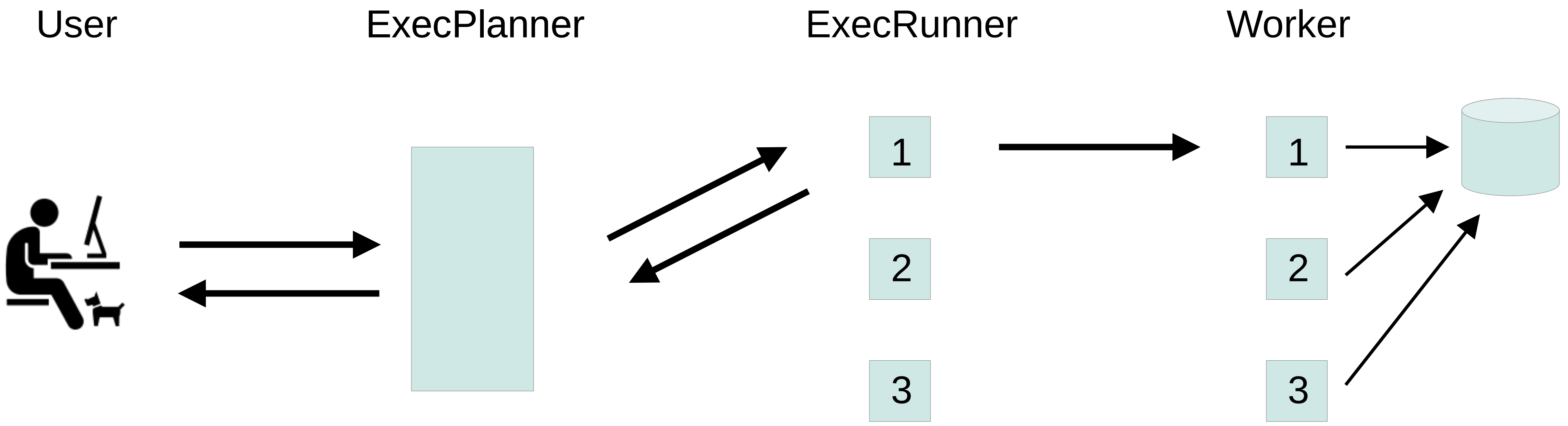
Worker

Some projects may choose to have multiple ExecPlanner and multiple ExecRunner services.



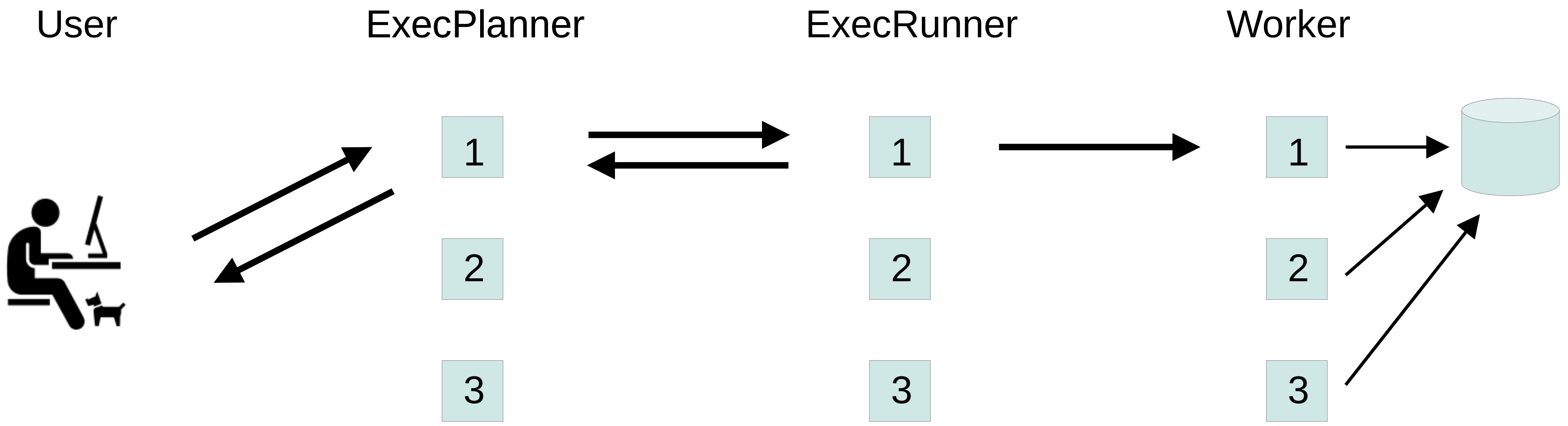
D.Morris
Institute for Astronomy,
Edinburgh University

If you have a metric for data proximity, the ExecPlanner can use it to choose which ExecRunner to use.



D.Morris
Institute for Astronomy,
Edinburgh University

If you have a metric for data proximity, the ExecPlanner can pass it back to the client, allowing the client to choose.



Any questions

dmr@roe.ac.uk