SRCNet ExecutionBroker prototype

Dave Morris, Bob Watson

SRCNet team Coral

SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

SRC | Net
SKAO Regional Centre Network

Where we started :

- 61 page document
- lots of theory
- no code

International
Virtual
Observatory
Alliance

IVOA meeting in May :

** name change **

~~IVOA ExecutionPlanner~~

IVOA ExecutionBroker

- slightly slimmer document
- still theoretical
- no code

IVOA Execution Broker

Version 1.0

IVOA Working Draft 2024-04-25

Working Group
GWS
This version
https://www.ivoa.net/documents/ExecutionBroker/20240425
Latest version
https://www.ivoa.net/documents/ExecutionBroker

https://github.com/ivoa-std/ExecutionBroker

SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Where we started :

2 separate web service endpoints

ExecutionPlanner

Resource booking and job scheduling

ExecutionWorker

In theory similar to UWS/CEA

Original document referred to a ‘UWS like’ service

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

What we have now:

Single web service endpoint

ExecutionBroker

Resource booking and job scheduling

~~**ExecutionWorker**~~

No longer required

Job control is automatic, based on time schedule

SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

OpenAPI

https://www.openapis.org/
https://swagger.io/specification/

IVOA P3T team developing the next generation of IVOA service specifications

Using OpenAPI to define the WebService API and data model.

This will effect the next versions of UWS, TAP and DataLink.

ExecutionBroker selected as a pathfinder for this process.

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

OpenAPI

https://www.openapis.org/
https://swagger.io/specification/

- Can we use OpenAPI to describe the web service ?
- Can we use OpenAPI to generate the code ?
- Does it help ?

SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

OpenAPI

- Can we use OpenAPI to describe the web service ?

  Yes. Simple language syntax, easy to learn and easy to understand.


```yaml
paths:
  /offerset:
    post:
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/OfferSetRequest'
          application/yaml:
            schema:
              $ref: '#/components/schemas/OfferSetRequest'
      responses:
        "200":
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/OfferSetResponse'
            application/yaml:
              schema:
                $ref: '#/components/schemas/OfferSetResponse'
```


Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

OpenAPI

- Can we use OpenAPI to generate the code ?

  Depends on the toolset.

  Java/Spring toolset is very good.
  Generates a full set of classes for the web service interface.

  Python/FastAPI toolset is patchy.
  Small simple messages are fine.
  Has problems with complex data structures, e.g. polymorphism.

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

OpenAPI

- Can we use OpenAPI to generate the code ?

  Depends on the toolset.

- **Human -> OpenAPI : good**

  Simple language syntax, easy to learn

- **OpenAPI -> Human : good**

  Simple language syntax, easy to understand

- **OpenAPI -> Java/Spring : good**

  Generated classes work ‘out of the box’, no modifications needed

- **OpenAPI -> Python/FastAPI : not good**

  Generated classes need a LOT of modifications

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

OpenAPI

- Can we use OpenAPI to generate the code ?

  Depends on the toolset.

- **OpenAPI -> Python/FastAPI : not good**

  Generated classes need a LOT of modifications

- **OpenAPI -> ChatGPT -> Python/FastAPI : better**

  Still needs a lot of modifications, but the starting code is better

  Not reproducable

  Getting it right involves a conversation with ChatGPT correcting errors and omissions

  Modify the OpenAPI, start the conversation again.

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

WebService API

Request for offers : POST to `/offerset`

Request contains 5 sections

```
executable:
  ....
schedule:
  ....
resources:
  compute:
    ....
  data:
    ....
  storage:
    ....
```

Creates a new set of offers on the server

Response is either a 303 redirect or a
200 response with in-line data

OpenAPI can represent both forms

Service implementation can swap between them

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

WebService API

Response with offers : GET to `/offerset/{uuid}`

Response contains YES/NO result and a list of offers


```
uuid: ....
result: YES
href: http://..../offerset/{uuid}
offers:
- uuid: ....
  state: "OFFERED"
  href: http://..../execution/{uuid}
  ....
  executable:
    ....
  schedule:
    ....
  resources:
```


Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Serialization format

Support for request and reply messages in JSON and YAML

Same code base, different serialiser/deserialiser

YAML is easier for Human clients


```
uuid: ....
result: YES
href: http://..../offerset/{uuid}
offers:
- uuid: ....
  state: "OFFERED"
  href: http://..../execution/{uuid}
  ....
  executable:
    ....
  schedule:
    ....
  resources:
```


JSON is easier for Python clients.


```
{
"uuid": "....",
"result": "YES",
"href": "http://..../offerset/{uuid}",
"offers": [
  {
  "uuid": "....",
  "state": "OFFERED",
  "href": "http://..../execution/{uuid}",
    ....
  "executable": {
    ....
    },
  "schedule": {
    ....
    },
  "resources": {
    ....
    },
  ....
  },
}
```


SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk

Bob Watson
bob.watson@manchester.ac.uk

Scheduling

Client can specify the start time and the duration

```
executable:
  type: urn:jupyter-notebook-0.1
  notebook: http://github.com/....
schedule:
  requested:
    executing:
      start: "2024-08-29T10:00ZPT1H"
      duration: "PT4H"
```

Request to run a JupyterNotebook,

- start time between 10:00 and 11:00
- session duration 4hrs

(*) Prototype code ignores the start range.

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

Schedule is optional, server will fill in with defaults:

```
executable:
  type: urn:jupyter-notebook-0.1
  notebook: http://github.com/....
schedule:
  # blank
```

Default schedule

- start time between now and +24hr
- session duration 30min

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Compute resources

Client can specify the number and size of compute resources

```
executable:
  properties:
  type: urn:jupyter-notebook-0.1
  notebook: http://github.com/....
resources:
  compute:
    - type: "urn:simple-compute-resource"
      cores:
        min: 4
      memory:
        min: 8
```


Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Compute resources

Compute resources are optional, server will fill in with defaults

```
executable:
  properties:
  type: urn:jupyter-notebook-0.1
  notebook: http://github.com/....
resources:
  compute:
    # blank
```

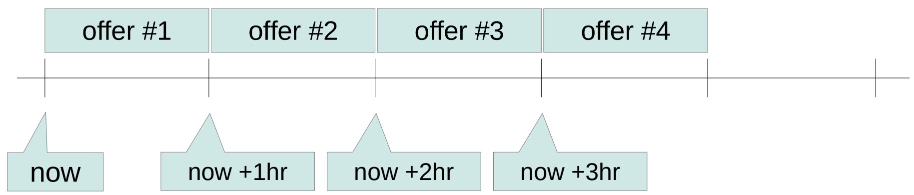
Default compute

- 1 compute node, 2 cores, 2GiB memory

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

Scheduling algorithm tries to find the best fit given the available resources
With no conflicts it will offer a sequence of consecutive start times

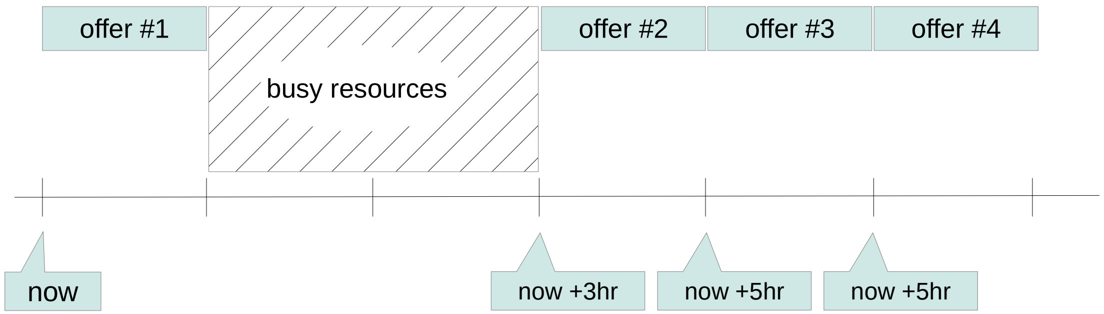


SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

If the database already has allocated blocks,
scheduling algorithm will try to fit around them

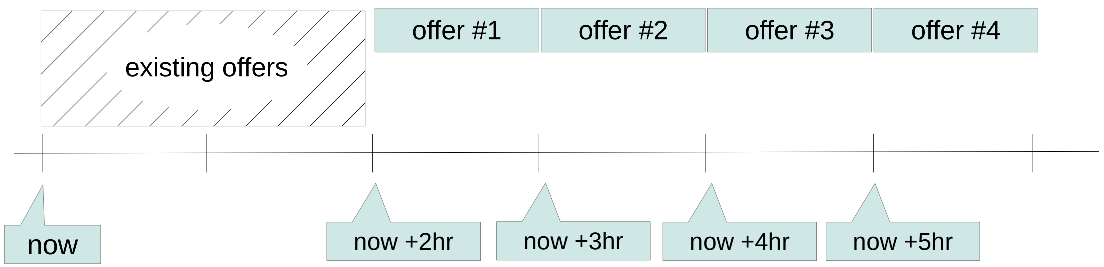


SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

If the resources are tied up in offers, new offers will be pushed forward to the next available slot

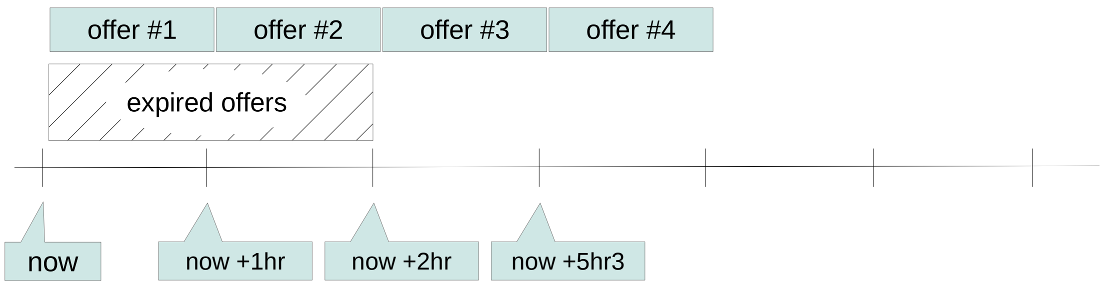


SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

Unused offers will automatically expire, releasing the resources back to the pool

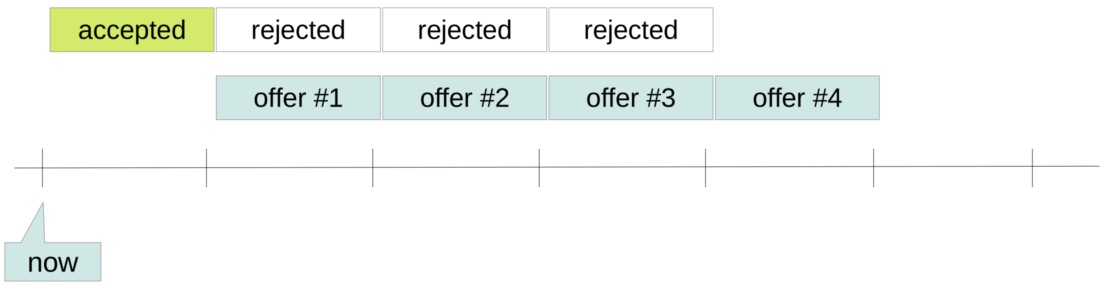


SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

Accepting one of the offers will automatically reject the others, releasing the resources back to the pool

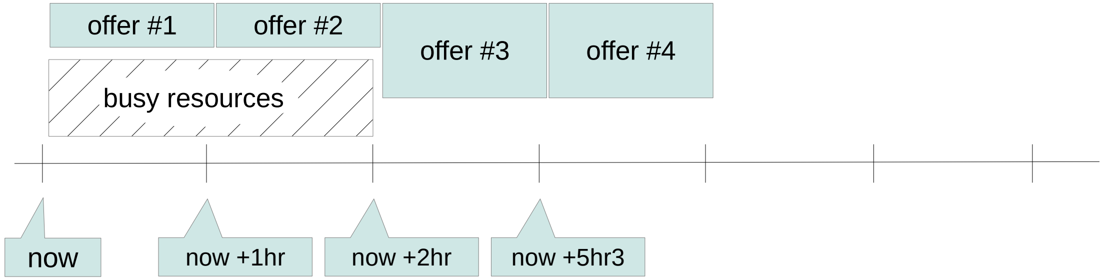


SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

The algorithm will try to offer twice the requested resources if they are available



Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Scheduling

Current algorithm tries to maximize start time, cores, memory and then duration.

Future work can look at refining the algorithm

Alternative algorithms can optimize for memory/core or duration.

Possible to include one offer from each algorithm in the same response

- Earliest
- Largest
- Longest
- etc ..

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

SRC Net
SKAO Regional Centre Network

IVOA

Data resources

Client specifies where to fetch a data from and links it to a volume on a compute resource

```
executable:
  properties:
  type: urn:jupyter-notebook-0.1
  notebook: http://github.com/....
resources:
  compute:
    - name: Compute 001
      type: urn:simple-compute-resource
      volumes:
        - name: Volume 001
          path: /my-data
          resource: Resource 001
  data:
    - name: Resource 001
      type: urn:simple-data-resource
      location: http://data.example.org/downloads/data-001
```

Current prototype

- code for parsing and validating data resources and linking them to volumes is implemented
- code for downloading data resources is not implemented yet

SRCNet demo
August 2024

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk

Still to do

Known unknowns

We have working Python code snippets for all of these operations

CANFAR prototype can use the VOSpace API to transfer the notebook into the user’s home directory

CANFAR prototype can use the VOSpace API to transfer the data into the user’s home directory

CANFAR prototype can use the Skaha API to launch notebook session using generic notebook container

IVOA standard document

We will not be able to complete this part of the work this iteration
New feature proposed for next iteration to cover this

Dave Morris
dave.morris@manchester.ac.uk

Bob Watson
bob.watson@manchester.ac.uk

Still to do

UnKnown unknowns

We have working solutions for all of the outstanding operations

The goal of building a prototype was to resolve the unknown unknowns
and verify that the web service API and data model met the requirements

Dave Morris
dave.morris@manchester.ac.uk
Bob Watson
bob.watson@manchester.ac.uk