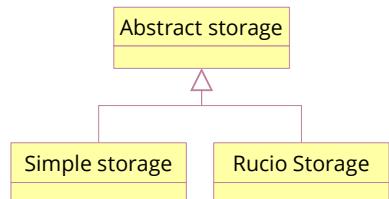




# Using OpenAPI for IVOA standards

## Lessons learned



Dave Morris  
Manchester  
University



Astro-CC meeting  
Trieste,  
October 2025



Dave Morris  
[dave.morris@manchester.ac.uk](mailto:dave.morris@manchester.ac.uk)



## GWS working group

Developing a new standard for remote execution of software.

Moving the code to the data.



*International  
Virtual  
Observatory  
Alliance*

### IVOA Execution Broker

Version 1.0

IVOA Working Draft 2024-11-15

Working Group

GWS

This version

<https://www.ivoa.net/documents/ExecutionBroker/20241115>

Latest version

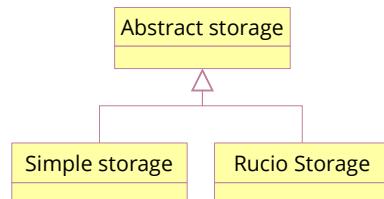
<https://www.ivoa.net/documents/ExecutionBroker>



New standard, new document structure.

*“This document explains the reasoning behind the design and uses examples to describe the service behavior.”*

*“The technical details of the data model and web-service API are defined in the OpenAPI specification published alongside this document.”*

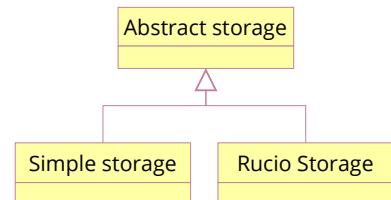
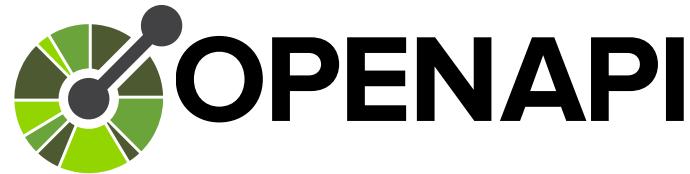


The image contains three main components:

- IVOA Logo:** A circular logo with a white sphere and a grid, surrounded by the text "IVOA".
- IVOA Execution Broker Version 1.0:** A document cover with the title "IVOA Execution Broker Version 1.0" and "IVOA Working Draft 2018". It includes sections for "Working Group GWS" and "This version".
- OpenAPI Specification:** A snippet of the OpenAPI JSON specification. It starts with the "openapi: 3.1.0" header and includes sections for "info", "license", "paths", and a detailed "post" request definition for the "/offersets" endpoint.

```
openapi: 3.1.0
info:
  title: IVOA Execution Broker
  version: "1.0"
  description: >
    IVOA Execution Broker web service
  license:
    Name: >
      Creative Commons Attribution
      Share Alike 4.0 International
    identifier: CC-BY-SA-4.0
paths:
  /offersets:
    post:
      requestBody:
        content:
          application/json:
            schema:
              $ref: 'OfferSetRequest'
          application/yaml:
            schema:
              $ref: 'OfferSetRequest'
        required: true
```

Using OpenAPI to specify the data model and web service API.



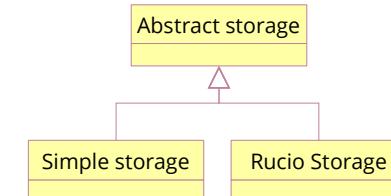
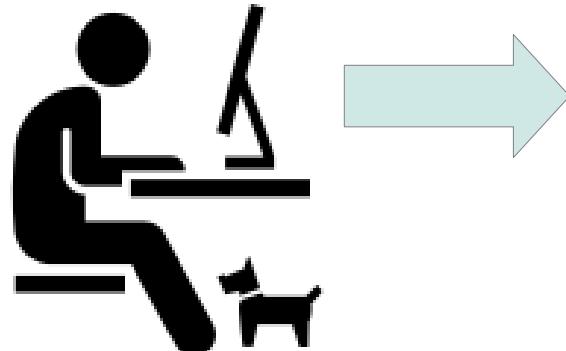
What worked

What didn't work

Would I use it again

# What worked

Using OpenAPI to describe the data model and service API

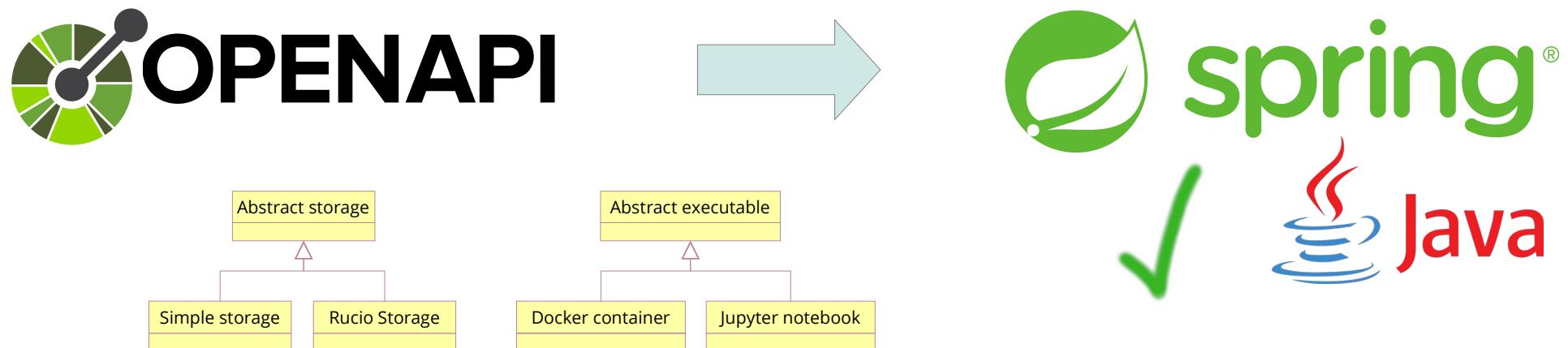


- Shallow learning curve
- Good documentation
- Clear and easy syntax
- Good feature coverage

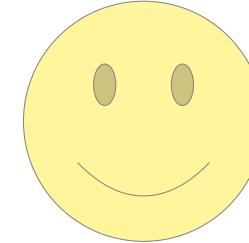
# What worked

Generating Java service code from the OpenAPI specification

Including support for polymorphic types in the message content.

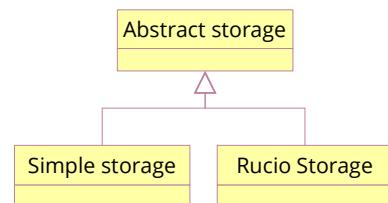
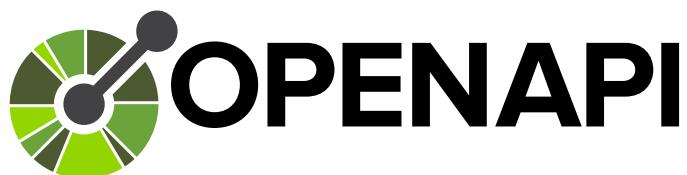


# What ~~didn't work~~ works



Generating Python service code from the OpenAPI specification

Including support for polymorphic types in the message content.



Content-type:

Accept:

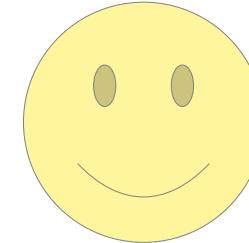


**Improvements to the code generators in 2025 mean this is no longer an issue**

(\*) Generated Python code supports JSON only, YAML and XML are not supported

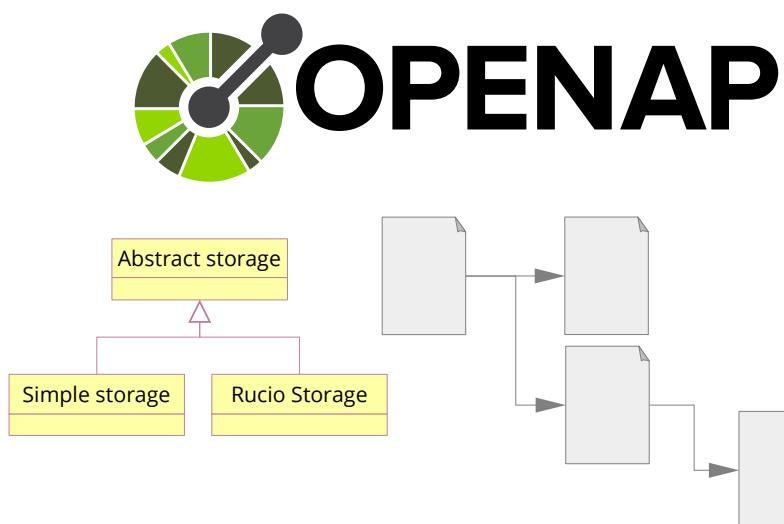
Dave Morris  
[dave.morris@manchester.ac.uk](mailto:dave.morris@manchester.ac.uk)

# What didn't work works



Splitting the OpenAPI specification into separate files.

Solved using a pre-processing tool to resolve \$ref links

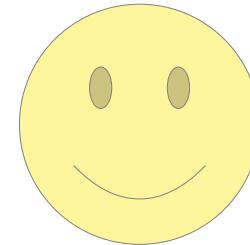


Pre-processor resolves \$ref links and puts everything into one large YAML file

<https://github.com/ivoa/Calycope-Isobeon>

Dave Morris  
dave.morris@manchester.ac.uk

# What works



Interoperable Python client and Java server  
generated from the OpenAPI specification



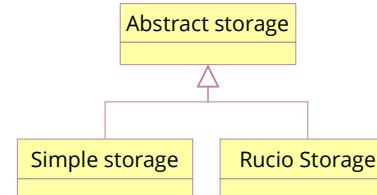
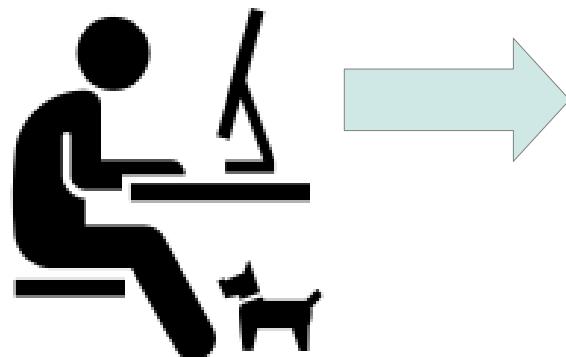
(\*) Generated Python code supports JSON only, YAML and XML are not supported

Dave Morris  
[dave.morris@manchester.ac.uk](mailto:dave.morris@manchester.ac.uk)

# Would I use it again ? YES

Using a structured schema to define the service API is a huge benefit.

Writing clear and precise technical specifications in text is hard.



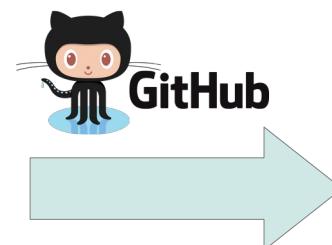
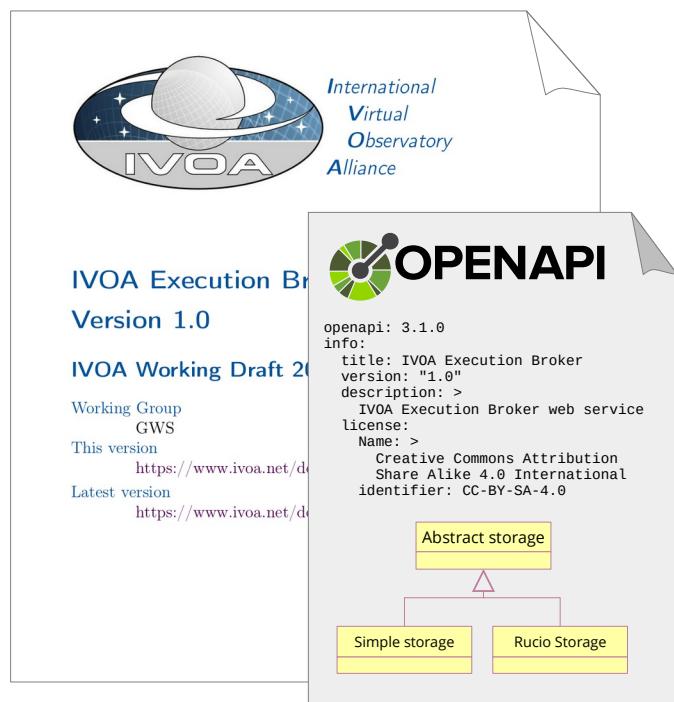
- Shallow learning curve
- Good documentation

- Clear and easy syntax
- Good feature coverage



# Where next ?

Automatically generate and publish libraries



Automatic CI workflow triggered on commit

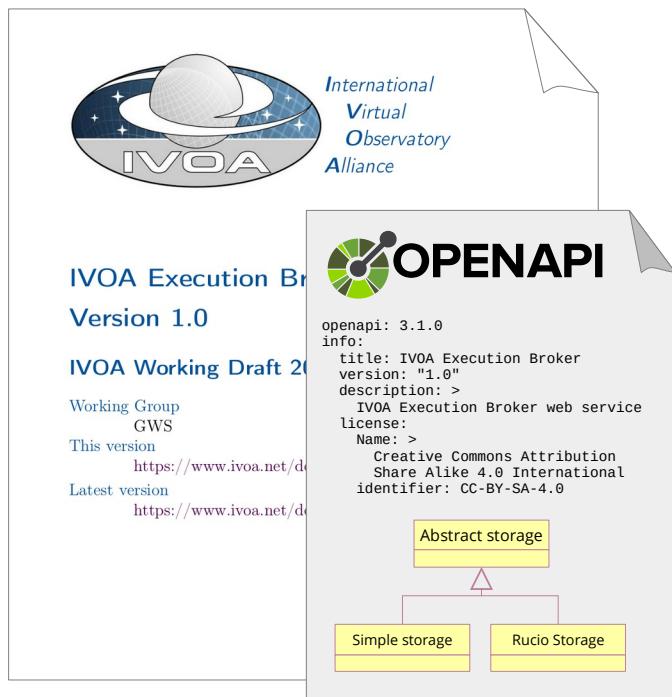
Same process as the preview PDFs





# What do we need ?

Permanent URLs to redirect to schema code



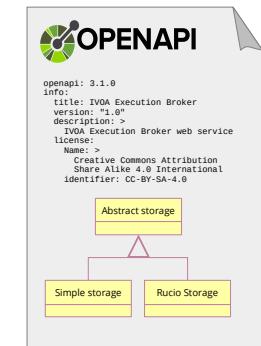
[www.purl.org works, but is a bit flakey](https://www.purl.org/ivoa.net/EB/schema/)

<https://www.purl.org/ivoa.net/EB/schema/>



Add IVOA to w3id ?

<https://w3id.org/ivoa/>



Setup our own ?

<https://purl.ivoa.net/>