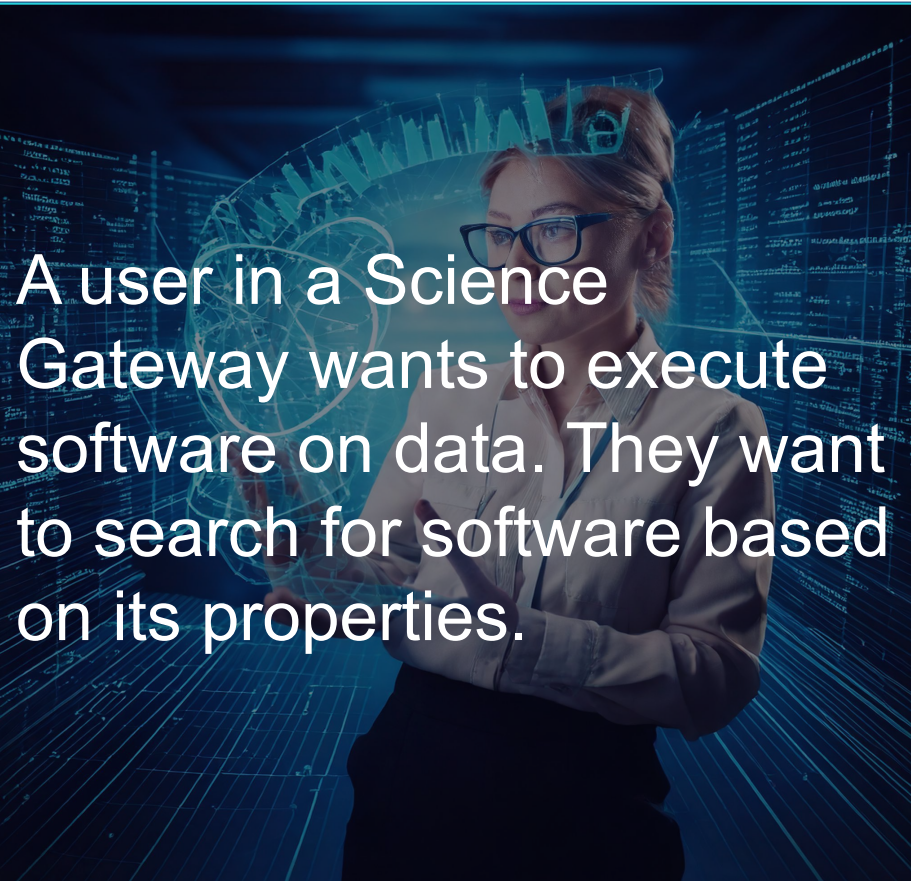


Software discovery characterisation

Tim Kok, Yan Grange

User-driven problem statement

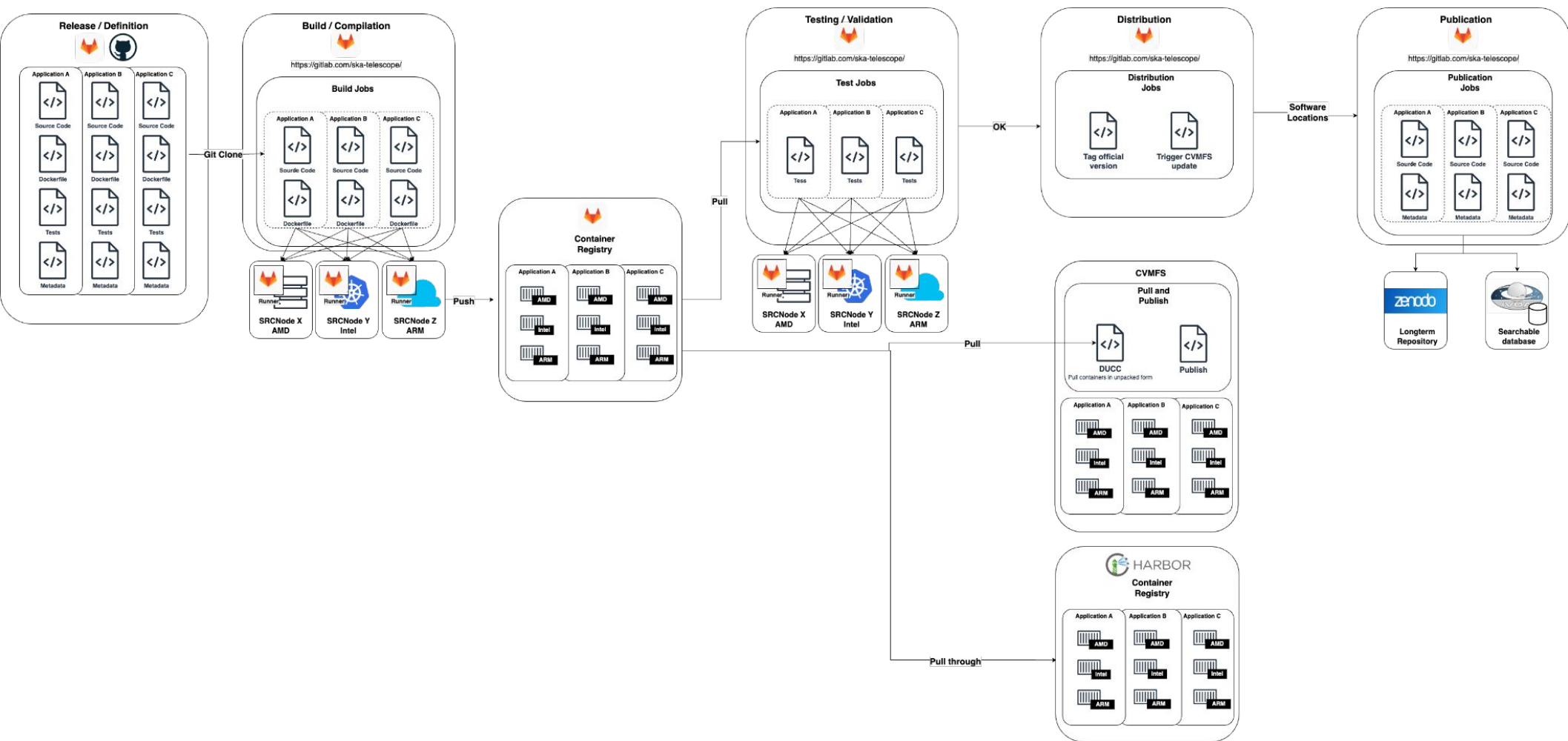


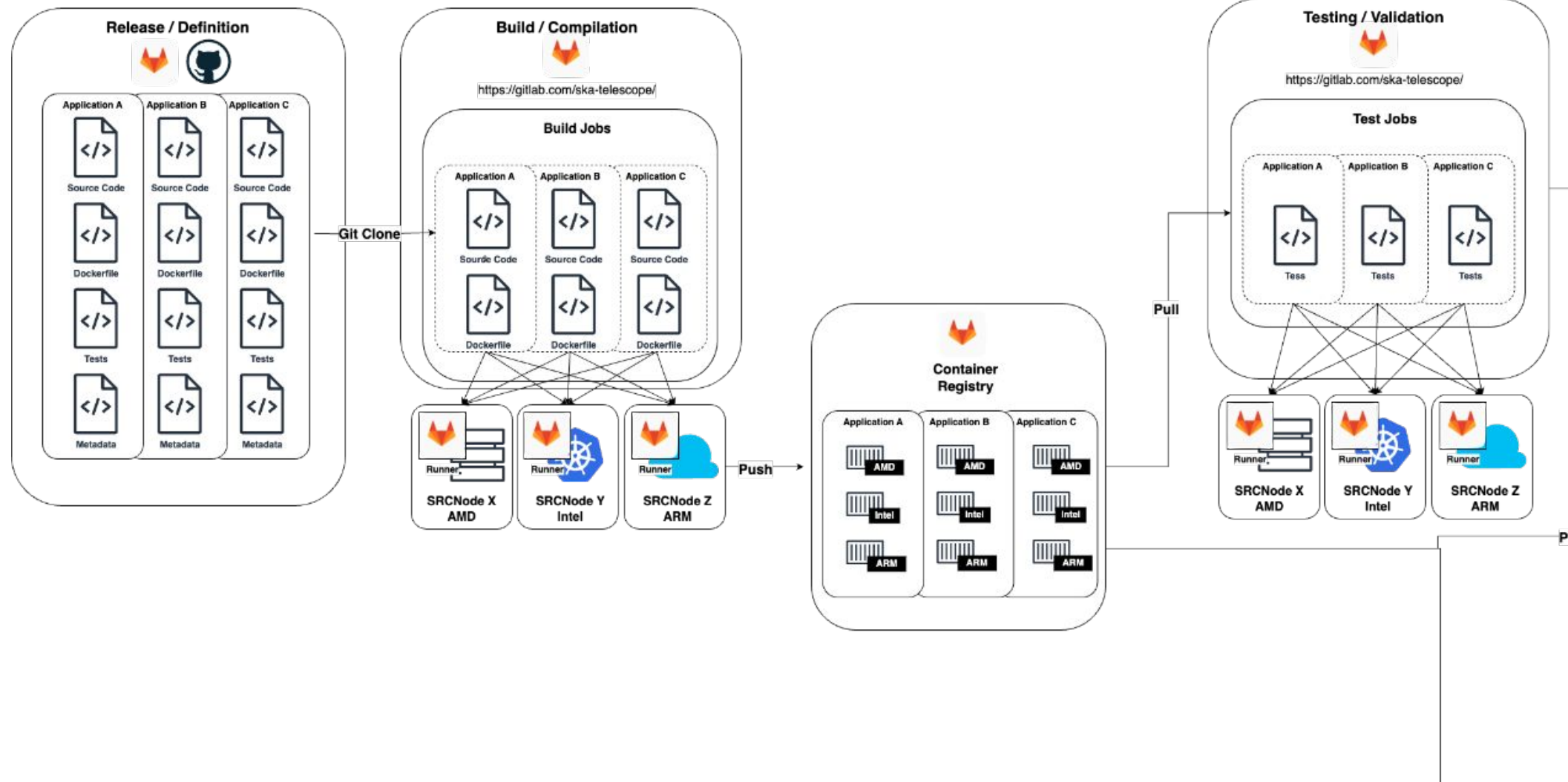
A user in a Science Gateway wants to execute software on data. They want to search for software based on its properties.

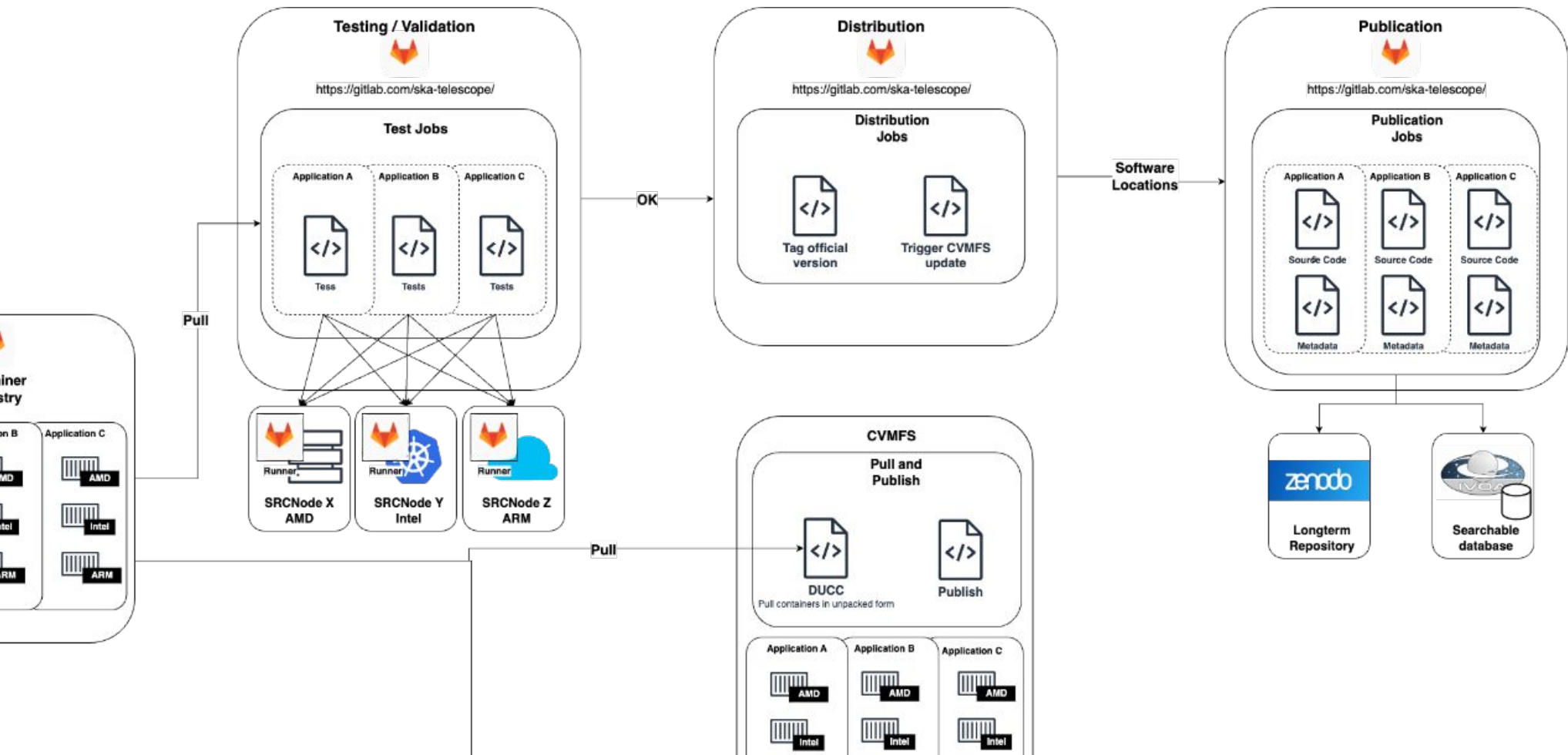


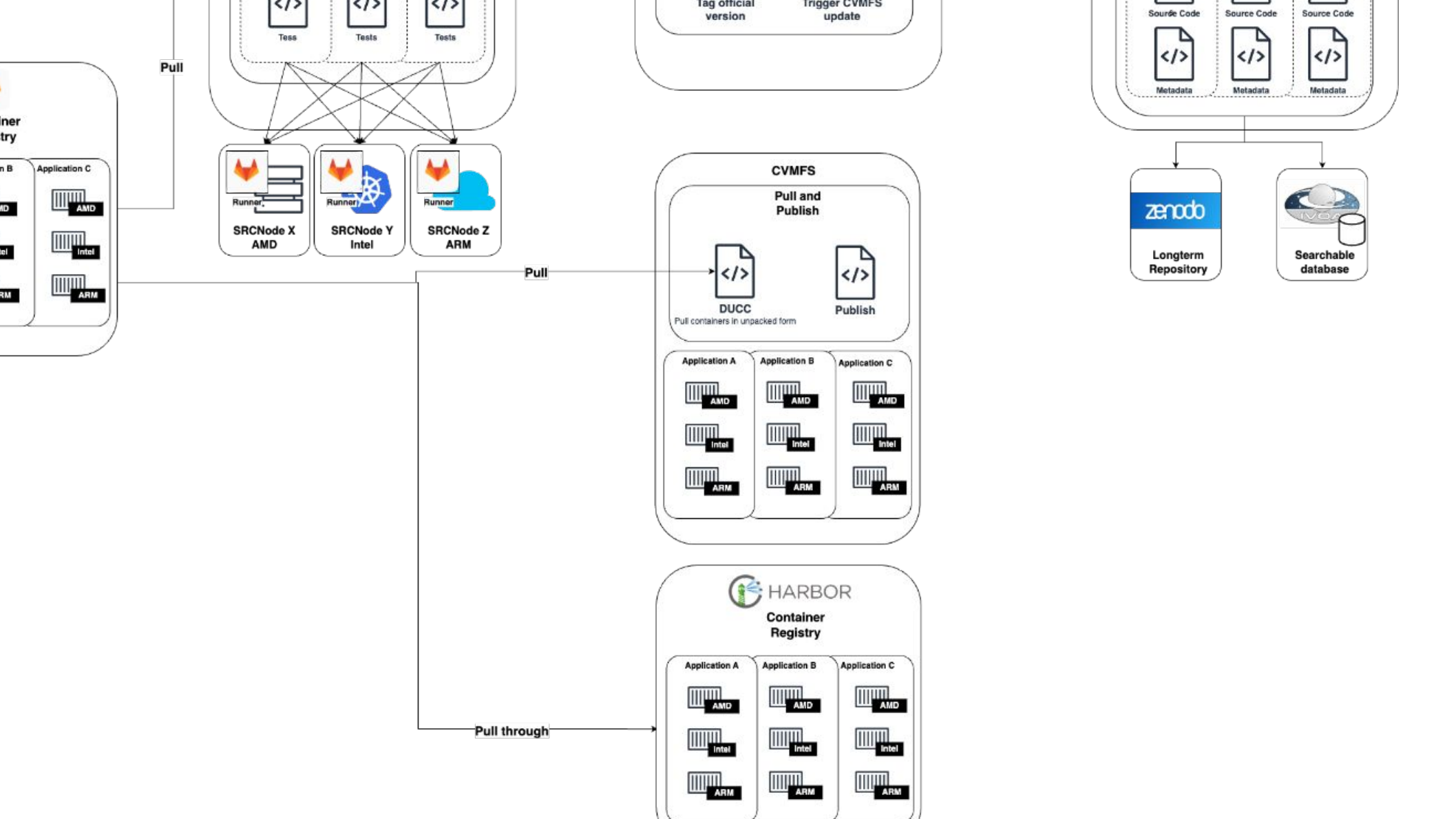
Science software needs to be described in a way that makes it findable and usable.

Software distribution in SRCNet









How is the software characterised

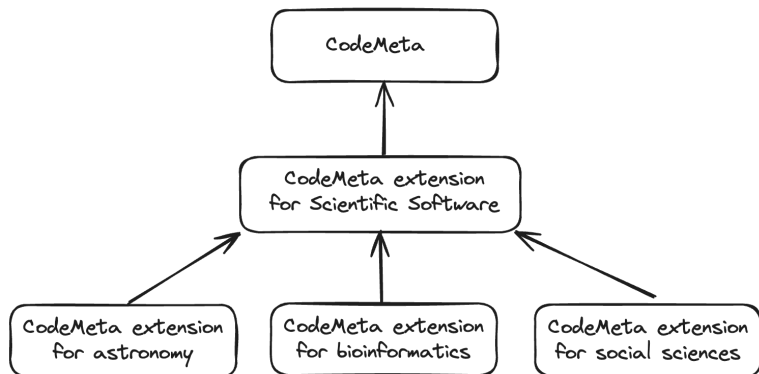
- But how to find software?
 - What does it do?
 - What does it run on?
 - How to execute it?
 - Who is responsible, is it supported, ...?
 - wrap this in a vocabulary.
- Some properties have already been defined in the EESSI definitions

codemeta.json

- + It is an existing standard, which seems widely used.
 - Provides some basic metadata
 - Authors, licens, support status (with vocabulary), provenance, keywords, ...
 - Provides interfaces to several repositories
- The “functional” parameters are missing.
- Afaik no controlled vocabulary for keywords

Can we extend it?

- Sure, it's JSON.



```
{
  "@context": [
    "https://raw.githubusercontent.com/codemeta/codemeta/2.0-rc/codemeta.jsonld",
    {
      "schema": "http://schema.org/",
      "codemetaforscience": "https://codemetaforscience.github.io/terms/",
      "researchDomain": {
        "@id": "codemetaforscience:researchDomain",
        "@type": "schema:DefinedTerm"
      },
      "researchSubdomain": {
        "@id": "codemetaforscience:researchSubdomain",
        "@type": "schema:DefinedTerm",
        "@container": "@list"
      },
      "methodology": {
        "@id": "codemetaforscience:methodology",
        "@type": "schema:Text"
      },
      "tags": {
        "@id": "http://schema.org/keywords",
        "@container": "@set"
      }
    }
  ]
}
```

<https://raw.githubusercontent.com/t1mk1k/codemeta-test/main/schema/codemeta-extended.jsonld>

Discussion points

- Extending codemeta makes the metadata go with the code. How to best index it so it is findable?
 - Do we need a standard for this so that we can expose it as a table?
- Need to define a (few) vocabulary(/-ies)
 - hardware types (optional, required, boundary conditions?)
 - operations
 - file types?
(e.g. “a tool that generates a FITS image, based on calibrated MS input for the radio domain. Requires 64 GB of memory, for data sets larger than 20GB, a GPU is advised”)