P3T Roadmap

Gregory Dubois-Felsmann
& P3T Tiger Team

May Interoperability meeting in Sydney, Australia
2024-05-22

P3T Implementation Road Map – overview

- What is the road we're setting out on here?

But first:

- What is the landscape we're traveling?

Reminder: Phases of the project

- As stated above:
  - We acknowledge *change is not easy for existing projects* --- that is why we have a development/evaluation phase between now and the next Interop
    - Phase 1 – Pilot project (no formally adopted changes)
    - Phase 2 – Phase for 1st set of recommendations
    - Phase 3 – Focus on other standards
- Now look at the larger picture…

The existing landscape – in brief

- Three perspectives: services*, clients*, and users

*and their maintainers

We believe:

- There are a lot of services in existence
  - Some are at large, relatively stable archives with (some) support for ongoing maintenance and upgrades
  - Some are very thinly supported, at small sites, but still valuable
- Relatively speaking, there are fewer clients (libraries and applications)
  - The best-known are associated with large organizations and/or relatively stably supported, and/or under active development
- Users don't want to be distracted by technicalities
  - Users want their tools and workflows to continue working, and
  - Users want access to new sources of data and services

Implications

We believe:

- This is a good situation for the sort of change we're suggesting:
  - Introduce a new way of specifying *and invoking* services
    - More rigorous specifications, *but also*
    - A change in the actual over-the-wire pattern for service invocation
  - Do not *require* any existing service to be migrated
    - Either the services/protocols covered by the pilot project (e.g., TAP),
    - or others – many of which we may not get to even in Phase 3
  - Burden is on client software to evolve to support the new style *and continue to support existing services*

  In other words:
- The only *required* migrations fall on the smaller and generally better-supported set of client libraries and applications

Distribution of benefits (and costs)

- Benefits to service developers and data publishers:
  - It becomes (much) easier to *deploy new services* in an IVOA-friendly way
    - Developers can concentrate on the substance of their services, spending their time proportionately more on the interesting new data and capabilities these will bring to light
  - Regarding existing services:
    - Data publishers can deploy new-style versions of existing services in parallel with the old ones, and feature development can be limited to the new one (at their discretion)
    - Once a critical mass of migrated, dual-capable *clients* exist:
      Data publishers can eventually retire the old-style versions of services to reduce their costs
    - However, this is not in the Phase 1-2-3 roadmap, and…
    - No existing service will be required to be migrated
- Benefits to users:
  - Nothing that works now goes away
  - New services appear sooner / there are more of them / they stay interoperable
  - Result data formats (e.g., VOTable) *are not changing*

Distribution of costs

- Costs to client maintainers:
  - Pressure to support the new style for service invocations
    - *BUT* we expect this to be compensated by increased ease of developing against the more rigorous definitions, and with the aid of code generation tooling
  - Indefinite commitment to support both styles
    - We do need PyVO, TOPCAT, Firefly, Aladin, etc. to be dual-capable for a long time
  - Note: most (not all) client maintenance comes from archives/data publishers that receive a balancing benefit from the simplification of developing services
- Service-side costs:
  - Registering and running old and new versions of the same services
    - Technical details need investigation in Phase 1
- Costs to users:
  - Minimal if we do everything well
  - Users should not be aware of which style of service they are accessing, if they are using a supported client

What about “informal clients”?

- We know some data services are accessed without using a “client” per se
  - The “there are only a few clients to migrate” argument doesn’t apply here
  - Examples:
    - Scripts using curl/wget to get simple tables in CSV
    - Coding directly against `requests` library in Python
    - HTML pages with web-1.0-style forms directly submitting form-parameter queries
      - Possibly some sophisticated ones, e.g. using XSLT to transform the result into HTML
  - “Form-parameter” service invocations will not be available for new services
    - But note that there are equivalently easy ways to submit new-style queries without full-fledged IVOA support client libraries

- When would this become a real problem? Probably not for years!
  - Not until old services are taken down, which is *not a part of the transition plan we are envisaging*. Still, it is plausible that eventually some data publishers will want to stand down obsolete, duplicated services.

Is there a “migration period” that has an end?

- Not in the obvious way

- New-style versions of existing service protocols will appear one at a time
  - No specific requirement to go through the whole body of standards and migrate every single one
  - Unlikely to have a sharp end

- Most important milestone: when are all common clients dual-capable?

Phase 1 implications

During Phase 1, we can identify some key goals:

- Continue prototyping implementation and delivery of concrete services in the new style (TAP, UWS, Execution Broker) and evaluate the results
- Develop one or more clients to be able to communicate with the new-style services (PyVO + ? … volunteers?)
- Work out how clients, services, and Registry interact to support clients in determining which style to use with which service, and how services deployed in both styles should be handled
- Prepare the documents for the formal standardization process in Phase 2

We’re still in the prototype/demonstration era

- No commitments are needed today
- We would like endorsement to continue this research project
- At the end of Phase 1 we’ll deliver a more concrete road map, and we’ll likely have an idea of the other client teams’ openness to and schedule for making the needed changes
- Only then will formal decisions be made, through a full standardization cycle in Phase 2