Interoperable Deliberative Tools Specification v0.7.1

```
---
title: Interoperable Deliberative Tools
description: A flatfile-based architecture for interoperating deliberative tools.
discussions-to: #interoperability in the Metagov Slack
status: Draft
author: Joshua Tan, Aviv Ovadya, Amy Zhang, Colin Megill
created: 2024-04-02
version: 0.7.2
---
```

Simple summary

Interoperability through flatfiles (e.g., JSON, CSV).

Motivation

The goal of this specification is to improve the interoperability of the deliberative and governance tooling space. Traditional API integrations are often considered the gold standard of interoperability, a strategy pursued by other efforts such as the Metagov Gateway. However, APIs and API integrations are often too expensive for many projects in the deliberative tooling space to maintain, especially at scale. Flatfile integration is a more accessible form of interoperability; many services across the space already generate some form of flatfile. Flatfile integrations are also often the first step in proving the value of a given integration between two tools or processes.

The ultimate vision is to reduce the need to replicate functionality across tools, speeding up innovation, and to enable the creation of more complex deliberative workflows for decision-making, involving a sequence of many tools providing specific functions. This specification is just one part of a larger effort, which may also involve creating libraries and sandboxes for tool development and evaluation.

Specification

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119.

Data export

A deliberative / governance tool, service, or library (henceforth, just tool) following this specification MUST publish a flatfile such as JSON or CSV at appropriate intervals of the deliberative process.

Tools MUST default to open, meaning that flatfiles generated from deliberative and governance processes are by default accessible without any tokens or prior authentication. If published openly, flatfiles SHOULD be indexable, i.e. all data should be published to a set of stable URLs that are publicly documented.

Tools MAY also provide the option to provide such flatfile data privately, e.g. with an obfuscated URL or URI, or with API tokens or authentication flows.

To facilitate integration, flatfiles SHOULD be accompanied by an appropriate data license and a data card.

Tools SHOULD provide a publicly accessible URL through which users can pull or request flatfiles, e.g. of the most recent state of a deliberative process.

Data import

Tools MUST include a method for importing flatfiles, of any format, to supply the inputs necessary for running the deliberative process.

Tools MAY pull flatfiles directly from a URL rather than through user upload. They may further support pulling data from this URL automatically.[1]

The expected format of imported flatfiles SHOULD be accompanied by a data card.

Rationale

Example

Here is an example of an integrated deliberative process where Society Library exports seed statements to Pol.is, which then exports conversations to Talk to the City and to Zoom Breakout Rooms.

1. **Society Library to Pol.is**. In the first stage of the deliberative process, a script takes a set of Society Library statement exports, launches a new Pol.is session, and imports those statements into the Pol.is as new seed statements.

[1] We strongly recommend this, but understand that it may provide potential avenues for attack.

2. **Pol.is to Talk to the City**. A script converts the exported flatfile from a Pol.is session and generates a Talk to the City report. This requires reconciling the Pol.is and Talk to the City formats.
3. **Pol.is to Zoom Breakouts**. A user in Zoom adds an embedded Zoom app that launches a Pol.is session from within Zoom, which then grabs the Pol.is flatfile and creates Zoom breakout rooms for the most popular statements or for the inferred consensus groups. A more hacky version involves starting a Pol.is session separately and then running a script through the Zoom API to generate breakout rooms.

Other candidates for integration include GitHub, All Our Ideas, Crowdwrite, Slideo, Mentimeter, Groupmap, various G Suite apps, etc.

More generally, there are multiple ways any two tools A and B may interact through flatfiles which don't require setting up dedicated APIs.

1. **Button-based approach**. User provides data (e.g. outputs from tool A) at a stable URL. User then enters that URL into tool B and clicks a button to import the data.
2. **Fetch-based script with obfuscated data URLs**. Tool A generates a flatfile at stable, obfuscated URL X. User copies URL X into a script for fetching and transforming data from tool A. On every update, script publishes data via obfuscated URL Y (e.g. from Dropbox, Google Drive, etc.). User copies in URL Y into tool B. Optionally, script pings service B to pull data at regular intervals.
3. **Fetch-based script with pushes**. Similar to the above, except that the script automatically pushes data into tool B.
4. **Script + cloud for private data**. User owns an S3 bucket or some other cloud data store that implements authentication. User provides data store authentication information to both A and B. Then for every update, A sends data to store, while B grabs data from store.

Flatfiles vs. APIs

Maintaining a good and consistent API service at global levels is not possible for most independent deliberative tooling. Taking as an example Pol.is's experience: the tool or service would need to maintain the API, to do version control, issue keys, and protect it. This is typically too costly for tools in the deliberative / governance tooling ecosystem, many of which are operated as nonprofits and/or as open-source libraries. While the problem could perhaps be resolved through a large API gateway that manages requests across many services, this would require substantial investment and probably necessitate some form of corporate stewardship to set up.[2]

[2] There is a possibility that an existing enterprise gateway system can be used for this, but we have not found software or a provider that is the ideal fit yet, thus the focus on flatfiles which provide a simple approach to interoperability.

API integrations enable fluid, real-time communication between two distinct services. Flatfile integrations enable a modular, “chunked” flow where a given process in one tool is fully completed and then handed off to the next tool(s). One of our models for this format was the chunked flow of vTaiwan, which mashed up 10+ services.

Access control

Flatfile access still needs to be rate-limited and protected, but this can be done scalably through services such as Cloudflare and S3. Accessibility is solved through the default open policy. An easy way of scaling up indexing while permitting some degree of protection and access control is to take a generated flatfile, save it to an AWS S3 bucket, and then return a unique ID to provide access to that flatfile.

The ability to pull new flatfiles is important for active integrations (active within the constraints of the modular, “chunked” flow for flatfiles.

Additional resources

This specification was launched with the Interoperable Deliberative Tools RFP, as part of the Interoperable Institutions Initiative at Metagov. Accompanying this specification will be a repository and website with documentation of the database and API schemas of existing deliberative tools, along with other metadata about such tools.

We may also provide additional utilities for data import, data migration, identity (e.g. xid), and a JSON-LD standard. Other possibilities include scripts / libraries for common tasks and a diagrammatic workflow creation tool that allows end-users to combine multiple tools through their flatfiles. The project may ship with a tool to facilitate discovery of new tools.

Acknowledgements

We would like to thank Eugene Leventhal for their help in refining this specification and the accompanying RFP.

Copyright