# Technology Options for Open Government Data Platforms

Draft: Jan 31, 2014  
Timothy Herzog, World Bank

**Disclaimer**

This paper discusses characteristics of several different products and services provided by different organizations. Nothing in this document should be interpreted as being an endorsement by the World Bank of a particular product or service.

# Overview

This document provides a summary of the key technical issues and options concerning the development of an Open Data Catalog. The intended audience is managers and decision-makers in governments that are in the process of implementing the technical considerations of an open government data initiative.

An Open Data Catalog is a web-based tool that provides public access to datasets provided by government agencies under Open Data terms of use. Open Data Catalogs are a centerpiece of almost any open data initiative, as they provide the “front door” for users to find and navigate available data. At a minimum, open data catalogs allow users to easily search for and download available datasets in open data formats, display terms of use (licensing) clearly and obviously, and provide attribution, metadata and other documentation. Open data catalogs typically have many other features as discussed under “Common Characteristics” below.

Open Data Catalogs are *not* data management systems that governments would use to manage public data throughout the complete life-cycle. Open data catalogs typically help fulfill the needs for data dissemination, user engagement and transparency within an open data initiative, as opposed to up-stream management of data production. For this reason, one important consideration for the open data catalog is how it integrates with external (and possibly multiple) data management systems.

# Common Characteristics of Open Data Catalogs

**Data Access and Storage**

All data catalogs provide access to publicly available data. Catalogs may have the ability to store data in its own management system (and provide users with basic visualization options), or simply manage metadata along with links to the complete dataset stored on other servers. Several catalogs provide both options.

* Provide data in open, non-proprietary formats such as CSV, XML, and JSON.
* For convenience of specialized users, may also provide data in proprietary formats such as Excel, STATA or shape files (GIS).
* On-site (internal) dataset storage, scalable to client needs
* External links for datasets stored on separate, publicly accessible servers
* Custom metadata fields
* Dublin Core/DCAT support (DCAT is an RDF-based standard for publishing metadata)
* Workflow: can restrict access to “embargoed” datasets
* Manual, bulk and automated data uploading
* Data interface to external/agency systems

**User Experience**

Data catalogs give users tools to access, explore and engage with the data.

* Searching: full-text search, faceted search, filtering and sorting of results
* Geographic searching (search for data in a defined geographic area)
* On-site data preview
* Data visualization via charts and maps
* Engagement: ability to “like” and/or comment on datasets; suggest new datasets; social media links to Facebook, Google+, Twitter, etc
* Ability for user to save visualizations and/or embed in blogs and other sites
* Multi-language support
* Mobile support

**Application Programming Interface (API)**

APIs provide a “low level” entry point for developers to access the data catalog and its contents directly, providing the same services as the catalog itself. APIs may allow data providers to update the data via external systems.

* Search/query data catalog via API
* API access in multiple formats (CSV, XML, JSON)
* Dataset additions and updates via API (for data providers)
* Update metadata via API (for data providers)

**Integration/Customization**

The flexibility to integrate or embed the data catalog with other websites, add additional pages, layouts, color schemes, logos, etc.

* Custom branding and theming
* Custom home pages and landing pages
* Integrates with external content management system
* Extensibility: add additional or custom features via modules
* Analytics: provide statistics on page views and downloads

# Technology Considerations

## Software Delivery Model: Open Source, Self-Managed, Cloud Hosting & Software as a Service (SaaS)

Total cost of ownership is a key evaluation criterion for data catalogs. Products distributed as “open source” software are nominally “free,” in that they may be acquired via download for no cost, and may be modified or customized without restriction or licensing fees. However, open source software still incurs management costs to provide hosting, maintain the source code, install updates and security patches, and provide training. Many consultants and businesses exist that provide all these services, and open source catalogs can be cloud hosted to mitigate server costs and provide performance scalability.

In contrast, SaaS products typically are proprietary products sourced from a single vendor that provides the software and hosting services, typically for a setup charge and recurring monthly or annual fee. Under the SaaS delivery model, the vendor is responsible for software maintenance, server availability and reliability, and provides scalable performance according to the contract. SaaS vendors also typically provide training and some measure of customization.

Decision makers should consider the following:

* Self-managed, open source catalogs can provide a high degree of customization and autonomy—provided the Ministry, Department or Agency (MDA) or its consultants has the necessary technological capacity to manage the technology stack and program customizations. Most open source catalogs are designed to run on Linux/Apache servers and are written in either the Python or PHP programming languages. Without technical proficiency in these areas, the MDA may quickly find open source products inflexible or difficult to manage.
* On the other hand, while SaaS products can provide a great deal of customization, their flexibility is by definition limited to what the vendor is capable of and willing to provide.
* Open data catalogs must be hosted on reliable, fast server architecture. Server outages and slow response times will quickly discourage users. If the MDA does not have the capacity to provide fast, reliable IT infrastructure, cloud hosting or SaaS may be a better option.
* MDAs may have procurement policies or laws that make it more difficult to acquire SaaS products and cloud hosting.

## Scalability

MDAs should anticipate the need to scale their Open Data catalogs in several respects. As data supply grows and new (and larger) datasets are added, the catalog must be able to accommodate the additional datasets without performance degradation. Similarly, as demand grows, the catalog must be able to accommodate the additional server load of more users. Since both supply and demand growth can place additional burdens on the technology stack, it is important to select a server infrastructure than can scale easily. Again, the MDA may already have this capacity in-house, in which case a self-managed catalog may be the best option. Otherwise, the MDA should consider cloud hosting or the SaaS approach.

Finally, as the Open Data initiative grows, the catalog should have the flexibility to add additional functionality, some of which may not be anticipated in the beginning. Some catalogs are intended to get clients “up and running” as quickly as possible, but may not scale well when the catalog grows to hundreds or thousands of datasets.

One way for the MDA to judge scalability is to first project the size of its data catalog at initial launch, after a year, two years, and 3-4 years. Then look at current catalogs to ascertain which ones work well at the needed scale. Each product’s website should provide lists of governments and organizations using that product.

## Data Management: “All-in-One” vs. Federated

One key design consideration for MDAs is how datasets are managed and stored in the data catalog. In an “all-in-one” catalog, datasets are stored in the catalog’s file or object architecture. By contrast, “federated” catalogs are designed so that datasets can reside on any publicly accessible file or web server; the catalog includes a link (URL) to the dataset, as opposed to including the dataset itself.

The chief benefit of the “all-in-one” model is that all datasets can be hosted and managed from a single IT platform, and thus the implementing agency can exercise strong oversight over the entire catalog infrastructure. This may be advantageous in cases where IT capacity and reliability varies significantly across MDAs that contribute data to the catalog, and some MDAs may not be able to provide reliable public access to their own datasets. A second benefit of the “all-in-one” model is that it presents a consistent interface; users need not navigate multiple websites and navigation to access datasets.

The chief drawback of the “all-in-one” model is that the agency responsible for the data catalog implicitly assumes responsibility for managing *every* dataset in the catalog, including updates. This can be a significant management issue, especially as the catalog grows, and especially if the catalog includes datasets from more than one level of government (national, state, municipal or city). It can also be an issue in cases where MDAs prefer to be responsible for their own data management (for political or practical reasons).

“Federated” data catalogs relieve the managing agency of the responsibility for curating every contributing MDAs’ datasets. They also allow greater autonomy to contributing MDAs since they would continue to host their datasets on their own systems. MDAs need only provide a URL and up-to-date metadata. However, the federated approach also requires that MDAs have the capacity to manage their own IT infrastructure.

Several data catalogs allow data to be stored either internally (all-in-one) or externally (federated) on a case-by-case basis, thus offering a hybrid approach. Still, the data management issue should be fully considered as the catalog is designed.

## Communities of Support

Support is an essential criterion for open data catalogs, not only while the catalog is being developed, but also after its launch, and well beyond. Without reliable and robust support services, the MDA will be left to its own means to keep the catalog running smoothly.

Many open data catalogs have an organization that “sponsors” (in the case of open source) or owns (in the case of proprietary) the catalog. These organizations play a lead role in promoting their products, fostering user communities, and providing support. Open source products often have a large, diffuse community of developers and vendors that are familiar with the product and can provide support.

One key advantage of open source software is that the MDA is not dependent on a single vendor; if needed, one can likely find another support vendor or even employ multiple support vendors. A common disadvantage of open source software is that there may not be an organization (or organizations) to coordinate the developer community and provide well-defined service delivery, which can introduce an element of uncertainty for MDAs.

The considerations are exactly the opposite for proprietary/SaaS products. Support is typically provided by a much smaller community, often a single vendor, on which the MDA would be reliant (often called technology “lock-in”). On the other hand, SaaS vendors have obvious incentives to provide regular improvements, upgrades, and corporate-level support consistent with vendors of commercial products.

The support model is also an important consideration for governments looking to create jobs linked directly to the Open Data initiative. MDAs may wish to hire local talent for essential tasks such as platform design, software development, theming, user interface design, site hosting, content generation, dataset curation, and customer service. Open source platforms offer the opportunity for local developers to assume nearly all responsibility for all of these areas, providing that sufficient local talent is available. For proprietary products, opportunities for local talent may be limited in some areas, particularly platform design, development and hosting, depending on the product’s design. All products, however, should afford opportunities for local talent to participate in managing the contents of the data catalog once it has been launched, as well as engaging the user community.

# Common Open Data Platforms

The leading open data platforms are listed and summarized below in alphabetical order. In addition, some governments and organizations have chosen to build custom software solutions, which are not listed here.

**CKAN**

CKAN is an open-source data catalog formally supported by the Open Knowledge Foundation (<http://okfn.org>). CKAN can be installed on any Linux server, including cloud-hosted configurations. The Open Knowledge Foundation also offers hosting services for a monthly fee. CKAN is written in the Python programming language, and is designed for publishing and managing data either through a user interface or through an API. CKAN has a modular architecture through which additional or custom features may be added. <http://ckan.org>

*Examples*

* Edo (Nigeria): <http://data.edostate.gov.ng>
* Brazil: <http://dados.gov.br>
* United States: <http://catalog.data.gov>
* Africa Open Data: <http://africaopendata.org>

**DKAN**

DKAN is designed to be “feature compatible” with CKAN; that is, its underlying API is identical (i.e., systems designed to be compatible with CKAN’s API should work equally well with DKAN). DKAN is also open-source, but is based on Drupal, a popular content management system written in PHP instead of Python. This may be more appealing to organizations that have already invested in Drupal-based websites. Drupal has its own modular architecture with thousands of modules available for download (custom modules are also possible), and a large developer community. <http://nucivic.com/dkan>

*Examples*

* Cologne (Germany): <http://www.offenedaten-koeln.de>

**Junar**

Junar is a cloud-based SaaS open data platform, so data is typically managed within Junar’s infrastructure (the “all-in-one” model). Junar can either provide a complete data catalog or can provide data via an API to a separate user catalog. <http://www.junar.com>

*Examples*

* Lima (Peru): [http://lima.datosabiertos.pe](http://lima.datosabiertos.pe/)
* Chile: [http://recursos.datos.gob.cl](http://recursos.datos.gob.cl/)
* Costa Rica: [http://datosabiertos.gob.go.cr](http://datosabiertos.gob.go.cr/)

**Open Government Platform (OGPL)**

Like DKAN, OGPL is an open-source Drupal-based data catalog, but not designed to be CKAN-compatible at the API level. OGPL was jointly developed by the Government of India and the U.S. Government. <http://ogpl.gov.in>

*Examples*

* Ghana: <http://data.gov.gh>
* India: <http://data.gov.in>

**Socrata**

Socrata is a cloud-based SaaS open data catalog platform that provides API, catalog, and data manipulation tools. One distinguishing feature of Socrata is that it allows users to create views and visualizations based on published data and save those for others to use. Additionally, Socrata offers an open source version of their API, intended to facilitate transitions for customers that decide to migrate away from the SaaS model. <http://www.socrata.com>

*Examples*

* Chicago (U.S.): <https://data.cityofchicago.org>
* San Francisco (U.S.): <https://data.sfgov.org>
* Kenya: <https://opendata.go.ke>
* UNDP: <https://data.undp.org>

# Open Data Platforms Comparison Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product (Technology)** | **Vendor/ Sponsor** | **Delivery Model** | **Data Management** | **Support Community** |
| CKAN  (Python) | Open Knowledge Foundation | Open Source (cloud hosting available) | All-in-One or Federated | Python developer community |
| DKAN  (PHP/Drupal) | Nuams | Open Source (cloud hosting available) | All-in-One or Federated | Drupal developer community |
| Junar | Junar | SaaS | All-in-One | Vendor |
| OGPL  (PHP/Drupal) | Funded by Gov. of India & Gov. of U.S. | Open Source | All-in-One or Federated | Drupal developer community |
| Socrata |  | SaaS | All-in-One or Federated | Vendor |