How to Implement a Hybrid Architecture for Open Source Geographic Information Systems

Tom Ingold

Tom Ingold is CTO of Boundless, an open source geographic information system (GIS) company. Tom joined Boundless from Booz Allen, where he was developing OGC services and architectures in the defense and intel markets. Tom has experience in agile systems and building GIS systems in production environments. He has a BA in Natural Sciences from Harvard and a postgrad certificate from Penn State in GIS.

More data is being generated than ever before, and much of it is location-based. The rise of the Internet of Things (IoT), from connected sensor networks to Smart Cities, has created an abundance of new data streams to be consumed, processed and analyzed on a massive scale.

Preceding the rise of IoT is that of open source software, as exemplified by operating systems such as Linux, container software such as Docker, and automation tools such as Ansible. In the geospatial world, open source has taken root over the past decade, in the form of what we call free and open source software for Geospatial (FOSS4G).

The concurrent proliferation of open source and IoT is driving a change in geographic information system (GIS) technology, which organizations use to visualize many different kinds of location-based data on one map in order to identify patterns and relationships. Geospatial professionals and what we call “geo-enabled users” (who use location information as part of their business function) require technology that can scale, both in technology performance and price, and also integrate with other systems through open and interoperable interfaces and standards.

Open source came about in part due to the challenges associated with closed source environments, like single vendor lock-in, increased costs for scaling architecture up or out, and a lack of interoperability with existing software and hardware. Open source is flexible enough to work alongside and improve the functions of pre-existing
architectures, so that IT teams do not have to rip and replace legacy systems, which often represent decades of significant investment.

Instead, organizations can migrate their enterprise gradually in phases, as operational tempo allows. The resulting architectures are called hybrid architectures, consisting of both proprietary and open source software.

**Migrating to a Hybrid Architecture with Boundless**

At the core of a hybrid architecture is the concept that GIS software can integrate seamlessly alongside existing proprietary software. Since open GIS software is largely built upon the use of standards, you naturally build in interoperability the more you migrate away from proprietary software. The *Boundless platform* includes software at the database, application server, and user interface tier. None of these have strict dependencies on each other, meaning that you can integrate open source one tier at a time without disrupting the entire system.

**Database Tier Migration**

It is most common to begin the migration process at the database level because changes are largely hidden from the end user. They still have the same user interface they are accustomed to, but are simply connecting to a different endpoint to retrieve their data (sometimes unknowingly). For example, an organization may swap out their proprietary database technology for an open source database. Despite a change at the database tier, customers still access content through the same custom web application the organization built using a proprietary web mapping library.

*Oracle Spatial*, for example, can be easily swapped out with PostGIS (the spatial extension for Postgres) with very little effort. In fact, the open source GDAL/OGR tools make the data migration a single command:

```bash
ogr2ogr -f "PostgreSQL" PG:"host=localhost user=pguser dbname=myspatialdb password=pgpassword port=5432" OCI:oracleuser/oraclepassword myspatialtable
```

In a few seconds, data moves from a proprietary database with a single vendor to an open source database —
available for free or with many different options for commercial support. The capabilities are nearly identical and most application servers (including the open source GeoServer) can switch over transparently.

Best of all, end users consuming data from the application server are unaware that you made the change, as they continue to use the front-end applications they are accustomed to using.

**Application Server Tier Migration**

Migrating to open source at the application server injects more interoperability into your architecture. Boundless Suite, for example, allows you to publish your spatial data in Open Geospatial Consortium (OGC) standard services and formats. The use of standards means a service published once can be consumed in any number of proprietary or open-source end-user applications at the same time.

By using these standard interfaces and data formats (and regardless whether you migrated your database described above) you insulated yourselves from any API changes added between a client and server. As long as the server implements the standard, any client can consume any data. OGC's Web Map Service, for example, is a well-defined standard that behaves the same regardless whether it is published from ArcGIS Server, Mapserver, QGIS Server or Boundless Suite.

**User Interface Tier Migration**

Migrating the user interface tier allows you to leverage the power of the 80/20 rule. The 80 percent of users that only require basic functionality can most likely perform their job using robust open source applications. Without the cost of licensing 80 percent of your user base, you open up your budget to support the remaining 20 percent of power users who require edgecase functionality found only in proprietary software.

**Boundless Desktop**, a cross-platform desktop GIS (meaning it works on Windows, Linux and Mac), allows users to manage, analyze, visualize and disseminate geospatial data from a variety of vector, raster, and database formats, including: PostGIS, Oracle, SQL Server, Shapefile, KML/KMZ, OGC WMS/WFS, GeoTiff, NITF and many more. It enables on-the-fly re-projection, data editing, spatial analysis, network analysis, and more. Boundless Desktop is extendable through an extensive plugin library, and an ability to create your own custom plugins using open source Python scripting.

For simple or repeatable workflows, consider migrating capabilities to purpose-built web applications. For example, a user who performs simple data collection or who uses a desktop GIS for situational awareness could migrate that workflow to an OpenLayers web application.

**A Phased Approach**

Successful hybrid migrations should set proper expectations regarding timeline and cost, while clearly articulating achievable milestones from the beginning. Phased approaches that include the following distinct steps: data consolidation, service enablement, and client/app development, are recommended.
Phase 1: Data Consolidation

Successful hybrid GIS architectures need a solid foundation, which requires a focus on the data and data storage first. This should consist of de-duplicating and conflating data sources as necessary, with the end goal being a centralized, authoritative set of data from which services can be published.

When preparing data, it’s important to undertake any possible optimizations that could be beneficial when exposing the data through a service. This might include reprojecting data into commonly requested reference systems, or creating small overviews of large images. This pre-optimization tends to pay off with big performance benefits to the end user, but caution must be exercised for any optimization that could inadvertently decrease precision or accuracy.

Phase 2: Service Enablement

Phase two is about service-enabling data into a map and analytical services. This allows organizations to maintain integrity and fidelity of their source data. Discussions around governance and policy are also necessary in this phase in order to determine an agreed upon set of policies that govern who can do what to the data and services within holdings.

Further optimizations can be realized here. Services are designed to provide a generic means for the distribution of data. While the service provider doesn’t always know the intended end use of the service, removing extraneous metadata or attribution is often recommended. For example, it is not uncommon for feature data to have multiple redundant fields — for example a state name, a state abbreviation and maybe even a state id. Reducing this to a single attribute field may seem like an insignificant step, but it can significantly reduce database IO, shrink bandwidth, and simplify client configuration.

Phase 3: Client/Application Development

Finally, one must consider how external users will access, connect and use the data and services. Beyond opening up and sharing service endpoints to users outside of the organization, this phase usually includes desktop, web and mobile application development, as it is a popular way to make use of data and services.

Hybrid GIS architectures consisting of both proprietary and commercially supported open source software empower organizations by reducing risk and adding value to their projects through better interoperability, scalability, availability and flexibility. By following the steps above, migrating to a hybrid architecture can be a simple and
effective way to improve one’s proprietary system and drive innovation within an organization.

Feature image: Boundless Desktop