Broadsea 3.0: "BROADening the ohdSEA"

Ajit Londhe[1,4], Lee Evans[2,4], Sanjay Udoshi[3,4]
[1] Boehringer Ingelheim, Ridgefield, CT USA, [2] LTS Computing, West Chester, PA, USA,
[3] Acumenus Data Sciences, USA,
[4] Observational Health Data Sciences and Informatics, New York, NY USA

Background

Broadsea[1] is a Docker-based project that allows for consistent deployment of OHDSI web applications (e.g. Atlas and RStudio) along with database and networking dependencies required to run those tools. Broadsea has allowed researchers to experiment with OHDSI tools without significant technical expertise or expense. Due to the Docker-based approach, the setup is consistent, regardless of the operating system or hardware. Still, many sites continue to struggle with deploying OHDSI tools, and many viewed Broadsea as not sufficient for production server usage. With Broadsea 3.0, we sought to improve Broadsea's ease-of-use and consistency through a simplified, a-la-carte configuration and expand its viability for production environments.

Methods

Based on our collective experience within our sites and support of the broader OHDSI community, we identified gaps in Broadsea. Broadsea 2.0 was designed to launch a predefined set of Docker services, which would not always be appropriate for all sites. Using Docker Compose profiles[2], we established a profile name for each Docker container so that a site could either use a "default" setup, or pick and choose which containers they wish to deploy for their site.

We also developed an environment variable driven cadence for running Broadsea. All configuration items are organized into commented sections documented by a more in-depth "Readme" file, resulting in needing just a single command pointing at a single environment file. In an effort to help new sites stand up an OMOP Vocabulary database quicker, we added a profile that launches a Docker container that can fully load Athena CSV files into a Postgres database schema.

With Broadsea 2.0, although Docker services could be launched on most systems, bringing it to a production server with a larger user group presented issues: insufficient routing of services, no SSL for securing web traffic, and no options to enable authentication support in Atlas. Routing multiple services had been available via the server proxy software Traefik[3], but it was not yet robust enough to handle adding more tools. To ensure a consistent networking experience in Broadsea, we expanded upon the Traefik utilization to add middleware routing logic that ensures each container is appropriately reachable from clients. We also leveraged the SSL capabilities in Traefik to allow sites the ability to enable secure connectivity to the server. We reviewed the authentication providers supported by WebAPI and added environment variables for enabling authentication. To help Broadsea sites test authentication behavior in Atlas, we added a profile for an OpenLDAP Docker container service.

To allow sites to evaluate new features available in Atlas and WebAPI, we leveraged Docker Compose build options in profiles that build Docker images in-place from a Git repository URL rather than the formally released Docker Hub images. We also examined more recent feature additions in Atlas and newer OHDSI web tools that have emerged in recent years. In Atlas, Apache Solr[4] powered OMOP Vocabulary search has become useful for improving the speed of Atlas concept searches. We added a Docker container service for this pattern that requires only a few environment variables set. The

PHOEBE[5] project has yielded concept set recommendations in Atlas; we added a profile for loading the files necessary into the OMOP Vocabulary to enable this feature. To aid in ETL, we have begun integration with Perseus[6]. Ares has emerged as a game-changing method for exploring data quality at the CDM and site level. We added a profile to Broadsea that handles deployment and exposes a file directory mount for adding the necessary Ares data files.

Results

Broadsea 3.0 has immediately improved the ease of deploying OHDSI tools, as it encompasses all facets of the OHDSI research approach, from converting data into the OMOP CDM, through performing analysis and sharing evidence.

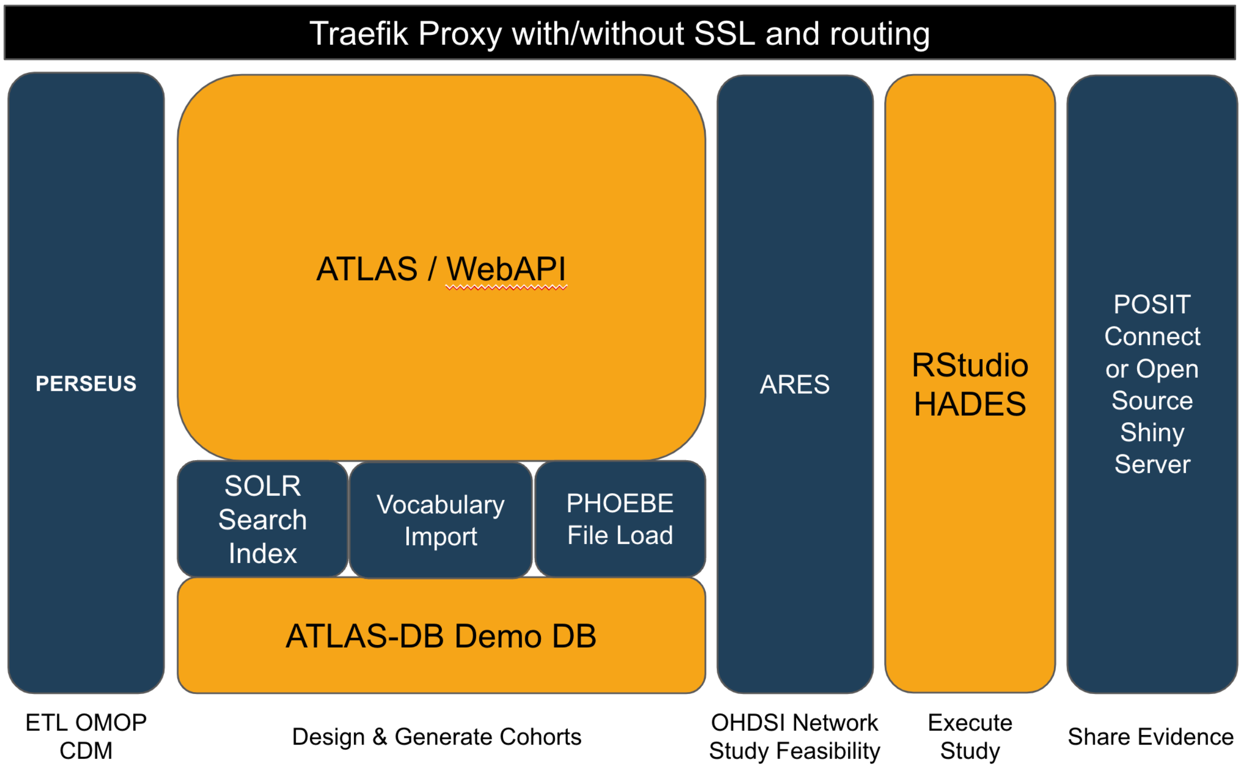

Figure 1. Architecture of Broadsea 3.0, with Traefik acting as a proxy atop several OHDSI tools and backend services.

Conclusion

Broadsea 3.0 provides a flexible approach to deploying OHDSI tools that are typically challenging to set up, and establishes a framework for supporting new OHDSI tools to come. Any site, regardless of size, can deploy a wide range of OHDSI tools on a laptop or a production server, and join us on the journey.

References

1. Broadsea GitHub repository. https://github.com/OHDSI/Broadsea.
2. Docker Compose. https://docs.docker.com/compose/.
3. Traefik. https://traefik.io/traefik/.
4. Apache Solr. https://solr.apache.org/.
5. PHenotype Observed Entity Baseline Endorsements (PHOEBE). https://data.ohdsi.org/PHOEBE/.
6. Perseus. https://github.com/OHDSI/Perseus.