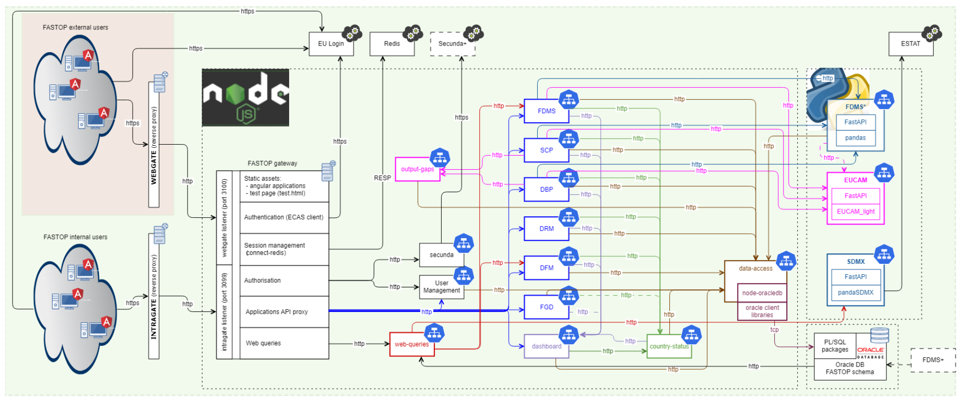
FASTOP microservices architecture high level overview



1. Overview

The FASTOP system is a set of web-based applications used by economists in the DG ECFIN and also by the member states staff. The applications provide tools for economic data reporting, analysis, forecasting, assessment, archiving, etc.

The system is built in the microservices architecture. It consists of number of services dedicated to performing specific tasks and communicating with each other. Most of these services are developed and maintained by the DG ECFIN team and they compose the core of the FASTOP system. There is also a number of services consumed by FASTOP and provided by third parties, e.g. *EU Login* (DG DIGIT) or *ESTAT* (Eurostat). All FASTOP services are stateless and every service exposes a REST API which may be consumed by other services. All communication between the services is done asynchronously in a non-blocking way over http and is secured by the API key so no direct access from outside is allowed to any service other than the *gat eway*. The *gateway* is the only entry point to the system and it serves several purposes which are explained in details below. The majority of the services are written in Node.js and others are written in Python.

2. Technology stack

Front-end applications are written in Angular (currently in version 10). We try to follow angular releases and migrate to the latest major Angular version some time (1-2 months) after the official release so it has some time to stabilize and get validated by the community. For the components library PrimeNG is broadly used.

For the back-end services two technologies are used: Node.js (version 12) and Python (version 3.7). In Node.js *Express* web application framework is used to build the REST API services. Similarly in Python *FastApi* framework is used to build the REST API services which then run on *uvicorn* (ASGI python server). All Node.js services run on one virtual machine (although this is not required, they might be spread across multiple hosts) and are orchestrated by the *PM2* process runner. They are typically run in a cluster mode with two instances of each service and with *PM2* distributing the load between the instances. Python services are run manually (there is an ongoing work to orchestrate these services as well but it is not finalized yet) on another host machine.

Redis service is used to store and share user session data.

Oracle DB (version 19c) is used as the persistence layer. Access to the data is always through the stored procedures which are logically grouped in PL /SQL packages.

3. Composition

The FASTOP diagram may be divided into several sections:

- User computers, tablets, phones, etc. running FASTOP angular applications in browsers. The applications communicate with the FASTOP system through reverse proxies *(intragate* for internal users and *webgate* for external users) which then direct traffic to the corresponding listener in the *gateway*. The angular applications are served from the *gateway.*
- Node.js ecosystem containing Node.js microservices *(gateway, data-access, authorization, application APIs, web-queries* and others).
- Python ecosystem containing Python microservices *(FDMS*, EUCAM, SDMX)*
- Oracle DB (persistent storage)
- Third party services which are consumed by FASTOP components (*EU Login*, *Redis, Secunda+, ESTAT)*

4. Node.js services

4.1. Gateway

The *gateway* is the only entry point to the FASTOP system. All incoming traffic into FASTOP has to pass through the *gateway*. It exposes two listeners: one for the *intragate* and one for the *webgate.* Users connecting through *webgate* have limited functionality (not all applications are accessible, admin actions not allowed) while users connecting through intragate have access to all functions (provided they possess corresponding authorizations).

The gateway is responsible for the following:

- Serving angular applications as static assets.
- Authentication: redirects users to the *EU Login* web site and verifies authentication tokens against the *EU Login* service.
- Authorization: retrieves authorization information from *user-management* and *secunda* and stores it in user session.
- User session management and persistence in *Redis.*
- Proxy applications API requests to corresponding services.
- Proxy web-query requests to the *web-queries* service.

4.2. Data-access service

It serves as the only interface to the persistence layer (Oracle DB). It uses the *node-oracledb* driver provided by Oracle. It maintains a pool of connections to the DB. Other services communicate with *data-access* using dedicated protocol which allows executing stored procedures in the DB.

4.3. Authorization services

User authorization information is currently provided by *user-management* and *secunda+.* This is a temporary solution and gradually *secunda+* will take over *user-management* responsibilities and eventually become the only service responsible for user authorization.

4.4. Front-end applications API services

These are services dedicated to front-end applications, serving their API requests. They include: *FDMS, SCP, DBP, DRM, DFM, FGD* and *dashboard* which serves all applications. In order to serve these requests they may need to send requests to other services for additional processing.

4.5. Web-queries service

It serves web-query requests. It collects data from other services and builds an html response which is sent back to the caller (usually an excel file).

4.6. Internal services

The *output-gap* service is responsible for calculating structural balance which is a step during output-gaps calculation process used by *SCP* and *DBP.* The plan is to move this functionality to a Python service in the future.

The *country-status* service is responsible for tracking country status changes across all applications. All requests for country status information as well as actions changing the status are handled by this service.

5. Python services

The FDMS* service is a set of functions performing forecast calculations, validations, aggregations, etc. It uses python pandas for building economic models. The goal is that this service eventually replaces the FDMS+.

The EUCAM service is maintained by another unit in DG ECFIN and is dedicated to performing output-gap calculation.

The SDMX service is used for retrieving the data from external sources that support the sdmx protocol. Currently it is used to retrieve the data from the Eurostat and other sources are to be added in the future.

6. Oracle DB

All FASTOP data that needs to be persisted is stored in the Oracle DB. The same schema *FASTOP* is used for all FASTOP components (this should be revisited, the more correct design would be to group logically the data and spread it into multiple schemes).

Access to the data is done exclusively via stored procedures which are logically grouped in PL/SQL packages. There is significant amount of business logic in the PL/SQL packages.

7. Third party services

EU Login is a SSO (single sign-on) service provided by Digit and widely used in EC for user authentication.

Redis is an in-memory datastore service provided also by Digit. It is used as the persistent storage for user sessions.

Secunda+ is the authorization service. It has its own interface for managing user access rights.

ESTAT is a service provided by the Eurostat for data access.

8. FDMS+

The legacy FDMS application which will be eventually replaced by the FDMS*.