Orchestrating ETL for a User-Facing Portal: An Integrated Narrative

At the heart of our portal lies a system designed for flexibility, scalability, and clarity of workflow: an ETL pipeline that can seamlessly extract data from multiple sources, transform it according to business and clinical rules, and load it into a target system where it can be consumed by users in near real-time. The orchestration of this system is not merely a sequence of scripts; it is a thoughtfully designed framework that abstracts complexity and provides a robust backbone for the portal.

Extraction Layer

The process begins at the extraction layer. Users may supply data from a variety of sources: relational databases such as MySQL, RESTful APIs, or local files in CSV or JSON format. Each source is defined in a structured dictionary that describes its type and configuration details, such as connection parameters or file paths. Behind the scenes, the system relies on a registry of extractors, each implementing a consistent run method. This abstraction allows the orchestration engine to handle a new data source without altering the core workflow — it simply looks up the appropriate extractor by type and invokes it. This design elegantly decouples the specifics of data access from the ETL process itself, supporting both modularity and future extensibility.

Transformation Layer

Once data is extracted, it often requires transformation before it can be useful for the portal. Transformations range from normalizing field names and types to constructing domain-specific structures, such as FHIR-compliant Patient or Immunization resources. Each transform is encapsulated in a class with a standard run(data) interface, ensuring that the orchestration engine can apply any transformation without concerning itself with the implementation details. This approach is particularly powerful in a healthcare context, where different tables or record types may require specialized logic. For example, patient records are carefully mapped to FHIR Patient objects with identifiers, addresses, and contact points, while immunization records are mapped to FHIR Immunization resources.

Other transformation outputs, such as CSV, SQL inserts, JSON, or Parquet, can be selected depending on downstream analytic needs. The dashboard provides users with a guided interface to choose the transformation method, ensuring that data is correctly prepared before analytics. This combination of user guidance and backend flexibility ensures the right data is available in the right format.

Loading Layer

After transformation, the ETL system proceeds to the loading phase. Loaders, also defined in a registry and adhering to the same run contract, handle the persistence of data into target systems. This could be writing to a relational database, saving to a local file for downstream

processing, or uploading to a cloud-based object store such as S3. The abstraction ensures that the orchestrator need only know what type of loader to invoke and what configuration to pass, leaving the mechanics of writing data encapsulated within the loader class. Loaders also return metadata about the operation — such as the number of records processed or the location of the stored data — which can be surfaced to the portal for user feedback and operational monitoring.

Orchestration with Celery

At the technical center of this orchestration is Celery, which coordinates ETL tasks. Celery executes tasks asynchronously or in parallel, resolving the appropriate extractor, transformer, and loader from their respective registries and applying them in order. Users interact with the portal through the dashboard to initiate ETL operations — selecting sources, transformations, and repositories — while Celery executes the workflow behind the scenes, returning structured results including status, record counts, and destination details.

Scalability and Modularity

This architecture supports scalability, modularity, and maintainability. Adding a new data source, introducing a new transformation, or extending the portal to write to a new type of repository requires only the creation of a new class that adheres to the run interface and an update to the relevant registry. The orchestrator itself remains untouched. This separation of concerns allows a complex healthcare portal to remain reliable and user-friendly, despite the intricacy of the underlying data landscape.

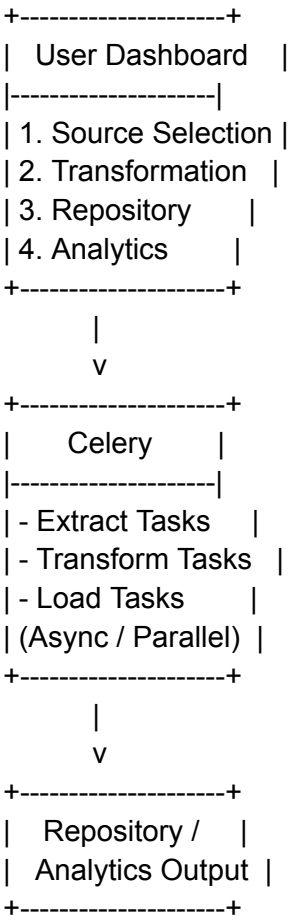
Dashboard Integration

The dashboard plays a central role in connecting the user to the ETL system. It abstracts complexity, guiding users step-by-step through the configuration process:

- Selecting data sources
- Choosing transformations
- Defining repositories
- Selecting standard or custom analytics

It ensures the user interacts with a simple interface, while the ETL system handles the heavy lifting in the background.
Figure: ETL Workflow with User-Facing Dashboard
The dashboard guides users through the ETL workflow, allowing selection of data sources, transformations, repositories, and analytics options. Behind the scenes, Celery orchestrates extraction, transformation, and loading tasks, ensuring data is processed asynchronously and accurately. This design separates user interaction from execution, providing simplicity and clarity for the user while maintaining modularity, scalability, and reliability in the underlying ETL pipeline.

Diagram Layout (for visual reference):



Conclusion

This sequence of tasks encapsulated in the ETL workflow empowers clinicians to investigate and obtain answers to their data questions at a reasonable cost, through PyxGen's forthcoming analytics Software-as-a-Service. Designed specifically for small and medium organizations that lack the resources of large IT departments, PyxGen levels the playing field, providing access to advanced data and interoperability tools without the need for a large in-house staff.

PyxGen has completed the design of the user interface that overlays the technical architecture, creating a clean, guided experience for subscribers. The next phase focuses on implementing the wiring necessary to support real-world use cases, including integration and user testing processes. Each subscriber has unique configuration needs, which are handled via a configuration profile accessible at login. These profiles securely store credentials and system-specific settings, ensuring that the ETL processes operate correctly for each environment.

The core ETL processes are standards-compliant, guaranteeing reliable data extraction, transformation, and loading. On the analytics side, clinician input drives the generation of meaningful reports. A basic report provides insights such as record counts, types of records, source distribution, and demographic statistics. Custom reports, on the other hand, require

closer collaboration with clinicians to ensure clarity and to meet specific decision-making requirements. By combining robust ETL workflows with an intuitive dashboard, PyxGen enables small and medium organizations to leverage their data effectively and efficiently.