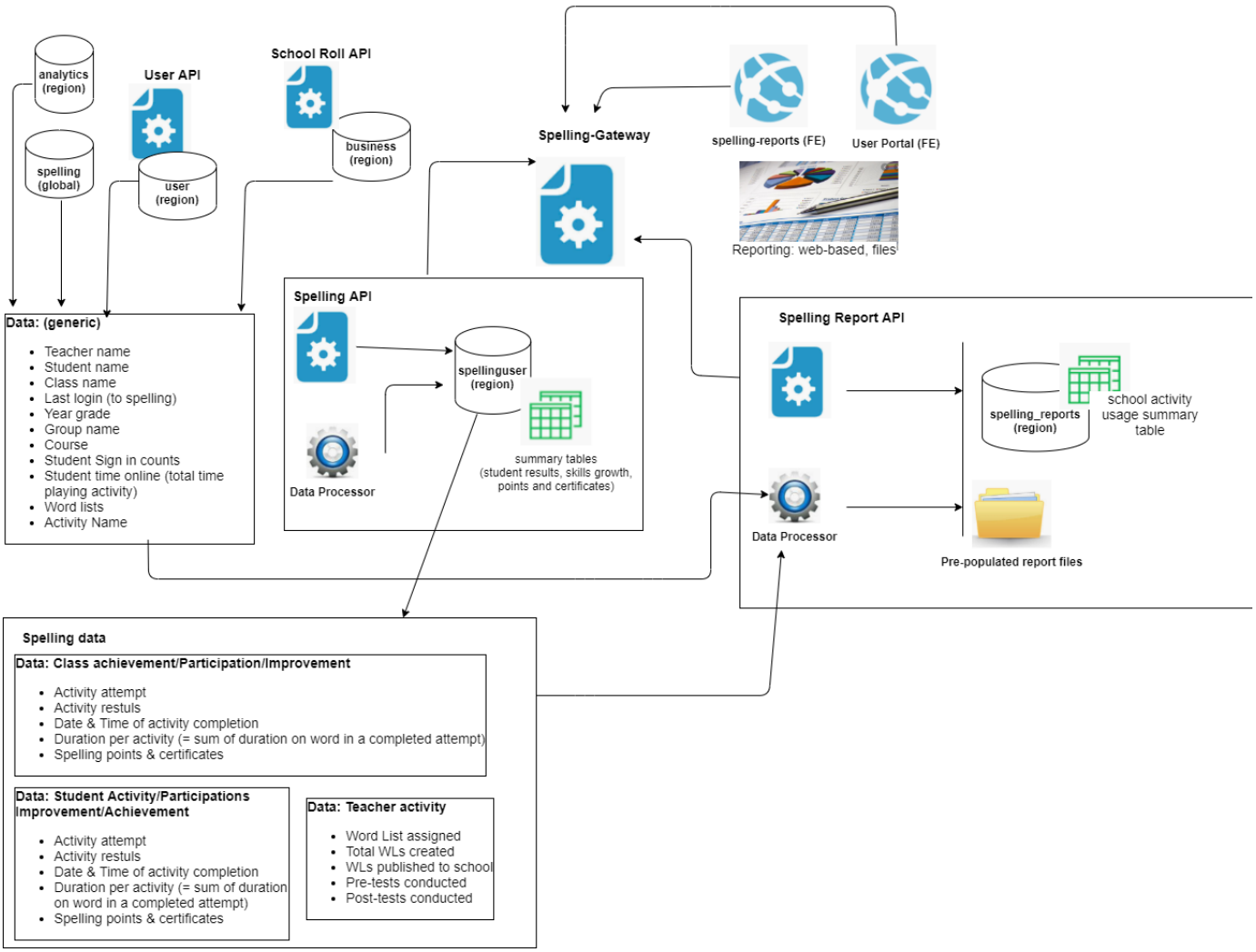
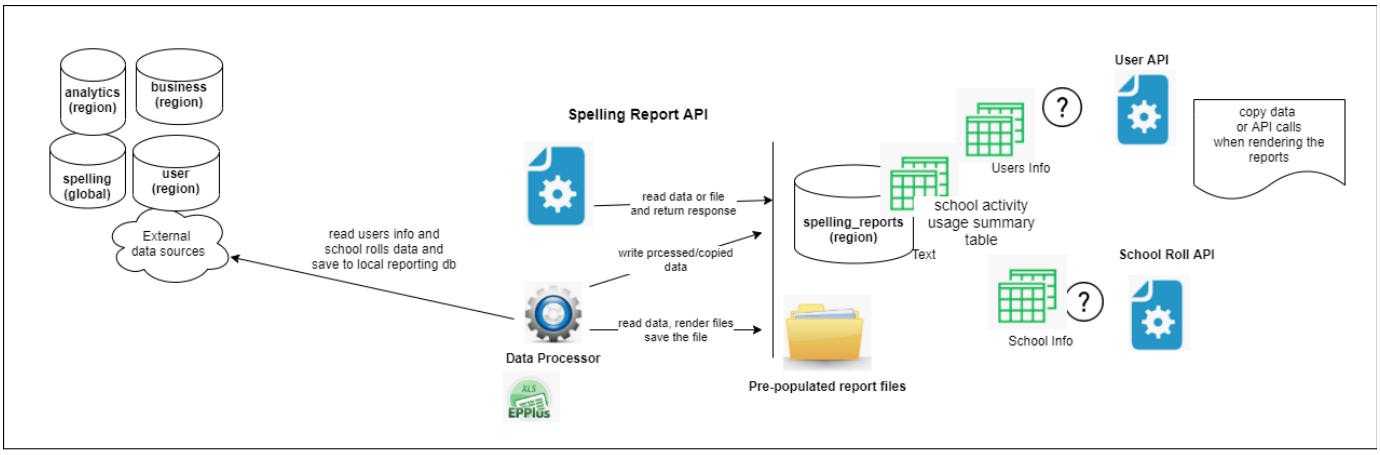
School Activity Usage Report

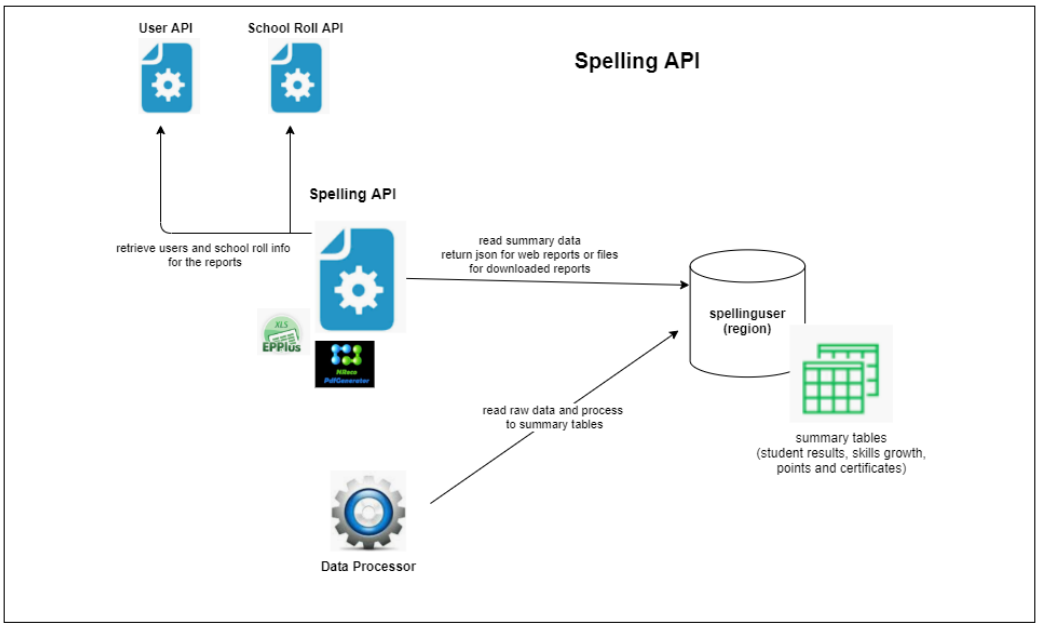
BE Design

High level components and data aggregation diagram



Components diagram





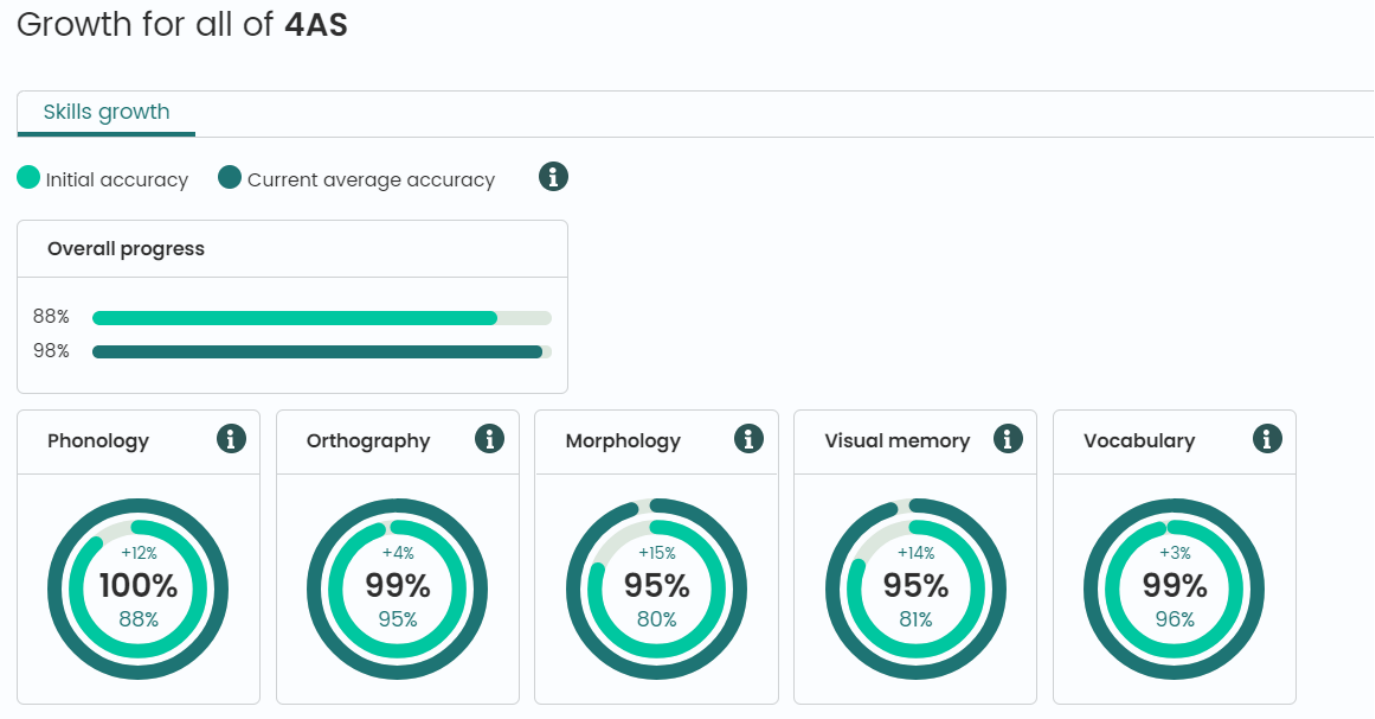
Design consideration

For spelling reporting BE side, we want to split this out of the current spelling API.

Spelling API

- This should be for main operations of spelling, e.g CRUD operations
- Data aggregations should be considered as reporting and move into spelling-reports component. Current data aggregation:
  - Student results
  - Points
  - Skills Growth

Years	Pre-test	Look, Say, Cover, Write, Check	Word list workout	Learning statement	Word facts	Finish the sentence		Post-test
Year 3		98%	81%	42s	✓	98%		
Year 4	44%	97%	81%	57s	✓	98%		64%
Year 5	32%	100%	83%	27s	✓	100%		
Year 6	54%	94%	77%	2m 53s	✓	99%		77%



Spelling Reports API

- This should contain all reporting functionalities for spelling
- DB design: regional sharding, contain necessary data for reporting such as aggregated data as well as data copied from other systems to avoid hitting the external APIs
    - Raw data (copied from other sources, such as User Info, LastLogin, School Rolls, etc.)
    - Aggregated data (such as Student Results, Skills Growth, Student Activity Summary, Teacher Activity Summary, etc.)
    - Time-based table sharing : flexible based on the reporting requirement
        - Per week
        - Per month
        - Per year
- Data Processor:
    - Import raw data from external sources
    - Retrieve spelling data, aggregate, then persist into local DB
    - Pre-populating reporting files: this is only when the reporting generation process could take long time (like for School Activity Usage report, we generate lot of spreadsheets and one file per Year Group and compress into single zip file).
- Files compression and storage for faster retrieveal
    - Storage decision:
        - Blob or network shared folder?
        - Should be protected: only accessible by the spelling-report API and its processor
    - Compressing requirement & library:
        - Basic compression (e.g no need password setup)
        - Library: System.IO.Compression

Spelling Report Processor

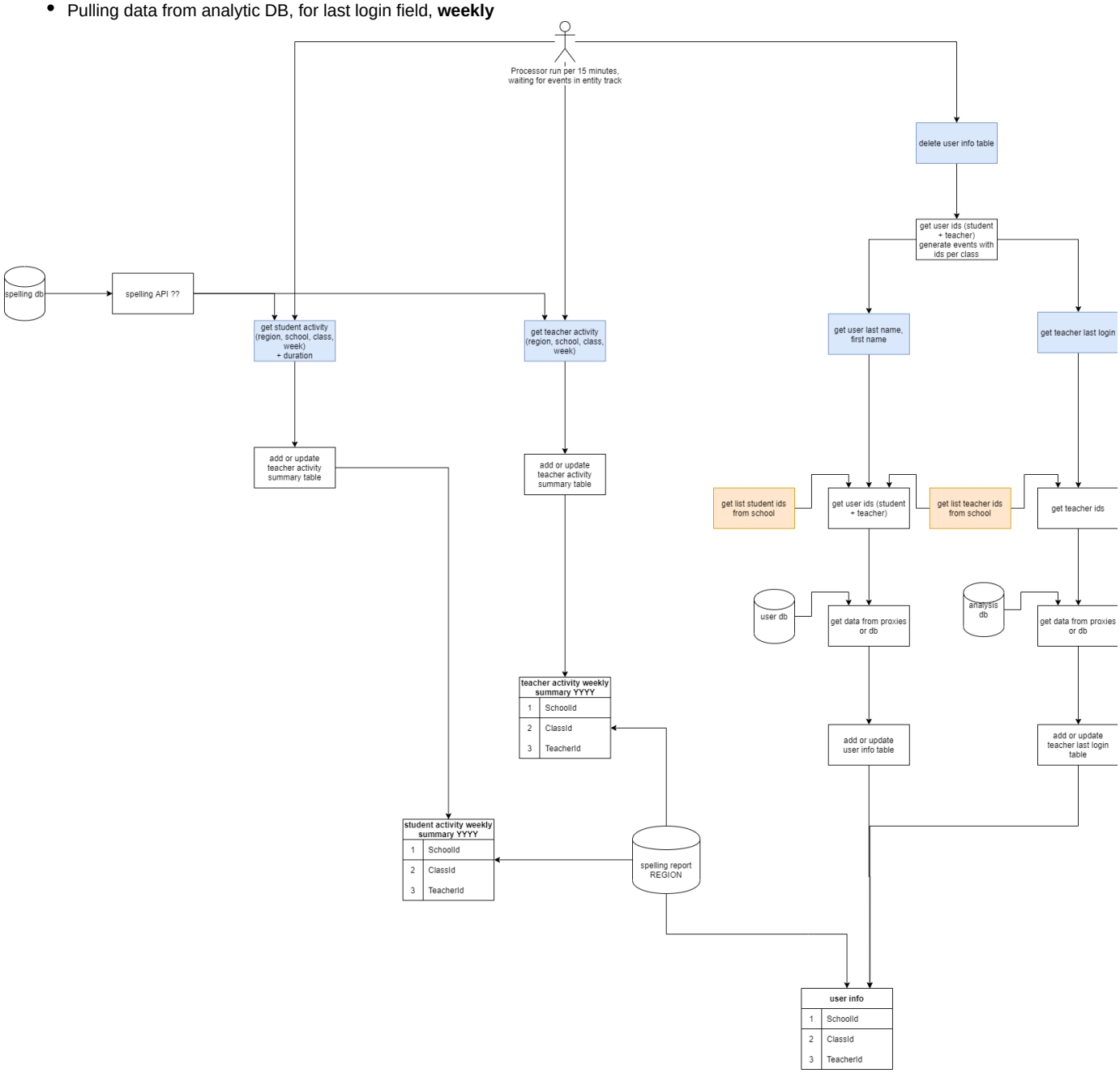
- **Data consolidation process:** Pulling data from different sources periodically (weekly for ActivityUsageReport) and save data into Spelling Report DB.
- **Data aggregation process:** On retrieved a report request, start building the report and save to CDN
- Manage a report request table to keep track of generated reports and their statuses. Additionally, maintain report request + artifacts life cycle
    - Generate new report request per school, do not generate new report request if existed
    - Cleanup these report request weekly
- Do we need to sync with REQUEST MANAGER API?

Data consolidation process

The diagram below depicts how we handling multiple data consolidation processes based-on entity track mechanism in Processor

- Run independently, pre-pulling data from multiple data sources
- Pulling data from spelling API, **weekly**

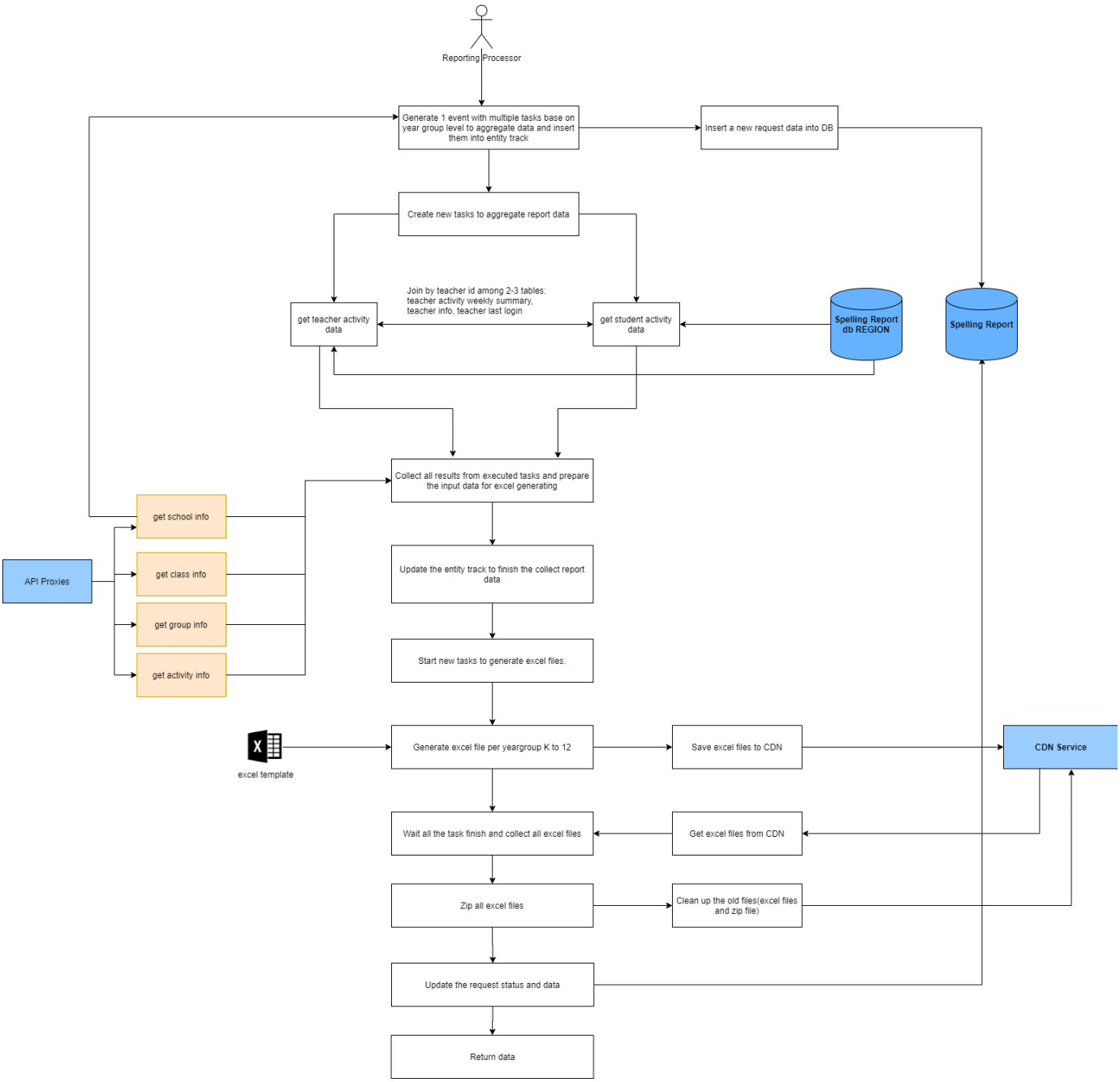
- Pulling data from user DB, **weekly**
- Pulling data from analytic DB, for last login field, **weekly**



Data aggregation process

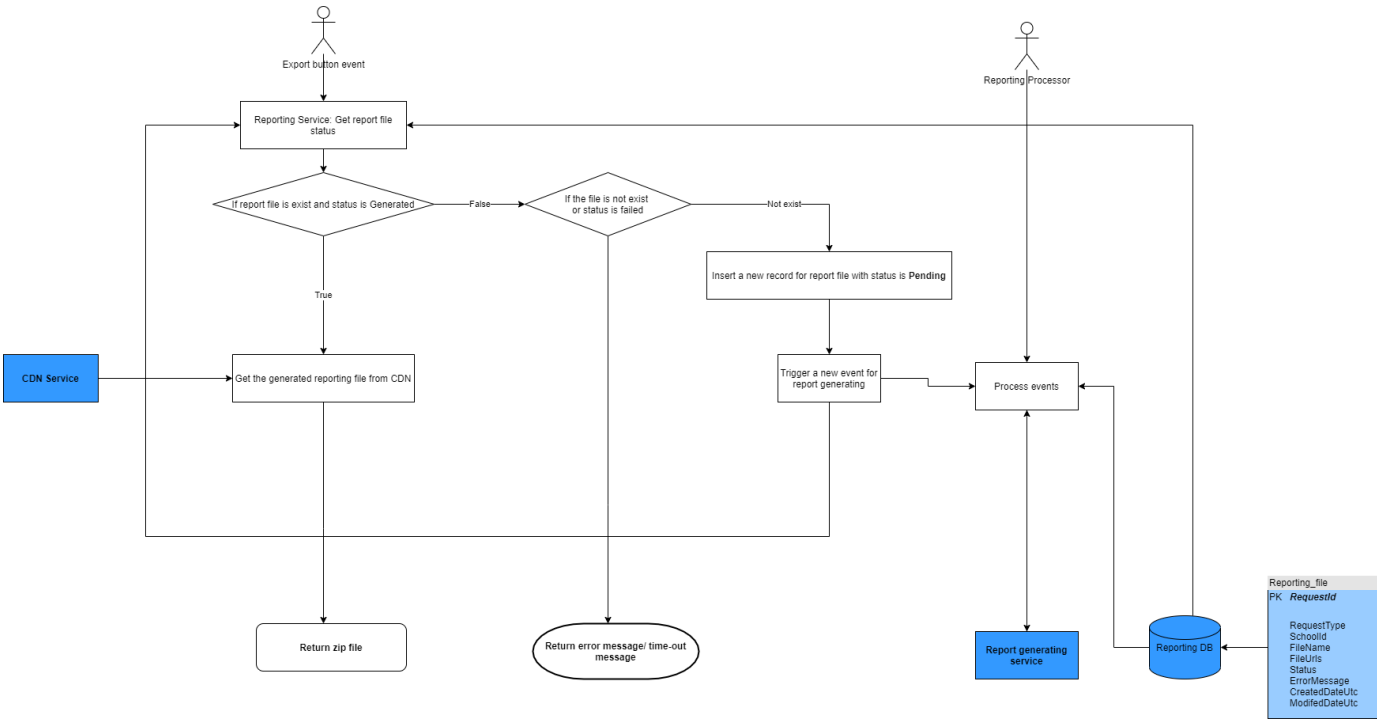
This diagram below depicts how we compose data from spelling report DB and build final excel/zip files for activity and usage report in Processor

- Build artifacts on demand
- Getting data from Spelling Report DB, compose excel file(s), zip file, upload to CDN
- update status (SUCCESS, ERROR), error reasons



How user trigger report request and retrieve the report file

- User click on a button to open a url in new tab
- Wait for report file ready
- return report file to the user as a stream, the user will be able to download the file immediately



Report request management

Manage a report request table to keep track of generated reports and their statuses. Additionally, maintain report request + artifacts life cycle

- Generate new report request per school, do not generate new report request if existed
- Cleanup these report request weekly
- Do we need to sync with REQUEST MANAGER API?