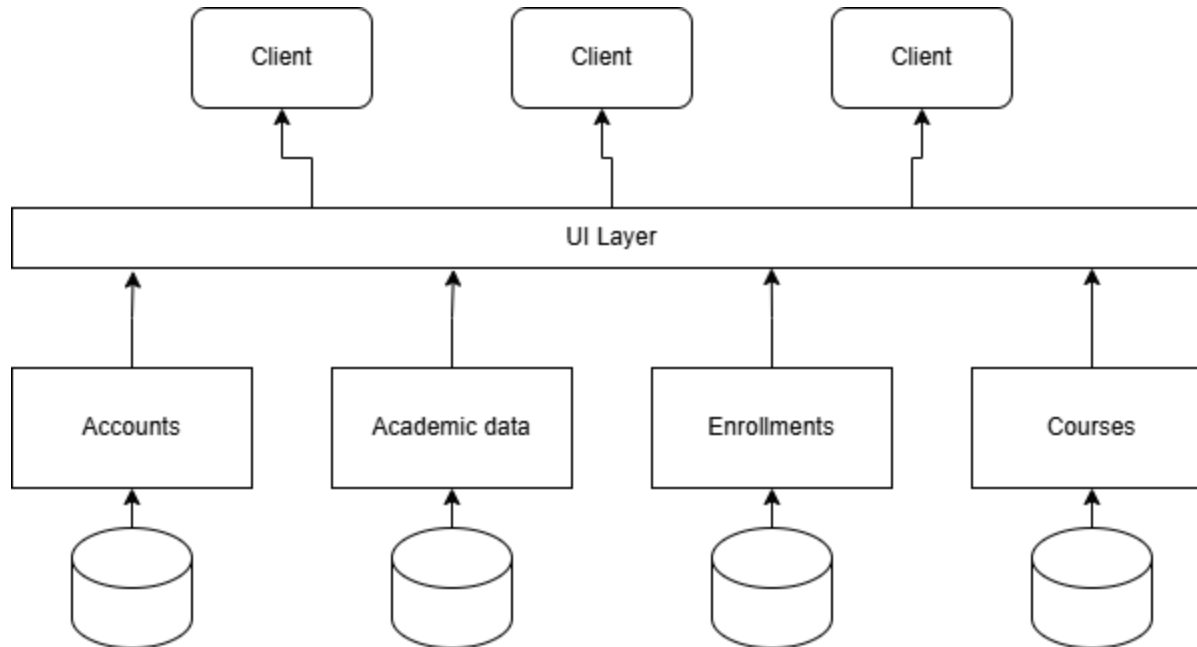


App development

System overview

- Each API connects to its own DB
- App (UI layer) connects to APIs
- Clients connects to App



Classic LAMP

- OS: Linux
- web server: Apache
- DB: MySQL
- Language: PHP

Classic LAMP explained

- Apache server can serve static or dynamic content
- static is just files
- dynamic content is generated using PHP and data from the DB
- e.g. a table that is populated with user information.

Modern JavaScript stacks

e.g., MEAN, MERN, etc.:

- web server: Express
- DB: MongoDB
- Language: JavaScript / Node.js
- Frontend framework:
 - Angular
 - React
 - etc.

MEAN stack explained

e.g.,

- Express server serves single page with JS
- JS code runs on the client side and makes API calls
 - Usually these are to back to the app server or an API gateway, rather than directly to microservices.
- JS code on the client side updates the page

Note

- LAMP vs. MEAN isn't as straightforward as it might seem.
 - LAMP servers can also serve JS.
 - MEAN servers can also generate dynamic HTML on the server-side.

Our App stack (FMP)

- Web server: Flask
- DB: MongoDB
- Language: Python
- Backend template engine: Jinja

How we will use our app stack

- We will generate dynamic HTML on the server-side
- We will avoid JavaScript
- These decisions are primarily made to stick to one language and keep things simple.

Sequence diagrams

- Sequence diagrams are a helpful tool to visualize interactions between the client, app server, and API(s).
- See 'notes1-sequence.md'