

Developer

The **Developer Experience**



The ***Developer Experience*** prioritises the specification and implementation of **robust, well documented, loosely coupled components & services**, integrated into a **coherent toolkit** open to contributions from **diverse skill sets**.



Application and Data (12)

- JavaScript
- Node.js
- TypeScript
- Firebase
- Tailwind CSS
- Svelte
- PostCSS
- Nunjucks

Skeleton

Showdown

SvelteKit

AG Grid

Utilities (3)

- Netlify
- Auth0
- Astro

DevOps (6)

- GitHub
- ESLint
- Prettier
- Cypress
- Vite
- Turborepo

<https://stackshare.io/tutors/tutors-core>

Tutors Architecture

Reference
Courses /
Tests

test



Command Line
Applications

apps



cli

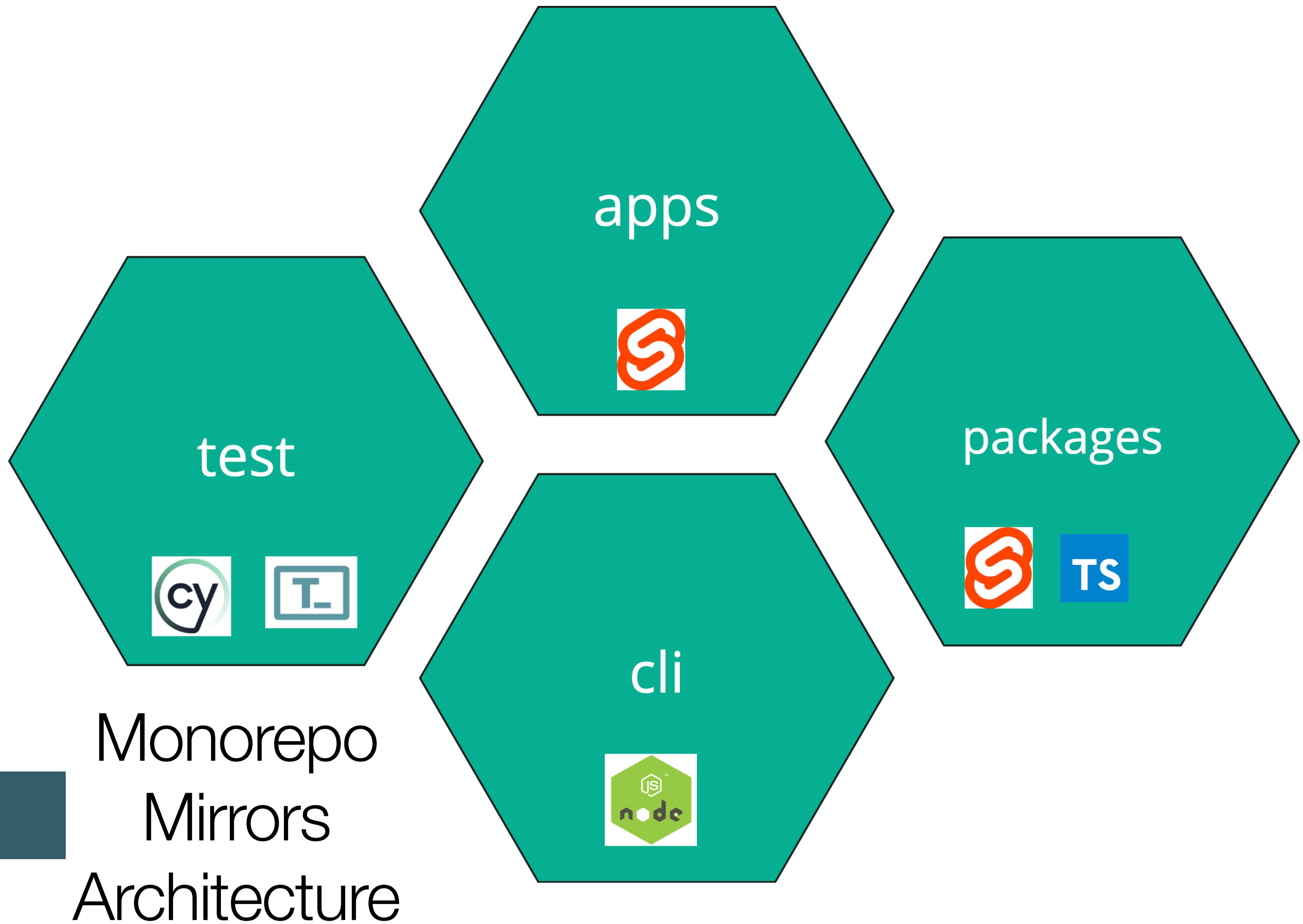
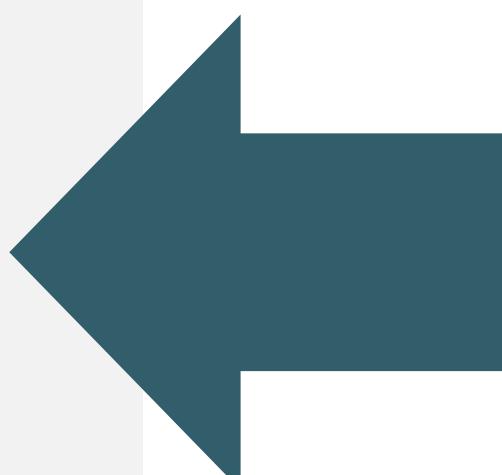
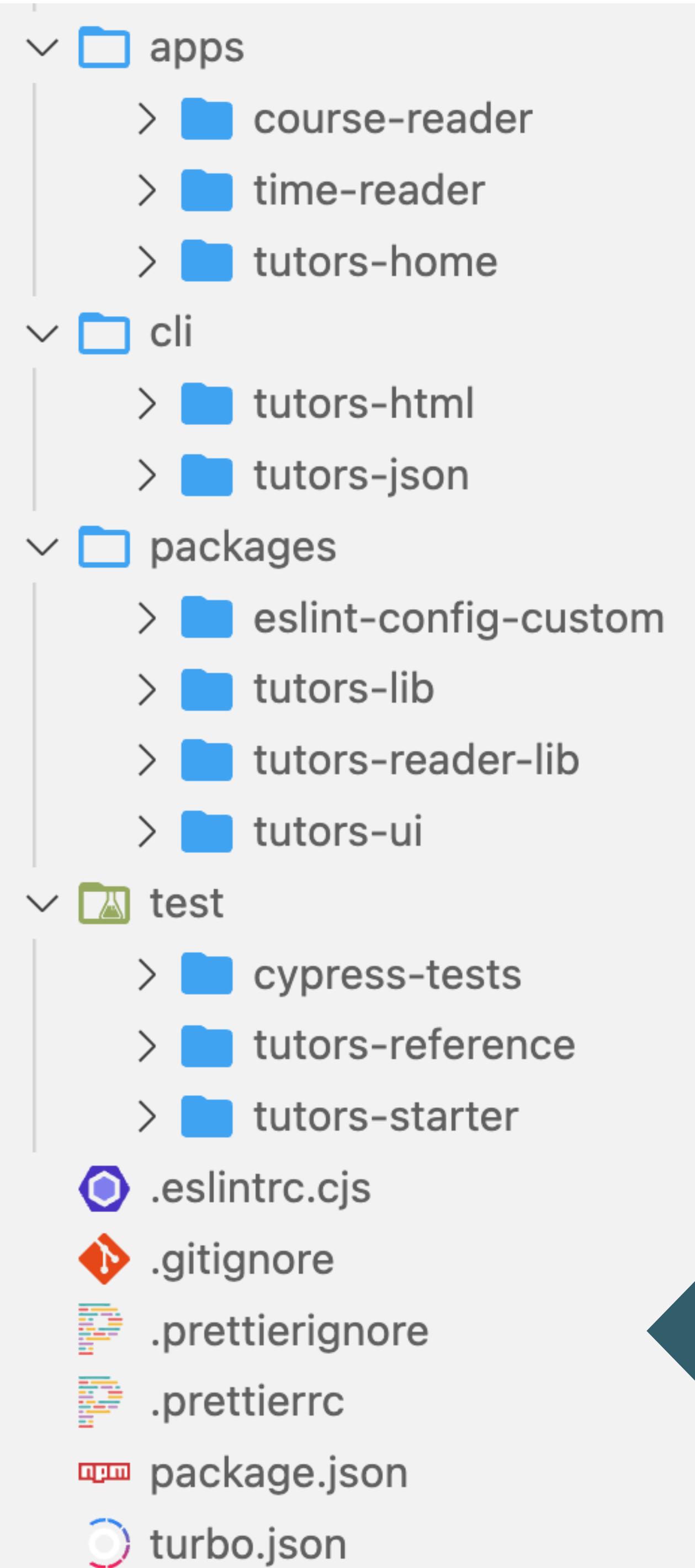


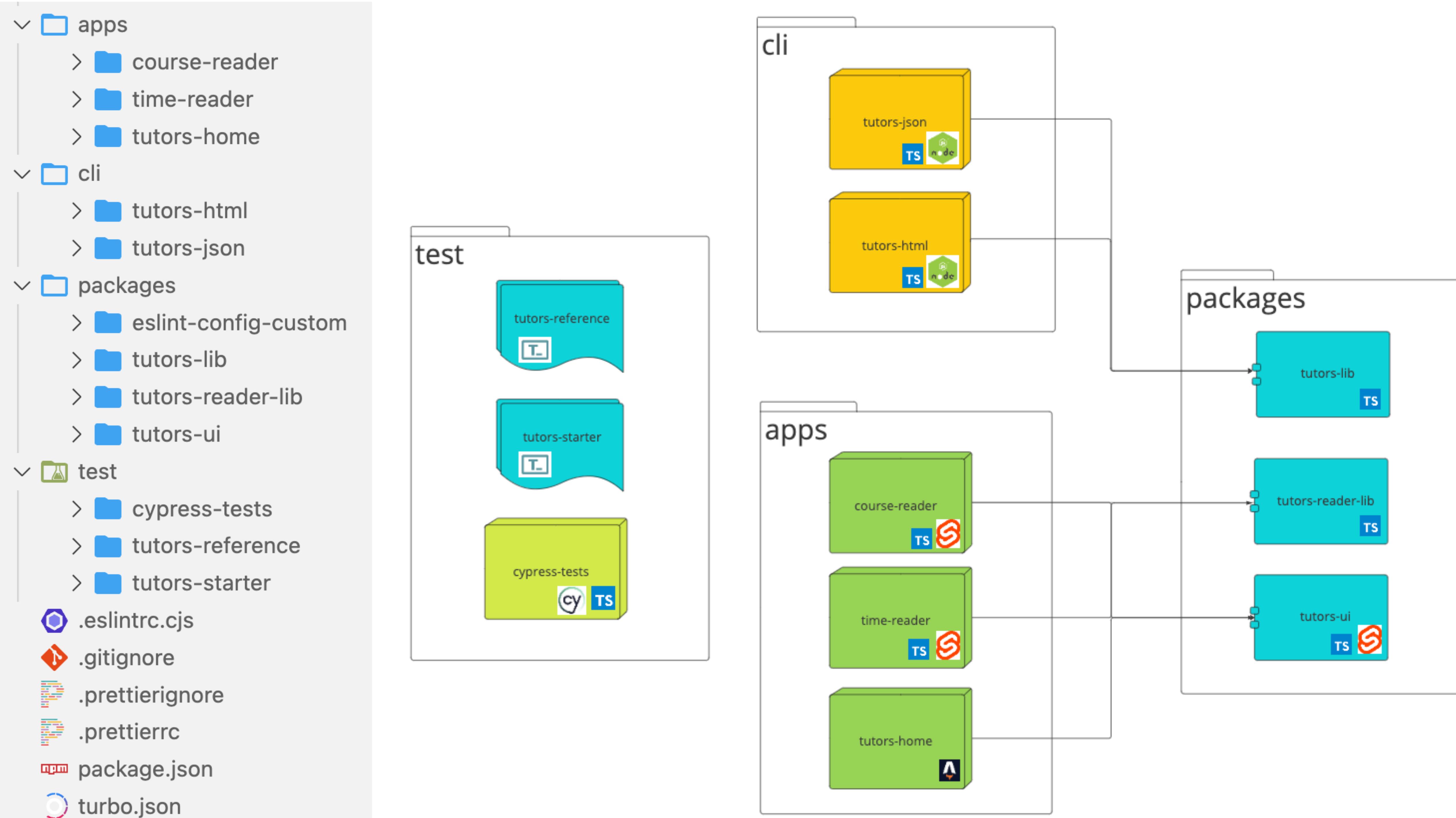
Web
Applications

packages



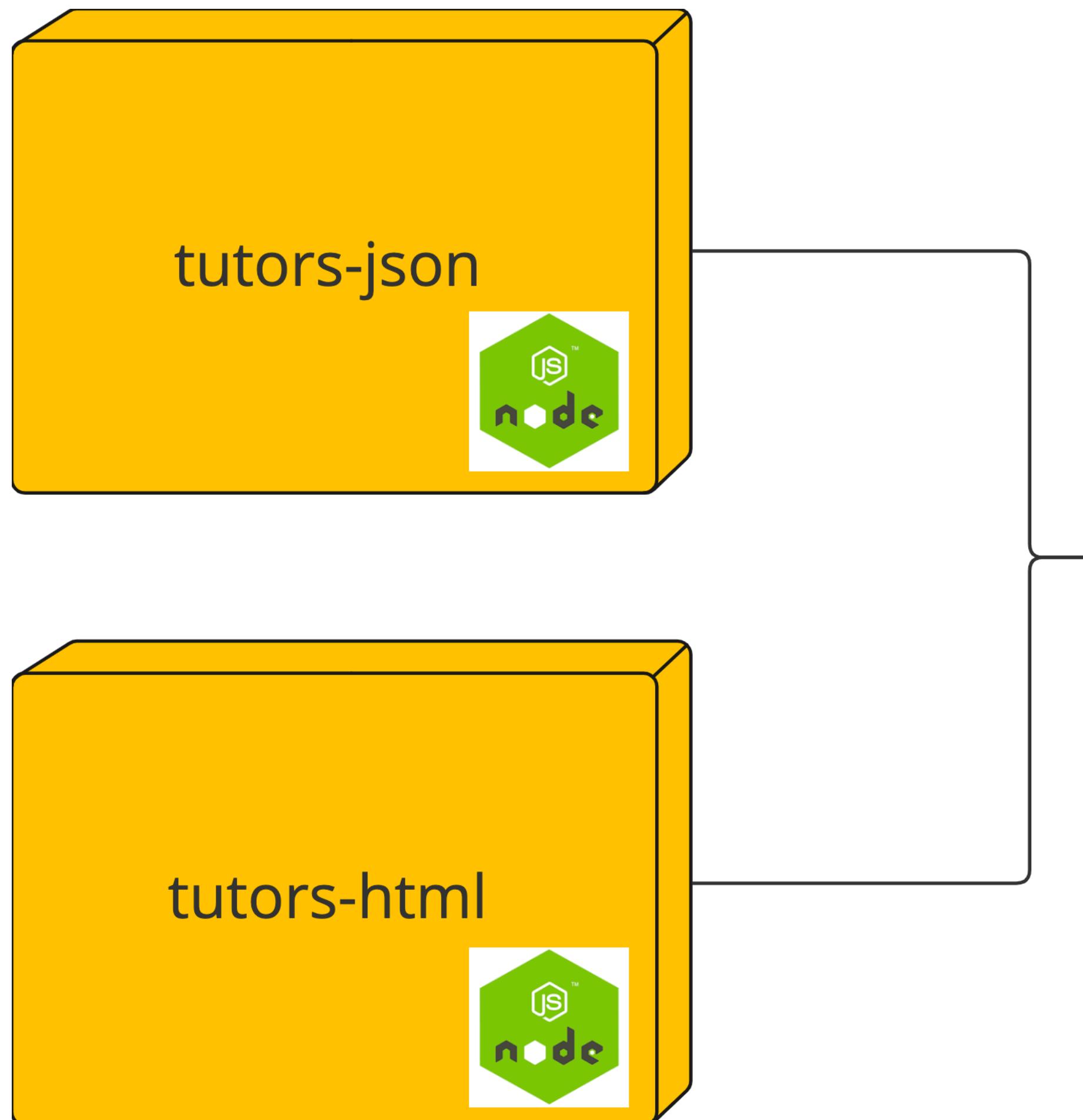
Shared
components





Node.js Command Line Applications

Generate JSON model for course



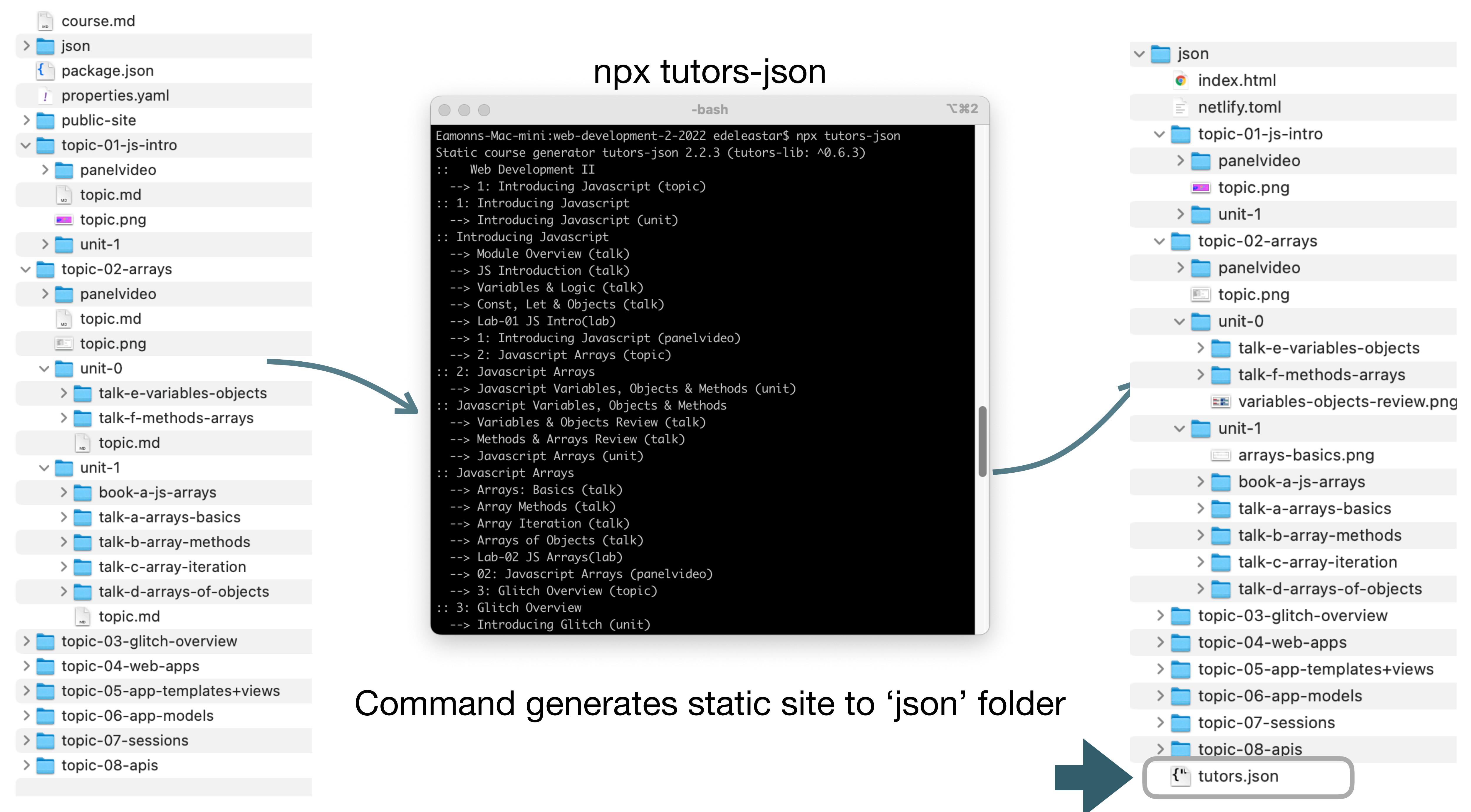
Generate HTML Static site for course

Course Memory Model

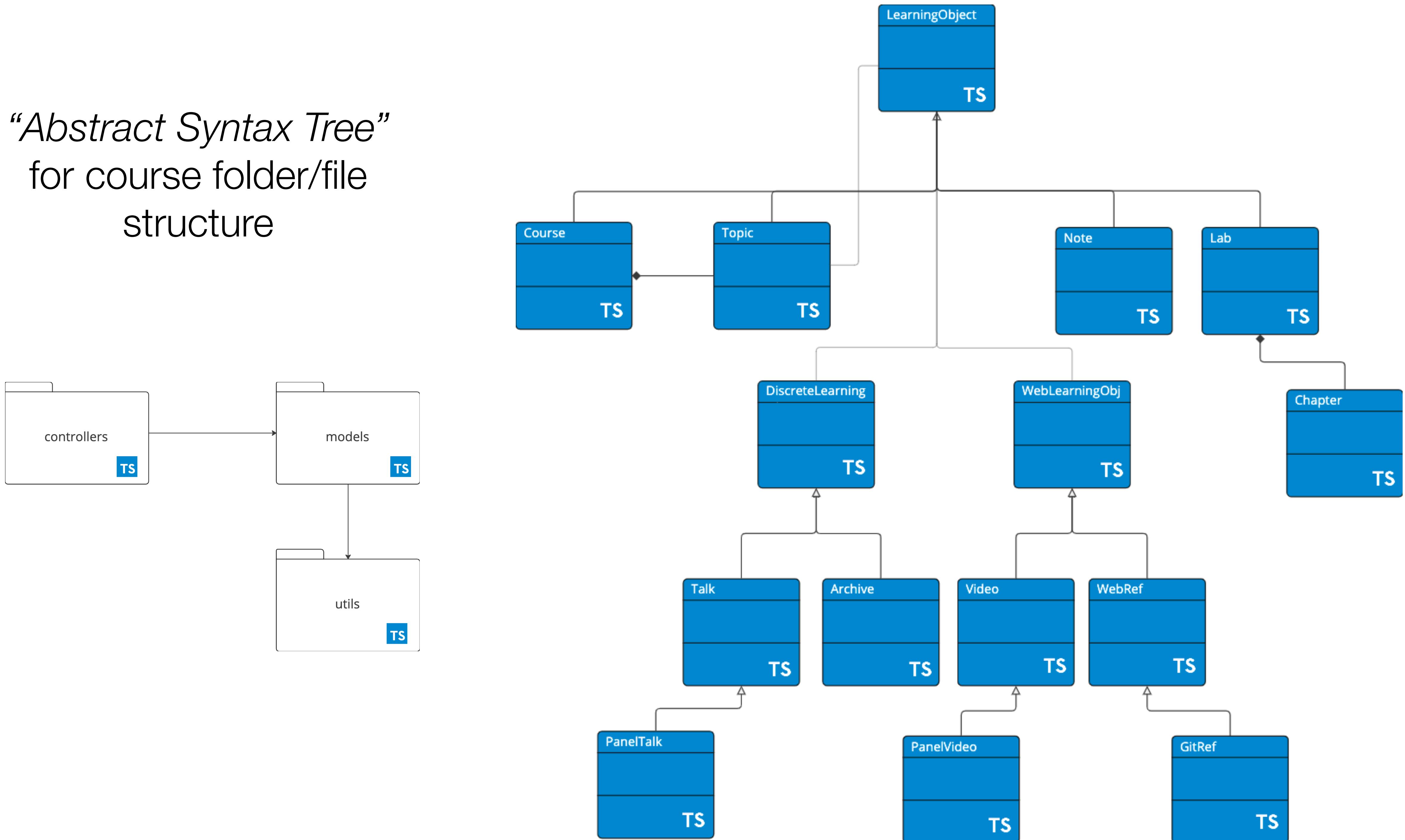
tutors-lib

TS

“Abstract Syntax Tree” for course folder/file structure



“Abstract Syntax Tree” for course folder/file structure



```

export class JsonEmitter {
  version = "0.0";

  emitLo(lo: LearningObject, url: string, jsonObj: any) { ... }
}

emitTalk(lo: Talk, url: string, jsonObj: any) { ... }

emitLab(lo: Lab, url: string, jsonObj: any) { ... }

emitNote(lo: Note, url: string, jsonObj: any) { ... }

emitPanelNote(lo: PanelNote, url: string, jsonObj: any) { ... }

emitPanelTalk(lo: PanelTalk, url: string, jsonObj: any) { ... }

emitArchive(lo: Archive, url: string, jsonObj: any) { ... }

emitPanelVideo(lo: PanelVideo, url: string, jsonObj: any) { ... }

emitUnit(lo: Unit, url: string, jsonObj: any) { ... }

emitTopic(lo: Topic, url: string, jsonObj: any) { ... }

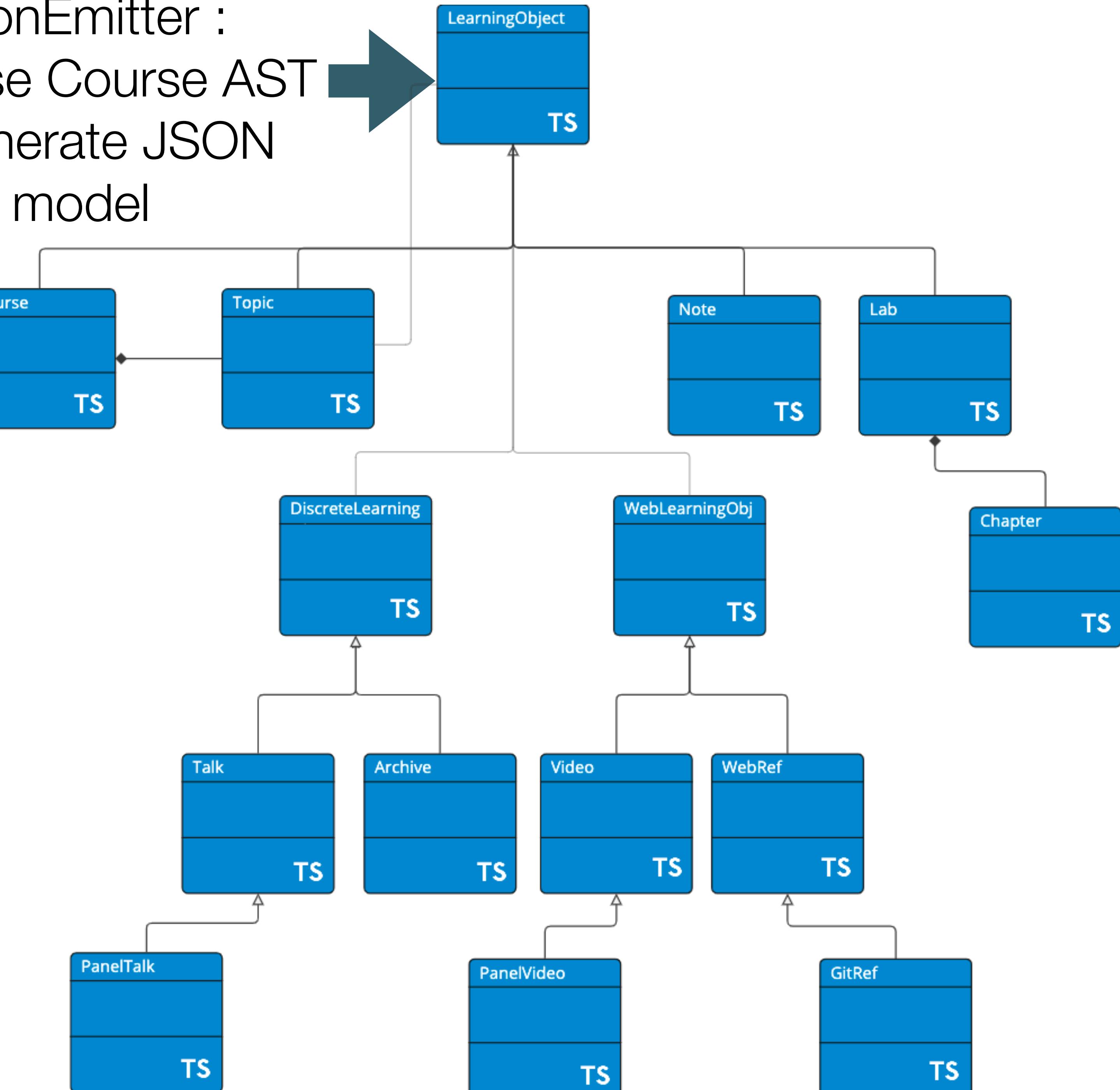
emitCourse(lo: Course, url: string, jsonObj: any) { ... }

generateCourse(version: string, path: string, course: Course) {
  const courseJson: any = {};
  courseJson.version = version.toString();
  this.emitCourse(course, "{{COURSEURL}}/", courseJson);
  writeFile(path, "tutors.json", JSON.stringify(courseJson));
}

```

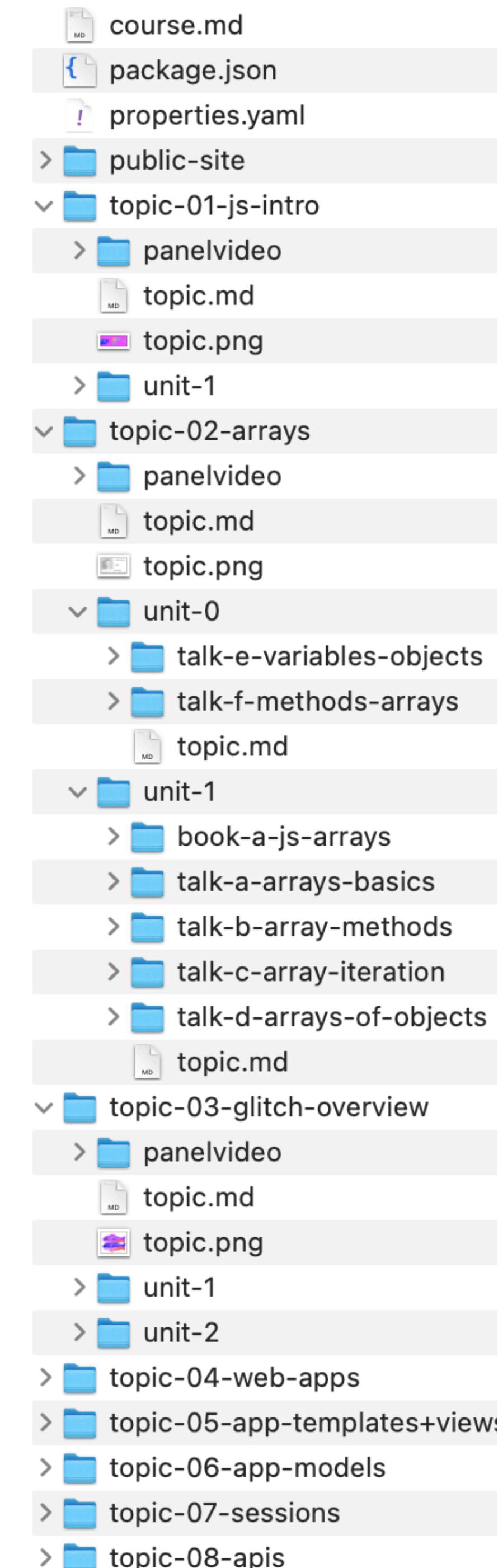
JsonEmitter :

Traverse Course AST
+ generate JSON model

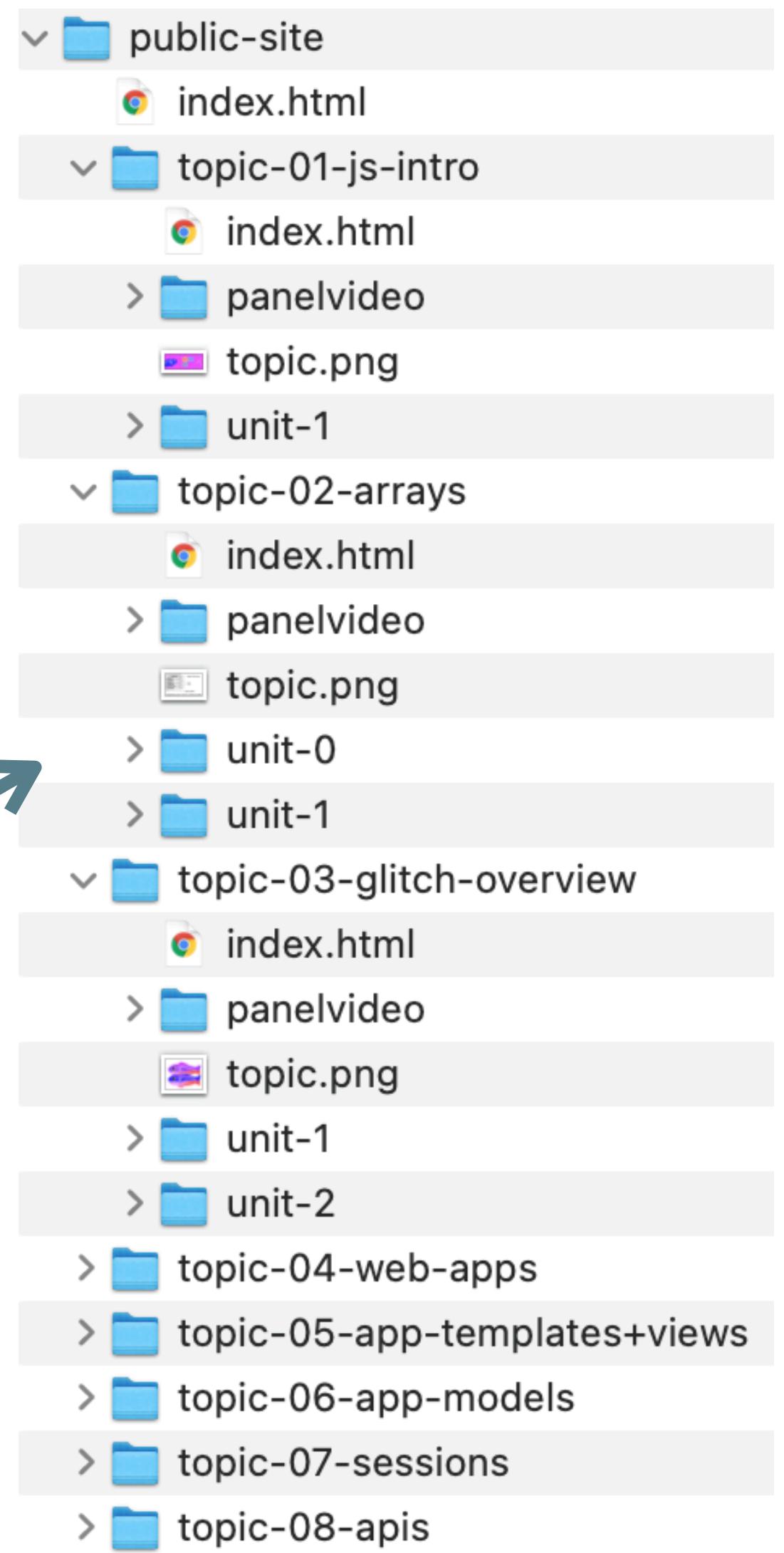


Static/HTML

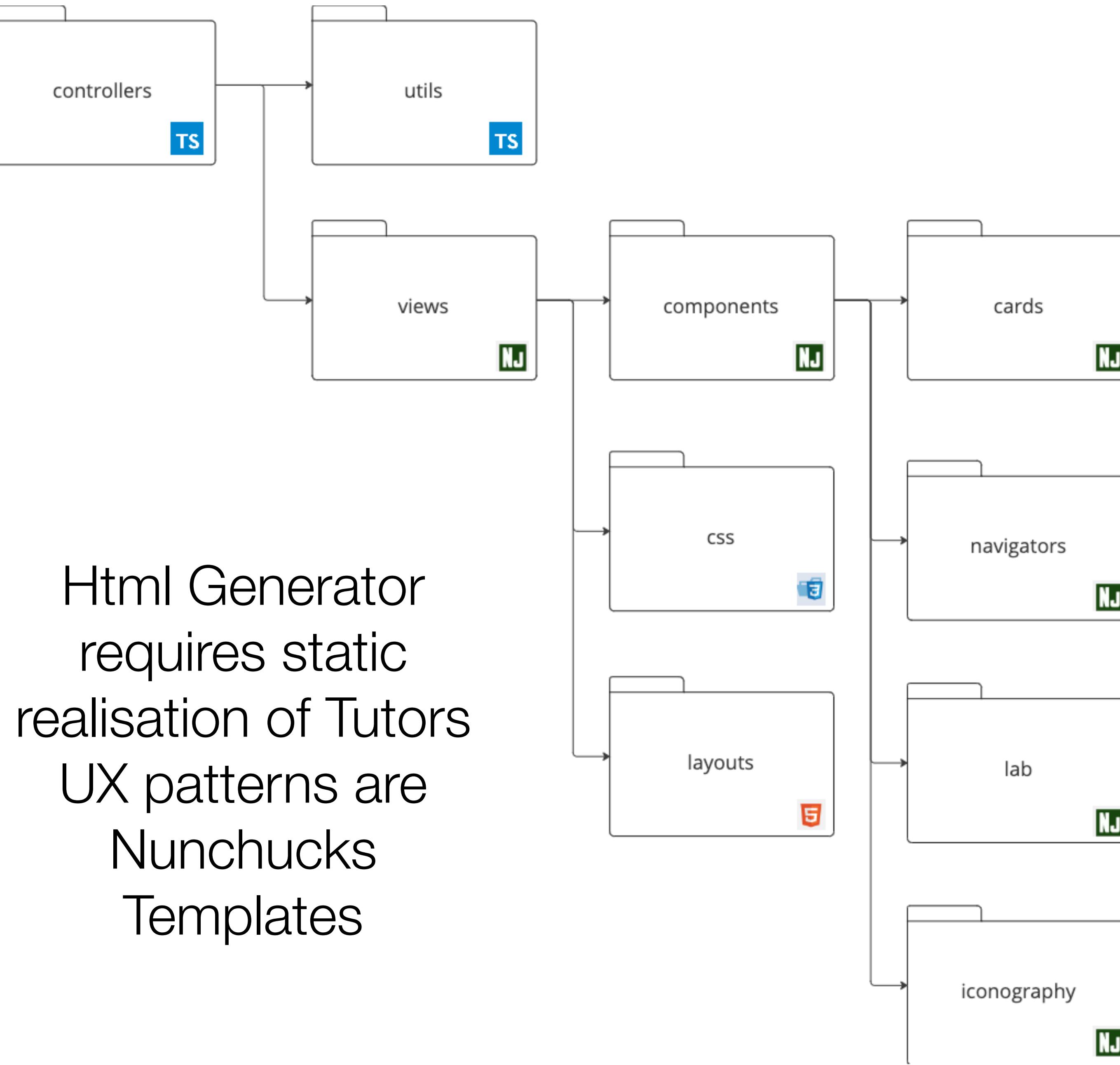
npx tutors-html



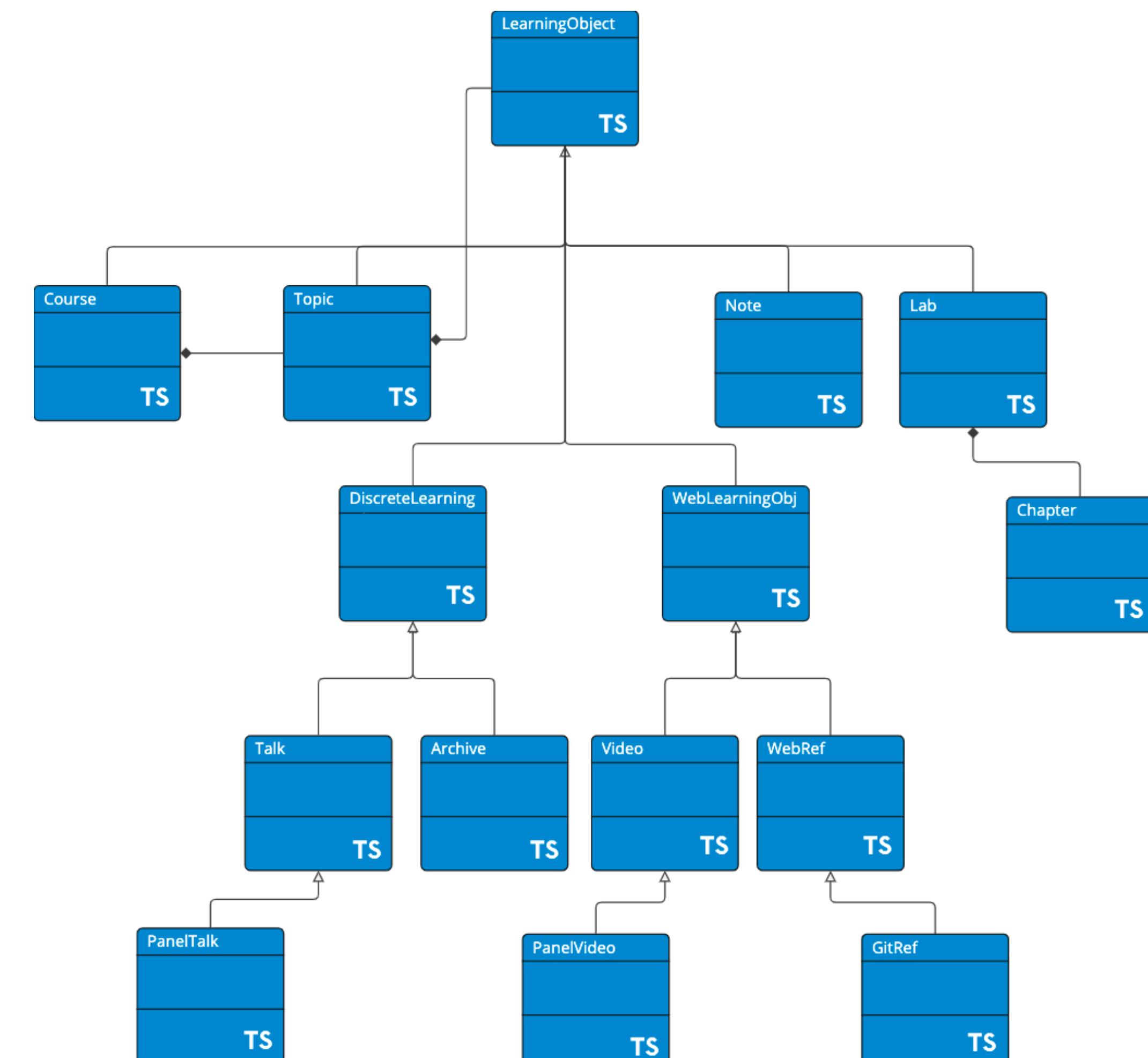
```
-bash
Eamonns-Mac-mini:web-development-2-2022 edeleastar$ npx tutors-html
Static course generator tutors-json 0.6.7 (tutors-lib: ^0.6.5)
::  Web Development II
  --> 1: Introducing Javascript (topic)
:: 1: Introducing Javascript
  --> Introducing Javascript (unit)
:: Introducing Javascript
  --> Module Overview (talk)
  --> JS Introduction (talk)
  --> Variables & Logic (talk)
  --> Const, Let & Objects (talk)
  --> Lab-01 JS Intro(lab)
  --> 1: Introducing Javascript (panelvideo)
  --> 2: Javascript Arrays (topic)
:: 2: Javascript Arrays
  --> Javascript Variables, Objects & Methods (unit)
:: Javascript Variables, Objects & Methods
  --> Variables & Objects Review (talk)
  --> Methods & Arrays Review (talk)
  --> Javascript Arrays (unit)
:: Javascript Arrays
  --> Arrays: Basics (talk)
  --> Array Methods (talk)
  --> Array Iteration (talk)
  --> Arrays of Objects (talk)
  --> Lab-02 JS Arrays(lab)
  --> 02: Javascript Arrays (panelvideo)
  --> 3: Glitch Overview (topic)
:: 3: Glitch Overview
```



Command generates static site to 'public-site' folder



Html Generator
requires static
realisation of Tutors
UX patterns are
Nunchucks
Templates



```

export class HtmEmitter {
  parser = new MarkdownParser();

  emitObjectives(lo: LearningObject) { ... }

  emitNote(note: Note, path: string) { ... }

  emitLab(lab: Lab, path: string) { ... }

  emitUnit(unit: Unit, path: string) { ... }

  emitLo(lo: LearningObject, path: string) { ... }

  emitTopic(topic: Topic, path: string) { ... }

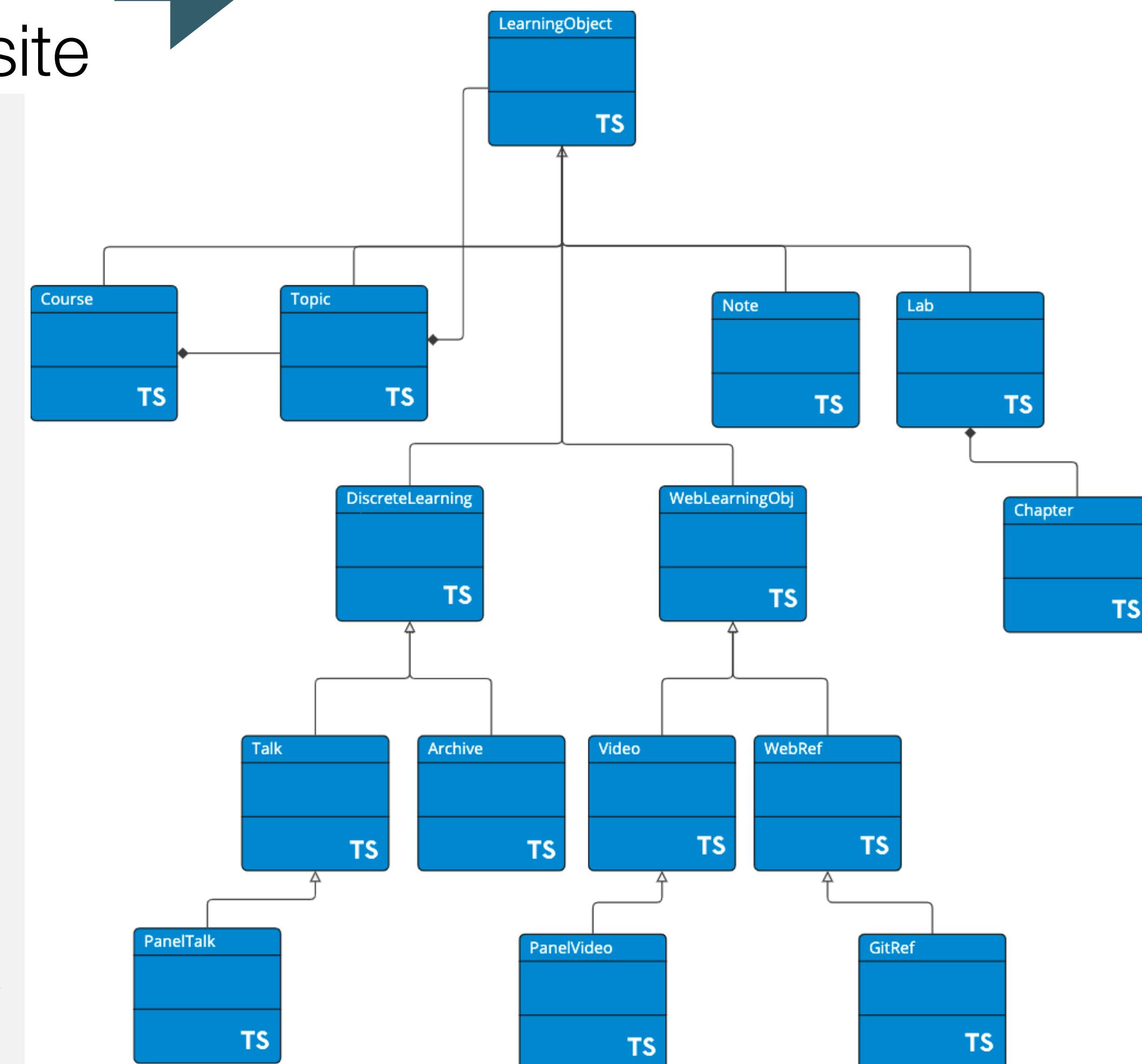
  emitCourse(course: Course, path: string) { ... }

  generateCourse(path: string, course: Course) {
    sh.cd(path);
    this.emitCourse(course, path);
  }
}

```

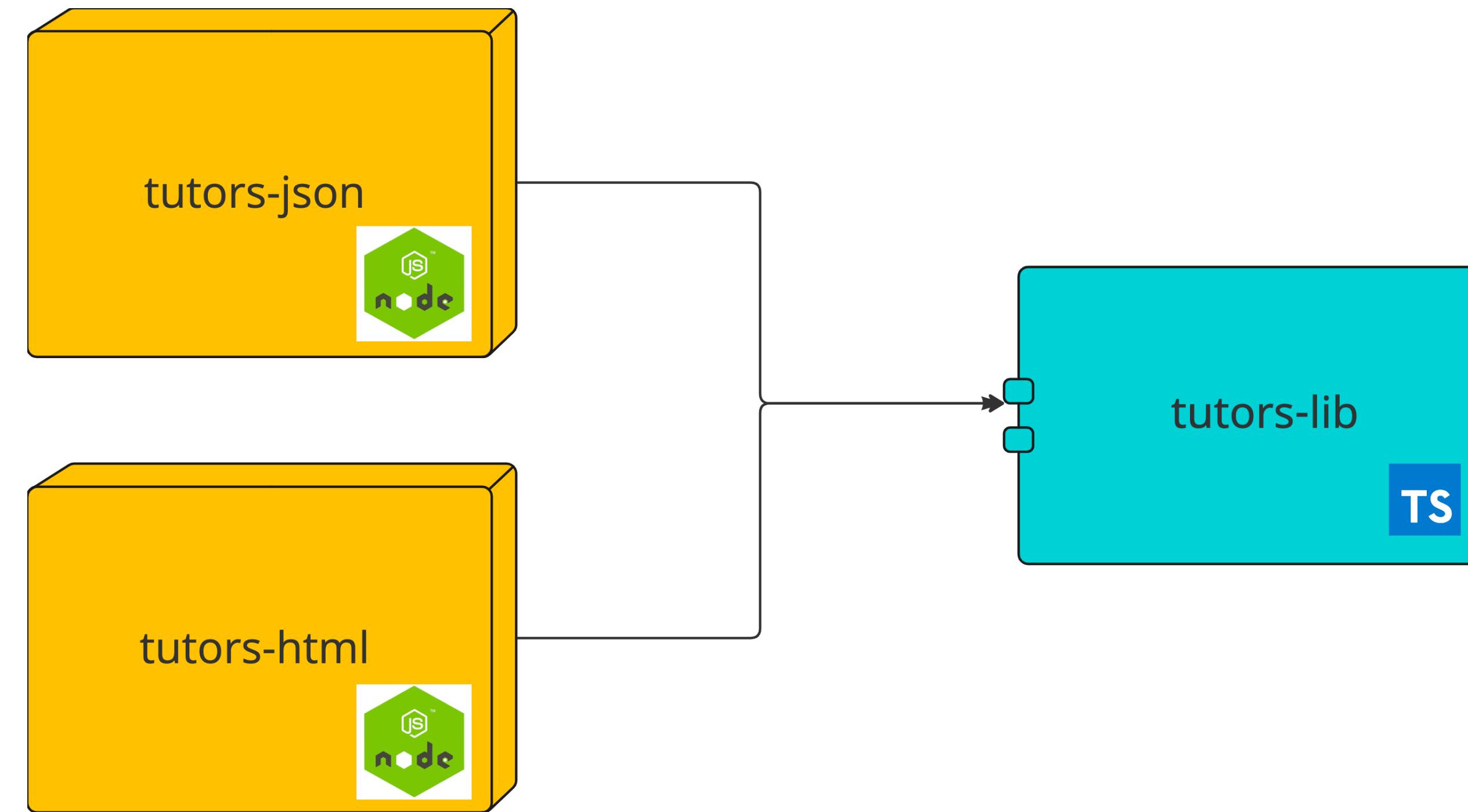
HtmEmitter : Traverse Course AST + generate Static site

components
 cards
 NJ Card.njk
 NJ CardDeck.njk
 NJ Image.njk
 NJ NoteCard.njk
 NJ PanelCards.njk
 NJ TalkCard.njk
 NJ UnitCard.njk
 NJ url.njk
 NJ VideoCard.njk
 iconography
 NJ Icon.njk
 NJ styles.njk
 lab-component
 NJ nav-bar.njk
 NJ nav-body.njk
 navigators
 support
 NJ MainNavigator.njk
 css
 layouts
 main.html
 NJ Course.njk
 NJ Lab.njk
 NJ Note.njk
 NJ Topic.njk



```
✓ tutotrs-json
  ✓ src
    ✓ controllers
      TS json-emitter.ts
      TS netlify.ts
      TS tutotrs-json.ts
    .eslintrc.js
    package.json
    tsconfig.json

  ✓ tutotrs-html
    ✓ src
      ✓ controllers
        TS html-emitter.ts
      ✓ utils
        TS markdown-parser.ts
        TS markdown-toc-lib.ts
      ✓ views
        > components
        > css
        > layouts
        NJ Course.njk
        NJ Lab.njk
        NJ Note.njk
        NJ Topic.njk
      TS tutotrs-html.ts
    .eslintrc.js
    package.json
    tsconfig.json
```



```
✓ tutotrs-lib
  ✓ src
    ✓ models
      TS course.ts
      TS lab.ts
      TS lo.ts
      TS los.ts
      TS note.ts
      TS topic.ts
      TS web-los.ts
    ✓ utils
      TS futils.ts
      TS loutils.ts
      TS properties.ts
      TS videoutils.ts
    .eslintrc.js
    package.json
    tsconfig.json
```

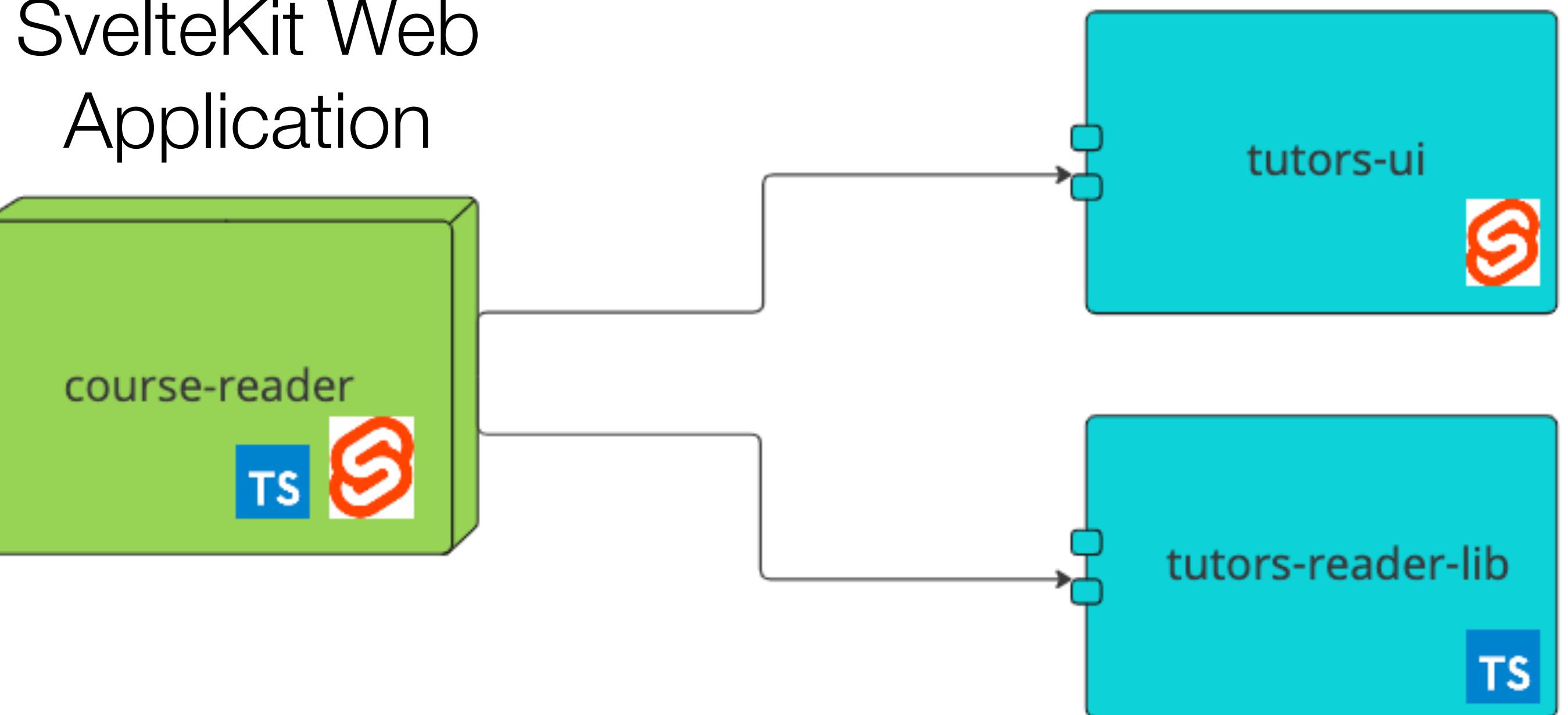
UX Components

The screenshot shows a web application interface. At the top left is a logo with a gear and the text "Web Development II". To its right is the title "Eamonn de Leistar, WIT Computing". On the far right of the header are search, layout, and user profile icons. Below the header is a navigation bar with links for "Home" and "Web Development...".

The main content area displays a grid of eight cards, each representing a module:

- 1: Introducing Javascript**: An introduction to the very basics of the Javascript Language. Includes a pink background icon with letters A, P, I.
- 2: Javascript Arrays**: Exploring Javascript arrays in detail. Includes a diagram of an array structure labeled "myCar" with elements "av", "d", "ick", "coln", "ick".
- 3: Glitch Overview**: Overview of the module + introduction to the Glitch platform. Includes an icon of two fish.
- 4: Web Applications**: Build your first Glitch app, a simple static playlist web site. Includes an icon of geometric shapes.
- 5: Templates & Routes**: Explore templating in more detail. Enhanced the routing behaviour. Includes an icon of three gears.
- 6: Forms**: Display a form, accept user input. Includes a diagram showing "User" interacting with "Model" and "Controller".
- 7: Sessions**: Implement user account management using sessions. Includes a diagram of session components like "HttpSession", "Cookies", "Hidden Field", and "URL-Rewriting".
- 8: Weather Top V2**: Solution · Specification · APIs · Charts. Includes an icon of a sun and clouds.

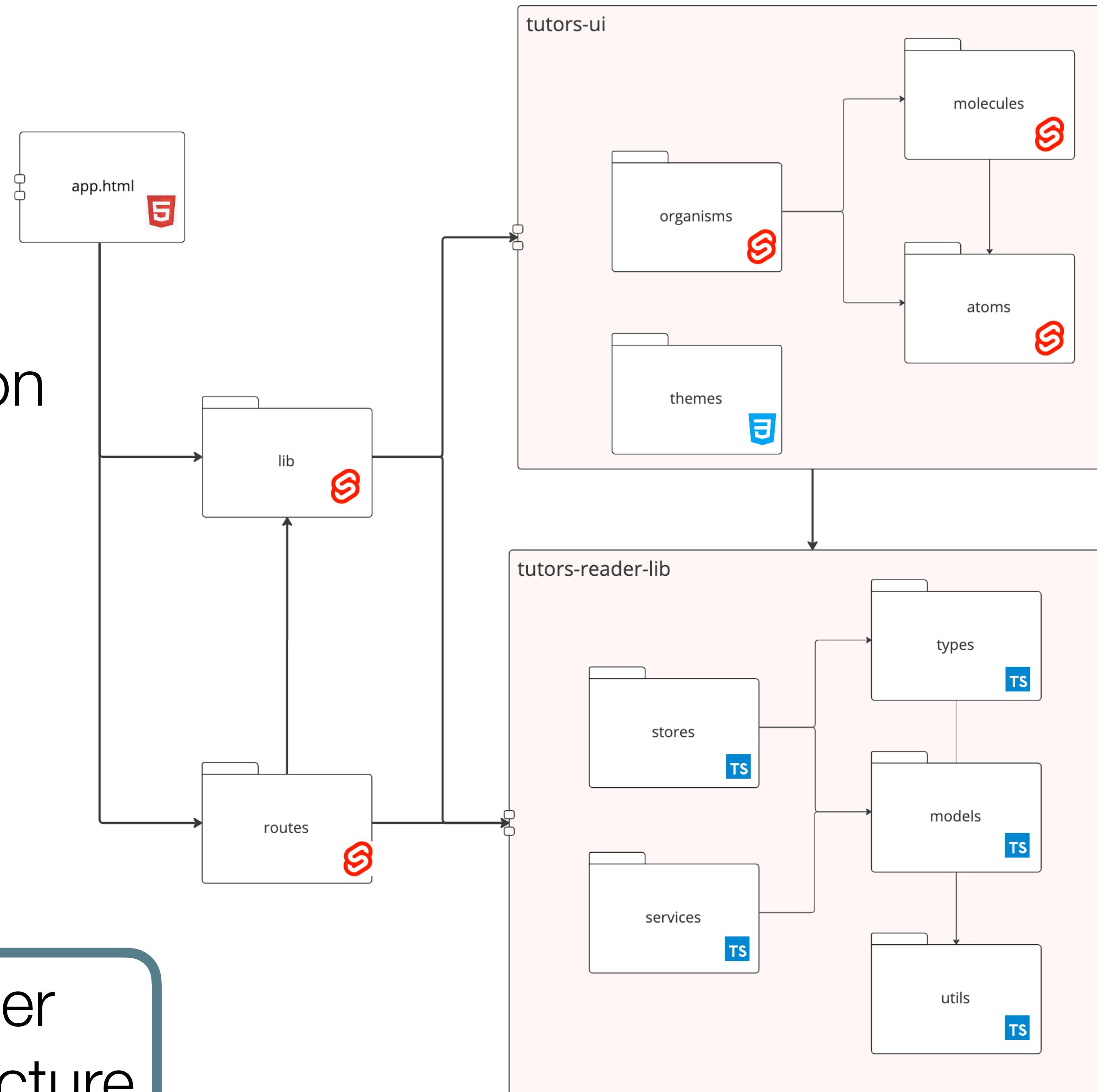
SvelteKit Web Application



Course, Analytics,
Metrics & Presence
Models

UI Components

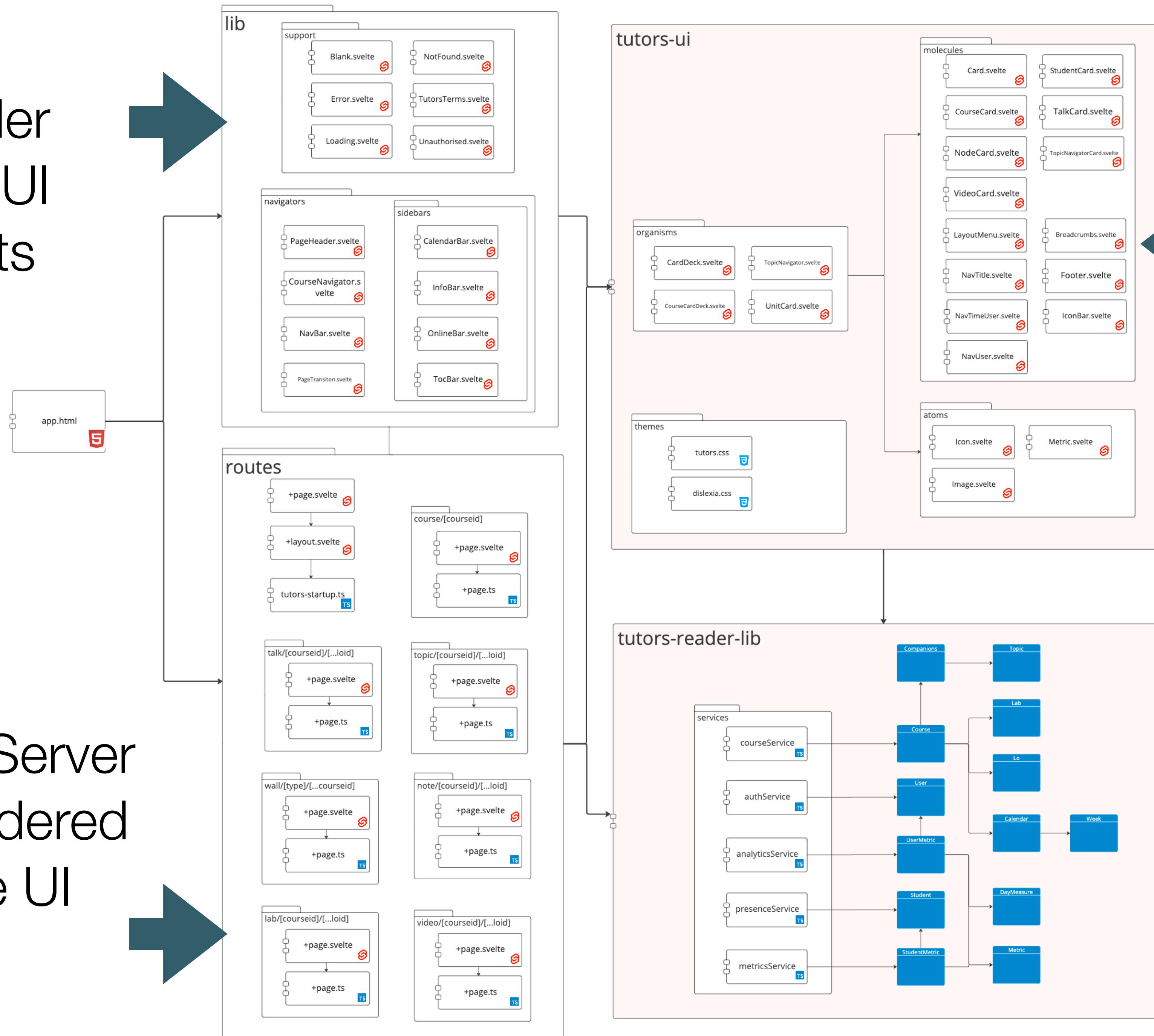
Application



Course Reader
Application Structure

Course
Models

lib: reader specific UI elements

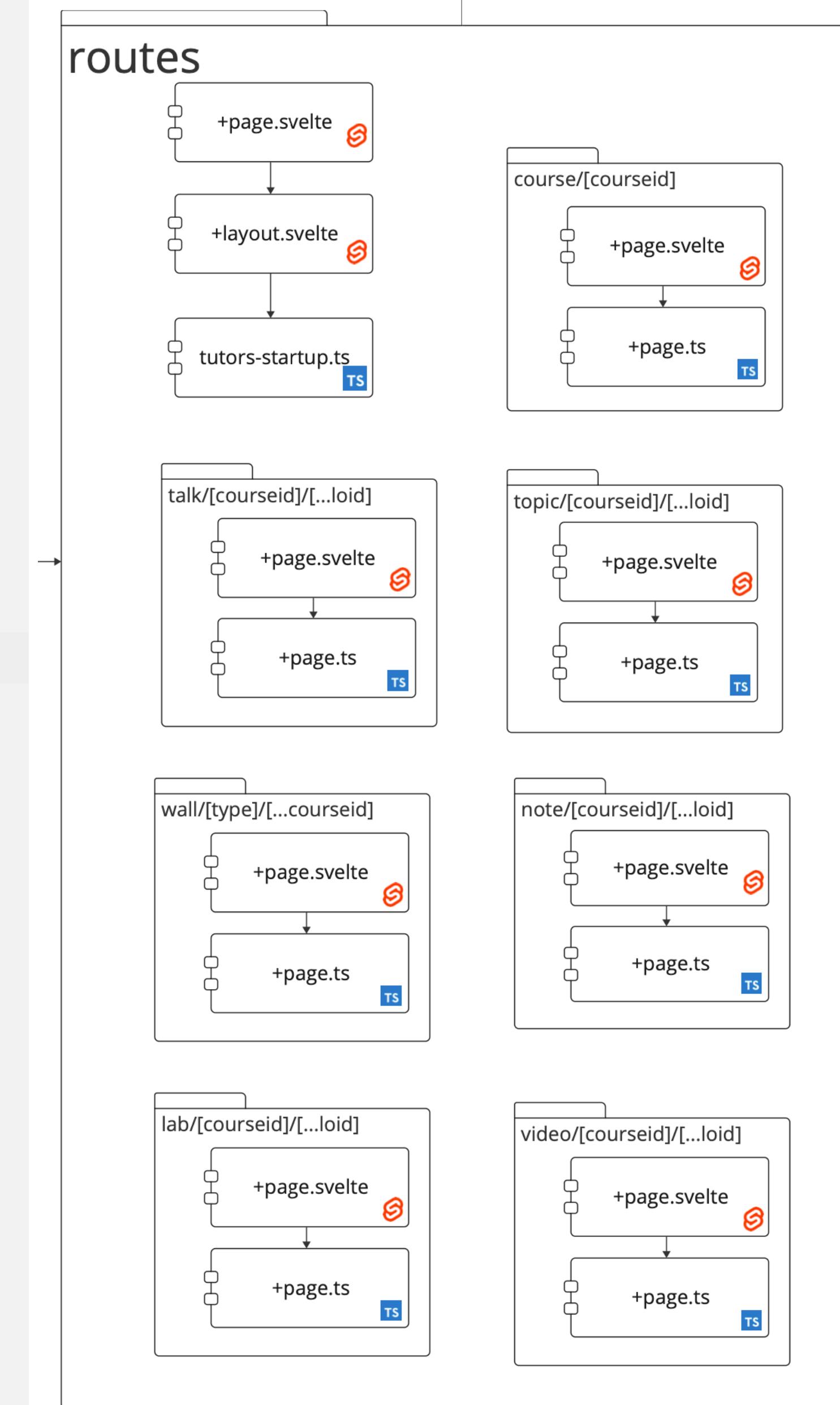
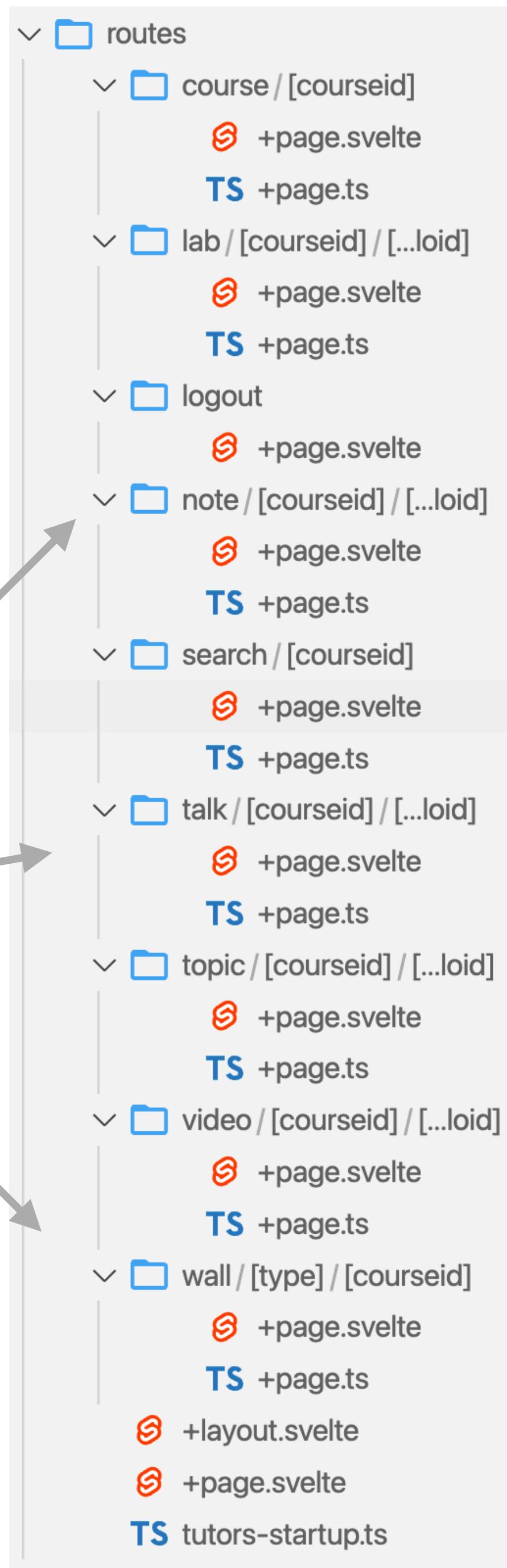


routes : Server Side Rendered Course UI

tutors-ui: Svelte Components reusable across multiple tutors Apps

tutors-reader-lib: Course Models + Analytics & Presence Engines

File based
routing:
Learning object
file patterns

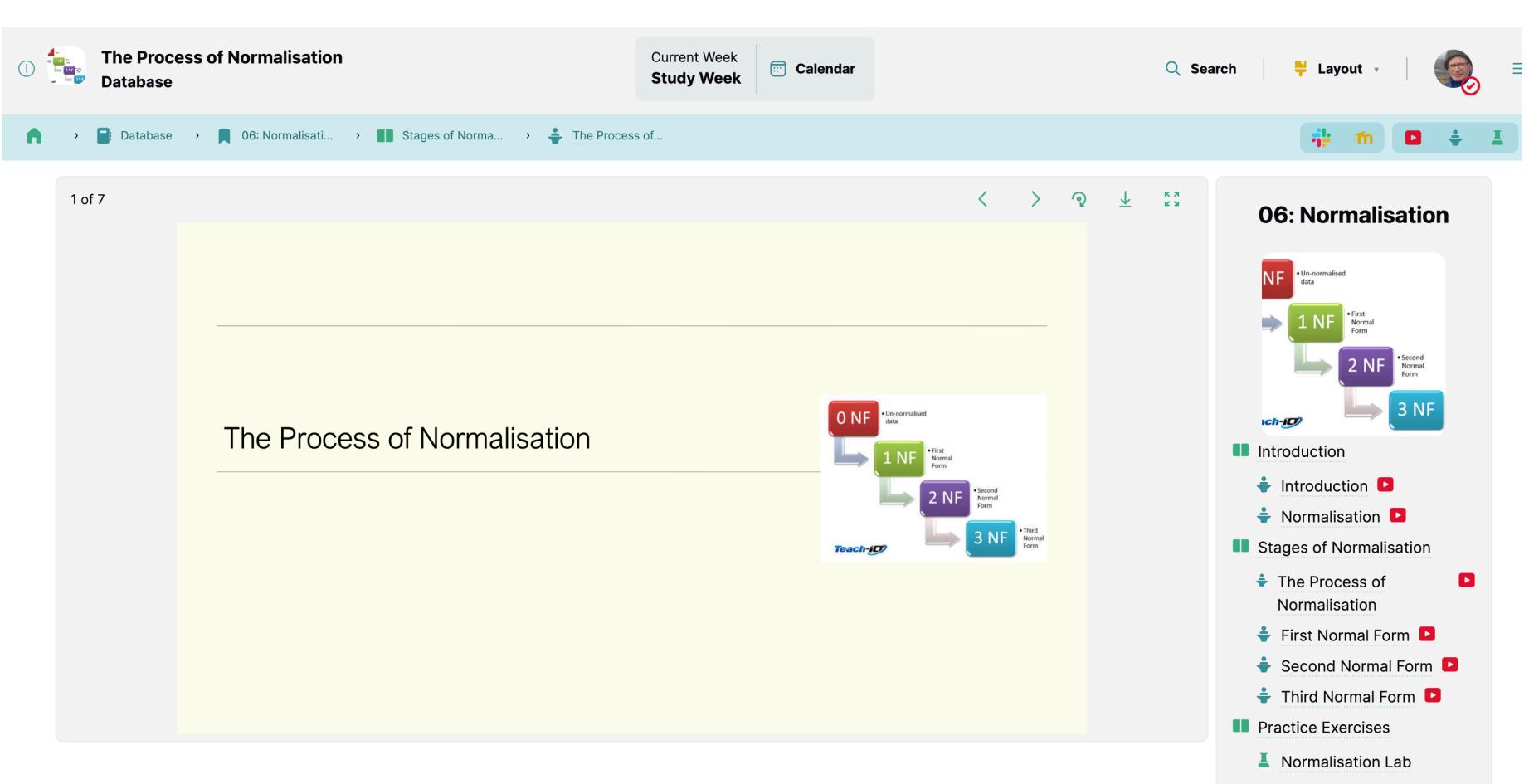


routes : Server
Side Rendered
Course UI

Server Side Rendering

Component
executed on the
server...

routes/talk/+page.svelte



```
<script lang="ts">
  import type { PageData } from "./$types";
  import { TopicNavigatorCard, TalkCard } from "tutors-ui";
  export let data: PageData;
</script>

#{if data.lo}
  <div class="flex w-11/12 mx-auto">
    <div class="w-full">
      #{key data.lo}
        <TalkCard lo="{data.lo}" />
      #{/key}
    </div>
    <div class="hidden md:block">
      <TopicNavigatorCard topic="{data.lo.parent}" />
    </div>
  </div>
{/if}
```

routes/talk/+page.ts

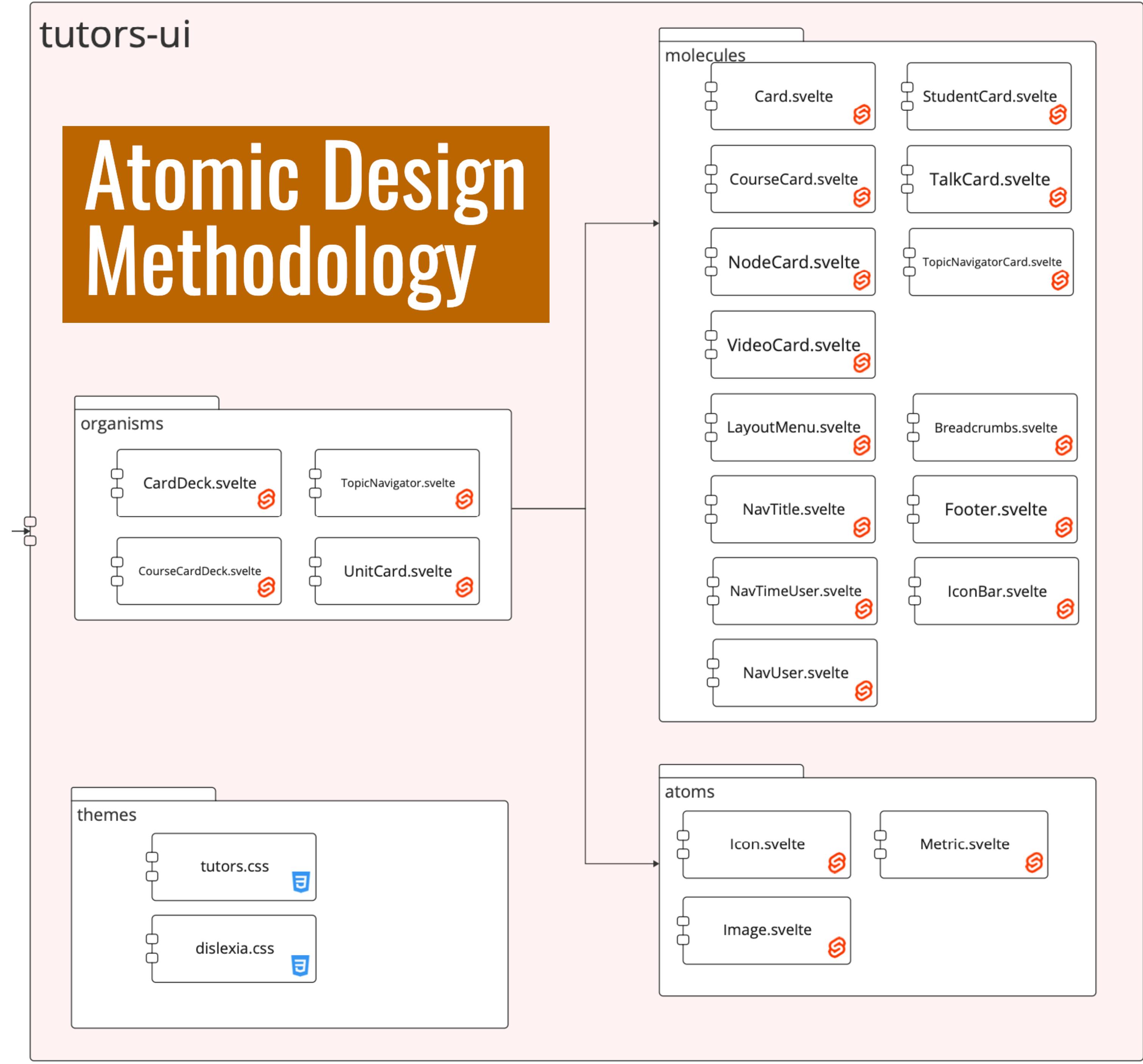
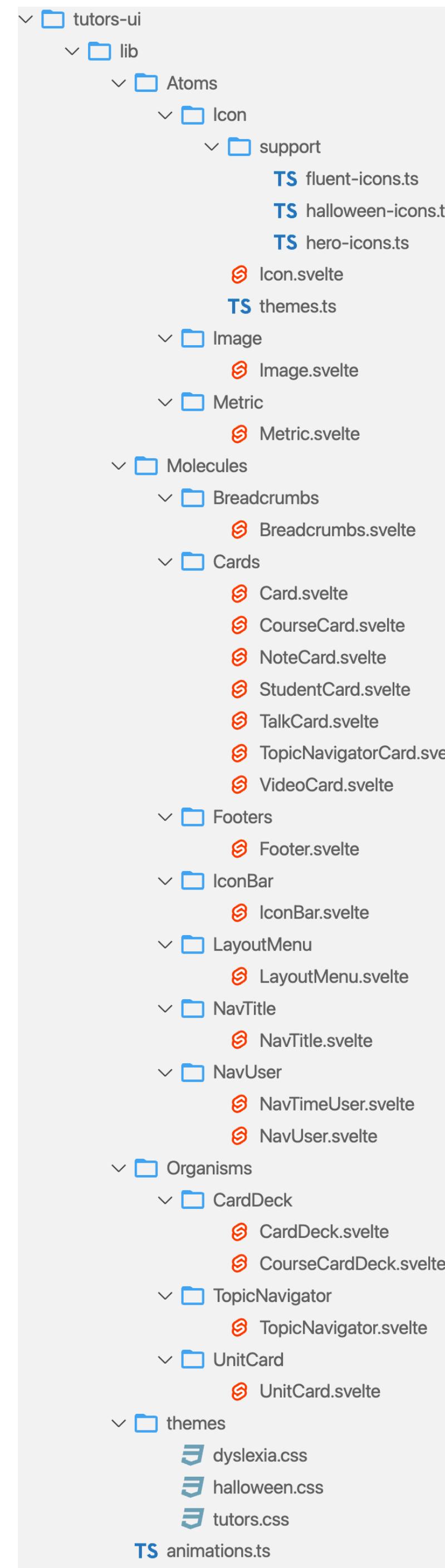
```
import type { PageLoad } from "./$types";
import { courseService } from "tutors-reader-lib/src/services/course-service";

export const load: PageLoad = async ({ url, params }) => {
  const lo = await courseService.readLo(params.courseid, url.pathname);
  return {
    lo: lo
  };
};
```

... and then
rendered to the
browser.

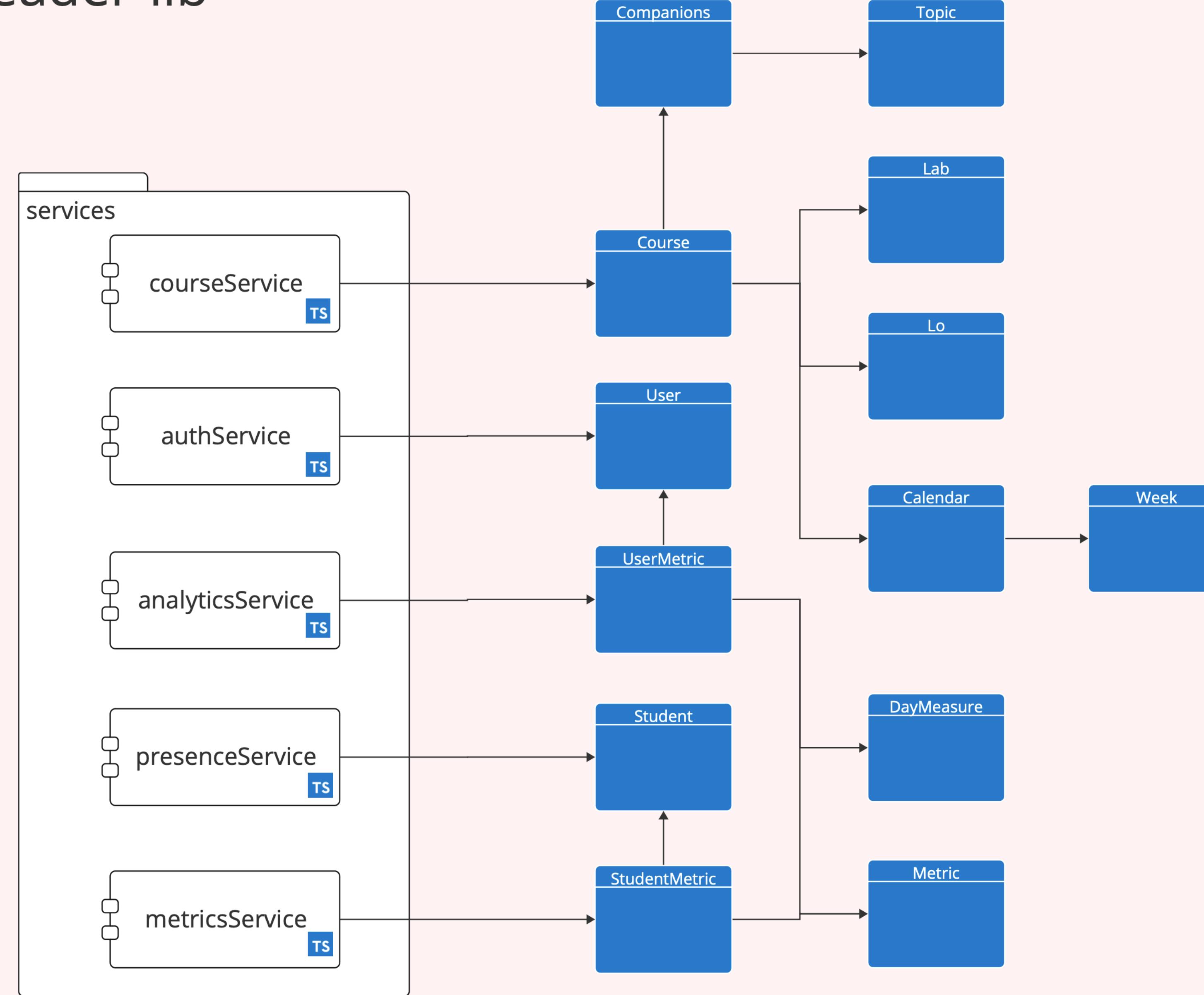
Subsequent
'hydration'
of new
content in the
browser

Important benefits
for performance &
SEO



tutors-reader-lib

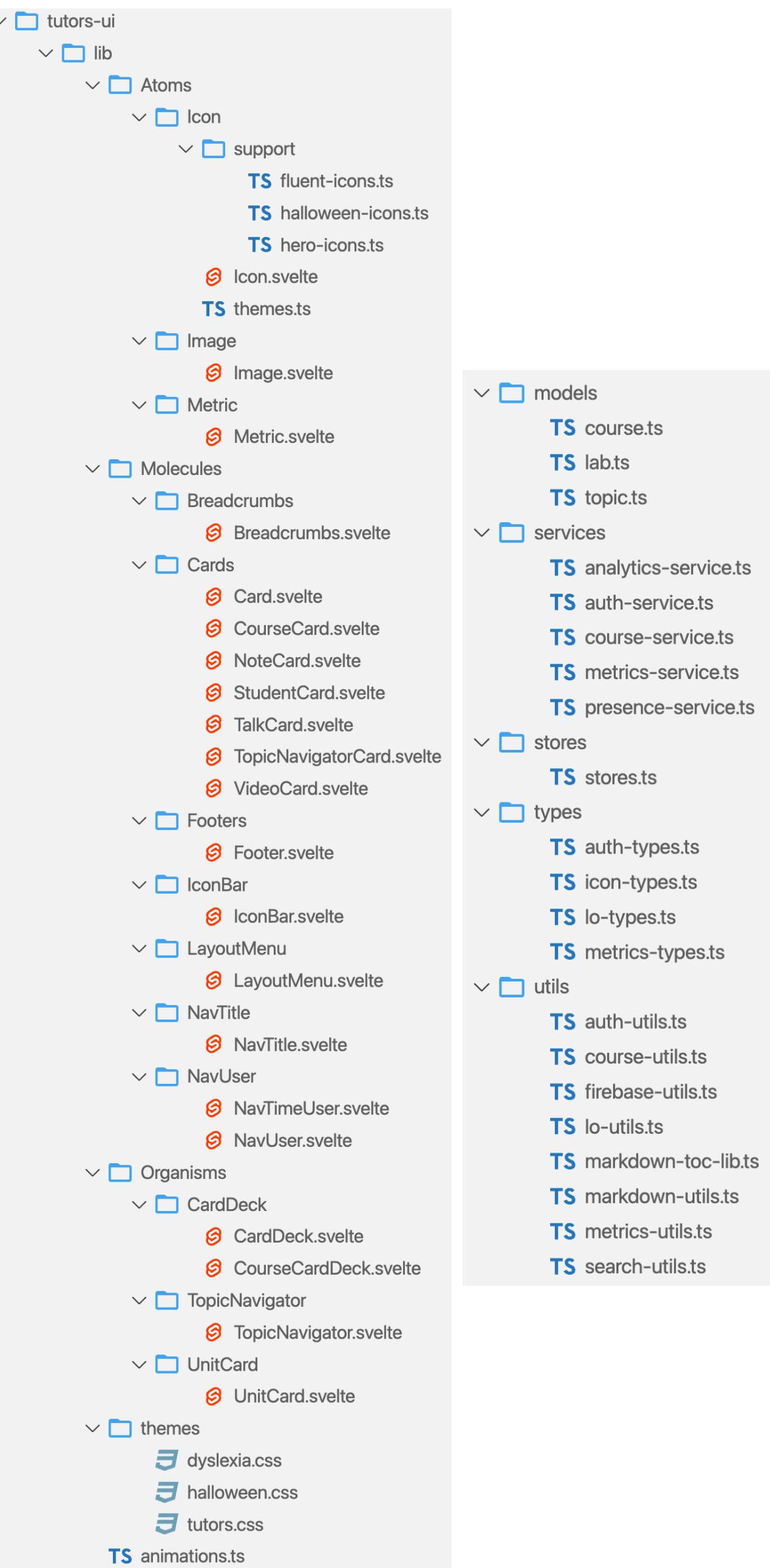
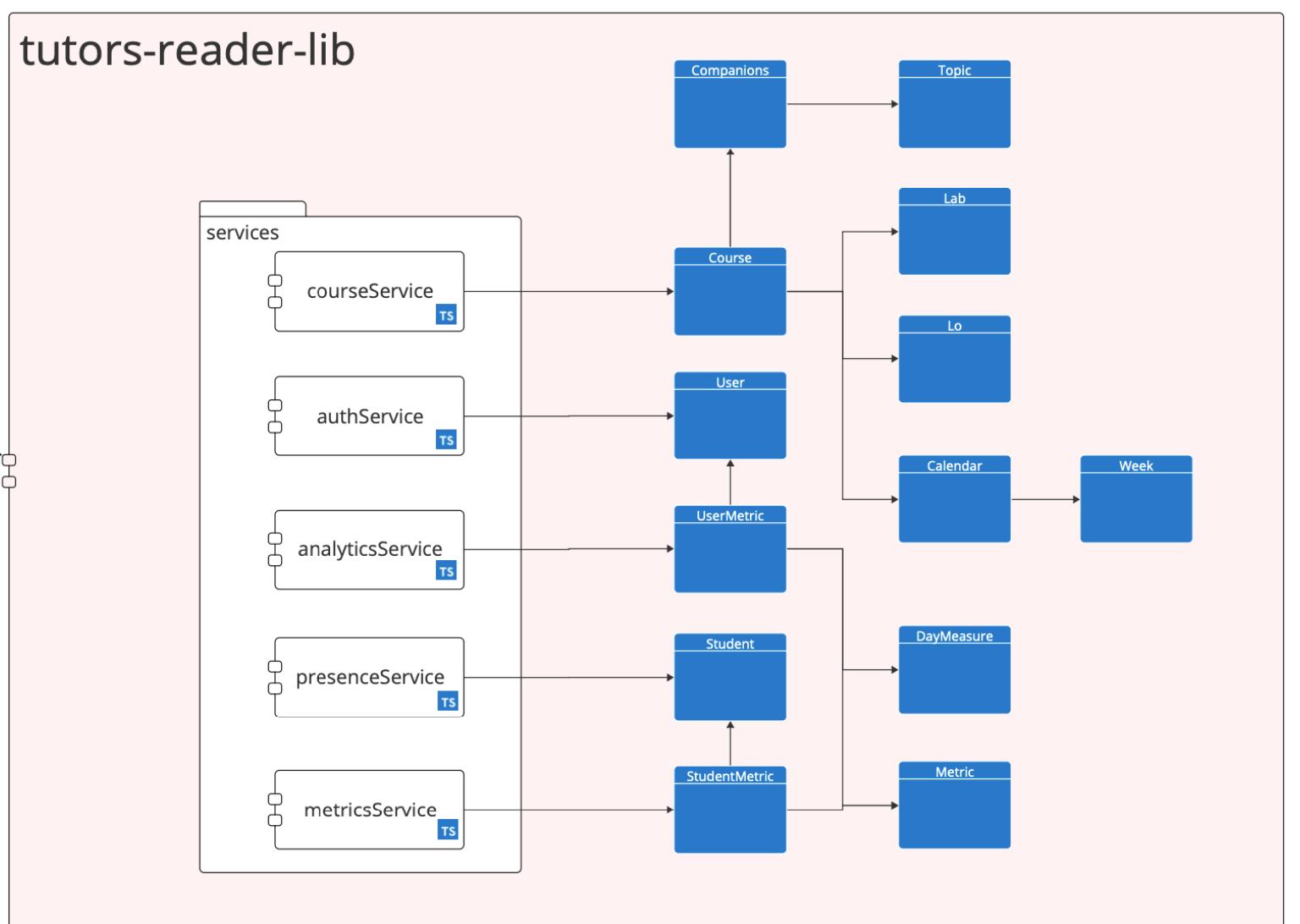
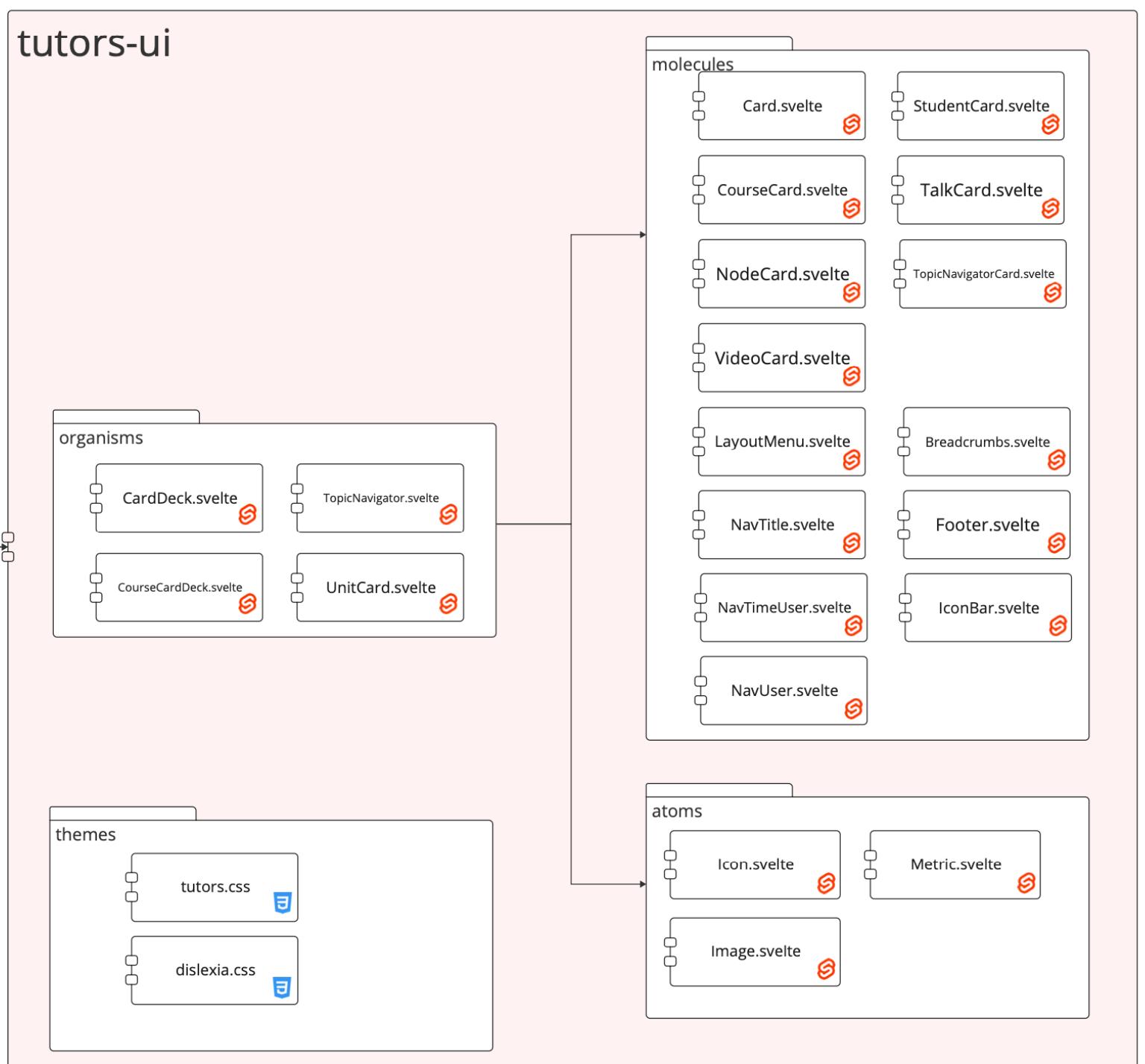
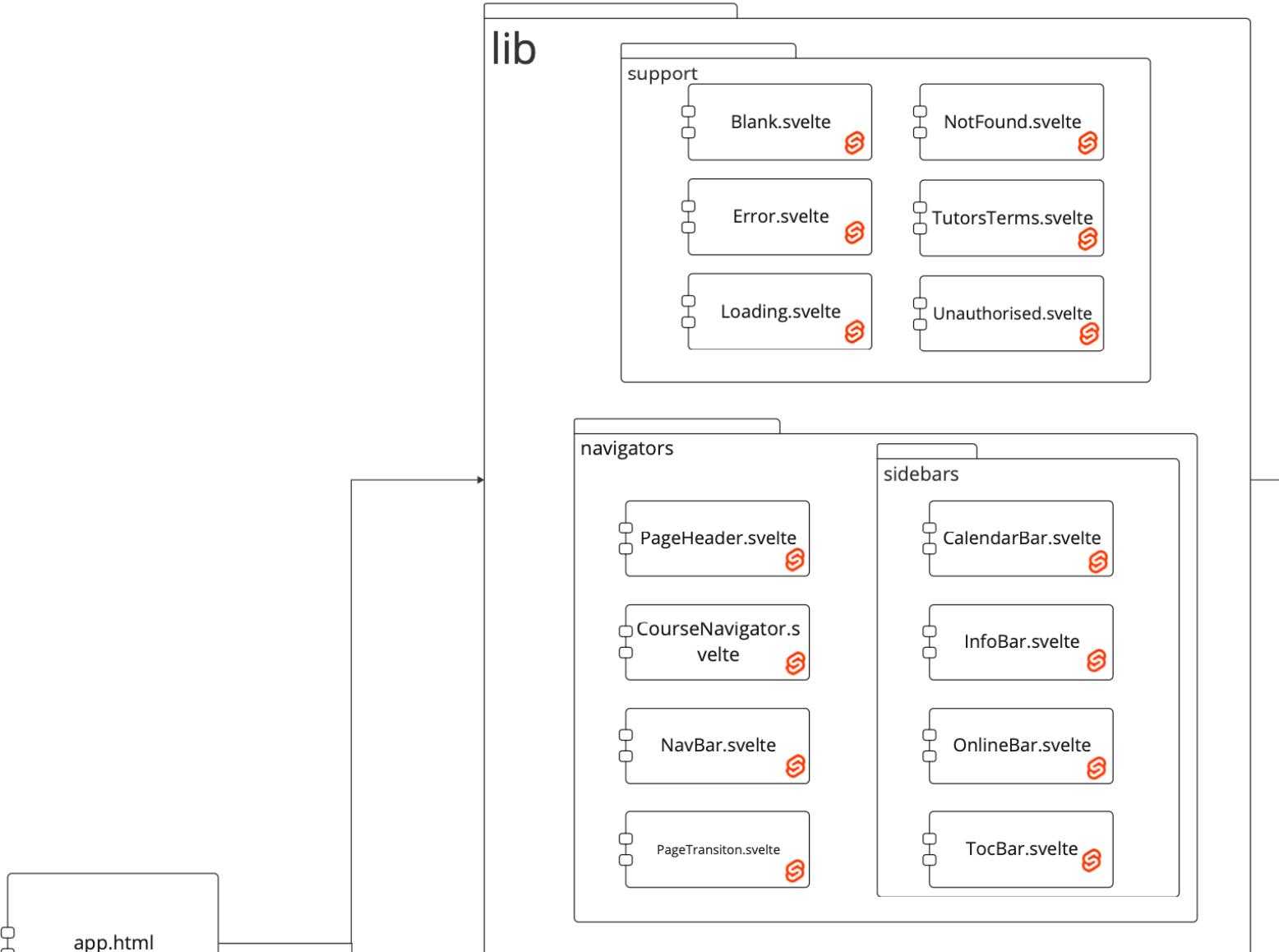
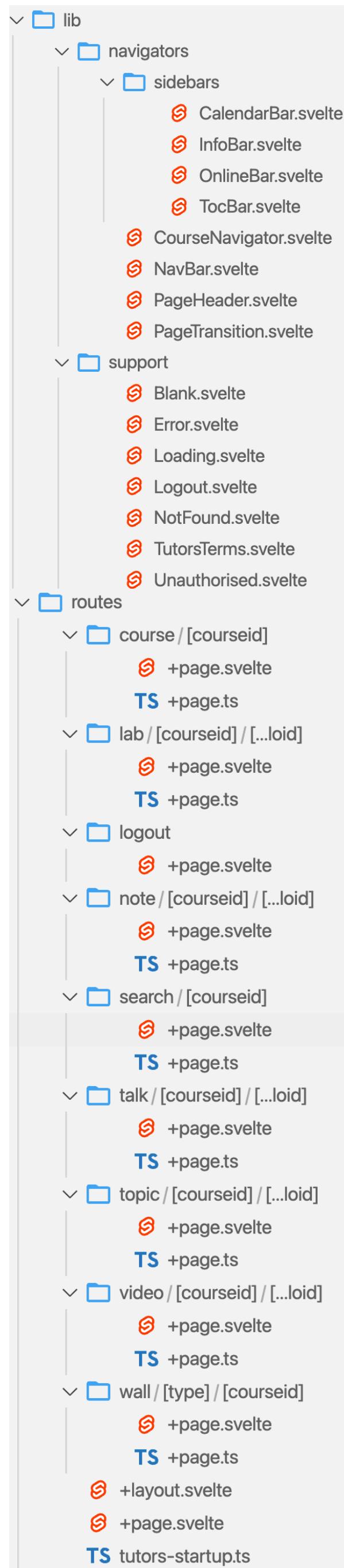
Course
Intelligence
Read course
+ prime
analytics,
presence &
metrics
services

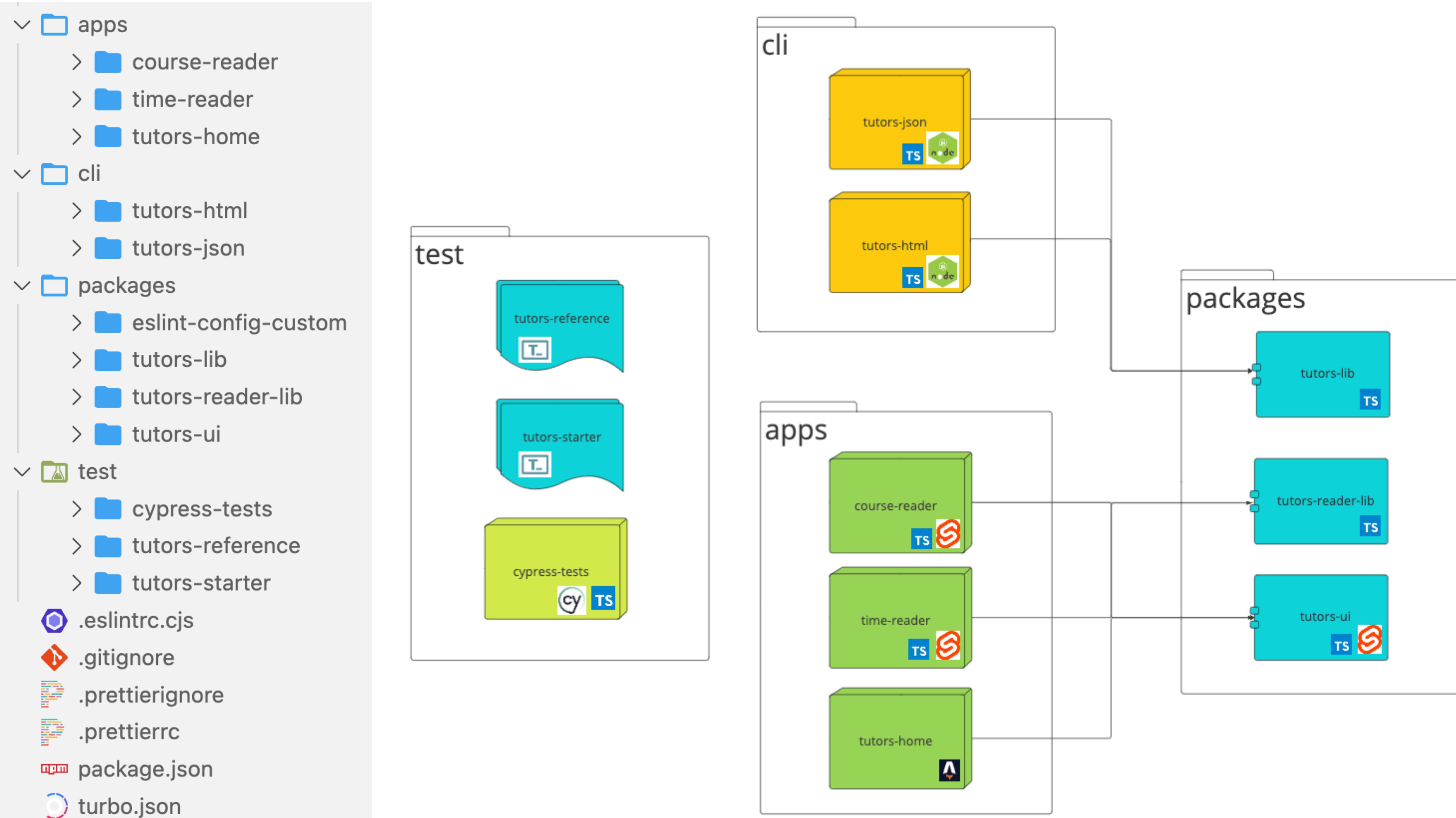


Run Time
version of
Course AST

Course Model
decorated
with caches +
indexes for url/
path resolution

Simplifies
Hydration
within reader





Development Roadmap

The screenshot shows the GitHub interface for the repository `tutors-sdk/tutors`. The user is viewing the `Issues` tab, which contains 11 open pull requests. The issues are listed in descending order of creation date. Each issue is accompanied by its title, labels (e.g., enhancement, help wanted, intermediate), and a brief description of who opened it and when.

Issue #	Title	Labels	Author	Created
#136	Eleventy Generator	enhancement, help wanted, intermediate	edeleastar	opened last month
#131	Implement new graphing feature to tutors-time-reader	enhancement, hacktoberfest-accepted, help wanted, intermediate	jordharr	opened on Oct 10
#89	tutors-home readme	beginner, documentation, good first issue	jordharr	opened on Sep 30
#87	Create Astro generator	advanced, enhancement, help wanted	jordharr	opened on Sep 30
#84	Centralise tailwindcss config in readers	beginner, enhancement, good first issue, help wanted	jordharr	opened on Sep 30
#82	Build electron app for tutors-course-reader	enhancement, help wanted, intermediate	jordharr	opened on Sep 30
#80	create new "quiz" learning object	advanced, enhancement, help wanted	jordharr	opened on Sep 30
#79	embed videos in labs in courses	enhancement, intermediate	jordharr	opened on Sep 30
#78	upgrade cli design for tutors-json	enhancement, help wanted, intermediate	jordharr	opened on Sep 30
#77	tutors-html readme	beginner, documentation, good first issue	jordharr	opened on Sep 30
#76	tutors-json readme	beginner, documentation, good first issue	jordharr	opened on Sep 30

Contributors



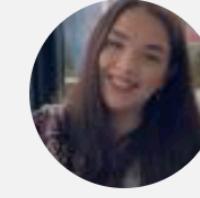
Eamonn de Leastar

SETU



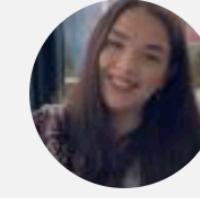
Colm Dunphy

SETU



Jordan Harrison

Charles River Labs



Emma Kidney



Ellen O'Carroll

Red Hat



Pete Windle

SETU



David Fan

Red Hat



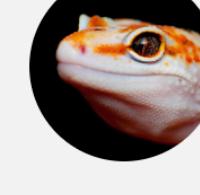
Bernard Thompson

Kargo



Leigh Griffin

Red Hat



Evgeniy Dainovich



Josh Bennett



Semester 1: January-June 2021

Workshop One

Life etc...

Web Development

Programming

Semester 1 Schedule

induction · structure · schedules · handbook

time · space · balance · experiences · tips · techniques

html · css · layout · web apps · web frameworks · deployment

algorithms · data structures · processing · java · classes · libraries

timetables · semester calendars · assessment schedules

Semester 2: June-December 2020

Workshop Two

ICT Skills 1

Computer Systems & Networks

Database

Semester 2 Schedule

introducing semester Databases & Computer Systems - project &...

Javascript · node · express · git · github · glitch

logic · computer organisation · os · networks · interfaces · sensors

entities · tables · rows · sql · er · nosql

timetables · semester calendars · assessment schedules

Semester Three

Workshop Three

Full Stack Web Development

Software Security

Developer Operations

Semester 3 Schedule

agile methods · cv preparation · introducing semester 3

javascript · node · apis · tdd · frameworks · front-end · svelte

crypto · vulnerabilities · authentication · pentesting

cloud computing · scripting · scaling · automation · monitoring

timetables · semester calendars · assessment schedules

Semester 4: June-December 2021

Workshop Four

ICT Skills 2

Mobile App Development

Semester 4 Schedule

Introducing semester 4, Project, Placement

Javascript · SPA · react · APIs · front-end

layouts · activities · resources · lifecycle · widgets · ux

timetables · semester calendars · assessment schedules

