Tutors

Open Web Learning Toolkit

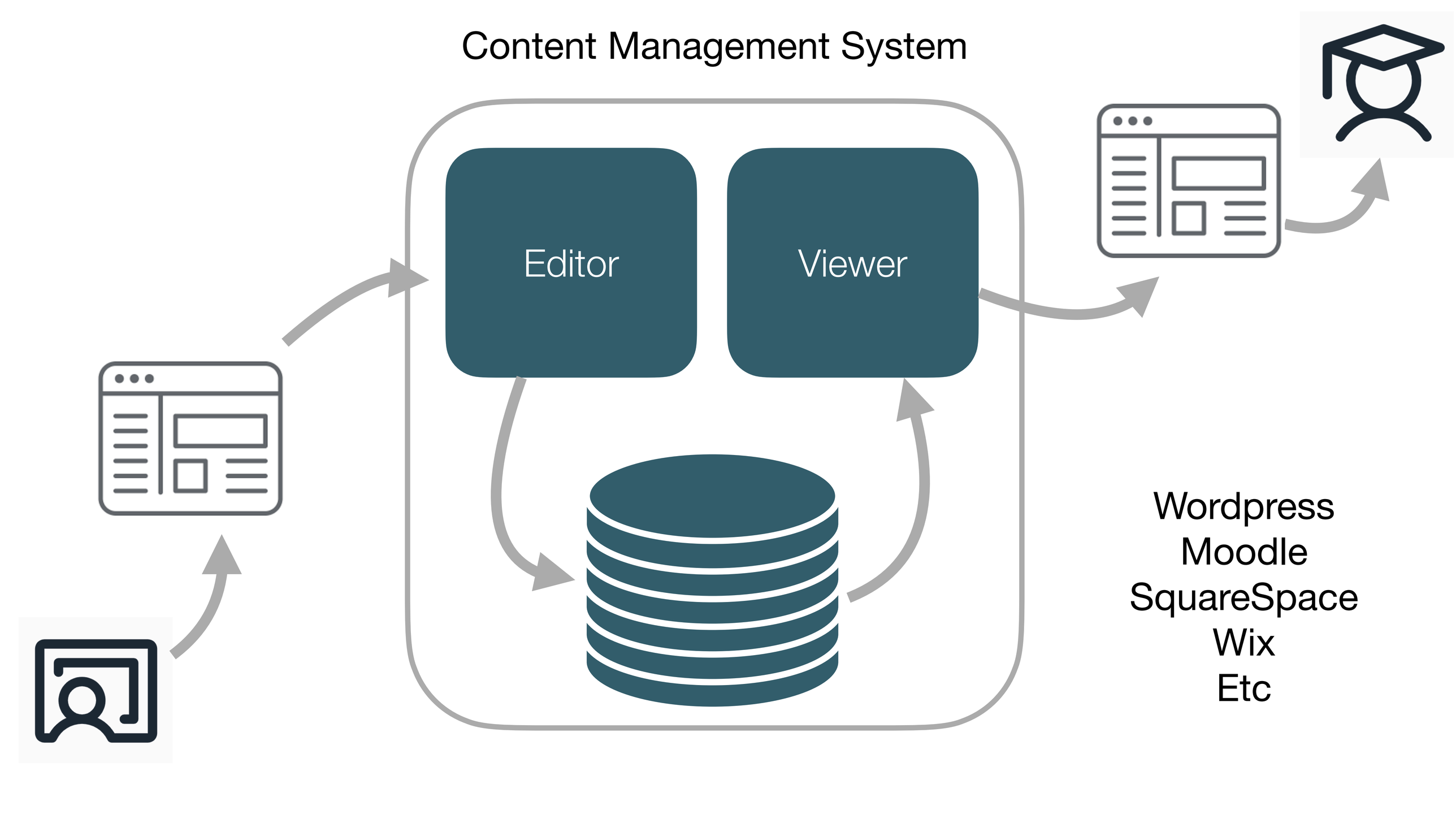
The Educator Experience

The ***Educator Experience*** prioritises the creation of a **guided paths** through a curriculum via the creation of learning materials that are **autonomous**, **structurally aligned**, **composable**, **auditable**, **extensible**, **versioned** and **independent**.

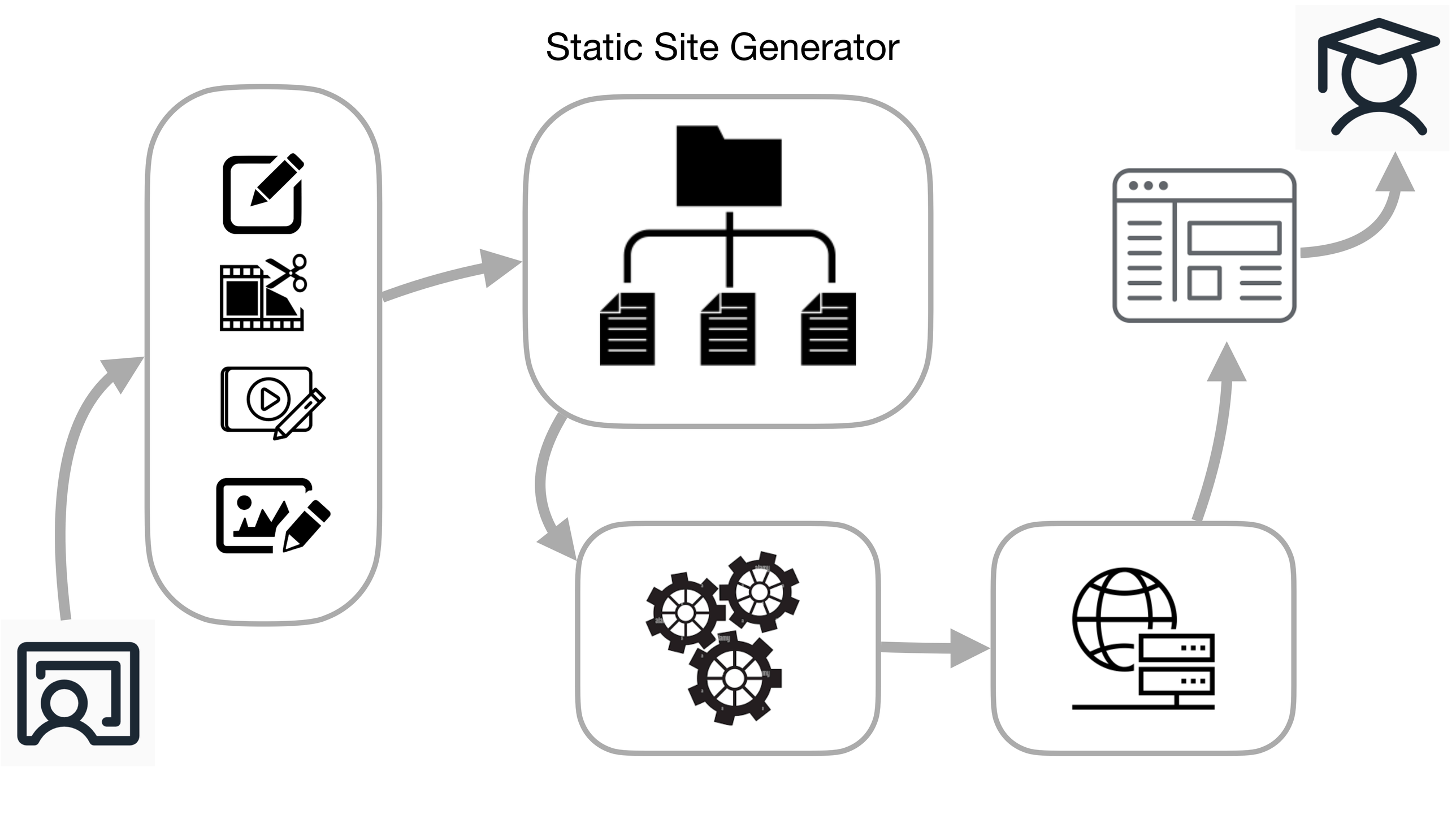
Established models for creating & managing web resources

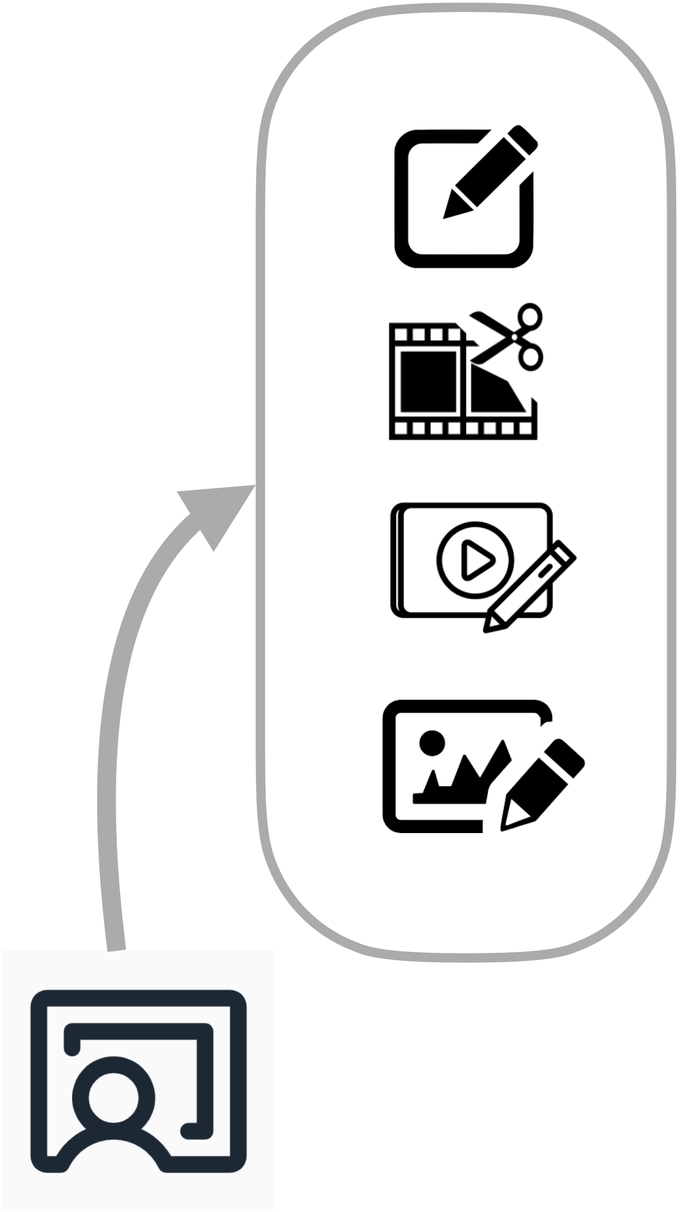
Content Management System

Static Site Generator


Content Management System
Editor
Viewer
Wordpress
Moodle
SquareSpace
Wix
Etc

Static Site Generator



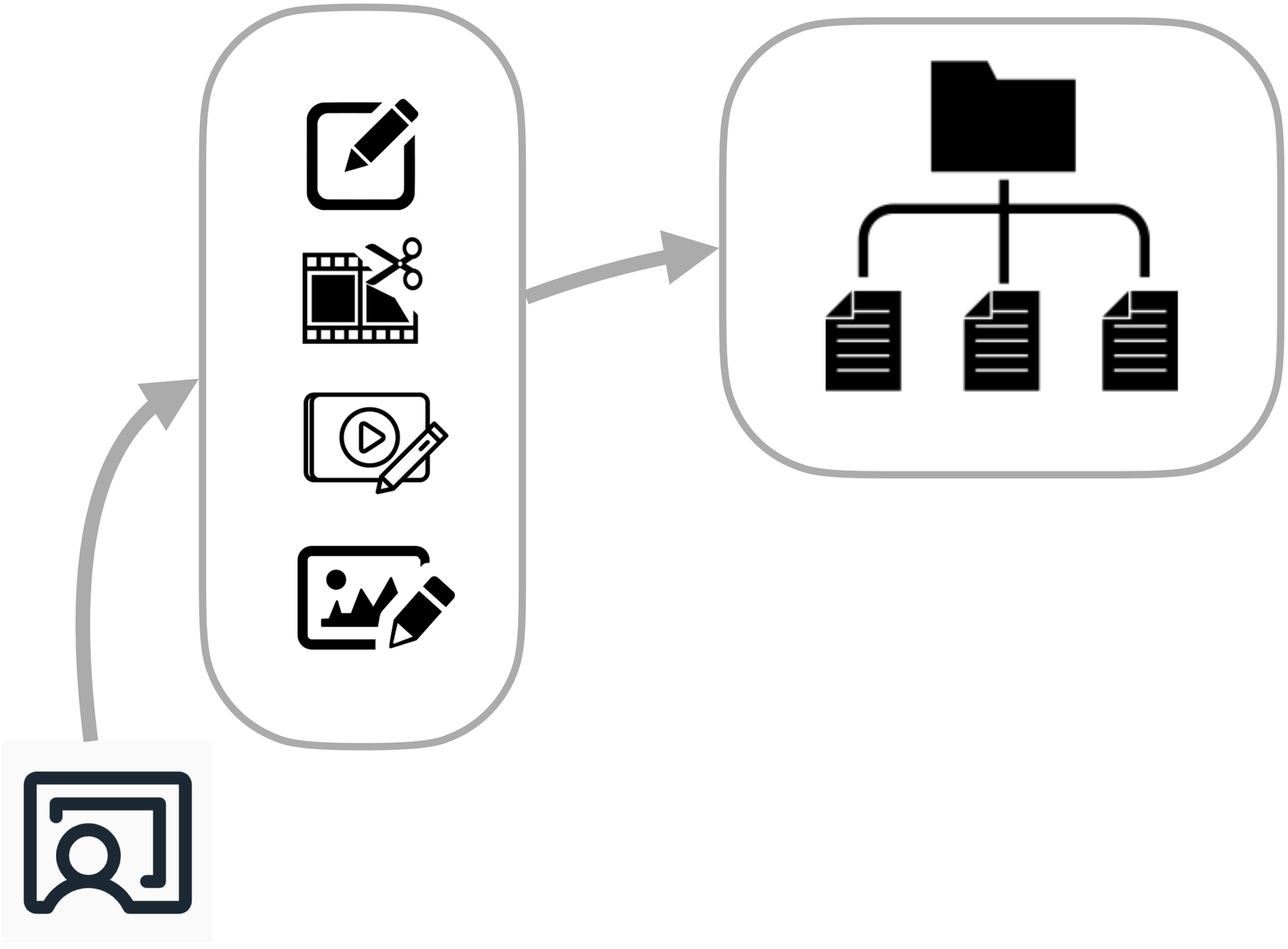


Use whatever tools you like:

- Editors
- Graphic Production
- Icon Sets
- Media

Emphasis on

- Markdown formatted documents
- Latex compatible markdown
- PDFs
- Video Players



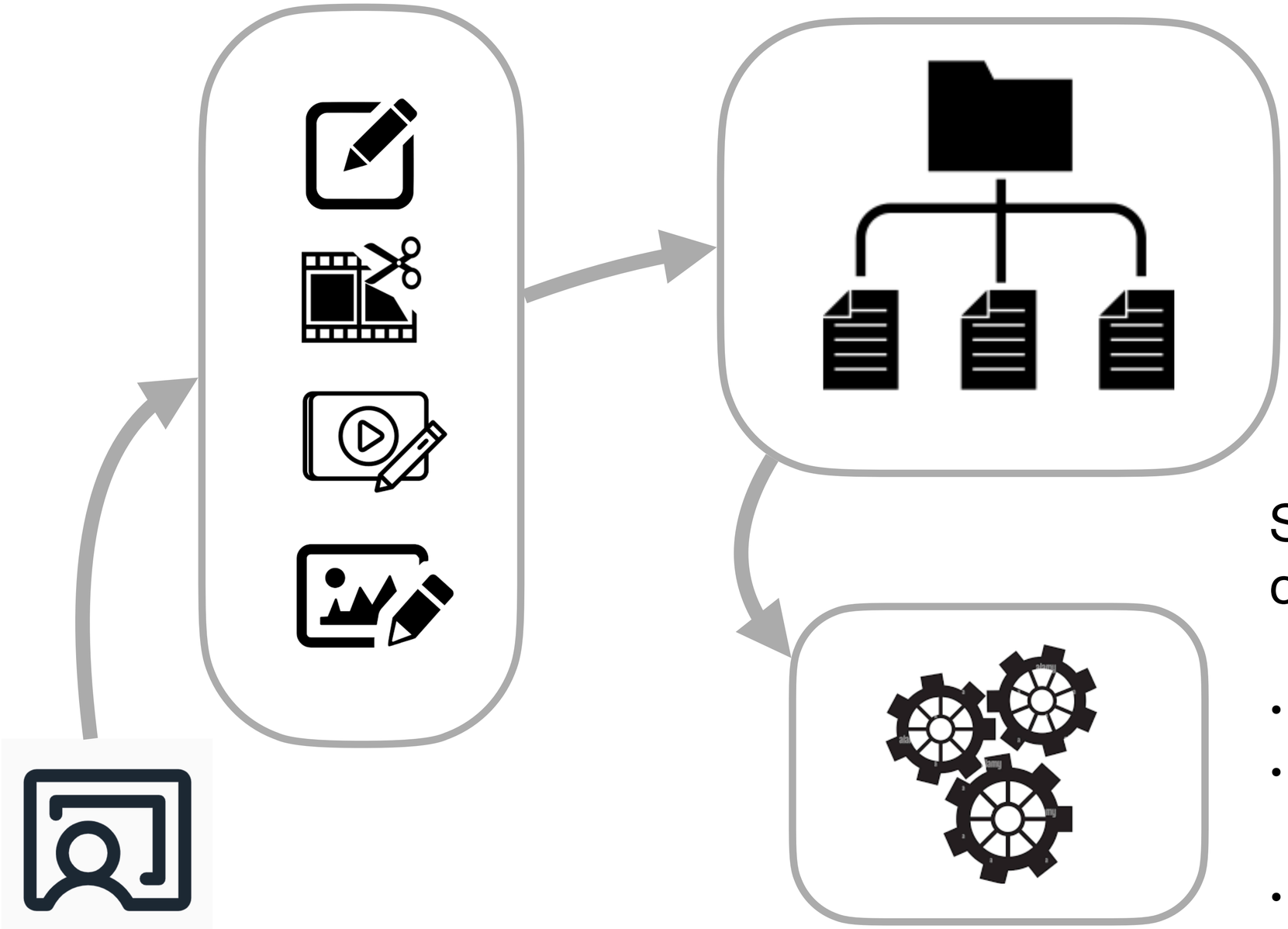
Content organised into Files & Folders on local HD

- Nested folder reflects course structure
- Adopt sensible naming conventions & file patterns
- Media & links

Can be versioned and stored anywhere

- Github
- Bitbucket
- OneDrive
- Dropbox

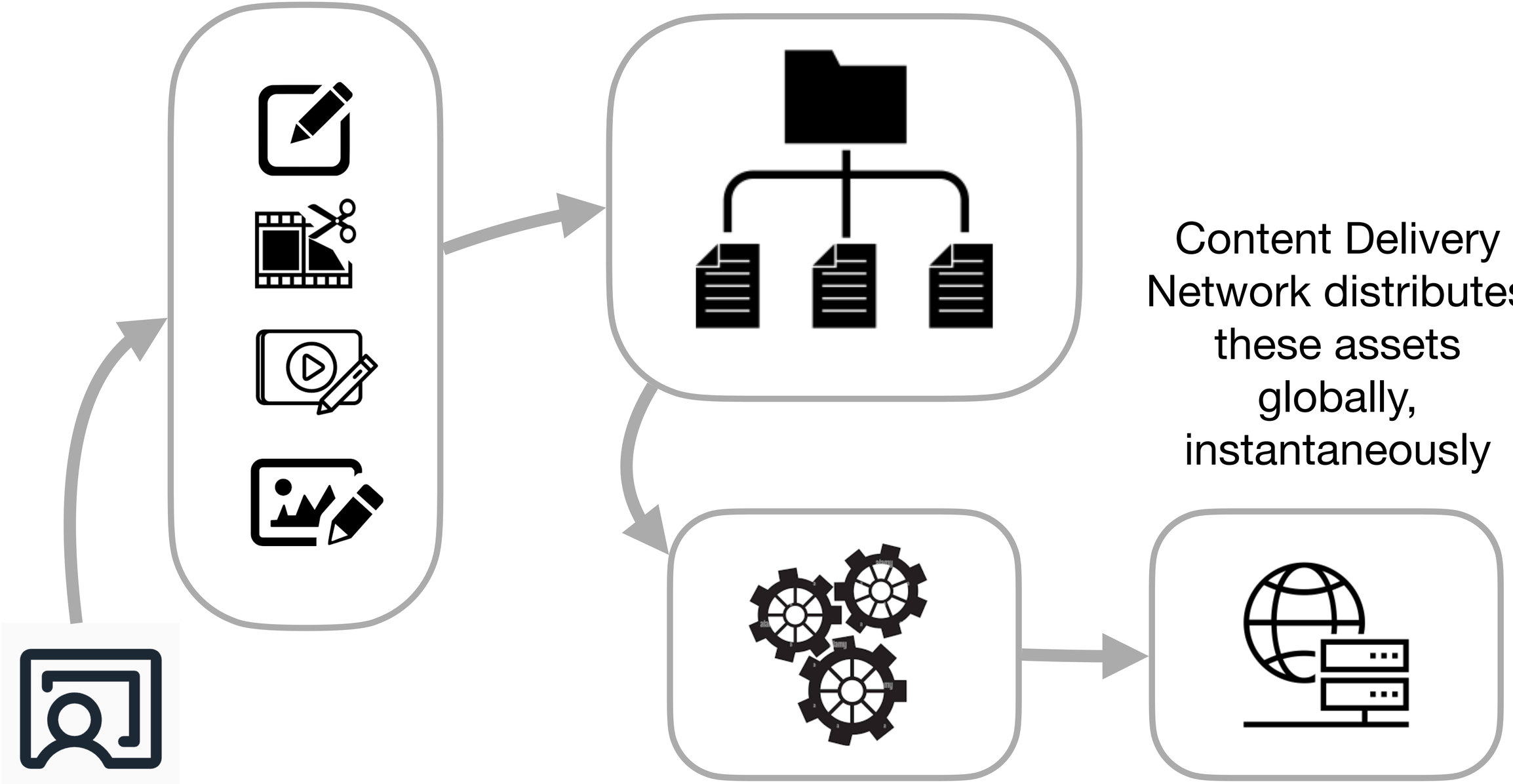
Static Site Generator



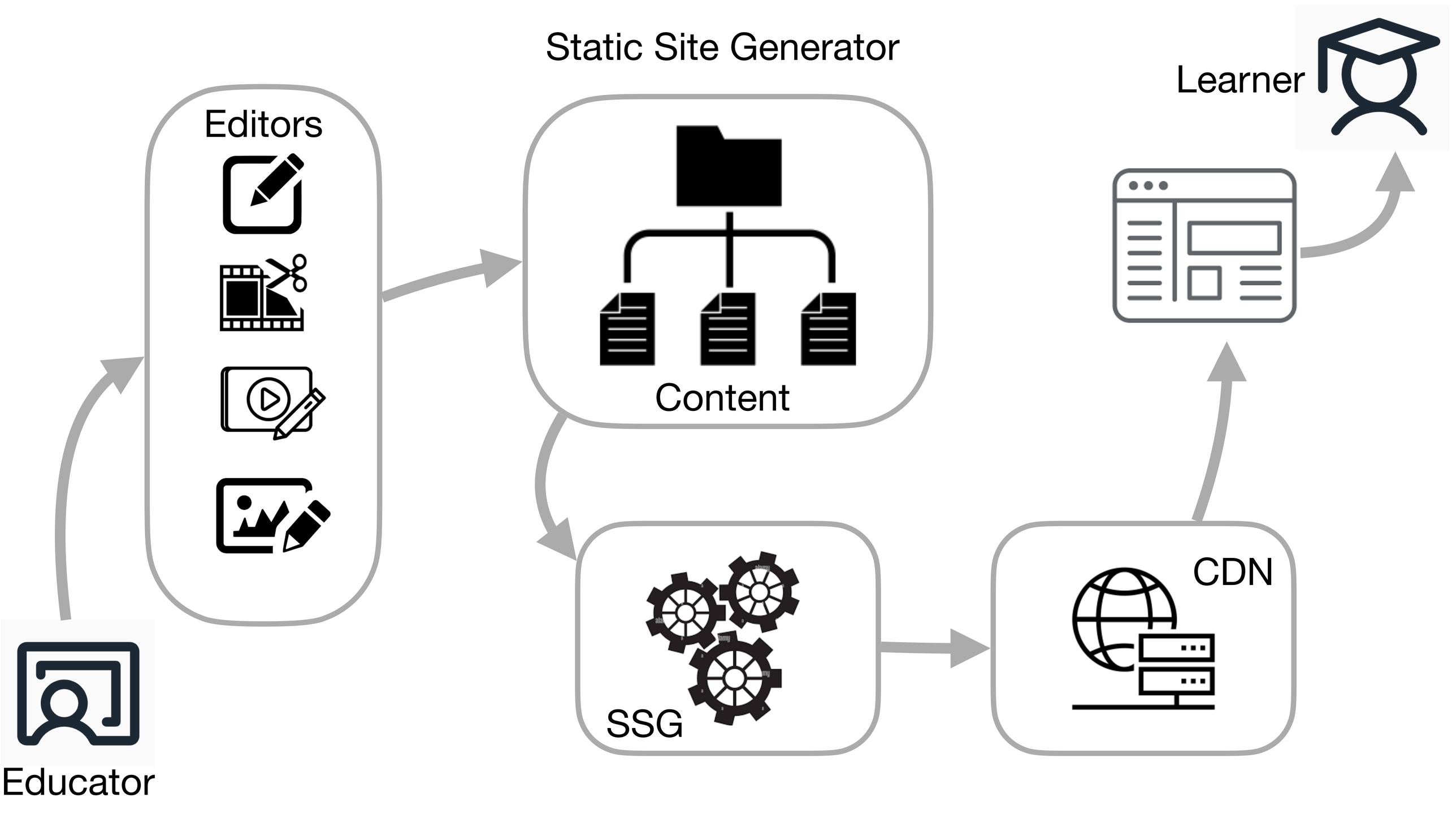
Static Site Generator ingests course structure:

- Produces a set of assets
- Can be directly loaded by a browser
- Can be rendered by a Web Application

Static Site Generator

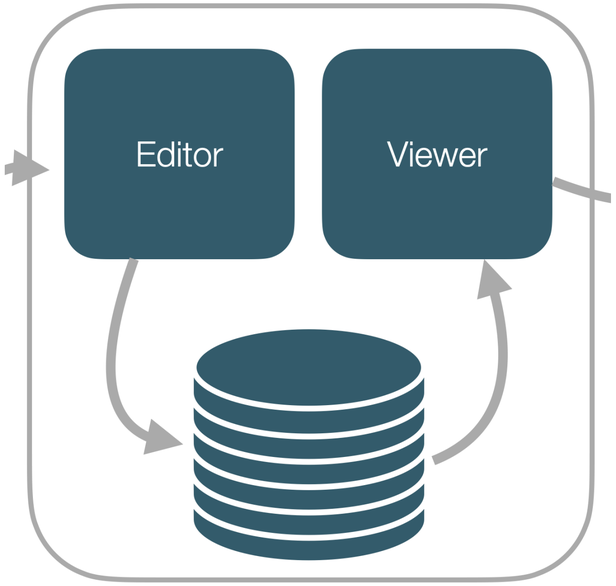


Content Delivery Network distributes these assets globally, instantaneously


Static Site Generator
Learner
Editors
Content
CDN
SSG
Educator

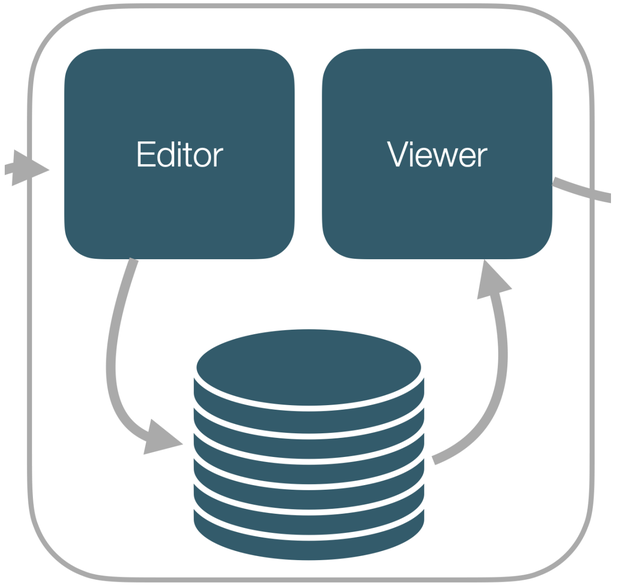
Content Management Systems

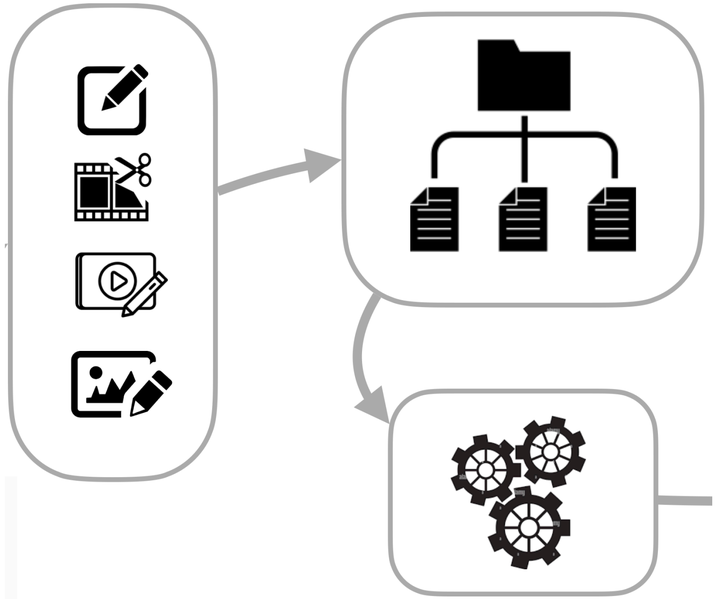
- Course material ‘atomised’ into opaque database
- Dependent on CMS, which must be hosted, secured, patched & maintained
- Courses must be ‘migrated’ each academic year, a painful process
- Highly restricted, often proprietary, design palette



Content Management Systems

- Course material ‘atomised’ into opaque database
- Dependent on CMS, which must be hosted, secured, patched & maintained
- Courses must be ‘migrated’ each academic year, a painful process
- Highly restricted, often proprietary, design palette



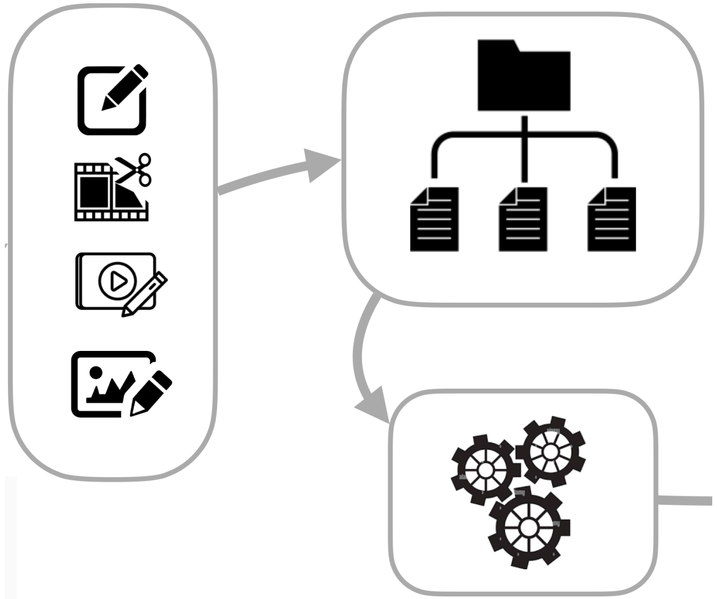


Static Sit Generators

- Course material hand crafted be structurally aligned to course learning paths
- Complete independence, there is no ‘server’ to be maintained
- Course ‘source’ can be managed & versioned (Github)
- New tools can be onboarded as needed

The Educator Experience

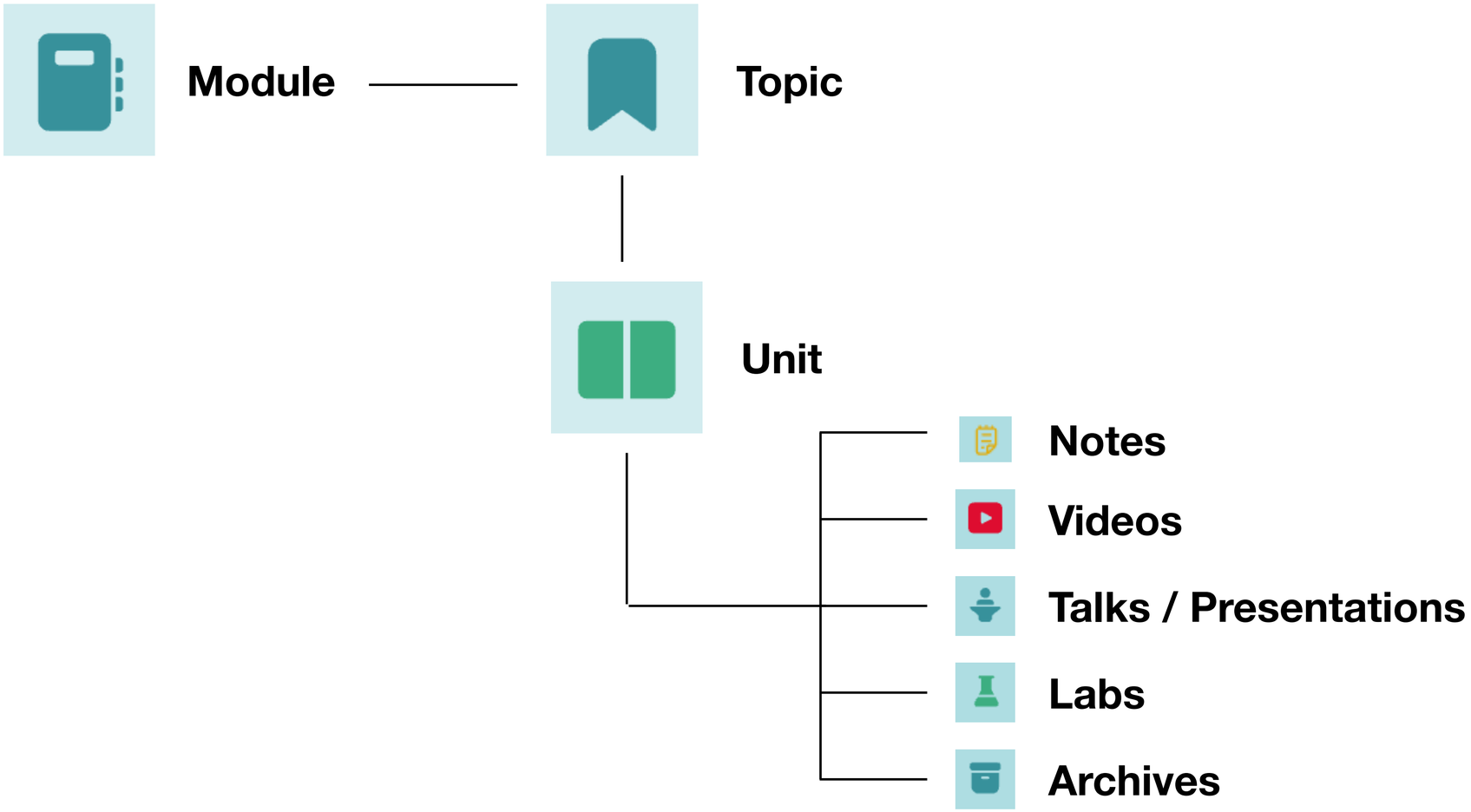
The ***Educator Experience*** prioritises the creation of a **guided paths** through a curriculum via the creation of learning materials that are **autonomous**, **structurally aligned**, **composable**, **auditable**, **extensible**, **versioned** and **independent**.



Static Sit Generators

- Course material hand crafted be structurally aligned to course learning paths
- Complete independence, there is no ‘server’ to be maintained
- Course ‘source’ can be managed & versioned (Github)
- New tools can be onboarded as needed

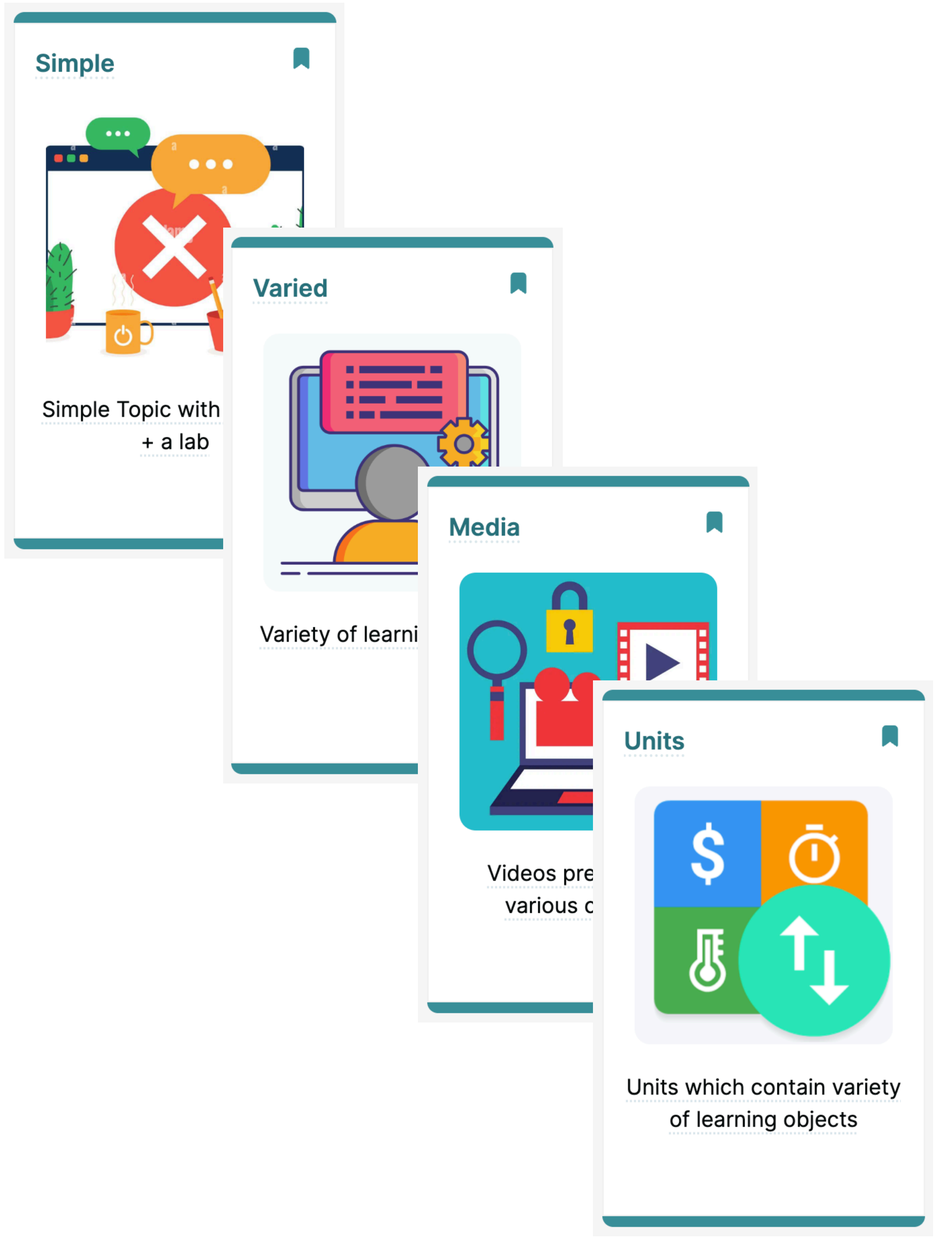
Module Structure



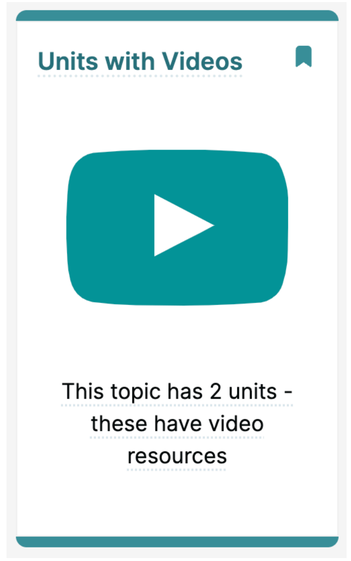
Topic Structure

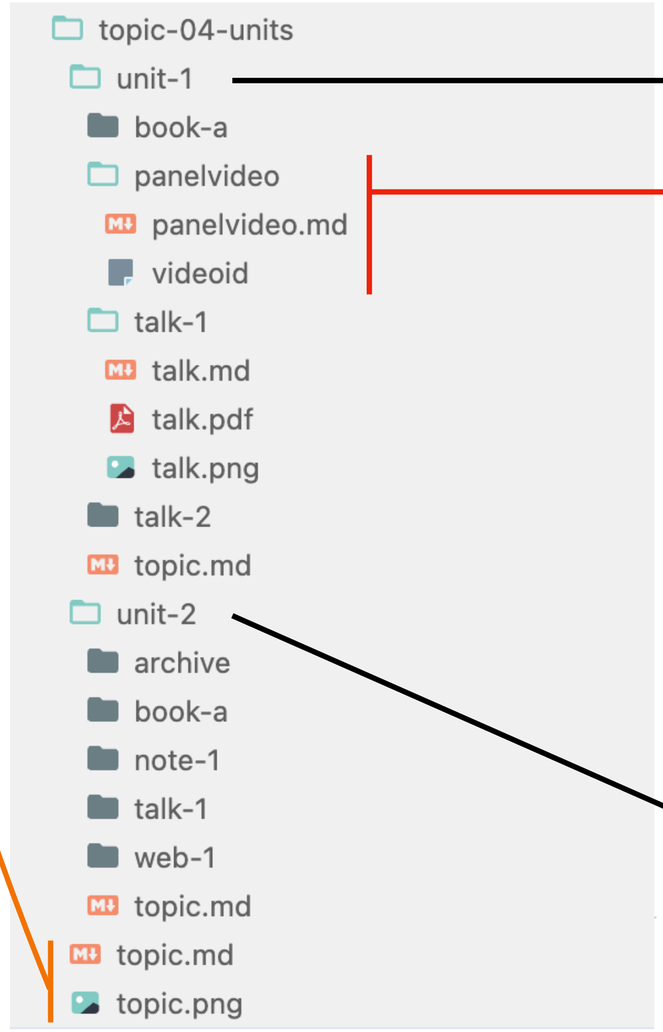
https://reference.tutors.dev

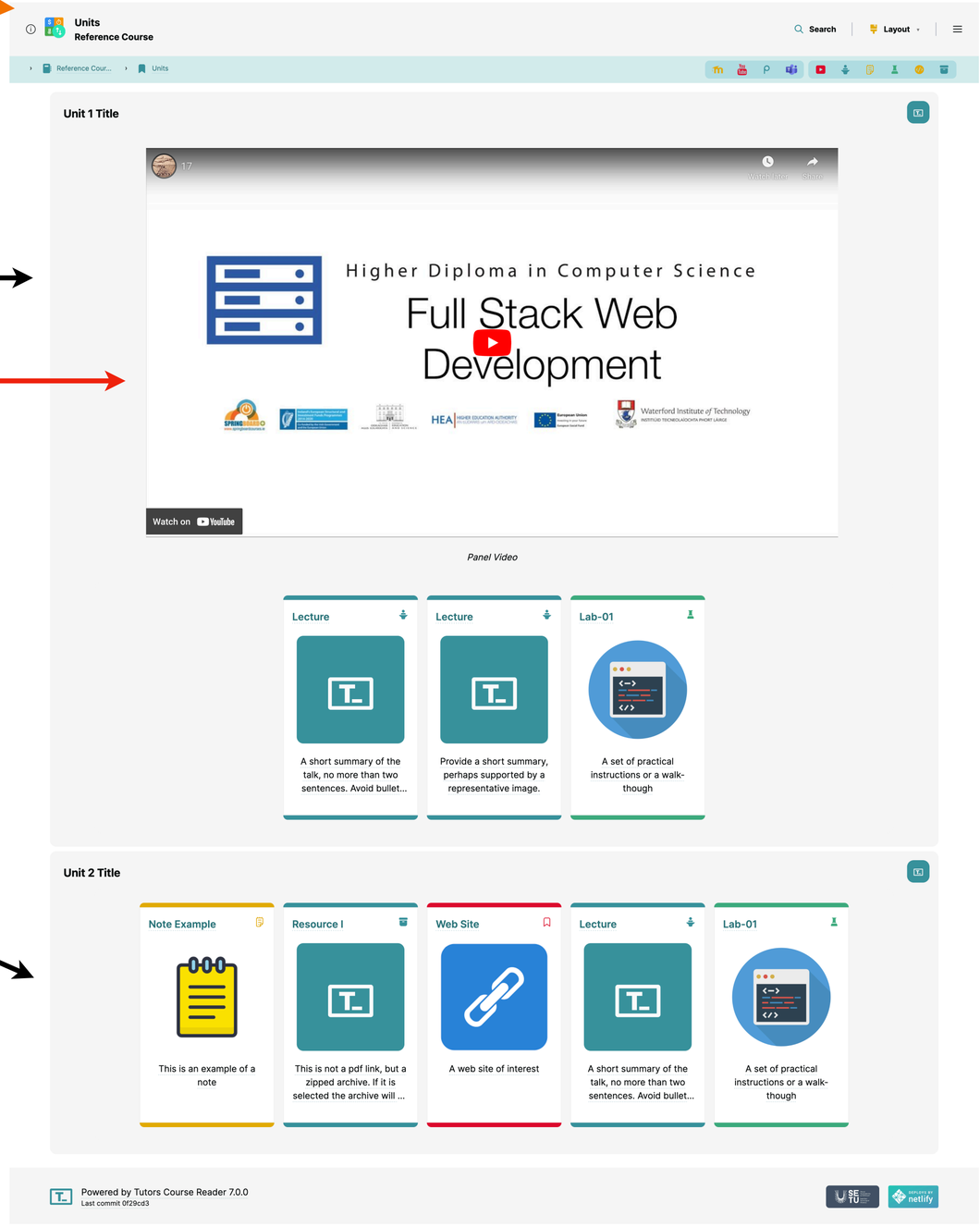
- courses / tutors-reference
  - topic-01-simple
    - book-a
    - talk-1
    - talk-2
    - topic.md
    - topic.png
  - topic-02-varied
    - archive
    - book-a
    - github-1-template-1
    - note-1
    - note-2
    - talk-1
    - web-1
    - topic.md
    - topic.png
  - topic-03-media
    - book-a
    - panelvideo
    - talk-1
    - talk-2
    - topic.md
    - topic.png
  - topic-04-units
    - unit-1
    - unit-2
    - topic.md
    - topic.png



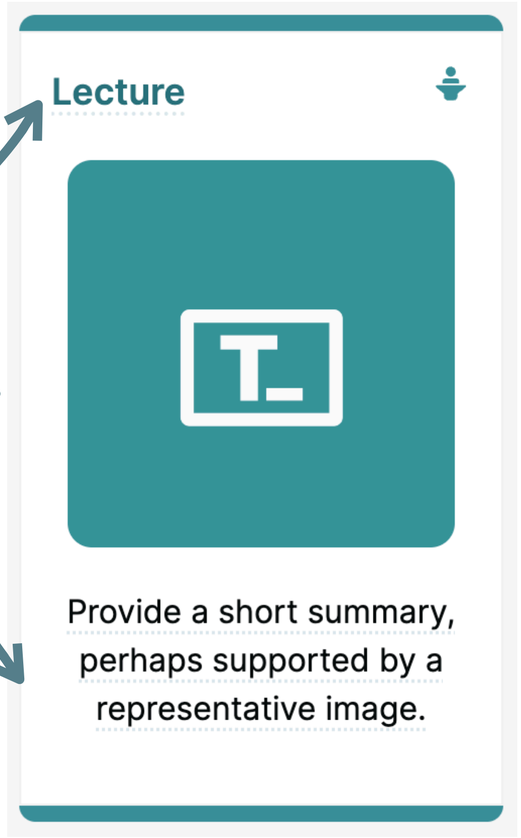
Single Topic Structure







Talk Structure



all files
same name

- image
- pdf
- markdown

```
Lecture

Provide a short summary, perhaps
supported by a representative
image.
```

Lab Structure

- book-a
  - archives
    - archive.zip
  - img
    - 01.jpg
    - 02.jpg
    - 03.png
    - 04.png
    - main.png
  - 00.Lab-01.md
  - 01.01.md
  - 02.02.md
  - 03.03.md
  - 04.04.md
  - 05.05.md
  - 06.Exercises.md

Links and Code Blocks

You can inert links in bullet points:

- http://github.com
- http://bitbucket.dom

More Code Block examples

```
publish(path, course) {
  const basePath = '../' + path + '/' + this.folder;
  futils.initEmptyPath(basePath);

  this.resources = this.talks.concat(this.labs);
  futils.copyFileToFolder(this.img, basePath);
  futils.writeFile(basePath + '/index.html', nunjucks.render('topic.html', this));

  futils.writeFile(basePath + '/ajaxlabel.html',
      nunjucks.render('ajaxlabel.html',
          { url: this.url.substring(5) + '/' + this.folder }));
  futils.writeFile(basePath + '/indexmoodle.html', nunjucks.render('indexmoodle.html', this));

  console.log(this.title);
  this.publishTalks(basePath);
  this.publishLabs(basePath, course);
}
```

This was autodetected as javascript. This one is html:

```
<div class="ui segment pushable">
  <div class="ui inverted labeled icon left inline ve
    {% for resource in course.allLabs.resources %}
      {% if resource.title == title %}
        <a class="active item"
href="../../{{resource.parentFolder}}/{{resource.fold
          {{resource.title}}
        </a>
      {% else %}
        <a class="item"
```

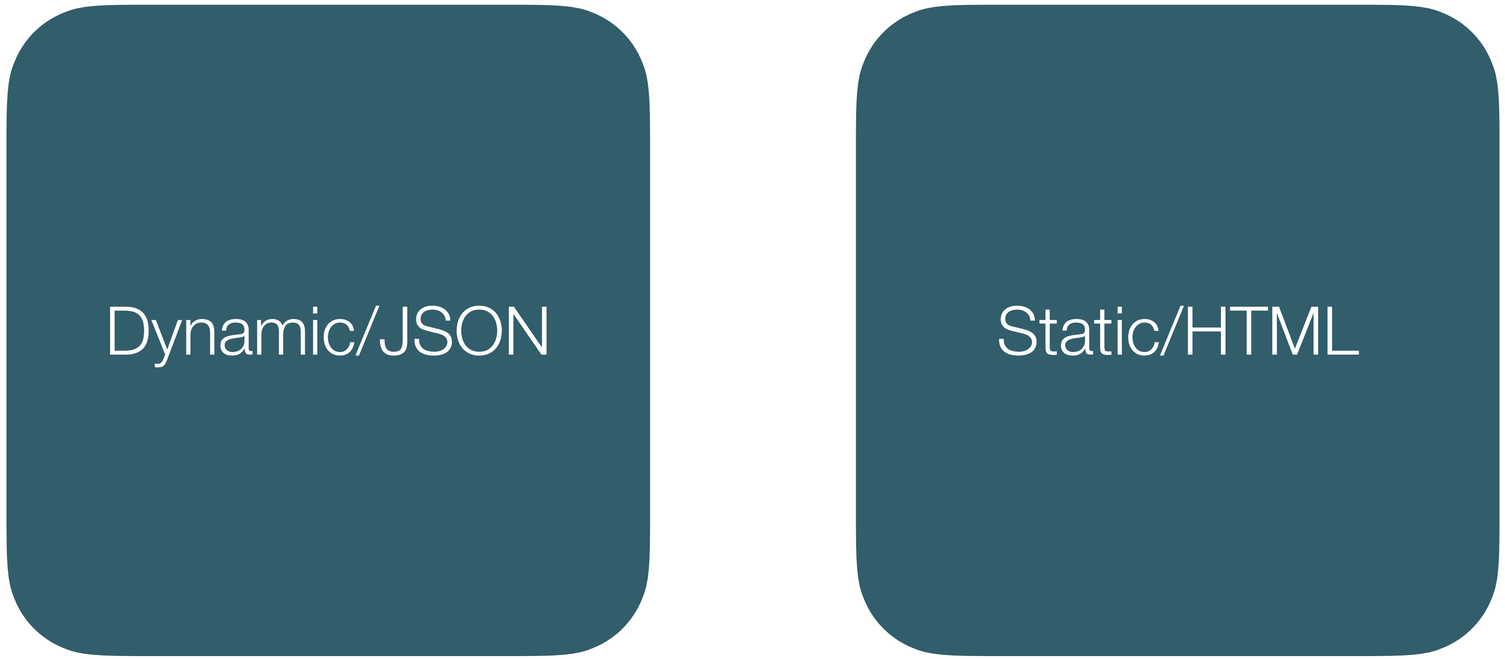
courses > tutors-reference > topic-01-simple > book-a > 03.03.md > # Links and Code Blocks

```
1  # Links and Code Blocks
2
3  You can inert links in bullet points:
4
5  - <http://github.com>
6  - <http://bitbucket.dom>
7
8
9  More Code Block examples
10
11 ~~~
12   publish(path, course) {
13     const basePath = '../' + path + '/' + this.folder;
14     futils.initEmptyPath(basePath);
15
16     this.resources = this.talks.concat(this.labs);
17     futils.copyFileToFolder(this.img, basePath);
18     futils.writeFile(basePath + '/index.html', nunjucks.render('topic.html', this));
19
20     futils.writeFile(basePath + '/ajaxlabel.html',
21         nunjucks.render('ajaxlabel.html',
22             { url: this.url.substring(5) + '/' + this.folder }));
23     futils.writeFile(basePath + '/indexmoodle.html', nunjucks.render('indexmoodle.html', this));
24
25     con
26     thi
27     this.publishLabs(basePath, course);
28   }
29 ~~~
30
31 This was autodetected as javascript. This one is html:
32
33 ~~~
```

~/Documents/GitHub/tutors/courses/tutors-reference/topic-03-media

Publishing

There are 2 ways to publish a module:



Dynamic/JSON Publishing

- Dynamic Site
- Browser through the Tutors Reader app
- Continuously updated and enhanced
- Included optional analytics or presence features

Dynamic/JSON Publishing

- course.md
- json
- package.json
- properties.yaml
- public-site
- topic-01-js-intro
  - panelvideo
    - topic.md
    - topic.png
  - unit-1
- topic-02-arrays
  - panelvideo
    - topic.md
    - topic.png
  - unit-0
    - talk-e-variables-objects
    - talk-f-methods-arrays
    - topic.md
  - unit-1
    - book-a-js-arrays
    - talk-a-arrays-basics
    - talk-b-array-methods
    - talk-c-array-iteration
    - talk-d-arrays-of-objects
    - topic.md
- topic-03-glitch-overview
- topic-04-web-apps
- topic-05-app-templates+views
- topic-06-app-models
- topic-07-sessions
- topic-08-apis

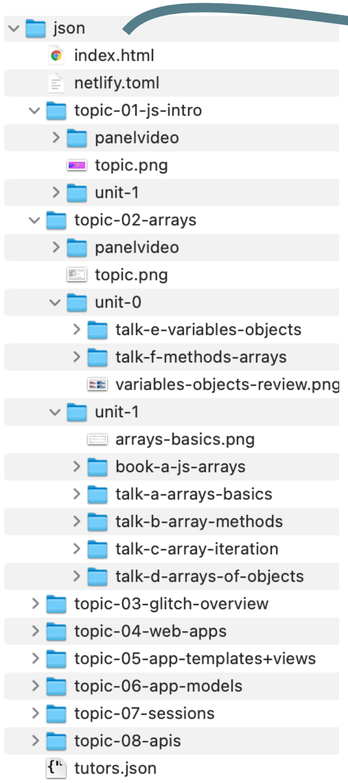
npx tutors-json

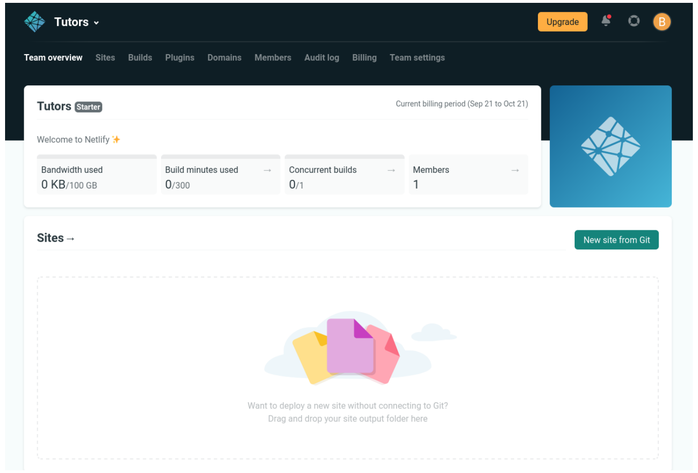
```
-bash                                                    ⌥⌘2
Eamonns-Mac-mini:web-development-2-2022 edeleastar$ npx tutors-json
Static course generator tutors-json 2.2.3 (tutors-lib: ^0.6.3)
::   Web Development II
  --> 1: Introducing Javascript (topic)
:: 1: Introducing Javascript
  --> Introducing 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
  --> Introducing Glitch (unit)
```

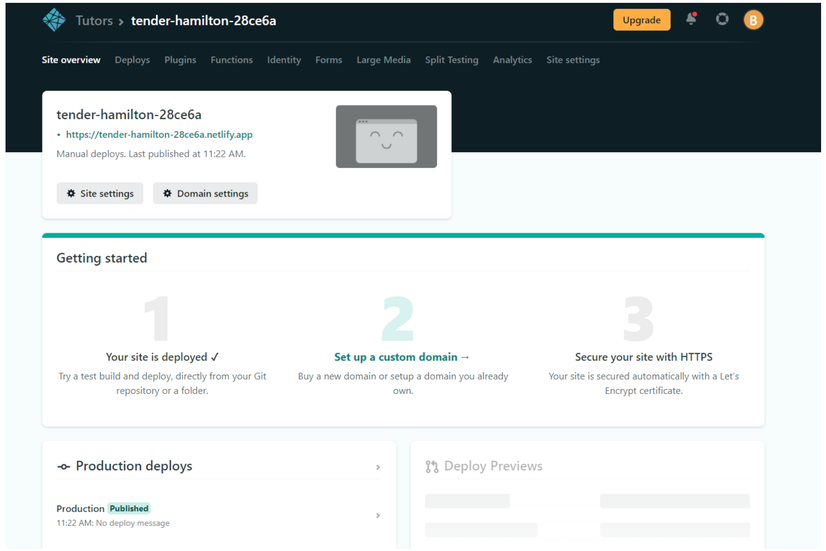
- json
  - index.html
  - netlify.toml
  - topic-01-js-intro
    - panelvideo
      - topic.png
    - unit-1
  - topic-02-arrays
    - panelvideo
      - topic.png
    - unit-0
      - talk-e-variables-objects
      - talk-f-methods-arrays
      - variables-objects-review.png
    - unit-1
      - arrays-basics.png
      - book-a-js-arrays
      - talk-a-arrays-basics
      - talk-b-array-methods
      - talk-c-array-iteration
      - talk-d-arrays-of-objects
  - topic-03-glitch-overview
  - topic-04-web-apps
  - topic-05-app-templates+views
  - topic-06-app-models
  - topic-07-sessions
  - topic-08-apis
  - tutors.json

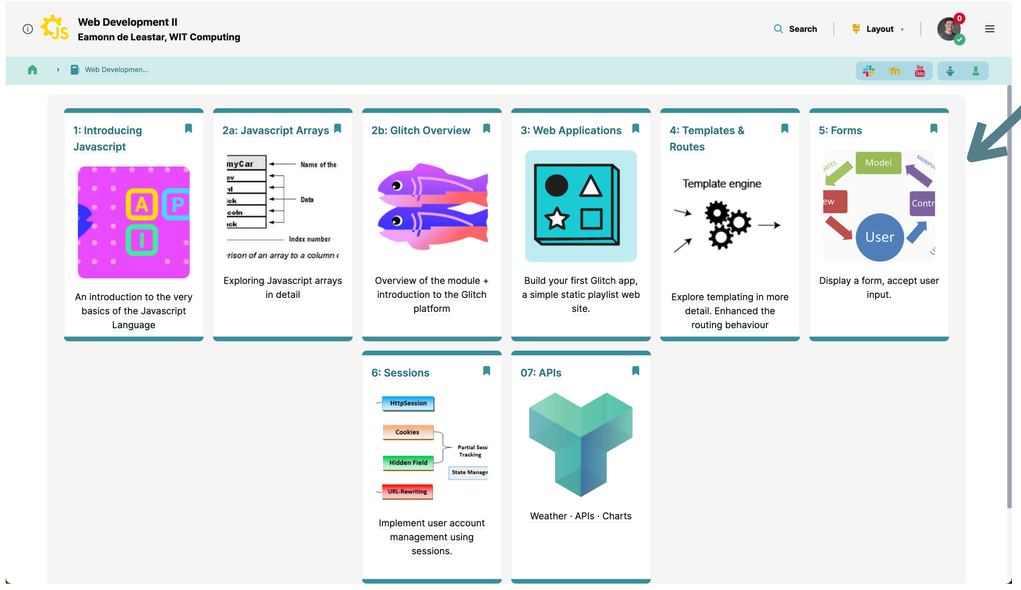
Command generates static site to ‘json’ folder

Dynamic/JSON Publishing









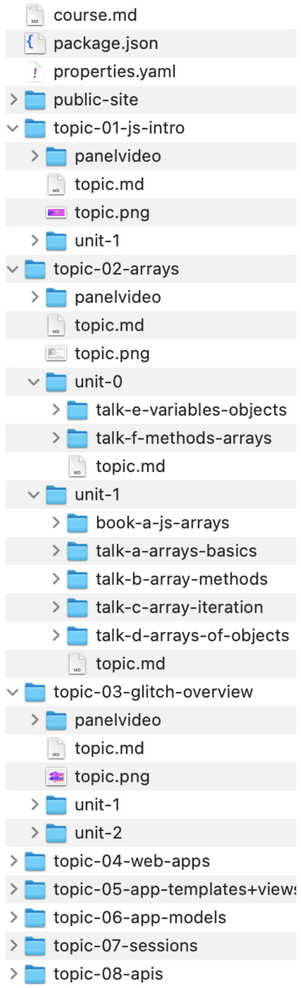
1. Netlify GitHub deployment
2. Netlify Drag & Drop

Deploys the Course to be read by the tutors reader

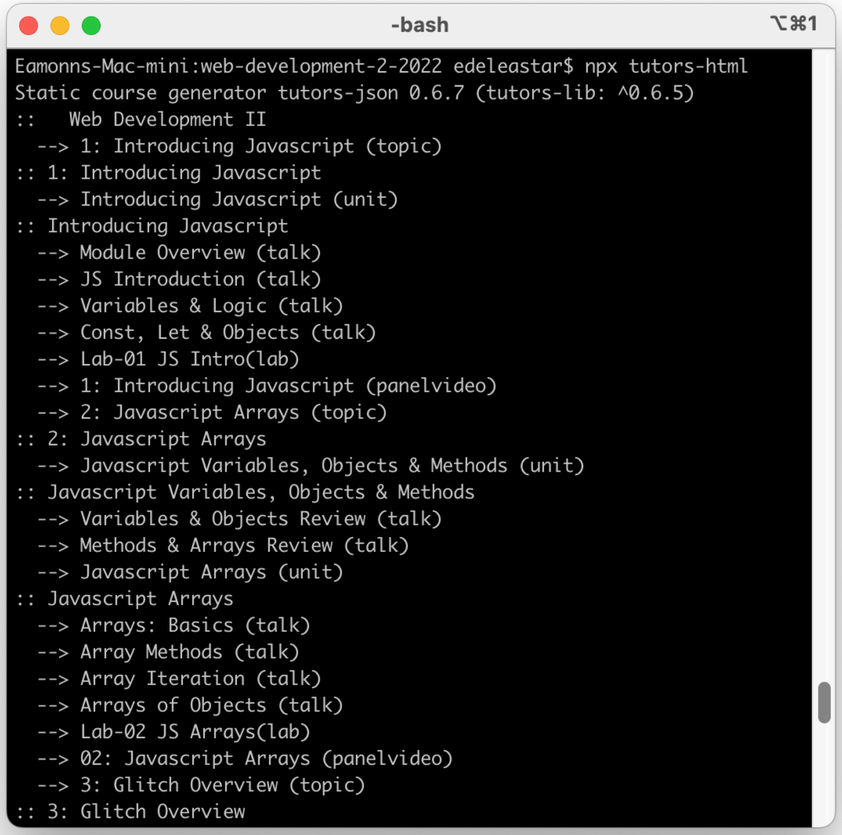
Static/HTML Publishing

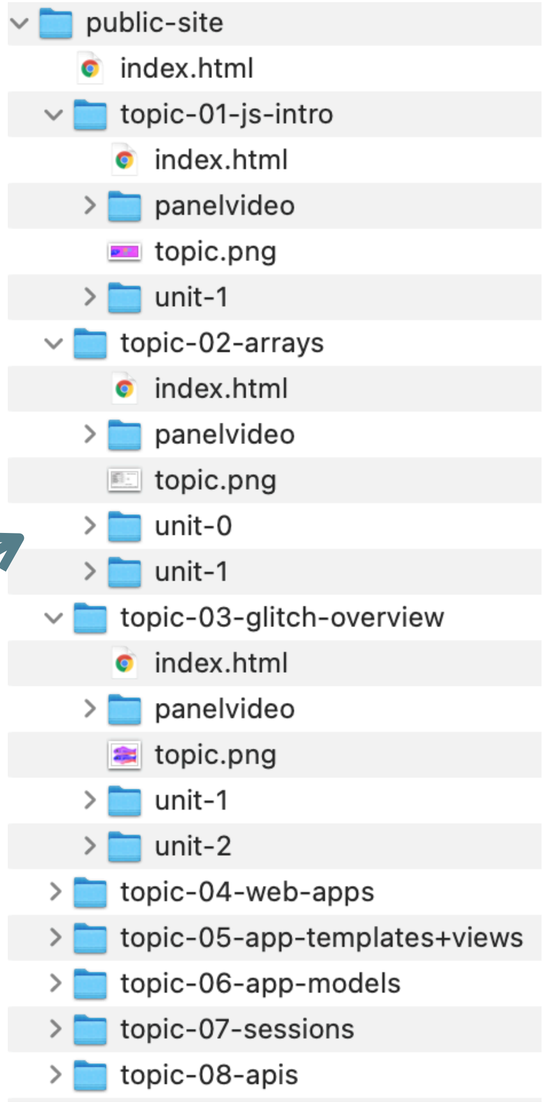
- Static Site
- Can be browsed offline (distribute zipped archive) / uploaded to traditional web host
- Format ‘frozen’ at time of generation
- No analytics or presence features

Static/HTML Publishing



npx tutors-html





Command generates static site to ‘public-site’ folder