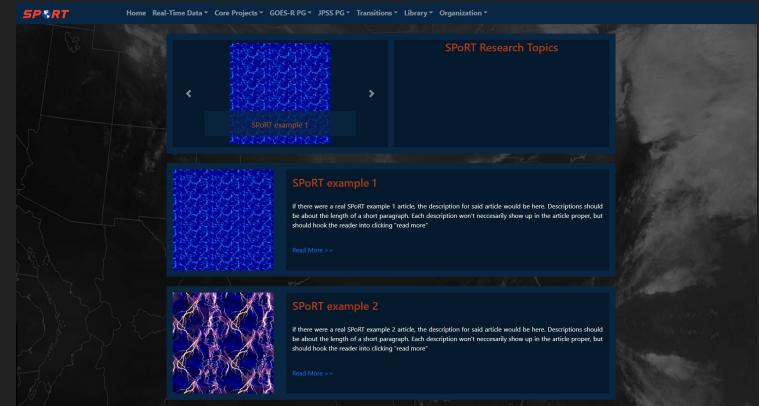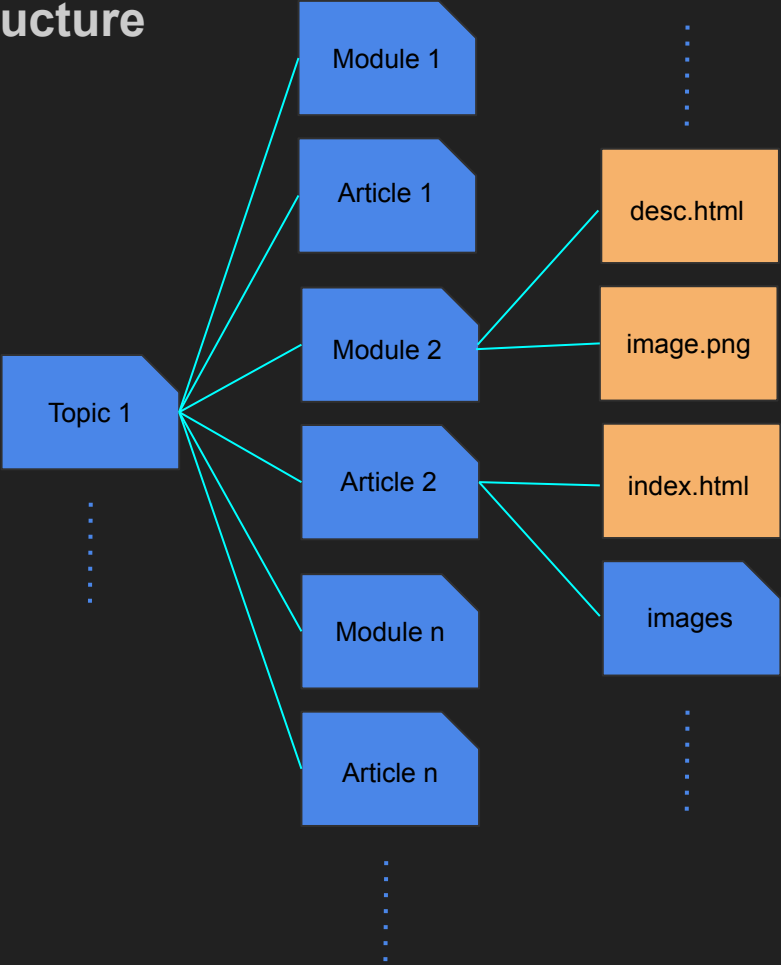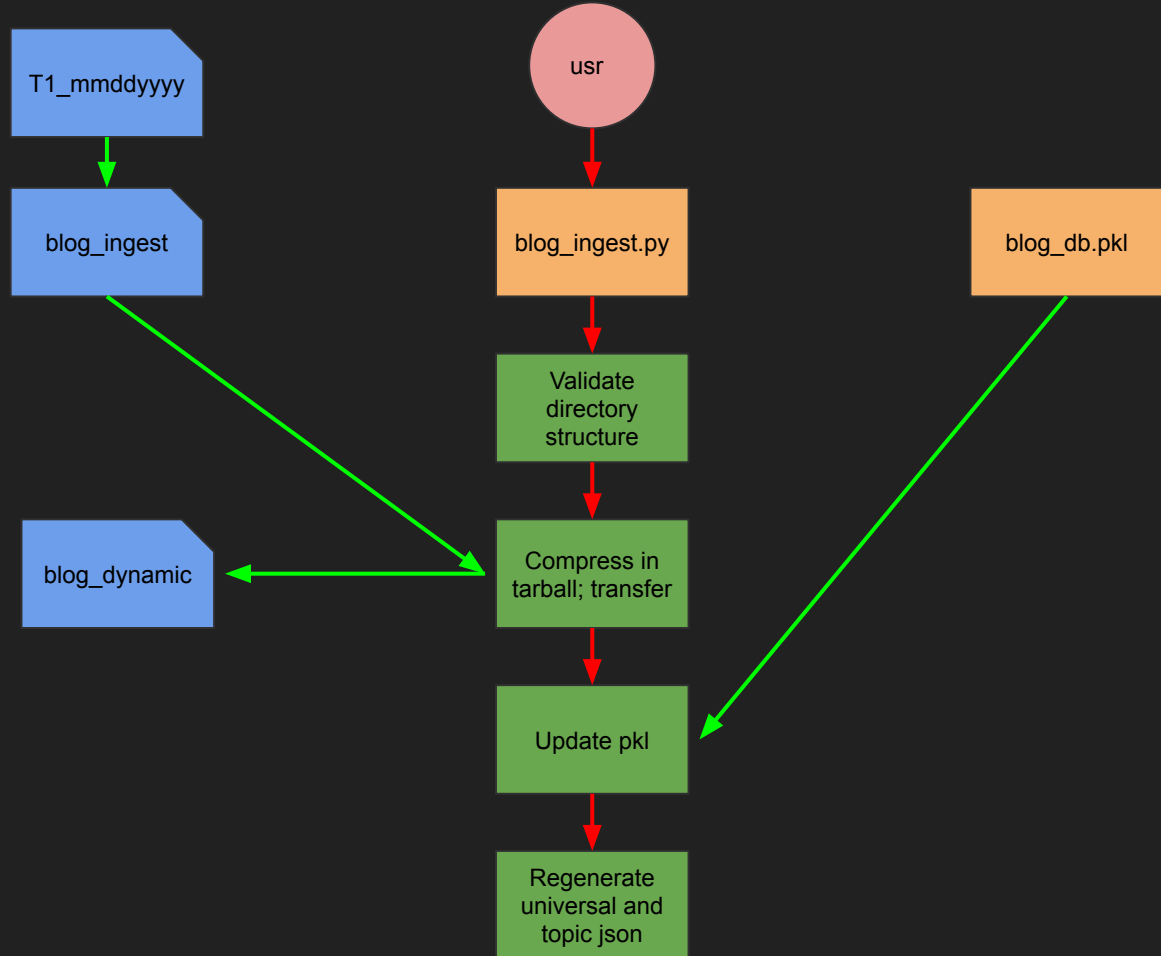# Front-End Prototype Overview

- Dark blue and grey color scheme
- Static background image
- Scroller for spotlighting specific articles
- Research Topics list to choose a category
- Articles appear as modules, which link to the article's dedicated page
- Possible infinite scrolling
- Mobile-friendly design
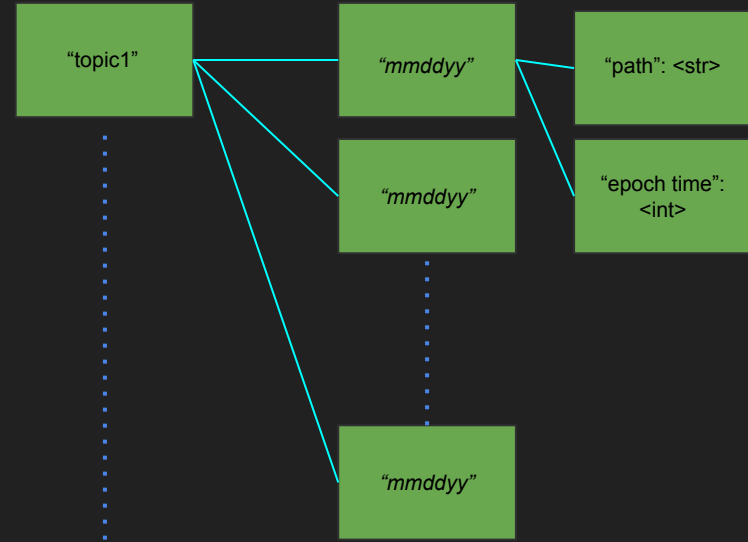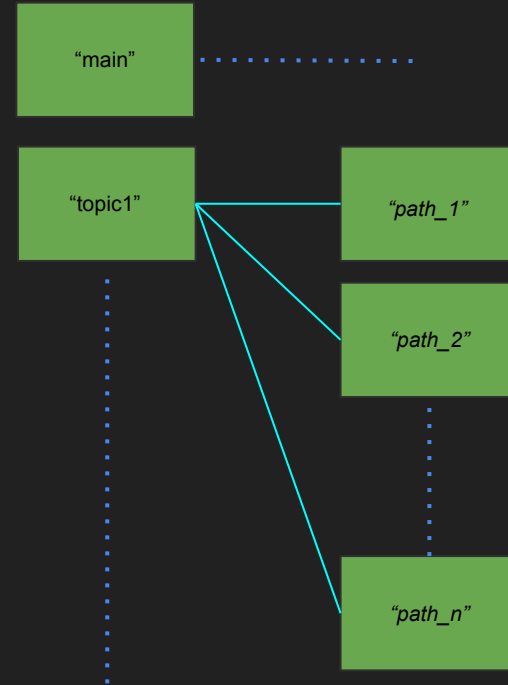
Proposed Directory Structure

Ingest Process

Article Upload Database (python pkl)

- Dictionaries of uploaded articles arranged by topic
- Uploaded articles are indexed by their upload date
- Each article is stored as a dictionary of their directory path and specific time of upload
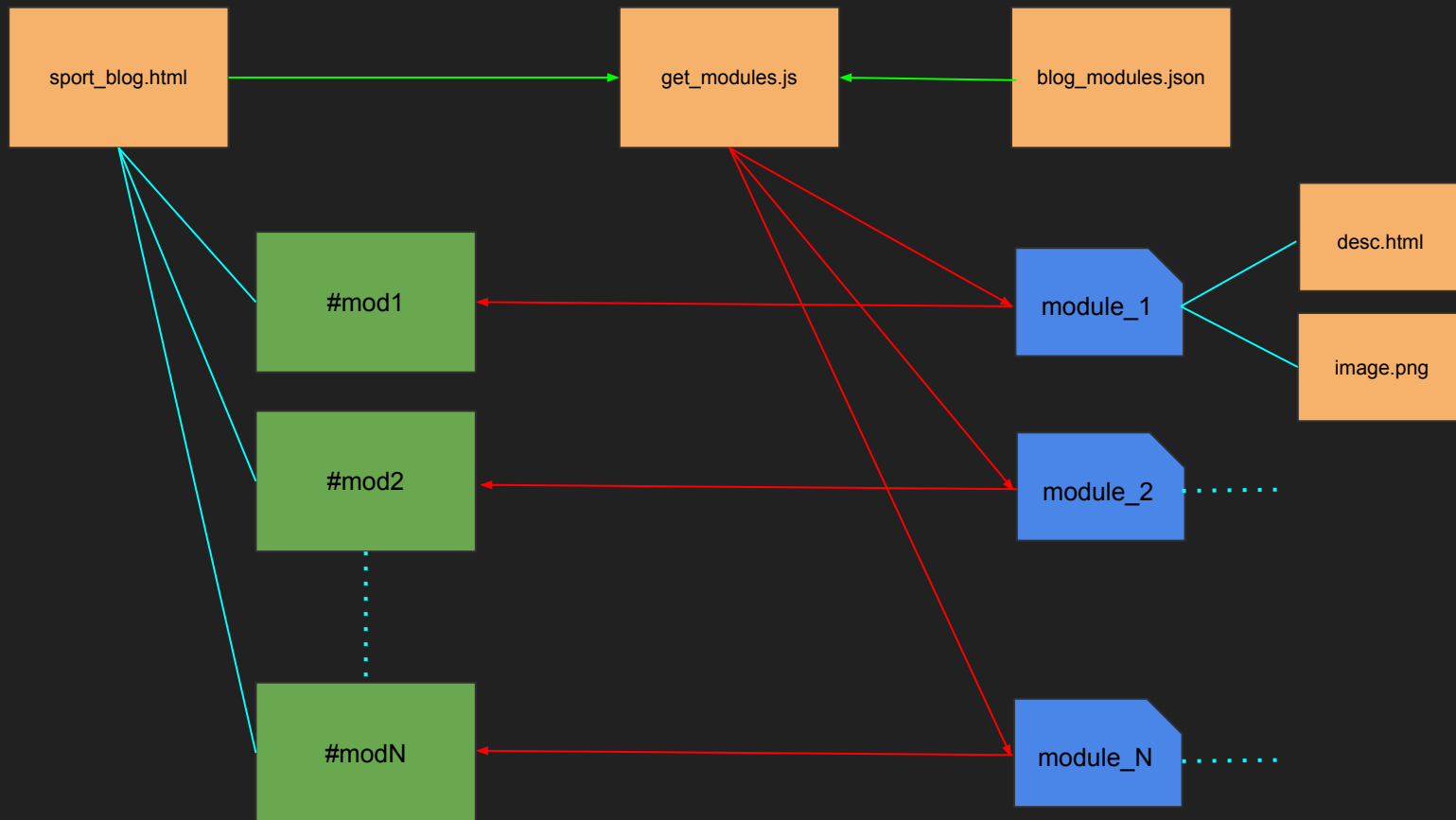- Articles will be easy to search and retrieve

Module Index (json)

- JSON - format file, regenerated upon new article upload
- Each topic has its own ordered array of paths to articles (most recent first)
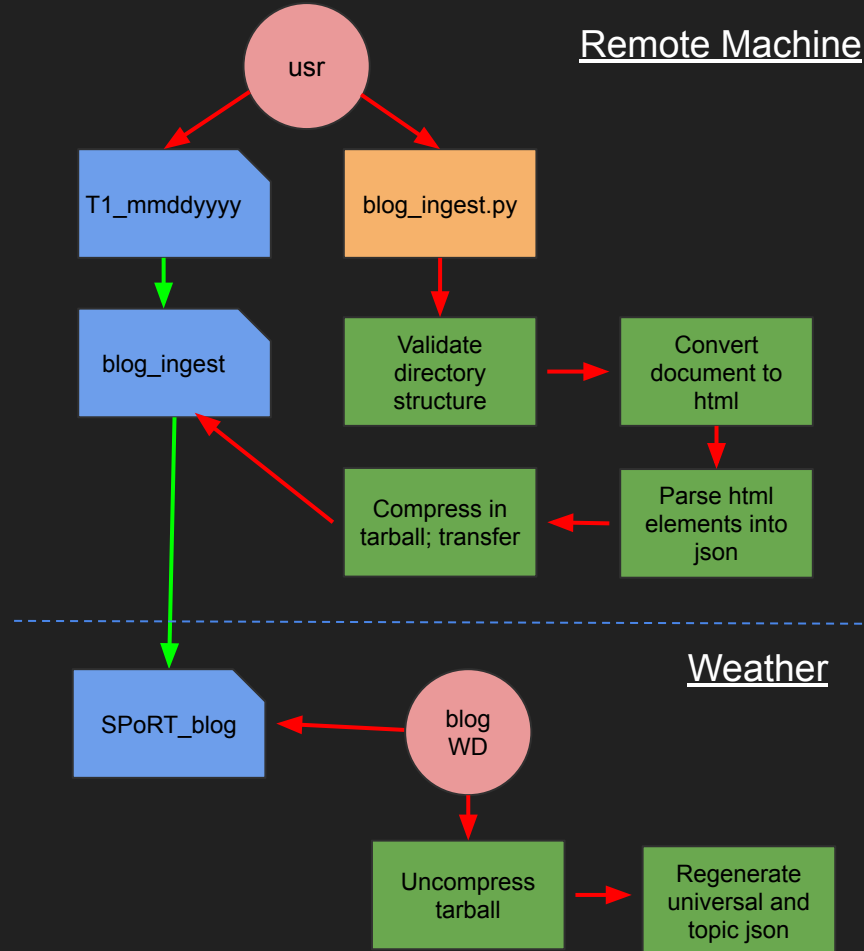- Landing page uses AJAX load to associate each module with a id tag

Module creation process

Problems to address

- Infinite scrolling may increase load time and make homepage difficult to navigate
- JSON array lengths will get very long if not split up
  - If JSON files are split up, multiple files will have to be regenerated per-ingest
- User is responsible for running ingest script (in order to handle format errors)
- Current file naming scheme makes it difficult to find a specific article based on subject
- Accidently removing pkl file would be pretty tragic. (backup method?)

Blog Publishing Process

usr

Remote Machine

T1_mmddyyyy

blog_ingest.py

blog_ingest

Validate directory structure

Convert document to html

Compress in tarball; transfer

Parse html elements into json

Weather

SPoRT_blog

blog WD

Uncompress tarball

Regenerate universal and topic json

Updated Proposed Directory Structure