Article Publishing Pipeline

lpmwfx, Denmark, EU

17.02.2026

Figure 1: lpmwfx articles

Article Publishing Pipeline

This article documents a fully automated publishing system that turns a single Markdown source into a responsive website and a professionally styled PDF, hosted on a custom domain with integrity verification.

Design Principles

The system follows one core rule: **Markdown is the single source of truth.** Everything else is generated. This means version control works naturally, diffs are readable, and there is zero risk of format drift between outputs.

Architecture

Every article lives in its own repository under the `articles-lpmwfx` GitHub organization. Each repository produces three outputs from one source:

1. **HTML** — a responsive, full-width web page served via GitHub Pages on a custom subdomain
2. **PDF** — a typeset document with Lato font, accent colors, and syntax highlighting, distributed as a tagged release
3. **Markdown** — the source file itself, readable directly on GitHub

All three formats link to each other and to a SHA256 checksum file for integrity verification.

The Pipeline

Each article passes through these stages:

- **Author** writes a draft in Danish or English
- **Claude** restructures the content into consistent Markdown with proper front matter

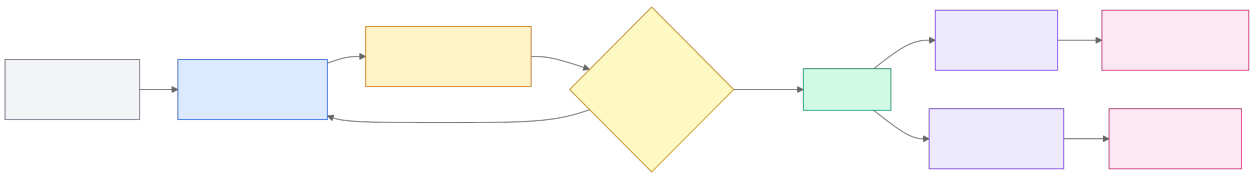

Figure 2: Publishing pipeline

- **Mistral Small** handles translation to English and proofreads spelling, grammar, and punctuation
- **Language Check** analyzes the text for AI-like patterns — if flagged, the text loops back for revision
- **Pandoc** generates responsive HTML (with Mermaid JS for diagrams) and styled PDF (with xelatex)
- **Local preview** via Python HTTP server confirms the result before publishing
- **Git push** and a **GitHub release** make everything public

Technology Stack

Component	Tool	Role
Source format	Markdown + YAML	Single source of truth
PDF engine	pandoc + xelatex	Typeset output with Lato font
HTML engine	pandoc + Mermaid JS	Responsive web with client-side diagrams
Diagrams	Mermaid (.mmd) + drawsvg	Flowcharts and programmatic SVG
Translation	Mistral Small API	Danish to English, proofreading
Formatting	Claude	Structure, consistency, front matter
Hosting	GitHub Pages	Static site per article
Domain	Njal.la DNS	CNAME per article subdomain
Language check	language-check.py	AI detection shield (burstiness, vocabulary, AI phrases)
Integrity	SHA256SUMS	Checksum for every output format
Feedback	GitHub Issues	Public, traceable, email-notified

Character Support

European characters render correctly across all formats:

- Danish: æ ø å — Æ Ø Å
- German: ä ö ü ß
- French: é è ê ë ç

Repository Structure

```
<article>/
```

```
├── article/
│   └── <name>.md          ← source of truth
├── assets/
│   ├── pipeline.mmd       ← Mermaid source
│   ├── pipeline.png       ← pre-rendered for PDF
│   └── badge.svg         ← drawsvg output
├── docs/
│   ├── index.html         ← generated HTML
│   ├── style.css         ← responsive CSS with dark mode
│   ├── CNAME             ← custom domain
│   └── SHA256SUMS        ← checksums
├── README.md
└── SHA256SUMS
```

Conclusion

The entire system runs from the command line with four scripts:

```
python3 app/mistral-proofread.py article/<name>.md
python3 app/language-check.py article/<name>.md
python3 app/build-html.py <article-dir>
python3 app/build-pdf.py <article-dir>
python3 app/build-checksums.py <article-dir>
```

No CMS, no database, no build server. Just Markdown, Git, and automation.

Also available as: HTML (.com) | HTML (.eu) | PDF | GitHub | Codeberg | SHA256 | Feedback