The `ltmermaid` package

安藤 遼哉（Ryoya Ando）
https://ryoya9826.github.io/

April 16, 2026

Abstract

The `ltmermaid` package renders Mermaid diagrams at LaTeX 2$_\varepsilon$ compile time when using **LuaLaTeX** and embeds them in the PDF. Rendering uses external tools such as the Mermaid CLI, so **shell escape** (`-shell-escape`) is required. The machine that runs `lualatex` must have a working **Node.js** installation, `mmdc` (or `npx`), and headless Chromium for the CLI.

1 Requirements

Item	Requirement
TeX engine	LuaLaTeX (`lualatex`).
Format	LaTeX 2$_\varepsilon$.
Shell escape	`lualatex -shell-escape` (the package invokes the CLI via `os.execute`).
External tool	Mermaid CLI (`mmdc`).

If the Mermaid CLI is missing from the environment, `ltmermaid` does **not** auto-install `mmdc` or similar. A typical setup uses **Node.js** with `npm` (or `npx`). If you pass a command that includes `npx -y` via the package option `Renderer`—for example `npx -y @mermaid-js/mermaid-cli …`—then `npx` may fetch and install packages as needed (**network access** may be required).

When you run `lualatex`, unless the `Renderer` option points to another command, the default renderer `mmdc` must be visible in the same environment (if you set `Renderer`, that command must be visible instead). The Mermaid CLI depends on headless Chromium (Puppeteer); see the Mermaid CLI documentation for details.

2 License and source

Source and issues: `https://github.com/ryoya9826/ltMermaid`

3 Usage

3.1 Minimal document

```
\documentclass{article}
\usepackage{ltmermaid}

\begin{document}
\begin{mermaid}
flowchart LR
  A --> B
\end{mermaid}
\end{document}
```

Compile with:

```
lualatex -shell-escape yourfile.tex
```

3.2 Layout adjustments (optional)

By default, included graphics are scaled with `adjustbox`. Example:

```
\MermaidAdjustBoxOpts{max width=0.8\linewidth,center}
\MermaidAdjustBoxOpts{max width=0.9\linewidth,center,valign=T}
```

3.3 Beamer

The package works with the `beamer` class. Any frame that contains a `mermaid` environment must use the `fragile` option (the environment relies on `fancyvrb`), for example `\begin{frame}[fragile]{Diagram}`.

```
\documentclass{beamer}
\usepackage{ltmermaid}

\begin{document}
\begin{frame}[fragile]{Mermaid}
\begin{mermaid}
flowchart LR
  A --> B
\end{mermaid}
\end{frame}
\end{document}
```

4 Package options (optional)

- **`Renderer`**: Prefix for the renderer command. If omitted, the default is `mmdc`. You can set it explicitly, for example `npx -y @mermaid-js/mermaid-cli`.

5 User macros

- **`\MermaidRendererOptions{...}`:** Extra CLI arguments before `-i` / `-o` (merged after the built-in `-f` when PDF fit is enabled).
- **`\MermaidNoPdfFit`:** Disables `-f` / `--pdfFit` for `mmdc` (by default this is *on*).
- **`\MermaidAdjustBoxOpts{...}`:** Full `adjustbox` key list around `\includegraphics` (default `max width=0.9\linewidth,center`).
- **`\MermaidGraphicsOpts{...}`:** Extra keys for `\includegraphics` (e.g. rotation, `trim`); width is usually set with `\MermaidAdjustBoxOpts`.

6 Output files

Intermediate `.mmd` and `.pdf` files are written under the `mermaid/` directory relative to the directory where compilation runs. With `-output-directory`, they are written under `mermaid/` inside the directory indicated by the environment variable `TEXMF_OUTPUT_DIRECTORY`.

7 Examples

Left: source for the `mermaid` environment as typed. **Right:** rendered diagram.

```
\begin{mermaid}
flowchart TB
  subgraph client["Client tier"]
    WEB["Browser / SPA"]
    CLI["CLI / batch"]
  end
  subgraph edge["Edge"]
    GW{{API Gateway}}
  end
  subgraph svc["Services"]
    AUTH["Auth"]
    API["Business API"]
    WORK["Workers"]
  end
  subgraph store["Data"]
    DB[("PostgreSQL")]
    CACHE[("Redis")]
    QUEUE["Job queue"]
  end
  WEB --> GW
  CLI --> GW
  GW --> AUTH
  GW --> API
  API --> WORK
  API --> DB
  API --> CACHE
  WORK --> QUEUE
  WORK --> DB
\end{mermaid}
```

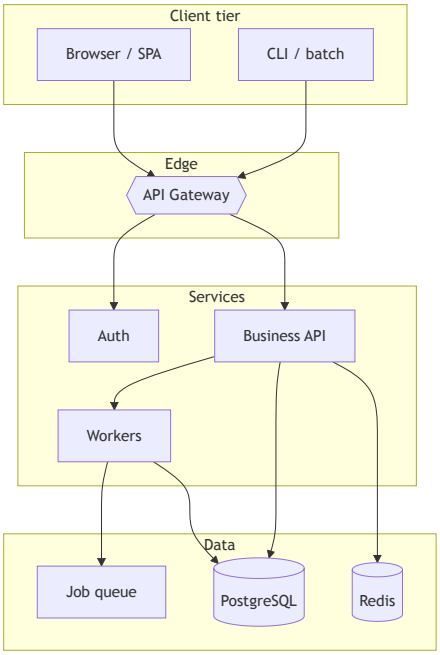

Figure 1: Layered architecture (subgraphs and node shapes)

```
\begin{mermaid}
sequenceDiagram
  autonumber
  actor U as User
  participant B as Browser
  participant A as Auth API
  participant S as Business API
  participant D as DB
  U->>B: Log in
  B->>+A: POST /token
  A->>D: Verify user
  D-->>A: Row
  A-->>-B: JWT
  B->>+S: GET /orders (Bearer)
  S->>A: Validate token
  A-->>S: Claims
  S->>D: SELECT
  D-->>S: Rows
  S-->>-B: 200 JSON
  B-->>U: Show list
\end{mermaid}
```

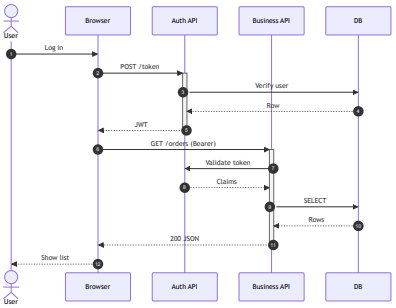

Figure 2: Sequence diagram (autonumber, async arrows, multiple participants)

```
\begin{mermaid}
stateDiagram-v2
  [*] --> Draft: Create
  Draft --> Review: Submit
  Review --> Draft: Send back
  Review --> Approved: Approve
  Approved --> Published: Publish
  Published --> Archived: Close
  Review --> Rejected: Reject
  Rejected --> [*]
  Archived --> [*]
\end{mermaid}
```

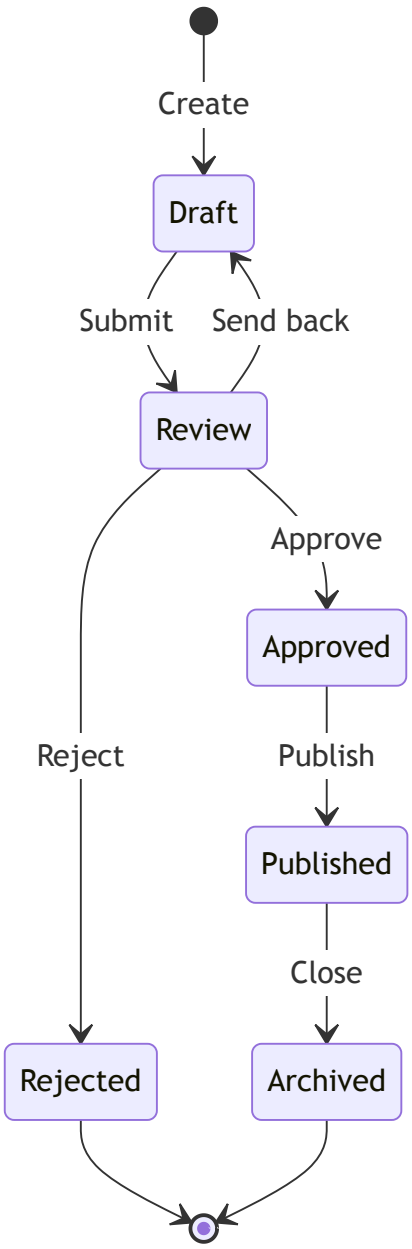

Figure 3: State machine (`stateDiagram-v2`)

8 Version history

Version 1.0 (2026-04-16): Stable release. LuaLaTeX Mermaid CLI integration via `os.execute`, `Renderer` option and `\Mermaid…` macros, plus `-output-directory` support.

Version 0.2 (2026-04-13): Removed `MERMAID_MMDC` and `MERMAID_MMDC_OPTIONS`.
Version 0.1 (2026-04-08): Initial release.