The `mermaid` package (`mermaid.sty`)

Ryoya Ando
https://ryoya9826.github.io/

April 20, 2026

Abstract

`mermaid.sty` invokes the Mermaid CLI via `shellesc` and `\write18`, renders Mermaid diagrams during a LaTeX 2_ε run, and embeds them as PDF. It can be used with engines such as upLaTeX, pdfLaTeX, and LuaLaTeX. Rendering Mermaid diagrams requires the Mermaid CLI (`mmdc`, etc.) and headless Chromium (Puppeteer). Compile with shell escape enabled (`-shell-escape`).

1 Operating conditions

Item	Condition
Format	LaTeX 2_ε
Shell escape	`-shell-escape` (invoke the CLI via `shellesc` / `\write 18`)
External tool	Mermaid CLI (`mmdc`)

If the Mermaid CLI is not present in your environment, `mermaid.sty` does not auto-install `mmdc` or similar. A typical Mermaid CLI installation uses **Node.js** with `npm` (or `npx`). If you set the package option `Renderer` to a command that includes `npx -y`, for example `npx -y @mermaid-js/mermaid-cli …`, then `npx` may fetch and install packages as needed (**network access** may be required).

When compiling, unless the `Renderer` option names 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.

Log output uses `\typeout` with the `[mermaid]` prefix. If something goes wrong, check the `.err` files under `mermaid/`.

2 License and source

LaTeX Project Public License (LPPL) 1.3c or later.
Source and issues: `https://github.com/ryoya9826/ltMermaid`

3 Usage

3.1 Minimal document

```
\documentclass{article}
\usepackage{mermaid}

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

Compile example:

```
pdflatex -shell-escape yourfile.tex
lualatex -shell-escape yourfile.tex
uplatex -shell-escape yourfile.tex && dvipdfmx yourfile.dvi
```

3.2 Layout adjustments (optional)

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

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

3.3 Beamer

The package can also be used with the `beamer` class. Every frame that contains a `mermaid` environment must use the `fragile` option (the environment depends on `fancyvrb`). Example: `\begin{frame}[fragile]{Figure}`.

```
\documentclass{beamer}
\usepackage{mermaid}

\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, `mmdc` is the default. You can specify it explicitly, for example `npx -y @mermaid-js/mermaid-cli`.

5 User commands

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

6 Output files

Intermediate `.mmd` files and rendered `.pdf` files are written under the `mermaid/` directory in the directory from which you run the compiler. File names are of the form `mermaid-doc-mermaid-1.mmd`, `mermaid-doc-mermaid-1.pdf`, …, using `mermaid-doc` (the job name) and a running index per diagram.

7 Diagram examples

Left: source typed into the `mermaid` environment. **Right:** rendered result.

```
\begin{mermaid}
flowchart TB
  subgraph client["Client tier"]
    WEB["Browser / SPA"]
    CLI["CLI / batch"]
  end
  subgraph edge["Edge"]
    GW{{API Gateway}}
  end
  subgraph svc["Service tier"]
    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}
```

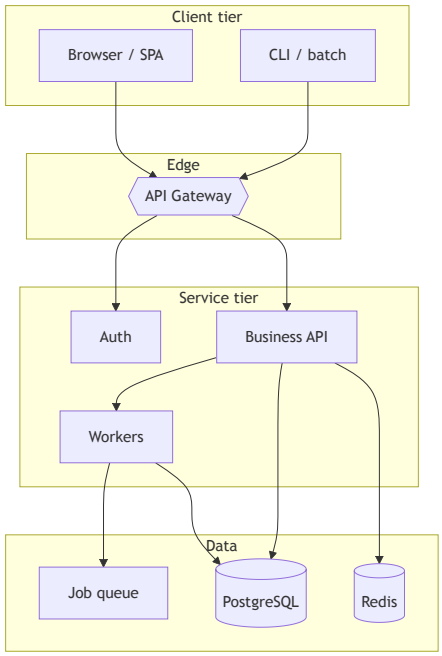

Figure 1: Multi-layer architecture (subgraphs and node shapes)

```
\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: List view
\end{mermaid}
```

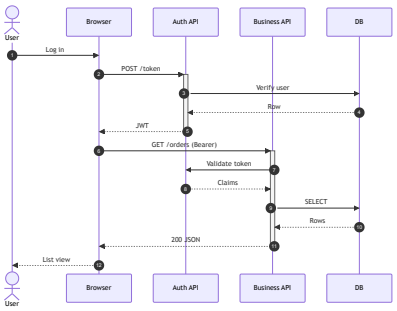

Figure 2: Sequence diagram (numbering, asynchronous arrows, multiple participants)

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

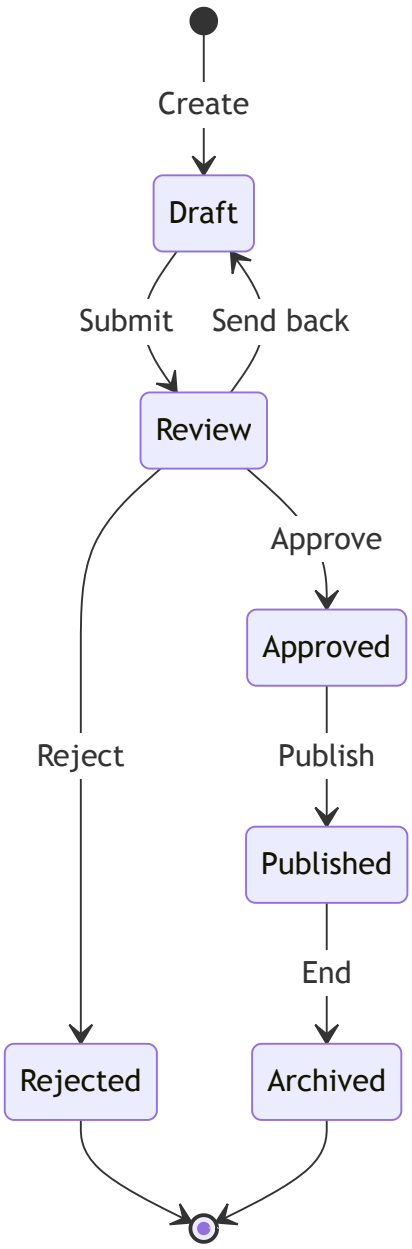

Figure 3: State transitions (`stateDiagram-v2`)

8 Revision history

Version 1.0a 2026-04-20 Renamed sample documents.

Version 1.0 2026-04-16 Stable release.