■ Internal Documentation Writing Guide

Unified rules and templates for writing technical documentation in Notion before migration to Docusaurus.

Part 1 — Writing Principles

Clarity & Consistency

- Use simple, professional English. Avoid slang or unnecessary jargon.
- Follow the same structure and headings across all documentation pages.

Audience

- Assume the reader is a new team member or someone from another department.
- Avoid assumptions; explain acronyms and context clearly.

Tone

- Write in active voice (e.g., 'The service validates data' not 'Data is validated').
- Keep sentences short and direct.

Formatting Rules

- Use headings (##) for major sections and subheadings (###) for subsections.
- Use bullet points for lists, and tables for structured information (like inputs, configs, parameters).
- Highlight filenames, commands, and code using backticks (`).

Diagrams & Visuals

- Each document should include at least one diagram or flow representation.
- Use consistent color-coding and minimal text in diagrams.

Links & References

- Add links to related services, repositories, or dashboards.
- Avoid broken links; check them before publishing.

Change Management

- Each update must be logged in the changelog with version and author.
- Mark old information as deprecated, not deleted.

Part 2 — Notion Documentation Structure

Each Notion page must follow this structure before being migrated to Markdown/Docusaurus.

1. Overview

Describe what the system/module/process is, why it exists, and who uses it.

2. Architecture / Design

Add high-level diagrams and describe the system structure, integrations, and data flow.

3. Responsibilities

List the specific tasks, objectives, or functionalities of this component.

4. Technical Details

Include tools, frameworks, databases, APIs, and deployment setup. For chemical/process teams, include operating ranges and units.

5. Data Flow / Workflow

Explain how data moves or how the process works step by step.

6. Dependencies & Interfaces

List services, APIs, data sources, or instruments this component interacts with.

7. Inputs & Outputs

Define input/output parameters with formats, units, and sources.

8. Configuration & Parameters

List environment variables, constants, or calibration parameters.

9. Monitoring & Validation

Describe how the system is monitored, KPIs, dashboards, and alerting.

10. Known Issues / Lessons Learned

Document any recurring problems and how they were solved.

11. Future Improvements / Roadmap

List upcoming features, optimizations, or ideas.

12. Ownership

Specify Owner, Reviewer, Team, and Last Updated Date.

13. References

Add related documents, code links, or academic references.

14. Change Log

Track version history, authors, and updates.

Part 3 — Team-Specific Rules

Backend Team

- Include API endpoints (method, path, purpose, authentication).
- Add database schema references (tables, keys, relationships).
- Mention Kafka or Redis topics used, with payload format.
- Document environment variables and service dependencies.

Frontend Team

- Describe routing structure and key pages/components.
- List dependencies and state management libraries used.
- Include screenshots or diagrams of UI components.
- Document integration with backend endpoints and websockets.

AI Team

- List all models used (type, task, input/output, location).
- Document training data sources and validation results.
- Explain serving and deployment pipelines.
- Include performance metrics (latency, accuracy, drift).

Chemical Engineering Team

- Document the process flow diagram (PFD) and main equipment.
- List main variables with units and operating ranges.
- Describe constraints, alarms, and interlocks.
- Add lab validation data and comparison with simulation.

Part 4 — Review & Approval Workflow

- Each page must have an assigned **Owner** and **Reviewer**.
- Status lifecycle: Draft → In Review → Approved → Published.
- Reviewer checks for accuracy, consistency, and formatting.
- After approval, set 'Published' tag and record update in changelog.

Part 5 — Consistency Checklist

- ■ Same heading order and names across all docs.
- ■ Each file includes Author, Owner, and Last Updated date.
- ■ All tables use consistent formatting (3 columns min).
- ■ All diagrams are labeled and easy to understand.
- ■ No secrets or tokens in documentation.
- ■ Each doc has a changelog entry.

Part 6 — Maintenance & Review

- Perform a documentation review every quarter per team.
- Archive outdated pages older than 1 year.
- Link documentation to release versions (e.g., 2025.10).
- Keep ownership information updated.