Technical Specification

CONFIDENTIAL – INTERNAL USE ONLY

Author: Matt Jeffcoat

Organization: [Organization Name]

Date: November 2025

Version: 1.0

This document is confidential and intended solely for authorized personnel and approved contractors. Unauthorized distribution is prohibited.

Documentation Pipeline: Technical Specification

Sample document demonstrating the complete docs-pipeline workflow: automated diagram generation, markdown processing, and professional PDF output.

Executive Summary

The docs-pipeline system provides an enterprise-grade documentation workflow that transforms Markdown sources into publication-quality PDFs with embedded architecture diagrams, syntax-highlighted code, and professional styling. This specification demonstrates the system's capabilities through automated generation of technical documentation.

Architecture Components

Pipeline Orchestrator

Coordinates multi-stage workflow execution using YAML configuration. The orchestrator loads configuration files, manages workspace execution, and coordinates between diagram generation and document conversion stages.

Diagram Generator

Converts Structurizr DSL to multiple formats (Mermaid, PNG, SVG, PlantUML). Uses Docker-based Structurizr CLI for reliable, consistent diagram generation across platforms.

PDF Generator

Uses Playwright for pixel-perfect rendering with Mermaid diagram optimization. Supports advanced features including: - Code syntax highlighting - Mathematical equations (KaTeX) - Mermaid diagram embedding - Custom styling profiles

CLI Interface

Provides command-line access to all pipeline features: - Single document conversion - Full pipeline execution - Dry-run mode for testing - Parallel processing for batch operations

Workflow

1. **Load Config** - Parse YAML configuration file
2. **Generate Diagrams** - Execute Structurizr export for each workspace
3. **Convert Documents** - Render PDFs with embedded diagrams
4. **Validate Output** - Ensure all artifacts generated successfully

This entire document, including architecture diagrams and professional formatting, was generated by the pipeline in a single command.

Usage Example

```
# Generate everything
python -m tools.docs_pipeline.cli --config docs-pipeline.yaml

# With parallel processing
python -m tools.docs_pipeline.cli --config docs-pipeline.yaml --parallel

# Dry run to preview
python -m tools.docs_pipeline.cli --config docs-pipeline.yaml --dry-run
```

Integration

The pipeline integrates seamlessly with: - **CI/CD Systems** - Generate documentation in build pipelines - **Web Interfaces** - Upload configs via web UI - **Version Control** - Track DSL and Markdown changes together - **Documentation Sites** - Publish generated PDFs automatically