High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **StoryWeaver - AI Creative Writing Workshop**. StoryWeaver is a collaborative platform enabling users to co-author stories with generative AI, receive real-time creative suggestions, manage versions, provide peer feedback, and export manuscripts. The system supports multiple genres and languages, with a focus on creative education, teamwork, modularity, and secure user management.

---

1. System Architecture Overview

Architecture Description:
StoryWeaver follows a modular, service-oriented architecture with a React-based frontend, Python backend (API & AI services), and PostgreSQL database. Key modules interact via RESTful APIs and secure authentication.

Module	Role
Frontend (React)	User interface, real-time collaboration, story editing, feedback
API Gateway (Python)	Orchestrates requests, authentication, and routing
AI Service (Python)	Provides generative AI suggestions (plot, dialogue, style)
Version Control	Manages story versions, tracks changes, supports branching/merging
Peer Feedback	Enables user comments, ratings, and collaborative review
User Management	Handles registration, authentication, roles, and permissions
Database (PostgreSQL)	Stores users, stories, versions, feedback, and metadata
Export Service	Generates exportable manuscripts (PDF, DOCX, etc.)

---

2. Component Interactions

Sequence Step	Interaction Description
1. User Login	Frontend → API Gateway → User Management → Database
2. Story Editing	Frontend ↔ API Gateway ↔ AI Service (for suggestions)
3. Version Save	Frontend → API Gateway → Version Control → Database
4. Peer Feedback Submission	Frontend → API Gateway → Peer Feedback → Database
5. Manuscript Export	Frontend → API Gateway → Export Service → Database (retrieve) → User

---

3. Data Flow Overview

Data Flow	Source	Destination	Purpose
User Auth Data	Frontend	User Management	Secure login, registration
Story Content	Frontend	Version Control	Save, update, retrieve story drafts
AI Suggestions	AI Service	Frontend	Real-time creative input
Feedback Data	Frontend	Peer Feedback	Store/retrieve peer reviews
Exported Manuscripts	Export Service	Frontend	Downloadable story files

---

4. Technology Stack

Layer/Component	Technology/Framework
Frontend	React, WebSockets
Backend/API	Python (FastAPI/Flask)
AI Service	Python (OpenAI/LLM APIs)
Database	PostgreSQL
Version Control	Custom logic, PostgreSQL
Authentication	JWT, OAuth2
Export Service	Python (docx/pdf libraries)
Deployment	Docker, Kubernetes (optional)
CI/CD	GitHub Actions, open-source tools

---

5. Scalability, Reliability & Security

- **Scalability:**
  Modular services enable horizontal scaling (e.g., AI Service, API Gateway). Stateless backend supports load balancing. Database optimized for concurrent access.
- **Reliability:**
  Version control ensures data integrity. Regular backups and monitoring. Graceful error handling and retry logic.
- **Security:**
  Secure authentication (JWT/OAuth2), encrypted data in transit, role-based access control, input validation, and audit logging.

End of Document