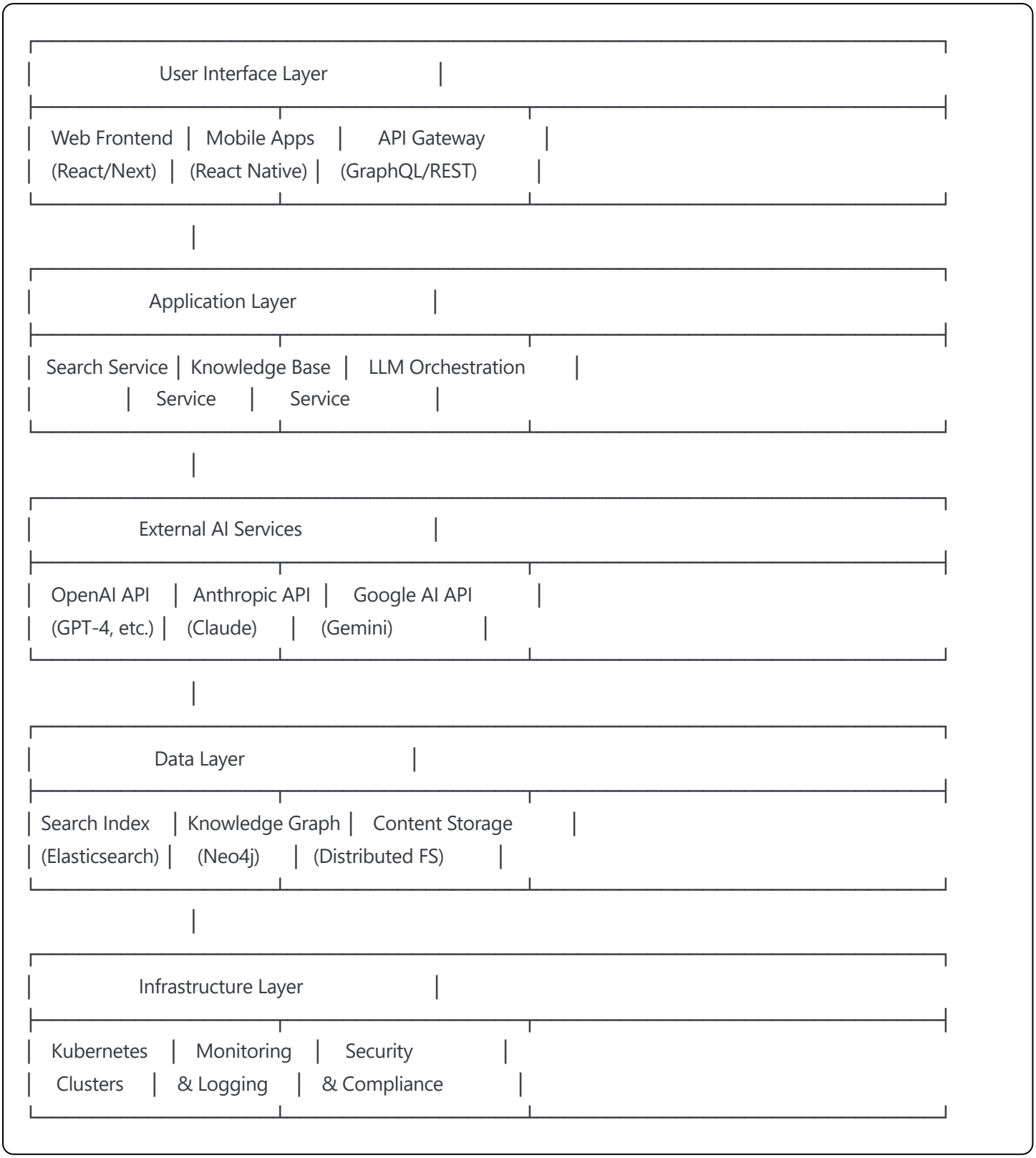
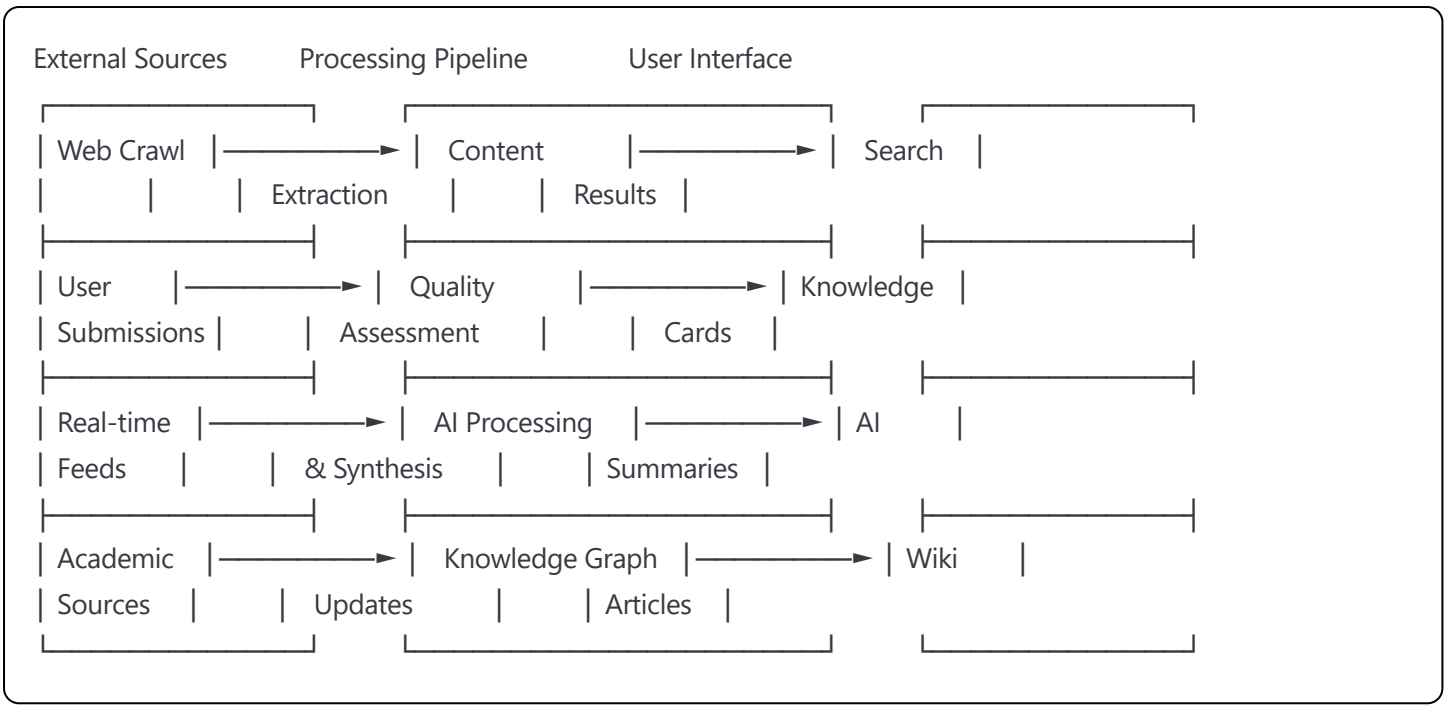
Universal Knowledge Platform - System Architecture (Using Existing LLMs)

High-Level Architecture

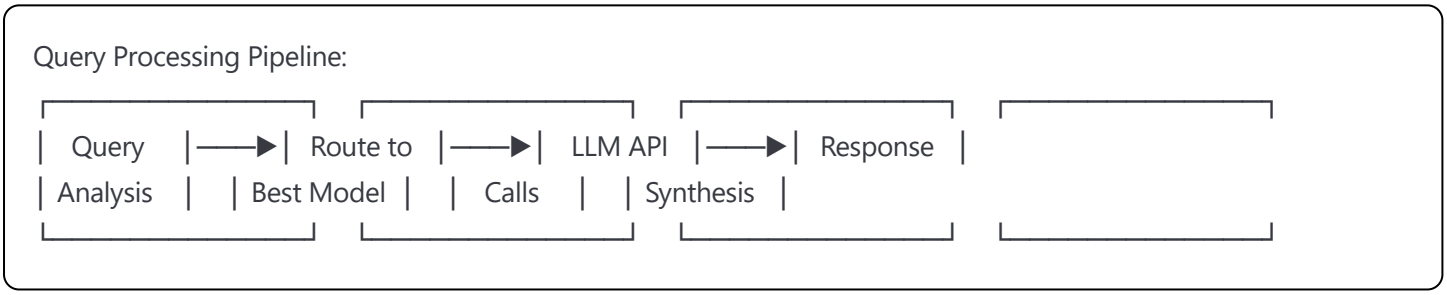


Data Flow Architecture



LLM Orchestration Strategy

Multi-Model Approach



Model Selection Logic

- **GPT-4**: Complex reasoning, creative tasks, code
- **Claude**: Long-form analysis, ethical reasoning
- **Gemini**: Multimodal content, real-time data
- **Specialized Models**: Domain-specific tasks (medical, legal, etc.)

Microservices Architecture

Core Services

1. **Search Service**
   - Query processing and understanding
   - Index management and ranking
   - Result aggregation and ranking
2. **LLM Orchestration Service**
   - Multi-model routing and load balancing

- Response quality assessment
- Cost optimization and rate limiting
- Fallback and retry logic

3. **Knowledge Management Service**
   - Wiki article management
   - Version control and collaboration
   - Editorial workflows
4. **User Management Service**
   - Authentication and authorization
   - User profiles and preferences
   - Reputation and trust scoring
5. **Content Ingestion Service**
   - Web crawling coordination
   - Content processing and extraction
   - Quality assessment and filtering

Supporting Services

- **Notification Service**: Real-time updates and alerts
- **Analytics Service**: Usage tracking and insights
- **Recommendation Service**: Personalized content suggestions
- **Translation Service**: Multi-language support
- **Moderation Service**: Content quality and safety

Data Storage Strategy

Primary Databases

1. **Search Index (Elasticsearch)**
   - Full-text search capabilities
   - Real-time indexing
   - Faceted search and filtering
2. **Knowledge Graph (Neo4j)**
   - Entity relationships
   - Semantic connections
   - Graph-based queries
3. **Content Store (Distributed File System)**

- Raw content storage
- Version control
- Media and document storage

4. **User Data (PostgreSQL)**
   - User profiles and preferences
   - Activity logs and history
   - Social graph data

Caching Layer

- **Redis Cluster**: Session management, frequent queries
- **CDN**: Static content and media delivery
- **Application Cache**: Computed results and summaries

Scalability Considerations

Horizontal Scaling

- Microservices can scale independently
- Load balancing across multiple instances
- Database sharding and replication

Performance Optimization

- Aggressive caching strategies
- Asynchronous processing pipelines
- Edge computing for global latency

Reliability

- Multi-region deployment
- Automated failover and recovery
- Data backup and disaster recovery