Contract Builder Platform - Product Requirements Document

1. Executive Summary

Vision

Build a modular, block-based contract management platform that enables businesses to create, manage, and execute contracts across multiple industries with embedded integration capabilities and a comprehensive template marketplace.

Key Differentiators

- **Modular Block Architecture**: Drag-and-drop contract building with reusable components
- **Template Marketplace**: Global template system with tenant customization
- **Industry Agnostic**: Adaptable to any business vertical
- **Multi-tenancy**: Separate test/live environments with role-based access
- **Version Control**: Complete audit trail with rollback capabilities
- **Embeddable Integration**: White-label contract creation for customer websites

2. Product Overview

2.1 Core Concept

The platform treats contracts as compositions of reusable, configurable blocks that can be arranged, reordered, and customized based on business needs. Templates serve as the foundation, providing pre-configured block arrangements for specific industries and use cases.

2.2 Target Users

- **Super Admins**: Platform owners who create and manage global templates
- **Tenant Admins**: Business owners who manage their organization's contracts and templates
- **End Users**: Team members who create and manage individual contracts
- **Integration Partners**: Businesses embedding contract functionality into their platforms

2.3 User Roles & Permissions

Super Admin (Platform Owner)

- Create and manage global templates
- Access template analytics across all tenants
- Platform configuration and maintenance
- Revenue and usage analytics

Tenant Admin

- Manage organization settings and users
- Create and edit local templates
- Access all contracts within organization
- Configure billing and integration settings

Tenant User

- Create contracts from available templates
- Manage assigned contracts
- View organization templates (read-only)
- Basic reporting and analytics

3. Template Architecture & Management

3.1 Template Ownership Model

Global Templates (Marketplace)

- **Owner**: Super Admin (Platform)
- **Storage**: `tenantId = "admin"` + `globalTemplate = true`
- **Access**: Read-only for all tenants
- **Purpose**: Industry-standard templates available to all users
- **Menu Location**: `Implementation Toolkit > Global Templates` (Admin only)

Local Templates (Tenant-Specific)

- **Owner**: Individual Tenants
- **Storage**: `tenantId = currentTenant` + `globalTemplate = false`
- **Access**: Full CRUD for owning tenant
- **Purpose**: Customized templates for specific business needs
- **Menu Location**: `Settings > Template Designer`

My Templates (Unified View)

- **Content**: Global templates + Local templates available to tenant
- **Access**: Browse, preview, clone, create contracts

- **Menu Location**: `Settings > My Templates`

3.2 Template Management Flows

Super Admin Template Flow

Implementation Toolkit → Global Templates

├── Template Gallery (All Global Templates)

├── Create Global Template (GlobalTemplateDesigner)

├── Template Analytics & Usage Stats

└── Marketplace Management

Global Template Actions:

- 📋 **Preview** - View template structure and blocks
- 📁 **Clone** - Create copy for testing/modification
- ✏️ **Edit** - Modify template (creates new version)
- 🔄 **Change Status** - Activate/Deactivate template

Tenant Template Flow

Settings → Template Designer

├── Create New Template (Template Builder)

├── Block Configuration System

├── Preview & Testing

└── Save as Local Template

Settings → My Templates

├── Global Templates (Marketplace view)

├── My Templates (Local templates)

├── Recently Used (Contract history)

└── Template Management

Local Template Actions:

- 📋 **Preview** - View template structure
- 📁 **Clone** - Create copy for modification
- 📄 **Create Contract** - Start contract creation flow
- ✏️ **Edit** - Modify template (version control)
- 🔄 **Change Status** - Activate/Deactivate

Global Template Actions (Tenant View):

- 📋 **Preview** - View template structure
- 📁 **Copy to My Space** - Clone to local templates
- 📄 **Create Contract** - Copy template and start contract

3.3 Template Copying & Version Control

Global to Local Template Copying

When tenant copies a global template:

-- Original Global Template

globalTemplate = true, tenantId = "admin", templateId = "global_001"

-- Copied Local Template

globalTemplate = false, tenantId = "tenant_123", templateId = "local_001"

globalTemplateId = "global_001" (traceability)

parentId = null (first record in tenant space)

version = 1, isActive = true

Template Version Control

When editing any template:

-- Original Template (marked inactive)

templateId = "template_001", version = 1, isActive = false

```
-- New Version (active)

templateId = "template_002", parentId = "template_001"

version = 2, isActive = true
```

Benefits:

- Complete audit trail and change tracking
- Rollback capability to previous versions
- Referential integrity maintained
- Safe concurrent editing

4. Contract Types & Architecture

4.1 Service Contracts (Buyer ↔ Seller)

Purpose: Direct business relationships with full lifecycle management

Components:

- Event tracking and SLA compliance
- Milestone-based billing and payments
- Service commitments with recurring cycles
- Equipment/asset dependencies and management
- Comprehensive timeline and reporting

4.2 Partnership Contracts (Seller ↔ Partner)

Purpose: Revenue sharing and commission-based relationships

Components:

- Percentage-based revenue sharing models
- Commission structures and payment terms
- Partner performance tracking
- Simplified workflow (reduced complexity vs service contracts)
- Revenue reporting and analytics

5. Block System Architecture

5.1 Block Categories

Core Blocks (Single Instance Per Contract)

Block Name	Description	Dependencies	Required
Contact Selection	Choose contract recipient	Contact database	Yes
Base Contract Details	Title, dates, description	Contact selection	Yes
Contract Acceptance	Payment/Sign-off/Auto criteria	Base details	Yes
Equipment/Asset Selection	Physical assets involved	Equipment catalog	Conditional
Billing Rules	Payment terms and cycles	Service commitments	Yes (Service)

Event Blocks (Multi-Instance)

Block Name	Description	Dependencies	Rules
Service Commitments	Calibration, maintenance, etc.	Equipment (if applicable)	Quantity + Cycle based
Milestone Events	Project checkpoints	Base details	Custom scheduling
Review Cycles	Periodic contract reviews	Base details	Configurable intervals

Content Blocks (Multi-Instance, Reorderable)

Block Name	Description	Dependencies	Validation
Legal Clauses	Terms, conditions, policies	None	Template library
Image Upload	Visual documentation	None	File type/size limits
Video Upload	Training, process videos	None	File type/size limits
Document Upload	Supporting materials	None	File type/size limits

Commercial Blocks

Block Name	Contract Type	Description
Billing Rules	Service	Monthly, quarterly, milestone-based
Revenue Sharing	Partnership	Percentage splits, commission rates
Payment Terms	Both	Net terms, late fees, discounts

5.2 Block Validation Framework

```
interface BlockRule {
  id: string;
  name: string;
  category: 'core' | 'event' | 'content' | 'commercial';

  contractTypes: ('service' | 'partnership')[];

  maxInstances: number | null; // null = unlimited

  dependsOn: string[]; // Required predecessor blocks

  equipmentDependent: boolean;

  validationRules: ValidationRule[];

  templateLibrary?: string; // For clause blocks

}
```

6. User Experience Flows

6.1 Template Creation Journey

1. Template Design Initiation

   ├── Super Admin: Implementation Toolkit → Global Templates → Create

   └── Tenant: Settings → Template Designer → Create

2. Template Configuration

   ├── Basic Information (name, industry, type, description)

   ├── Block Selection & Arrangement

   ├── Block Configuration & Validation

   └── Preview & Testing

3. Template Finalization

   ├── Save & Activate

   ├── Version Control

└── Usage Analytics Setup

6.2 Contract Creation Journey

1. Template Selection

├── From My Templates → Select Template

├── From Contracts Menu → Create → Select Template

└── From Global Templates → Copy & Use

2. Contract Configuration (Dynamic based on contract type)

├── Contract Type Confirmation

├── Recipient Selection & Verification

├── Acceptance Criteria Definition

├── Block Assembly & Configuration (Service Contracts)

├── Timeline Preview & Validation

├── Billing Configuration (Service Contracts)

└── Final Review & Approval

3. Contract Execution

├── Send for Acceptance

├── Track Status & Progress

└── Manage Lifecycle Events

Note: Partnership contracts follow simplified path (steps 1, 2a-2c, 2g, 3)

6.3 Template Management Journey

```
1. Template Discovery
   ├── Browse Global Templates (Marketplace)
   ├── Search & Filter Options
   └── Preview & Evaluation

2. Template Acquisition
   ├── Copy Global Template to Local Space
   ├── Clone Existing Local Template
   └── Create New Template from Scratch

3. Template Customization
   ├── Edit Template Structure
   ├── Modify Block Configuration
   ├── Test & Validate Changes
   └── Version Control Management

4. Template Utilization
   ├── Create Contracts from Template
   ├── Monitor Usage Analytics
   └── Optimize Based on Performance
```

7. Technical Architecture

7.1 Enhanced Database Schema

Templates Table

```
CREATE TABLE templates (
  id VARCHAR PRIMARY KEY,
  name VARCHAR NOT NULL,
  description TEXT,
  industry VARCHAR,
  contractType ENUM('service', 'partnership'),

  -- Ownership & Access Control
  tenantId VARCHAR NOT NULL,
  globalTemplate BOOLEAN DEFAULT false,
  globalTemplateId VARCHAR NULL, -- Traceability for copied templates

  -- Version Control System
  parentId VARCHAR NULL, -- Reference to original template
  version INTEGER DEFAULT 1,
  isActive BOOLEAN DEFAULT true,

  -- Template Configuration
  blocks JSON NOT NULL,
  complexity ENUM('simple', 'medium', 'complex'),
  estimatedDuration VARCHAR,
  tags JSON,

  -- Analytics & Performance
  usageCount INTEGER DEFAULT 0,
  rating DECIMAL(3,2) DEFAULT 0,
  isPopular BOOLEAN DEFAULT false,
  successRate DECIMAL(5,2) DEFAULT 0,

  -- Metadata
  createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  createdBy VARCHAR,
  lastModifiedBy VARCHAR,

  -- Indexes
  INDEX idx_tenant_global (tenantId, globalTemplate),
  INDEX idx_parent_version (parentId, version),
  INDEX idx_active_templates (isActive, tenantId),
  INDEX idx_usage_stats (usageCount, rating, isPopular)
);
```

Contracts Table

```
CREATE TABLE contracts (
  id VARCHAR PRIMARY KEY,
  templateId VARCHAR NOT NULL,
  contractNumber VARCHAR UNIQUE,
  name VARCHAR NOT NULL,
  description TEXT,
  status ENUM('draft', 'pending_review', 'pending_acceptance', 'active', 'completed', 'cancelled', 'expired'),
  contractType ENUM('service', 'partnership'),

  -- Parties Involved
  buyerId VARCHAR NOT NULL,
  buyerDetails JSON,
  sellerId VARCHAR NOT NULL,
  sellerDetails JSON,
  partnerId VARCHAR NULL, -- For partnership contracts
  partnerDetails JSON NULL,

  -- Contract Timeline
  startDate DATE,
  endDate DATE,
  acceptedAt TIMESTAMP NULL,
  completedAt TIMESTAMP NULL,

  -- Configuration & Blocks
  acceptanceCriteria ENUM('payment', 'signoff', 'creation'),
  blocks JSON NOT NULL,
  billingConfiguration JSON,

  -- Financial Information
  currency VARCHAR(3) DEFAULT 'USD',
  totalValue DECIMAL(15,2),
  priority ENUM('low', 'medium', 'high', 'critical'),

  -- Metadata
  tenantId VARCHAR NOT NULL,
  isLive BOOLEAN DEFAULT false,
  tags JSON,
  attachments JSON,
  createdBy VARCHAR,
  lastModifiedBy VARCHAR,
  version INTEGER DEFAULT 1,
  createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

  -- Foreign Keys
  FOREIGN KEY (templateId) REFERENCES templates(id),

  -- Indexes
  INDEX idx_tenant_status (tenantId, status),
  INDEX idx_contract_dates (startDate, endDate),
  INDEX idx_contract_parties (buyerId, sellerId, partnerId)
);
```

7.2 API Architecture

Template Management APIs

```
// Global Templates (Super Admin)

POST   /api/v1/admin/templates/global          // Create global template

GET    /api/v1/admin/templates/global          // List global templates

PUT    /api/v1/admin/templates/global/{id}     // Update global template

DELETE /api/v1/admin/templates/global/{id}     // Deactivate global template

GET    /api/v1/admin/templates/analytics       // Global template analytics


// Local Templates (Tenant)

POST   /api/v1/templates                       // Create local template

GET    /api/v1/templates                       // List tenant templates

PUT    /api/v1/templates/{id}                  // Update local template

DELETE /api/v1/templates/{id}                  // Deactivate local template

POST   /api/v1/templates/{id}/clone            // Clone template

POST   /api/v1/templates/copy-global/{id}      // Copy global template


// Template Operations

GET    /api/v1/templates/{id}/preview          // Generate template preview

GET    /api/v1/templates/marketplace           // Browse global templates

POST   /api/v1/templates/{id}/validate         // Validate template configuration
```

Contract Management APIs

```
// Contract CRUD
```

```
POST   /api/v1/contracts                       // Create contract
GET    /api/v1/contracts                       // List contracts
GET    /api/v1/contracts/{id}                  // Get contract details
PUT    /api/v1/contracts/{id}                  // Update contract
DELETE /api/v1/contracts/{id}                  // Cancel contract


// Contract Lifecycle
POST   /api/v1/contracts/{id}/send             // Send for acceptance
POST   /api/v1/contracts/{id}/accept           // Accept contract
POST   /api/v1/contracts/{id}/complete         // Mark completed
GET    /api/v1/contracts/{id}/timeline         // Get contract timeline
GET    /api/v1/contracts/{id}/events           // Get contract events
```

8. Menu Structure & Navigation

8.1 Super Admin Navigation

```
{
  id: 'implementation-toolkit',
  label: 'Implementation Toolkit',
  icon: 'Settings',
  hasSubmenu: true,
  submenuItems: [
    {
      id: 'global-templates',
      label: 'Global Templates',
      icon: 'FileText',
      path: '/implementation/global-templates'
    },
    {
      id: 'template-analytics',
      label: 'Template Analytics',
      icon: 'BarChart',
      path: '/implementation/analytics'
    },
    {
      id: 'platform-settings',
      label: 'Platform Settings',
      icon: 'Settings',
      path: '/implementation/settings'
    }
  ]
}
```

8.2 Tenant Navigation

```
// Settings Menu
{
  id: 'settings',
  label: 'Settings',
  icon: 'Settings',
  hasSubmenu: true,
  submenuItems: [
    {
      id: 'template-designer',
      label: 'Template Designer',
      icon: 'Edit',
      path: '/settings/template-designer'
    },
    {
      id: 'my-templates',
      label: 'My Templates',
      icon: 'FolderOpen',
      path: '/settings/my-templates'
    },
    {
      id: 'business-profile',
      label: 'Business Profile',
      icon: 'Building',
      path: '/settings/business-profile'
    }
  ]
},

// Contracts Menu
```

```
{
  id: 'contracts',
  label: 'Contracts',
  icon: 'FileText',
  path: '/contracts',
  hasSubmenu: true,
  submenuItems: [
    {
      id: 'contracts-list',
      label: 'All Contracts',
      icon: 'List',
      path: '/contracts'
    },
    {
      id: 'contracts-create',
      label: 'Create Contract',
      icon: 'FilePlus',
      path: '/contracts/create'
    },
    {
      id: 'contracts-drafts',
      label: 'Draft Contracts',
      icon: 'FileEdit',
      path: '/contracts/drafts'
    }
  ]
}
```

9. Integration Points & Extensibility

9.1 External System Integrations

- **Accounting Systems**: Automated invoice generation (QuickBooks, Xero, SAP)
- **CRM Platforms**: Contact and opportunity synchronization (Salesforce, HubSpot)
- **Equipment Management**: Asset tracking and maintenance (CMMS systems)
- **Payment Gateways**: Contract payments and billing (Stripe, PayPal, Square)
- **E-signature Platforms**: Legal acceptance (DocuSign, Adobe Sign)
- **Document Storage**: Contract document management (Google Drive, SharePoint)

9.2 Embeddable Widgets

```
// Website Integration Example
<script src="https://platform.contractnest.com/embed.js"></script>
<div data-contract-templates="industry:healthcare,type:service"></div>

// Configuration Options
{
  tenantId: "customer_tenant_id",
  templateFilters: {
    industry: "healthcare",
    contractType: "service",
    complexity: "simple"
  },
  theme: "light", // or "dark"
  whiteLabel: true,
  callbacks: {
    onTemplateSelect: (template) => {},
    onContractCreate: (contract) => {},
    onContractComplete: (contract) => {}
  }
}
```

9.3 API Ecosystem

- **RESTful APIs** for all core operations
- **Webhook System** for real-time event notifications
- **GraphQL Endpoint** for complex data relationships
- **Bulk Operations API** for enterprise integrations
- **Analytics API** for custom reporting and dashboards

10. Success Metrics & Analytics

10.1 Template Performance Metrics

- Template usage rates by industry and tenant
- Template completion rates and success ratios
- Time-to-contract reduction measurements
- Template customization frequency
- User satisfaction and rating scores

10.2 Business Impact Metrics

- Customer onboarding time reduction
- Contract compliance rate improvements
- Revenue per contract optimization
- Partner ecosystem growth rates
- Platform adoption and retention metrics

10.3 Technical Performance Metrics

- API response times and availability

- Template rendering performance
- Contract generation speed
- System scalability metrics
- Error rates and resolution times

11. Development Roadmap

Phase 1: Foundation (Months 1-2)

Template Management System

- ✅ Enhanced database schema implementation
- ✅ Basic template builder interface
- ✅ Template gallery and card system
- 🔄 Global vs local template management
- 🔄 Template version control system
- 🔄 My Templates interface

Phase 2: Contract Creation (Months 3-4)

Block System & Contract Builder

- Block validation framework
- Drag-and-drop contract builder
- Service commitment logic implementation
- Timeline preview and billing configuration
- Contract review and approval workflow

Phase 3: Advanced Features (Months 5-6)

Integration & Analytics

- External system integrations
- Embeddable widget system
- Advanced analytics dashboard
- Template marketplace optimization
- Performance monitoring and optimization

Phase 4: Enterprise Features (Months 7-8)

Scalability & Customization

- Multi-tenant optimizations
- Advanced permission systems

- Custom block development tools
- Enterprise reporting suite
- API ecosystem expansion

12. Risk Mitigation & Quality Assurance

12.1 Technical Risks

- **Template Complexity Management**: Implement progressive disclosure and guided workflows
- **Performance at Scale**: Implement caching strategies and database optimization
- **Data Consistency**: Use event sourcing for audit trails and data integrity
- **Integration Reliability**: Build resilient API connections with fallback mechanisms

12.2 Business Risks

- **User Adoption**: Strong onboarding process and comprehensive template library
- **Template Quality**: Moderation system for global templates and quality guidelines
- **Competitive Differentiation**: Focus on unique block system and industry specialization
- **Scalability Planning**: Architecture designed for multi-tenant scale from day one

12.3 Security & Compliance

- **Data Protection**: GDPR and SOC 2 compliance implementation
- **Access Control**: Role-based permissions with audit logging
- **Contract Security**: End-to-end encryption for sensitive contract data
- **Backup & Recovery**: Automated backup systems with point-in-time recovery

13. Conclusion

The Contract Builder Platform represents a comprehensive solution for modern contract management, combining the flexibility of a modular block system with the convenience of a template marketplace. The enhanced template architecture provides a scalable foundation that supports both platform-wide standardization and tenant-specific customization.

The platform's design prioritizes user experience while maintaining enterprise-grade security and performance requirements. The phased development approach ensures rapid time-to-market while building toward a fully-featured contract management ecosystem.

With its embeddable integration capabilities and comprehensive API system, the platform is positioned to become the backbone of contract management across diverse industries and business models.

Document Version: 2.0
Last Updated: Current Date
Next Review: Quarterly
Stakeholder Approval: Pending