# 🔌 Modular Plugin System

## ✅ Overview

The Modular Plugin System allows the SaaS platform to scale flexibly by letting developers and partners build, install, and manage additional feature modules as plugins. Brands can selectively enable these modules via their dashboard. Superadmins maintain control over approvals, access levels, usage limits, and billing.

## 🧩 Core Goals

* Let third-party developers or internal teams add new features without changing core code
* Allow brands to selectively enable or disable modules based on need or plan
* Provide a plug-and-play architecture for extending system capabilities
* Maintain centralized governance over module integrity, permissions, and API usage

## 🏗️ Architecture & Components

### 1. Plugin Registry

* Central database of all available plugins
* Fields:
  + plugin\_id, name, version, developer\_id
  + category (e.g., Marketing, Analytics, Engagement)
  + description, required\_permissions, dependencies, status
  + is\_approved, is\_default\_enabled, price\_model

### 2. Plugin Loader Engine

* Dynamically loads plugins into the dashboard and system
* Supports:
  + UI injection (dashboard widgets, feature tabs)
  + Backend APIs (routes and logic injection)
  + Scheduled jobs (CRON-compatible plugins)

### 3. Plugin Sandbox (Security Layer)

* Each plugin runs in an isolated sandbox environment
* No direct access to global variables, core services unless allowed via API proxy
* Rate limiting and logging enforced

### 4. Superadmin Plugin Panel

* Approve or deny plugin submissions
* Assign modules to brands or packages
* Set limits: API tokens, GPT usage, UI features
* Disable plugins across brands if security/bugs are detected

### 5. Developer Dashboard (External Partners)

* Submit plugin via manifest + zipped code repo
* See sandbox results
* Test plugins using test tenant
* Submit version upgrades
* Documentation guidelines and plugin rules available

## ⚙️ Plugin Manifest Example

{  
 "name": "Pinterest Poster",  
 "version": "1.0.2",  
 "category": "Marketing",  
 "main": "index.js",  
 "permissions": ["post\_generation", "media\_upload"],  
 "ui\_hooks": ["campaign\_editor", "post\_scheduler"],  
 "api\_hooks": ["/create-post", "/upload-media"],  
 "scheduled\_jobs": ["weekly\_summary\_report"],  
 "dependencies": ["image\_generator"],  
 "config\_schema": {  
 "apiKey": "string",  
 "boardId": "string"  
 }  
}

## 🔐 Security & Permissions

* Plugins can request only approved permissions
* Plugins with AI or API usage are metered and billable
* Superadmins can disable plugins immediately across tenants
* All plugin logs stored per brand

## 📦 Usage Flow

### For Superadmin:

1. Review plugin submission
2. Approve and categorize plugin
3. Enable plugin for specific brand(s) or tiers
4. Monitor plugin usage (metrics + logs)

### For Brands:

1. Visit Plugin Marketplace in dashboard
2. Enable/disable available plugins
3. Configure per plugin settings (API keys, toggle on/off, schedules)
4. Access plugin features inside core modules (if UI hooks defined)

## ✅ Example Plugins

* Pinterest Poster
* WhatsApp Reply Handler
* Review Booster
* TikTok Content Syncer
* UGC Heatmap Analyzer
* LinkedIn Auto-Scheduler

## 💡 Benefits

* Extend core SaaS features without bloat
* Build a developer ecosystem
* Offer exclusive features as upsells
* Easily launch limited-time experiments
* Support region- or industry-specific innovations

✅ Modular Plugin System documentation is now complete.