Context Engineering with Prompt Registry

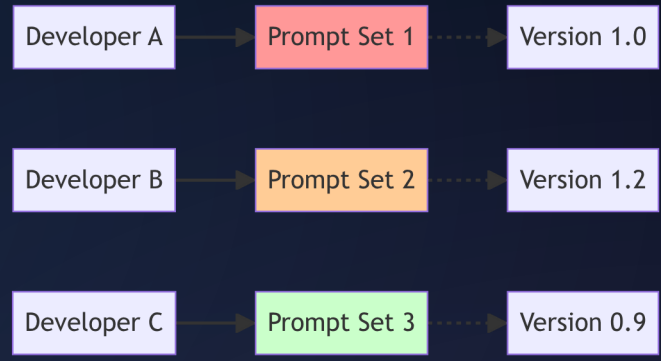
Unified prompt management for modern development teams →

The Problem

Context Chaos

- **Scattered prompts** across projects and workspaces
- **Inconsistent AI assistance** between team members
- **Manual maintenance** causing version drift
- **Collaboration friction** when sharing configurations

Teams spend hours managing prompts instead of building software



Fragmented prompt ecosystems

The Vision

Unified Prompt Management

- **Single source of truth** for prompt collections
- **Team collaboration** through Git and hubs
- **Automatic updates** with version tracking
- **Enterprise-grade** reliability and security

Prompt Registry transforms how teams discover, share, and manage GitHub Copilot prompts

Marketplace Tour

Visual Discovery

- **Search & filter** by name, tags, source
- **Multi-source support**: GitHub, GitLab, APM, Local
- **One-click installation** with progress tracking
- **Real-time updates** in the background

Browse collections like an app store

🔍 **Search Examples:**

- "react hooks" → 12 bundles
- "testing" → 8 bundles
- "enterprise" → 5 bundles

📦 **Bundle Types:**

- 🎨 Prompts
- 📋 Instructions
- 💬 Chat Modes
- 🤖 Agents

layout: section

Enterprise

Enterprise Features

Production-Grade Capabilities

- **Version management** with auto-update options
- **Lockfile tracking** via `prompt-registry.lock.json`
- **Multi-scope installations** (user/workspace/repository)
- **Repository-level sharing** through Git
- **PR-based updates** via tooling like Renovate

Treat prompt collections as software packages

Lockfile = Dependency Management

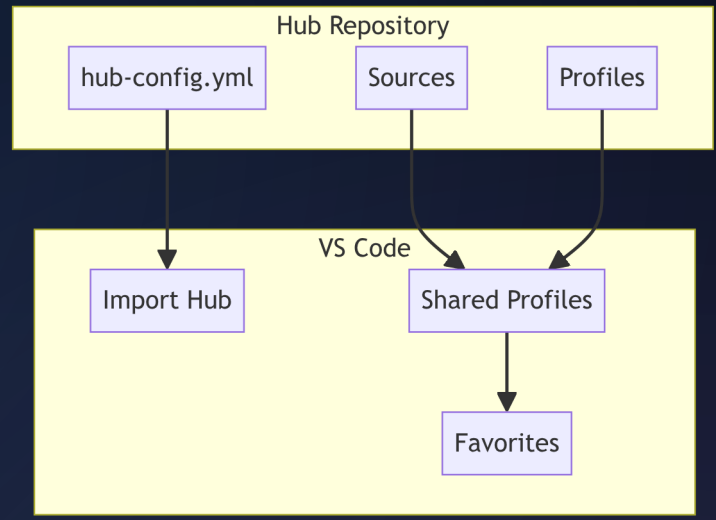
Treat collections like packages

- `prompt-registry.lock.json` is the **source of truth** in repositories
- Enables **reproducible** setups for a team
- Plays well with **automation** (update PRs)

Hubs & Profiles

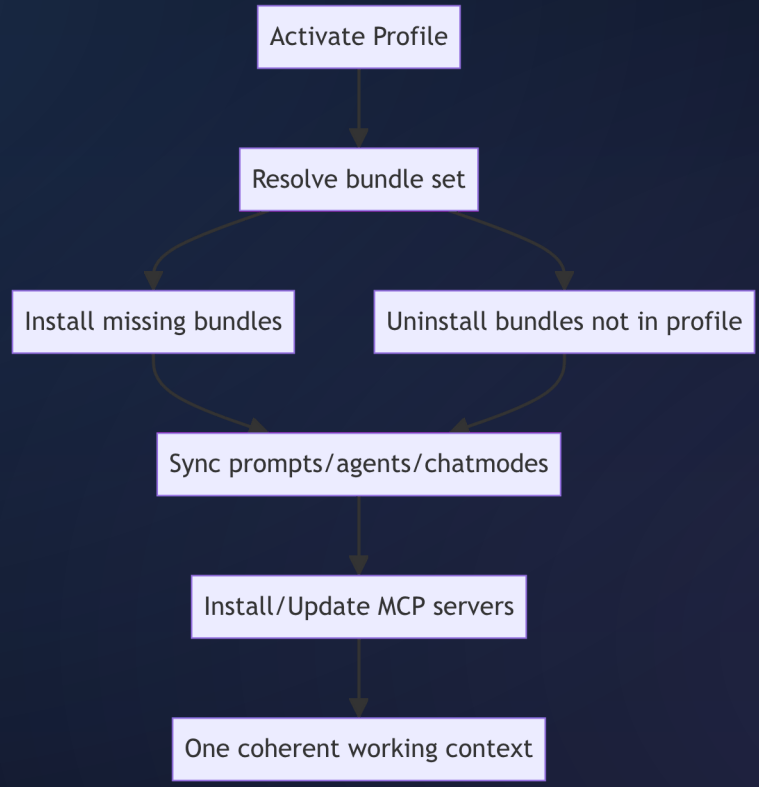
A curated catalog, not a wiki page

- A **Hub** centralizes:
  - curated **sources**
  - versioned **profiles**
  - an easy contribution surface (PRs)
- Profiles bundle everything for a role/task



Atomic Profile Activation

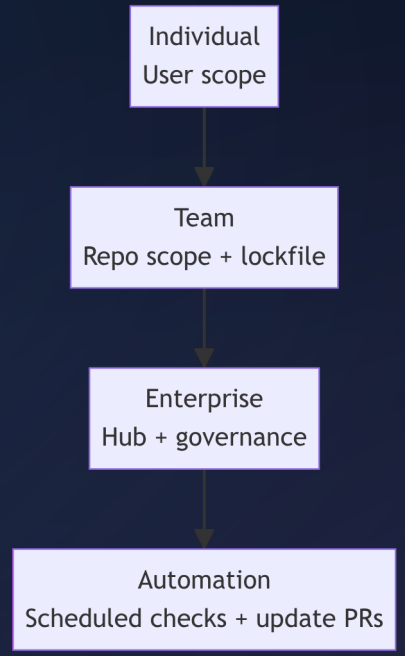
No more context clutter



Enterprise Rollout Pattern

Adopt incrementally

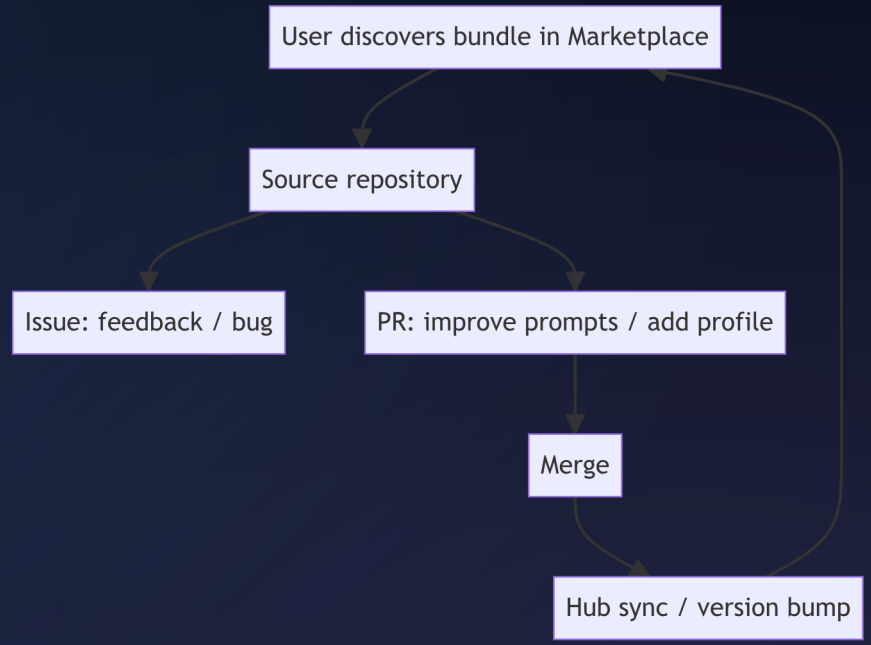
- **Individual**: use Marketplace + user scope
- **Team**: repository scope + lockfile
- **Enterprise**: hub governance + curated profiles + update automation



Collaboration Surface

Make contribution the default

- Every bundle **comes from a source repo** (GitHub/others)
- Contributors can:
  - open **issues** (feedback)
  - submit **PRs** (improvements)
  - propose **new profiles** (team use-cases)
- Hubs make the “where is this hosted?” question **easy to answer**



Reliability & Trust

Enterprise-grade by design

- **Versioned bundles**: know what you installed
- **Validation**: collections must include `deployment-manifest.yml`
- **Lockfile checksums**: detect drift / unexpected changes
- **Safe updates**: warn before overwriting local modifications

Sustainable & Inclusive

Built to last

- **Markdown-first** resources (easy to author/review)
- **Transparent governance** via hubs + PRs
- **Cross-platform** support (macOS, Linux, Windows)
- **Low barrier to entry**: start with the Marketplace, scale up later

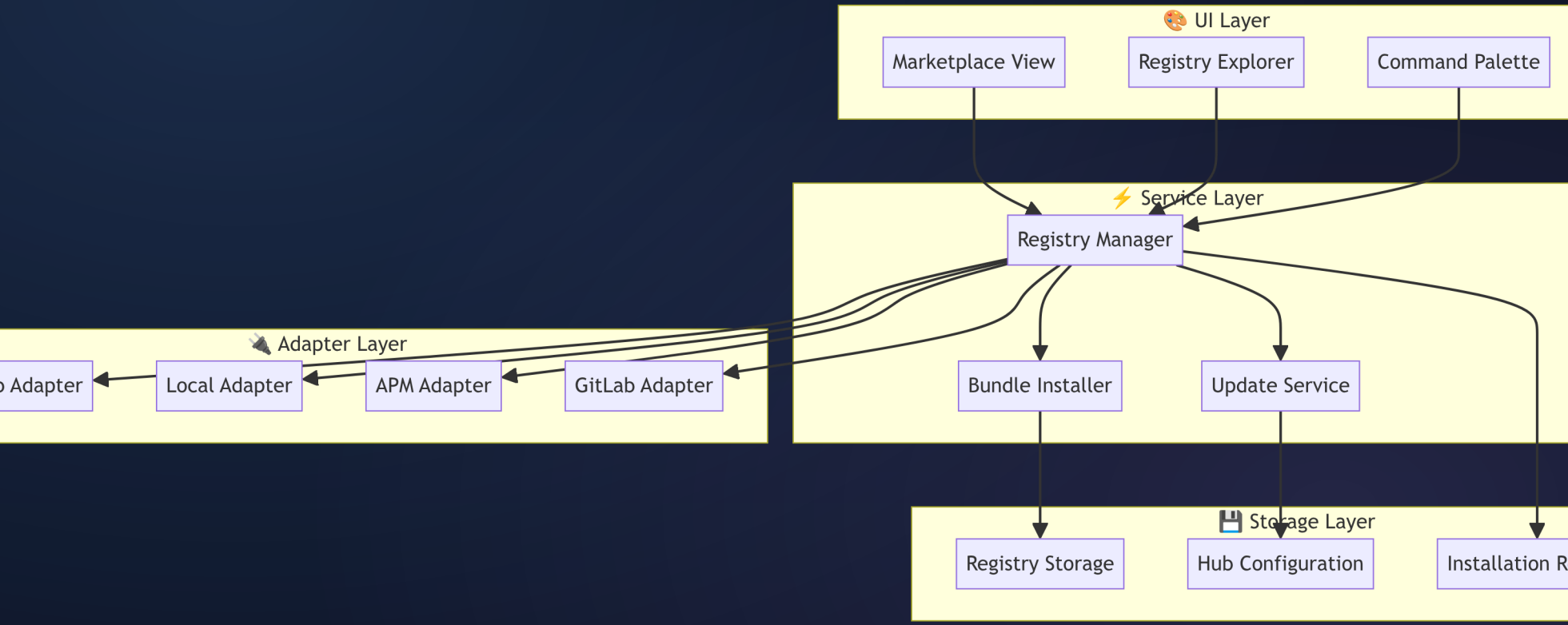
Simplified Install Pipeline

From click to usable context

```
Error: Parse error on line 5:
...-> E[Sync to scope\n(user or .github)]
----------------------^
Expecting 'SQE', 'DOUBLECIRCLEEND', 'PE', '-)', 'STADIUMEND', 'SUBROUTINEEND',
'PIPE', 'CYLINDEREND', 'DIAMOND_STOP', 'TAGEND', 'TRAPEND', 'INVTRAPEND',
'UNICODE_TEXT', 'TEXT', 'TAGSTART', got 'PS'
```

itecture Overview

t Works



Atomic Profiles

One-Click Context Switching

- **Role-based collections** (Frontend, Backend, DevOps)
- **Instant activation** with single click
- **Clean deactivation** - no context clutter
- **Team synchronization** across workspaces

Switch from frontend to backend context in seconds

🎨 **Frontend Profile**
React, TypeScript, CSS

⚙️ **Backend Profile**
Node.js, Python, Database

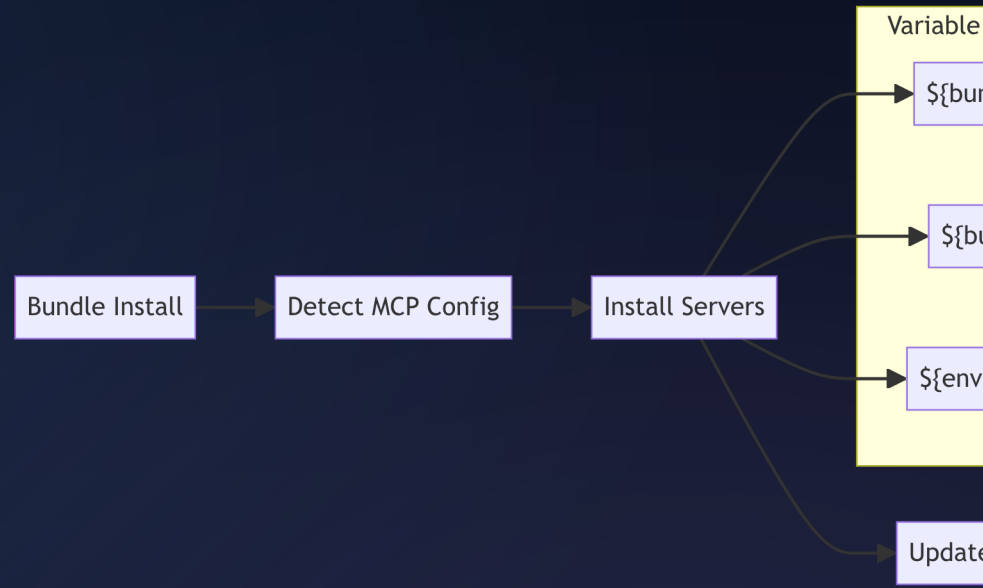
🔧 **DevOps Profile**
Docker, Kubernetes, CI/CD

MCP Integration

Self-Contained Tooling

- **Automatic server installation** with bundles
- **Variable substitution** for environments
- **Zero configuration assumptions** by authors
- **Complete working environments** out of the box

Bundles include required MCP servers automatically



Automatic tool setup

Getting Started

Easiest Adoption Path

1️⃣ **Install**
VS Code Extension

2️⃣ **Select Hub**
Or use defaults

3️⃣ **Browse**
Marketplace

4️⃣ **Install**
One-click

5️⃣ **Use**
Enhanced Copilot

From installation to productive use in under 2 minutes

Enterprise Setup

Production Configuration

🏢 Organization Hub

- Custom hub configuration
- Curated prompt collections
- Team-specific profiles
- Access control policies

🔧 Repository Integration

- Git-based configuration sharing
- Automated dependency management
- CI/CD integration
- Team onboarding workflows

Enterprise-grade features for teams of any size

Join the Community

Contribution Pathways

Create Collections

Share your prompt expertise

Contribute Code

Build the platform

Shape the Future

Context engineering movement

🚀 Start your context engineering journey today

github.com/AmadeusITGroup/prompt-registry

Thank You

Questions?

GitHub: github.com/AmadeusITGroup/prompt-registry

VS Code Marketplace: "Prompt Registry"

Community: Discussions and Issues welcome

Transform your AI development workflow

Unified prompt management for modern teams