

SFG NEXUS Orchestration System

Repository: sfg-app-portfolio

Purpose: Zero-drift orchestration hub for 40+ satellite applications

Status: Production

Last Updated: November 3, 2025

What Is This Repository?

This repository serves as the **single source of truth** for the entire SFG NEXUS ecosystem. It orchestrates 40+ satellite applications using a GitHub-first architecture where all instructions, documentation, and app metadata are version-controlled.

Key Concept: Zero-Drift Architecture

Traditional systems suffer from “configuration drift” where different components become out of sync over time. This repository eliminates drift by:

1. **Centralizing all instructions** in `/instructions` directory
2. **Version-controlling everything** with Git
3. **Automating distribution** via webhooks and MCP protocol
4. **Tracking compliance** through GitHub issues

When an instruction changes, all affected apps are automatically notified and update themselves. No manual synchronization needed.

Repository Structure

```
sfg-app-portfolio/
├── instructions/          # □ Operational procedures
│   ├── SATELLITE_APP_REGISTRATION.md # Autonomous registration prompt
│   └── README.md
│
├── docs/                  # □ Technical documentation
│   ├── ECOSYSTEM_OVERVIEW.md # High-level architecture
│   └── README.md
│
├── analysis/              # □ Business intelligence
│   └── README.md
│
├── templates/             # □ Standardized templates
│   ├── app-registration-template.json
│   ├── README-template.md
│   └── README.md
│
└── apps/                  # □ Application backups
    ├── app-001/ to app-050/ # Placeholder slots
    ├── chronoshift-pro/    # Time tracking (191 files)
    └── sfg-nexus/          # Central conductor
    └── ORCHESTRATION.md     # This file
```

How It Works

1. NEXUS (The Conductor)

SFG NEXUS is the central orchestration hub. It:

- Reads instructions from `/instructions/`
- Maintains persistent memory (PostgreSQL database)
- Connects to satellite apps via MCP protocol
- Coordinates workflows across multiple apps
- Provides Warren's Brain (ML decision support)

Technology: Next.js 14, TypeScript, PostgreSQL, AWS S3, Abacus.AI

2. Satellite Applications

40+ specialized apps perform specific business functions:

Core Apps: sfg-vertex, sfg-esp, sfg-sync

Support Apps: chronoshift-pro, heathcote-hub, company-wiki

Experimental Apps: sfg-brand-engine, sfgcomms-hub

3. Orchestration Flow

1. Warren updates instruction in `/instructions/`
2. GitHub webhook triggers notification workflow
3. NEXUS pulls latest instructions from GitHub
4. NEXUS distributes to satellites via MCP

-
5. Satellites implement instructions autonomously
 6. Satellites report completion via GitHub issue
-

Key Features

Persistent Memory

NEXUS never forgets conversations. It stores all conversations, messages, plans, decisions, and context in 8 PostgreSQL tables.

Autonomous Registration

Satellite apps register themselves using a standardized 9-step prompt (~30 minutes per app, fully automated).

MCP Integration

Model Context Protocol enables NEXUS to connect to satellite apps, call app-specific tools, access resources, and coordinate workflows.

Business Intelligence

The system tracks:

- **Portfolio Value:** £5.17M+
 - **Annual Savings:** £260k-320k from automation
 - **ROI:** 300%+ projected
 - **Efficiency:** 50% reduction in quote processing time
 - **Staff Replaced:** 7 roles eliminated
-

Getting Started

For Warren (Repository Owner)

1. Review the structure
2. Read the documentation (start with `docs/ECOSYSTEM_OVERVIEW.md`)
3. Configure repository privacy (make private, add collaborators)
4. Brief NEXUS on persistent memory implementation

For NEXUS (Conductor)

1. Load instructions from `/instructions/`
2. Implement persistent memory (8 database tables)
3. Register satellite apps (send autonomous prompt)
4. Set up MCP client (connect to first 5 apps)

For Satellite Apps

1. Receive registration prompt from NEXUS or Warren
 2. Execute 9-step registration process
 3. Set up webhook endpoint (`/api/github-webhook`)
 4. Implement MCP server for NEXUS communication
-

Security & Access Control

Repository Privacy

- **Status:** Private repository
- **Owner:** sfgaluminium1-spec (Warren) - Full access
- **Collaborators:** Manus AI agent (read), Machine user (write)

Authentication

- GitHub Personal Access Tokens with appropriate scopes
 - Webhook HMAC SHA-256 signatures
 - Environment variables for secrets (never committed)
-

Business Impact

Financial Metrics

- **Annual Savings:** £260k-320k (7 staff roles eliminated)
- **Portfolio Value:** £5.17M+
- **ROI:** 300%+ projected

Operational Metrics

- Warren works 10 hours/week (down from 24/7)
 - 50% reduction in quote processing time
 - 100% data accessibility
 - Zero rework (nothing gets lost)
-

Implementation Timeline

- **Week 1:** Foundation (repository setup, persistent memory, privacy config)
 - **Week 2:** First 10 Apps (registration, MCP integration, testing)
 - **Week 3-4:** Full Rollout (remaining 40 apps, gap analysis, full orchestration)
-

Support & Resources

- **Documentation:** [docs/ECOSYSTEM_OVERVIEW.md](#)
 - **Templates:** [templates/](#) directory
 - **Contact:** Warren Heathcote (warren@sfg-aluminium.co.uk)
 - **Repository:** <https://github.com/sfgaluminium1-spec/sfg-app-portfolio>
-

This repository transforms the SFG NEXUS ecosystem from a collection of disconnected apps into a unified, orchestrated system that never forgets, never drifts, and continuously improves.

Document Owner: SFG NEXUS Team

Last Updated: November 3, 2025

Version: 1.0