**Technical Justification for Central Administration Access (Development Environment)**

**Name:** Jessica Graf  
**Title:** SharePoint Architect and Developer  
**Scope:** SharePoint Server 2016/2019/SE – Multi-server Farm  
**Environment:** Development (Tier 1 – Non-Production)

**Objective**

**Technical Justification for Central Administration (CA) Access in Development**

This document provides the technical justification for granting Farm-level Central Administration (CA) access to the undersigned SharePoint Architect within the development environment. Direct access to CA is required to configure, validate, and troubleshoot farm-scoped components, including service applications, custom timer jobs, web application configurations, and farm-wide deployment processes.

**Benefits of CA Access in Development**

1. **Accurate Service Application Configuration**  
   Many SharePoint services (Search, BCS, Secure Store, User Profile Service) require provisioning and tuning via CA. Without CA, configuration is limited to default behaviors, which leads to instability or mismatched behavior across environments.
2. **Custom Solution Deployment & Validation**  
   Farm solutions (WSPs) and timer jobs often require full-trust deployment through CA. Farm-scoped feature activation and validation must be tested under controlled conditions that mirror production.
3. **Troubleshooting & Diagnostics**  
   Access to CA allows for immediate inspection of ULS logs, health reports, and service application status. Without this, triaging failures often depends on delayed coordination with administrators, increasing development cycle time and potentially masking critical issues until late-stage testing.
4. **Web Application and AAM Management**  
   CA is required to manage web applications, alternate access mappings, zone-level authentication methods, and service bindings — all critical for load-balanced and claims-based authentication environments.
5. **Farm Configuration Consistency**  
   Aligning the development environment with production requires the ability to configure managed paths, quotas, and global settings — all of which are centralized in CA.

**Risks of Denied Access**

Without CA access in development, the following issues are likely:

* **Configuration Drift** between Dev and Production, leading to deployment failures or untestable behaviors
* **Inability to Test Farm-Scoped Features**, including timer jobs, full-trust code, or managed service dependencies
* **Blocked DevOps Workflows**, particularly those involving scripted provisioning, health monitoring, and service proxy registration
* **Delayed Debug Cycles** due to restricted access to log configurations, service health dashboards, and error trace visibility
* **Deployment Failures** due to missing configurations that must be validated against the farm’s actual state

**Alignment with Best Practices**

This request aligns with Microsoft’s recommended DevOps and ALM (Application Lifecycle Management) practices, which emphasize parity between development and production farm topologies, as well as direct access for developers to validate infrastructure-dependent features.

Access will be limited to the development farm only, and all activities will adhere to governance and change control policies defined by the organization.

**Justification – Technical Use Cases**

**1. Service Application Infrastructure Configuration**

* Provision, configure, and bind Service Applications:
  + Search Service Application
  + User Profile Service Application (UPA)
  + Managed Metadata Service (MMS)
  + Secure Store Service
  + Business Connectivity Services (BCS)
* Manage Service Application Proxies and cross-service bindings
* Validate endpoint accessibility and custom service app registration

**2. Custom Timer Jobs & Job Definitions**

* Deploy and register SPJobDefinition-based timer jobs
* Monitor jobs via Timer Job Status and Definitions in CA
* Debug failures via job logs, schedules, and history

**3. Search Topology & Crawl Configuration**

* Manage content sources, crawl rules, and crawl schedules
* Configure managed properties and result sources
* Diagnose crawl errors and schema mismatches through CA

**4. Web Application Management**

* Create web applications with specific zones and SSL bindings
* Set authentication providers (Kerberos, Claims, NTLM)
* Configure AAMs and test multi-tenancy setups

**5. Farm Solution Deployment & Feature Registration**

* Deploy WSPs scoped at farm or web app level
* Confirm propagation across WFEs and app servers
* Restart app pools and force config updates through CA

**6. Logging and Diagnostic Monitoring**

* Set diagnostic logging levels per category
* Use ULS Viewer and CA logs for trace analysis
* Enable Health Analyzer rules for dev validation

**7. Secure Store & Claims Configuration**

* Manage Secure Store target applications for BCS
* Map external credentials and validate claims identity flows
* Test custom claims providers and federation configs

**8. Farm Health & Performance Monitoring**

* Review and configure Health Analyzer thresholds
* Track timer service responsiveness and app pool failures
* Validate uptime and performance during dev load testing

**Access Scope and Controls**

* Access limited strictly to **development** environment
* No access requested for QA, staging, or production
* All actions auditable via ULS logs, Health Analyzer, and Windows Event Logs
* Access mapped to named individual role (not generic admin group)

**Compliance Alignment & Governance**

**Principle of Least Privilege**

Access is scoped only to the dev environment and tied to role-based responsibility.

**Segregation of Duties**

Developers configure and test in Dev only. Admins maintain staging and prod.

**Change Management**

All CA activity is tied to change requests or internal ticketing processes.

**Auditability**

All CA configurations and diagnostics are captured by SharePoint and OS-level logs.

**Appendix A – Compliance References for Auditor Review**

**A.1 – CIS Microsoft SharePoint Benchmark (v1.1.0)**

| **Control ID** | **Description** | **Justification** |
| --- | --- | --- |
| 1.1 | Secure SharePoint services | CA access required to validate service provisioning and security |
| 3.3 | Restrict privileged account use | Dev-only access minimizes production risk |
| 5.1 | Review timer jobs | Enables inspection of default and custom timer jobs |
| 6.1 | Enable diagnostic logging | CA required to configure log levels by workload |
| 8.2 | Review application logs | Crawl logs, ULS logs require CA-level access for dev analysis |

**A.2 – Microsoft Security Compliance Toolkit (SCT)**

| **Policy Area** | **Justification** |
| --- | --- |
| Farm account isolation | Dev CA access helps validate service accounts and least privilege |
| Central Admin lockdown | Dev tier allows relaxed testing before production hardening |
| Claims identity & SSO | Testing claims providers and Secure Store requires CA control |

**A.3 – Internal IT Governance**

| **Policy** | **Relevance** |
| --- | --- |
| Development Access Policy | CA access requested only in dev tier |
| Change Management Policy | Configs logged through ticket/change tracking |
| DevSecOps Policy | Farm validation must occur before promoting to QA/prod |

**A.4 – Traceability and Audit Readiness**

| **Audit Category** | **Method** |
| --- | --- |
| Action Logs | ULS, Health Analyzer, Windows Event Logs |
| Access Logs | AD Group Membership and Server Security Logs |
| Deployment History | PowerShell logs and WSP deployment status |

**Appendix B – USGS & DOI IT Security Guidelines Compliance**

**B.1 – USGS IT Security Handbook Compliance**

| **USGS Policy Area** | **Relevance** |
| --- | --- |
| Section 400.3.4 – RBAC | Access is tied to a named developer role (Jessica Graf) |
| Section 600.2 – Config Mgmt | Required for testing baseline service configurations |
| Section 700.5 – Logging | CA actions are auditable via ULS and event logs |
| Section 800.1 – SDLC Security | CA access supports secure feature testing and review |

**B.2 – DOI Departmental Manual (375 DM 19)**

| **Control** | **Justification** |
| --- | --- |
| 19.6 C (1) | Developmental configurations must be tested prior to promotion |
| 19.6 D (4) | Developer access must be traceable and controlled |

**B.3 – FISMA & NIST 800-53 Alignment**

| **NIST Control** | **Control Name** | **Justification** |
| --- | --- | --- |
| AC-6 | Least Privilege | Dev-only access ensures minimum elevation |
| AU-2 / AU-12 | Auditable Events | CA activities are logged and attributable |
| CM-3 | Config Change Control | Supports configuration management testing in Dev |
| CM-5 | Access Restrictions | Only Architect has necessary access scope |
| SA-11 | Developer Testing | Farm-level testing is required prior to production use |

**B.4 – Summary of USGS Governance Alignment**

* No production access requested
* All actions traceable and accountable
* CA access tied to role and reviewed through change control
* Ensures secure SDLC and audit readiness per federal mandates