NCDOT GIS Architectural Standards

v3.4 (July 07, 2022)

System Architecture

- Platforms
  - Esri - verify the GIS Unit's current version plans before planning!
    - Except Roads and Highways:
      - ArcGIS Pro 2.9.3 (with latest patches)
      - ArcGIS Enterprise 10.9.1 (with latest patches). [1]
    - Roads and Highways for immediate deployment:
      - ArcGIS Desktop 10.8.1 (with latest patches)
      - ArcGIS Enterprise 10.8.1 (with latest patches)
    - Roads and Highways near future deployment
      - ArcGIS Desktop 10.8.1 (with latest patches) [2]
      - ArcGIS Enterprise 10.8.1 (with latest patches)

    NOTE 1: ArcGIS Desktop is a retiring technology. Do not plan any new work for ArcGIS Desktop.

    NOTE 2: As the Roads and Highways implementation in ArcGIS Pro nears completion by Esri, please plan to (re-)implement any Roads and Highways work to be compatible with ArcGIS Pro.
  - Web application and Web services (NCDIT-T hosted):
    - IIS (on premises) [1]
    - Azure Web App Service (cloud)
    - .NET Core

    NOTE 1: On premises deployment is a retiring technology. An exception is required for new work to be planned for on premises deployment.
  - Open Source Software
    - Requires prior approval
    - Latest stable release
  - Operating System
    - Desktops - Windows 10
    - Servers (on-premises) - Windows Server 2019 Standard
    - Servers (cloud) - TBD; consult with the NCDIT-T GIS Unit
  - Containers
    - Azure Kubernetes Service [1]

    NOTE 1: Infrastructure containers are not supported on premises.
- Applications must conform to the "NCDOT Technical Architecture Specifications"
- Network communication must be encrypted with a State (NCDIT) approved protocol.
- All non encrypted endpoints must be disabled. e.g. HTTP
- Data loading, editing, and usage functions must be implemented as web services.
- Initial data migration may use database endpoints providing the process is executed by database administrators who are not application administrators and are not end users of the system.
- Atomic business rules requiring more than one database transaction, must be implemented using a transaction manager.

- Data loading and editing functions must require a, securely authenticated, authorized user to allow access.
- GIS clients must support integration with OGC compliant services, *ArcGIS server* services, and *ArcGIS Online* services.
- GIS services must be compatible with Esri's *ArcGIS Desktop* and *ArcGIS Online*, and *ArcGIS API for JavaScript* viewers.
- All web applications and services must only be accessible using HTTPS.
- GIS data stores and services used for the collection and maintenance of data must be separated from data stores used for the publication and analysis of data.
    - Separating the two environments (editing and publication) helps to ensure availability, and reliability of each environment for its intended purpose.

Software - Architecture

- Software must be developed using n-tier service based design principles.
- File based artifacts must not be stored on a file server. For file based artifacts, use an approved document management technology such as SharePoint.
- Enterprise data must be stored in an approved enterprise data storage platform.
- Enterprise data must be created, updated, and used via web based services.
- Web based services must be implemented with the business rules to control who, what, how, and when data is accessed or updated.
- Client applications and system integrations must access enterprise data via these web services.
- Application functions must be organized into logical groups and implemented as potentially reusable web services.
- Whenever possible, look for logical groups of application functions that could make sense to be decoupled from the core application.
- Integrations with other applications must avoid "back-end" integrations that would prevent relocating one of the applications to another data center; e.g. moving from hosted servers to a cloud or moving from a private cloud to a public cloud, or vice versa.

Software - Installation

- Software written for the Windows operating system must be provided as a self-contained windows installer file. This includes providing an installer for all sample, intermediate, test, and final work products from all iterations, sprints, projects, etc.
    - NOTE: Microsoft Web Platform Installer is NOT available.
- Windows' installer files must be compatible with msiexec.exe, SCCM, and PowerShell DSC.
- Software installer must be named <ApplicationName>-<ApplicationVersion>.exe or <ApplicationName>-<ApplicationVersion>.msi.
- The <ApplicationName> must not contain spaces or special characters.
- Software installers must support a scriptable “silent install” process with full command line parameterization.
- Software installers must not require access to the internet to complete the installation process.
- The application and the install must not "know" anything about its target environment. NOTE: The following are acceptable restrictions on the supported Target Environment:
    - Supported operating systems and versions.
    - Supported databases and versions.
    - Supported versions of runtime environments and libraries, such as CLR, JVM, .NET, etc.
    - Supported authentication and authorization.
- The MSI must perform a complete installation; e.g.
    - The MSI must set secure folder permissions.
    - The MSI must not create user accounts, unless parameterized for the user to opt-in to create the account.

    - The MSI must use parameterized service account information where custom accounts are required.
    - The MSI must support managed and unmanaged service accounts.
    - For Web Applications, the MSI must
        - add the necessary application pool
        - configure all web handlers and security
    - The MSI must configure any required database connections
    - If there are other servers / services configured in the application, those endpoints must be parameterized. The installer must not use hard coded pre-configured service endpoints.
    - If there are many configuration parameters, they (except security restricted information such as passwords, tokens, keys, etc.) may be put in an XML or JSON file (external to the MSI) and make the path to the file a parameter to the MSI.
    - The MSI must create (if necessary) all required local directories and ensure the ACL is sufficient and is aligned with the principal of least privilege.
- Uninstall
    - The MSI must support silent uninstall.
    - A "standard" uninstall (e.g. when preparing for a reinstall or upgrade) should retain any needed configuration data, log files, history, etc.
    - A "clean" uninstall option may be provided to remove everything associated with the application including configuration data, log files, history, etc.
- Each delivery of software must have a unique version number.
- The installer must be considered part of the software; i.e. a change to the installer requires a new version number be assigned to the application. i.e. The version number attached to the install must be the same as the version number attached to the application. This version number must be the version number used in common communication about the version of the application plus a build number.
- Version numbers must:
    - be part of the installer file name e.g. <ApplicationName>-1.1.msi
    - be documented in the programs and files list by the installer
    - be identifiable in the application; e.g. On a “Help -> About” page

Cloud

- Cloud deployments must be fully automated in Azure DevOps

Software - Testing

- Automated unit test, system test, and regression test suites must be provided with custom software.
    - Graphical user interface testing may be excluded from automated testing, provided:
        - The system is deployed as an n-tier application.
        - User interface test scripts (AKA test plans) must be provided to ensure complete manual testing of the user interface.
    - There must be sufficient business logic testing (including fail cases) to ensure all business logic is correct, complete, resilient to improper inputs and ensures transactional operations meet ACID, BASE, or equivalent requirements.
        - If automated user interface tests are not provided, manually executed user interface test scripts (AKA test plans) must be provided.
- User acceptance criteria must include all aspects of the product; including: packaging, installation, security, performance, administrative, and functional testing.
- User acceptance testing must be performed on the customer's permanent installation (QC stage servers and representative PC).