

|  |
| --- |
|  |
| Development Considerations & Lessons Learned |
|  |

|  |
| --- |
| September 28, 2018 |

Table of Contents

[1 Introduction 2](#_Toc526145737)

[2 Application/Service Development Considerations 2](#_Toc526145738)

[2.1 E-Enterprise Portal Collaboration 2](#_Toc526145739)

[2.2 Design and Build for Sharing/Scalability 2](#_Toc526145740)

[2.2.1 Reference data used to populate dropdown lists or other user interface controls. 3](#_Toc526145741)

[2.2.2 Additional help and guidance data such as contact information, resources, etc. 3](#_Toc526145742)

[2.2.3 Business logic components 4](#_Toc526145743)

[2.2.4 Dynamic user interface (UI) based on service configuration 5](#_Toc526145744)

[2.3 Demonstrated Need and User-centric Design 6](#_Toc526145745)

[2.3.1 Demonstrated Need 6](#_Toc526145746)

[2.3.2 User-centric Design 7](#_Toc526145747)

[2.4 Comprehensive Service Documentation 7](#_Toc526145748)

[2.5 Onboarding/Provisioning Resources 7](#_Toc526145749)

[2.6 Versioning 7](#_Toc526145750)

[2.7 Pagination/Performance Considerations 8](#_Toc526145751)

[2.8 Transport Security 8](#_Toc526145752)

[2.9 Cross-Origin Resource Sharing (CORS) 8](#_Toc526145753)

[2.10 Hosting Considerations 8](#_Toc526145754)

[2.11 Required Libraries Considerations 8](#_Toc526145755)

[2.12 Ownership/Stewardship/Sustainability 9](#_Toc526145756)

[2.13 Compliance with .gov Standards 9](#_Toc526145757)

[2.14 Technology Agnostic 9](#_Toc526145758)

[2.15 Traceability and Respectful Use of Data 9](#_Toc526145759)

[2.16 Track Usage and Adoption 10](#_Toc526145760)

[2.17 Widgets Not Meeting Criteria 10](#_Toc526145761)

[3 Retrospective from previous co-development/collaboration efforts 10](#_Toc526145762)

[3.1 Incorporating Existing Functionality 10](#_Toc526145763)

[3.2 On General Observations 11](#_Toc526145764)

[3.3 On Policy/Guidance 12](#_Toc526145765)

# Introduction

The [E-Enterprise Portal](https://e-enterprise.gov) is an important component of the [E-Enterprise for the Environment](http://e-enterprisefortheenvironment.net/) model. The E-Enterprise Portal provides a means by which users can customize the types of information presented to them, find and select tools and other resources, and conduct transactions with environmental co-regulators. The E-Enterprise collaborative leadership is actively exploring means for various partners and collaborators to contribute functionality to the Portal.

This document uses the lessons learned from various E-Enterprise Portal collaboration efforts, such as the

* Be *Well* Informed Guide,
* Assistance Gateway,
* U.S. EPA Construction General Permit and Multi-Sector General Permit Search, and
* Federal Regulation Finder implementations

The document describes the considerations that should be made during development of applications and services intended for integration with the E-Enterprise Portal or shared across the Exchange Network. For additional information, please visit the E-Enterprise Portal to access the [E-Enterprise Portal User Guide](https://e-enterprise.gov/sites/default/files/E-Enterprise%20Portal%20User%20Guide.docx) and other resources.

# Application/Service Development Considerations

A key principle of this E-Enterprise product is to build and deploy Information Technology (IT) services that are built once and reused many times. This facilitates the flexibility and scalability of solutions within a single organization and establishes the capability to truly share services between partners.

Developers should incorporate the following system development practices to facilitate and integrate into the E-Enterprise Platform, as well as, the services and capabilities shared between E-Enterprise partners.

## E-Enterprise Portal Collaboration

Teams considering the development of E-Enterprise shared services should consider the extent to which their services will integrate with the E-Enterprise Portal. The development guidelines throughout this document apply to all E-Enterprise service development, but special consideration must be taken when building services intended for exposure through the E-Enterprise Portal. All service design should include early collaboration with the E-Enterprise Portal Coordinator in order to determine the appropriate level of E-Enterprise Portal integration, and to establish appropriate coordination with E-Enterprise governance and alignment with the most current E-Enterprise Portal standards and to ensure adoption of a consistent look (or at least the ability to modify the visual components). E-Enterprise Portal collaboration will also establish the process and governance for determining “production readiness” for the introduction of a new service into the E-Enterprise Platform.

## Design and Build for Sharing/Scalability

The fundamental lesson learned across E-Enterprise Portal collaboration efforts is that applications and services must be designed and built with the goal of sharing or distributing the capabilities among partners if we are going to truly embrace - and enable - a culture of "build once reuse many times". Service-oriented architecture (SOA) approaches [e.g., microservices architecture and Application Program Interface (API)-first development strategy] enable capabilities to be shared effectively and efficiently among partners.

Developers should build applications using services that can be exposed individually, support a specific business goal, and keep any business logic decoupled from the user interface components. When developing something, it is recommended to update configuration and not alter or hard code. This will facilitate future changes and facilitate the usage of the service by other Exchange Network and E-Enterprise partners.

Additionally, developers should consider multiple uses within the area of application (e.g., multiple permits or media).

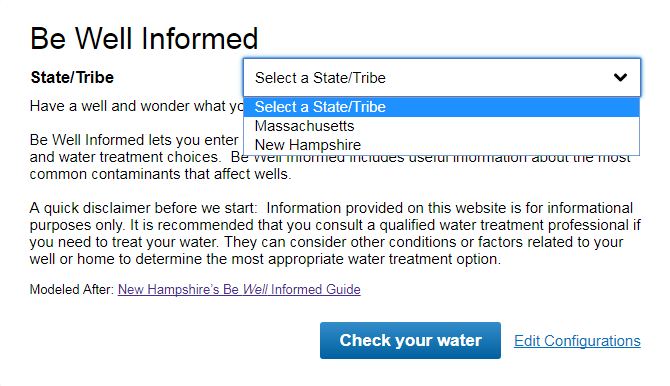
**Guiding Principle:** When building services to support your systems, consider what future partners might desire or need to change in order to implement as a shared service. To the extent possible, engage other organizations early in your design process and/or make those aspects of your services configurable. Maximize use of reusable component services and other shared services and use open source coding, when possible, for easy reuse by others or use middleware to decouple applications from specific software solutions. Developers should also post all technical documentation for others to adopt (e.g., XML Schema, Shared Services Strategy Service Level Commitment, Onboarding Kit).

Developers/service providers are also encouraged to consult the E-Enterprise Shared Services Strategy.

The following are examples of the types of configurable elements that should be provided, wherever possible, during the provisioning of shared services:

### Reference data used to populate dropdown lists or other user interface controls.

Provision lists in a configuration resource (i.e., table) or associate them with a standard service definition to expose city or town name. The preference is to use information as close to the source as possible to enable access to the most up-to-date information (e.g., use of an API to surface information from a registry for Tribal names instead of importing a list of tribal names into a static table.) Be *Well* Informed uses a configuration table to expose the State, Tribe or Territory-specific information and allows for additional entities to be added as the application is adopted by E-Enterprise partners.



### Additional help and guidance data such as contact information, resources, etc.

Coding should be configurable. For example, Be *Well* Informed presents information and resources specific to that State, Tribe or Territory. The partner configures the information within their XML file during their onboarding and provisioning process.

Figure 1: State/Tribe/Territory Resources Page

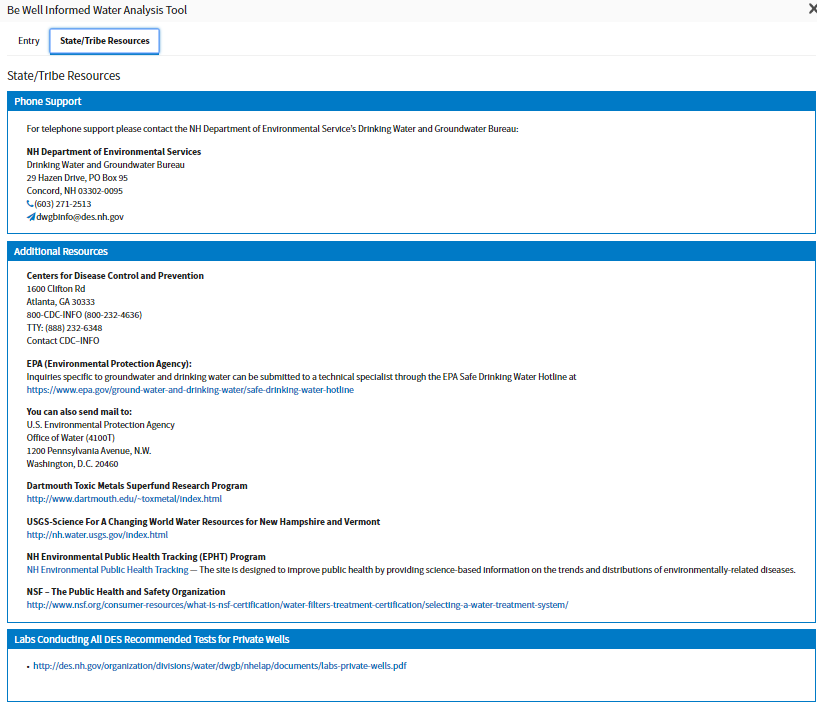
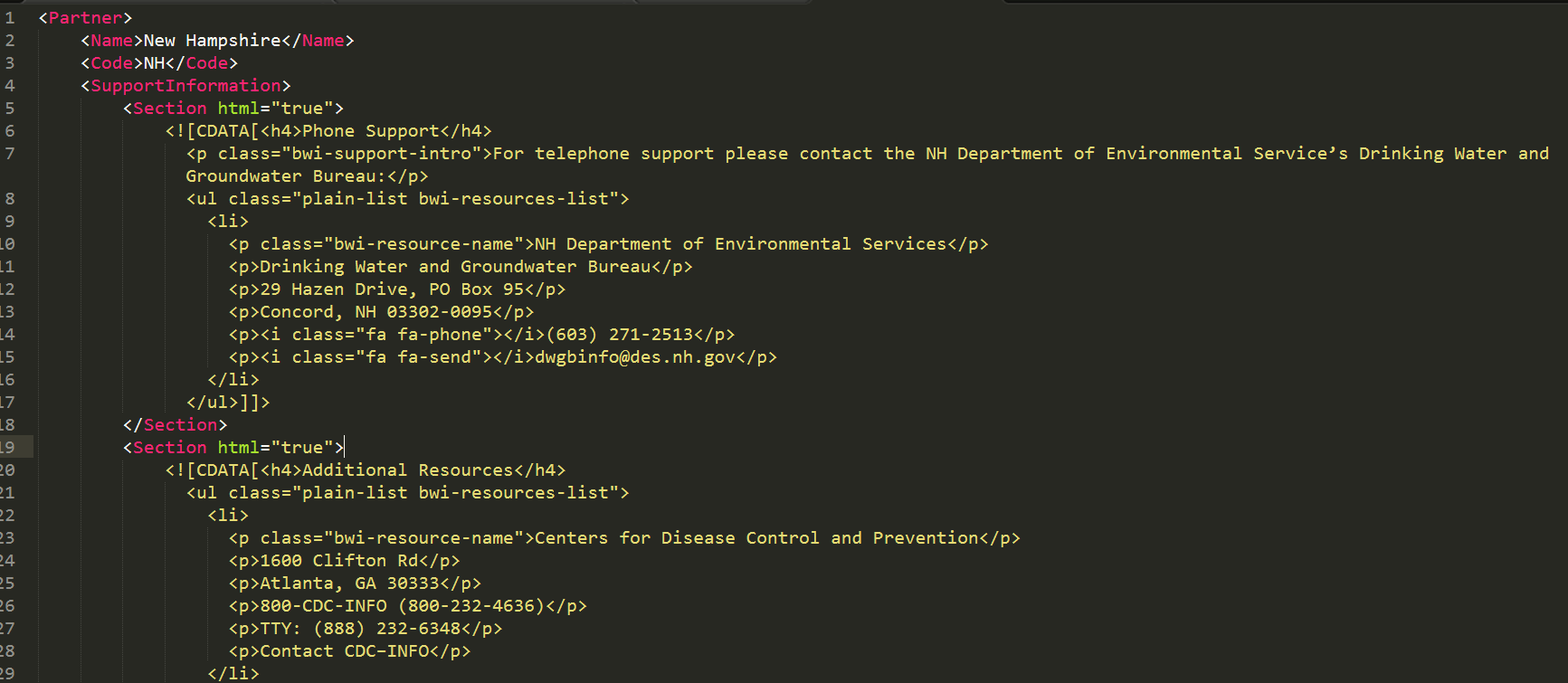


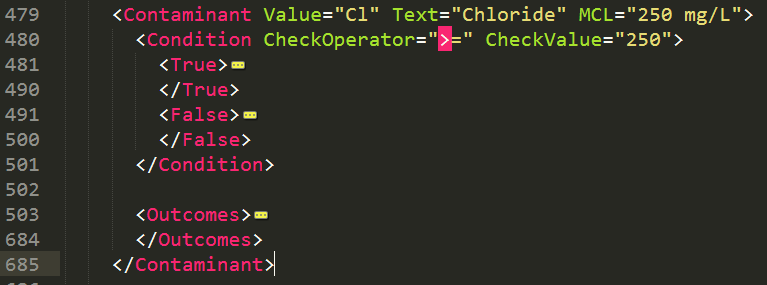
Figure 2: State/Tribe/Territory Resources Configuration XML



### Business logic components

Allow for configurable system components when business logic elements may change over time or vary when provisioned for another partner. For example, Be *Well* Informed functionality allows for partners to configure maximum contaminant levels (MCLs). Contaminant levels may be configured for a specific partner by modifying the XML/flowcharts.xml. This supports the business case where MCLs may differ between partners.

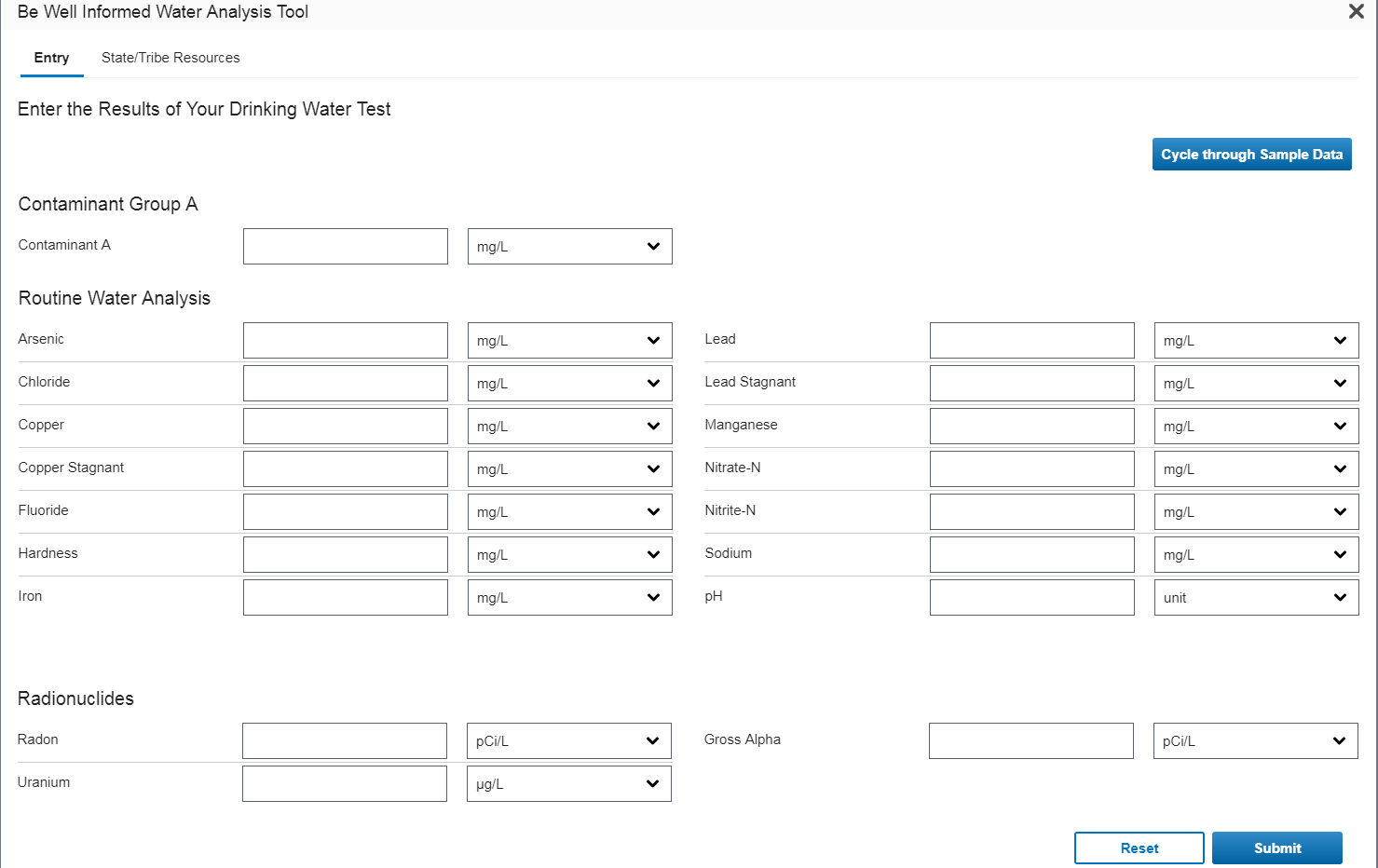
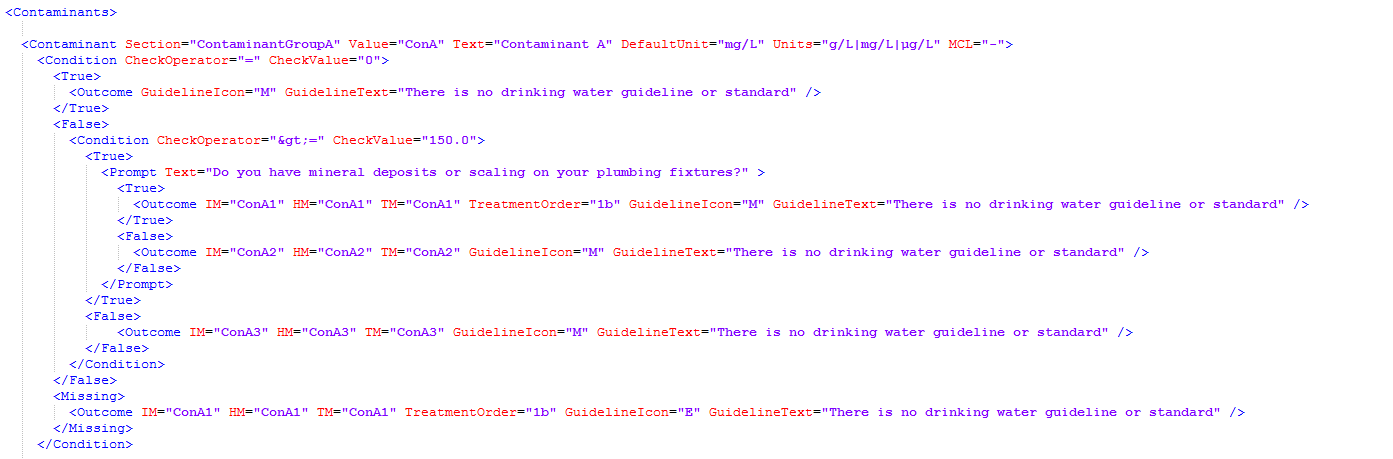
Figure 3: 2.1.3 Be Well Informed Component MCL Configuration XML



### Dynamic user interface (UI) based on service configuration

Design and build the UI to render dynamically based on the configuration made during onboarding/provisioning. This will reduce constraints on partner's use or branding due to limitations built into, or resulting from, UI components of your systems. For example, Be *Well* Informed dynamically renders the data entry form. When a partner's XML files are provisioned, the data entry form will display the contaminants and associated MCLs specified. This supports the business case where the contaminants of interests may differ between partners.

Figure 4: Be Well Informed contaminant input

** **

## Demonstrated Need and User-centric Design

### Demonstrated Need

Consider if other States, Tribes, Territories or EPA regions and programs may have similar business needs; a product should meet the needs of multiple partners. Determine whether the proposed service is duplicative of an existing service and if other entities are developing or expect to develop or use similar services and leverage those as appropriate. Engage potential partner organizations early to identify and prioritize needs and ease of use, including determining what configuration may be desired. It is helpful to obtain feedback from potential State partners that may surface the metrics used (e.g., use of the tool by individuals on a state-by-state basis) and may enable them to quantify the impact to that State's customers and their return on investment.

This is an opportunity to determine if multiple partners can leverage resources to build a shared service as well as the long term needs for service sustainability and scalability.

### User-centric Design

Employ a user-centric design. Use plain language to provide information to general or public users, and to the extent possible for services that have a more targeted audience. Engage potential users and consumers for information early in the development process. This will help to determine their needs and subsequently to ensure that any services developed meet their needs in an intuitive way. Continue to engage potential users through testing to ensure their ease-of-use needs are met.

Partners should describe their user audiences and select them from a configured list (if possible) and make available in E-Enterprise. This will facilitate future plans for recommending audience-specific widgets, enable future user discussions with the appropriate audiences, and help identify any audience-specific design aspects that should be considered.

*The E-Enterprise Portal is intended to provide a standard user experience for multiple user communities. At the highest level these include regulators, the regulated community, and the public. Whenever possible, services and interfaces intended for the E-Enterprise Portal should design services to be scalable and consider designing for the user with the lowest level of institutional knowledge and understanding, which is likely the public user.*

## Comprehensive Service Documentation

Services should be fully documented. Comprehensive service documentation will benefit service providers and consumers within the Exchange Network and E-Enterprise. Service documentation will make system operations and maintenance more efficient for the service provider, and facilitate the adoption and usage of services by partners. The OpenAPI Specification (formerly known as Swagger RESTful API Documentation Specification) is a powerful and effective way to document and represent RESTful APIs. The E-Enterprise Shared Services Strategy also provides a template for E-Enterprise Portal adoption. This service documentation is included in the following types of information:

* Service endpoints
* Query parameters, with sample values shown as examples
* Response data array fields that are available, with example or descriptions, if appropriate

## Onboarding/Provisioning Resources

Comprehensive onboarding/provisioning infrastructure, documentation, and support are necessary to ensure efficient onboarding and adoption of shared services by partners. These materials will reduce the amount of time and resources required to onboard partners by the service provider. Examples of onboarding/provisioning materials include:

* Step-by-step instructions to provision a service from the service provider
* Sample configuration materials for the configurable elements
* Frequently Asked Questions (FAQs)

## Versioning

As software evolves, service providers need to version Exchange Network and E-Enterprise service APIs. As APIs evolve due to routine maintenance or to support changes within their organization business process, consumers of the services require protection from changes. Versioning allows the service provider to alter behavior, communicate changes, and protect consumers from changes. Multiple versioning strategies exist; the most common is based on the request URL or the request headers. The method should be well documented.

## Pagination/Performance Considerations

When the service consumes a large dataset(s), employ a strategy or method for pagination or consumption of the data without deteriorating performance. Methods could include incremental loading or progressive rending. Metadata should enable service consumers to calculate the volume of data and determine whether or not fetching subsequent data is required.

## Transport Security

Any service APIs must use and require HTTPS encryption (using TLS/SSL). HTTPS encryption provides many benefits including security, privacy, stronger trust, and compatibility.

## Cross-Origin Resource Sharing (CORS)

Service providers must support cross-origin requests. The concept of cross-origin resource sharing is that resources from both within and outside the original resource's domain may be accessed and displayed to the user. For more information, the following link provides more on cross origin resource sharing: <https://en.wikipedia.org/wiki/Cross-origin_resource_sharing>.

## Hosting Considerations

Plan your service strategy to include hosting considerations. A shared services model that distributes services between partners introduces complexities. The service provider's infrastructure must be able to onboard and/or provision new partner service consumers, and plan for the increased load and server usage. Cloud hosting strategies provide a modern hosting infrastructure that can help provide scalability. It is possible a service may not be available, such as a critical service that functions through shutdowns is not guaranteed.

Service Level Commitment or Agreement (SLC/SLA) is standard documentation for all shared services that includes a service description with hosting platform, service availability, point of contact (POC) or service manager, sponsoring organization, user support resources and timeline for future availability. SLC/SLA provides the detail needed for partners to effectively evaluate and implement a shared service. A template is provided in the E-Enterprise Shared Services Strategy.

*The E-Enterprise Portal is an option for the hosting of services and corresponding user interfaces. The Portal is intended to provide lightweight user interfaces that expose shared services and their corresponding data. It is not intended to provide a storage space or repository for large volumes of data or user interfaces with significant embedded business logic.*

## Required Libraries Considerations

If your services include dependencies on required libraries, offer access to minified versions of the library over HTTPS. Minification is a process for removing unnecessary characters from source without impacting functionality. Minified libraries can help improve service performance.

In addition, it is recommended to use a version naming convention to prevent issues related to client caching of the script(s). This can prevent clients from running out-of-date script versions due to client-side caching.

## Ownership, Stewardship, and Sustainability

The strategy for sustainability should be detailed of your service offering and your service level commitment. This includes managing the resources and/or information published, provisioning funding resources to support operations & maintenance, as well as fielding and addressing reported issues. Contact with the E-Enterprise Portal team points of contact is also recommended. Additional guidance on service level commitment may be found in Section 2.9. Hosting Considerations.

## Compliance with .gov Standards

Widgets should be accessible according to Section 508. The more recent Web Content Accessibility Guidelines (WCAG) 2.0 guidelines can be found in the links provided below:

* <https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-ict-refresh>
* <https://www.w3.org/WAI/intro/wcag>

Ensure that proposed apps/tools comply with product review guidelines and web best practices established by the Office of Web Communications in the Office of Public Affairs (OPA).

For shared services or widgets specifically targeted for display within the E-Enterprise Portal:

* The E-Enterprise for the Environment logo will be placed on the site
* A disclaimer icon must be placed on all links to non-federal websites
* Site content will be subject to EPA's records management and archiving requirements and procedures for EPA information and data
* The site will be hosted by EPA’s National Computer Center at Research Triangle Park (RTP), NC or on a federally approved cloud. As the site will be placed on an EPA server/federally-approved cloud, security audits will not be necessary.
* The site will comply with EPA and federal security requirements.
* The site will include a privacy and security statement on its pages, as required by OMB policies.
* EPA will follow EPA’s records management and archiving requirements and procedures for EPA information and data.

## Technology Agnostic

Technology agnostic approaches support portability and align with the overarching guideline to design and build for sharing, and include:

* APIs to enable legacy applications to interoperate with applications and operating environments developed under the enterprise architecture
* Use middleware to decouple applications from specific software solutions
* Utilize Service Oriented Architectures (SOA)

## Traceability and Respectful Use of Data

* Identify data sources
* Use data as close to the source as possible to minimize variability
* Security needs must be identified and developed at the data level, not the application level. Shared services can and should enforce/enhance non-exposure of secure information and provide additional safeguards.
* Employing data caching strategies in order to improve performance and user experience may be employed. When doing so, the type of data should be carefully considered and in cases where the user should be made aware of possible data latency, the service description and related documentation should provide the user with the appropriate information regarding possible data latency.

## Track Usage and Adoption

Build mechanisms into your services to track both usage and adoption.

Utilization indicators provide information on the extent to which a service is being used. Adoption indicators provide information on the extent to which partners find the service useful for their needs. Performance indicators provide more details into individual service performance and how well the service tracks to its service level commitment (SLC). Customer feedback indicators provide information if a service is working for users and helps identify potential enhancements or modifications that may be required in the future.

## Widgets Not Meeting Criteria

Some widgets do not appear to meet all of the criteria above. All widgets that were in the Showcase Portal as of June 2017 were grandfathered.  This includes: Be Well Informed, Assistance Gateway, EPA’s 2017 Construction General Permit Search, Progress Tracker, To Do List, Favorite Links, Interactive Maps, My Air, My Environment Mapper, My Facility Manager, and Trending Air. The E-Enterprise Portal Coordinator and/or governance team will work with the tool’s owners to make improvements during future releases or enhancements.

# Retrospective from previous co-development/collaboration efforts

The text below reflects the experiences gathered from previous co-development/collaboration efforts such as Be *Well* Informed and Assistance Gateway.

## Incorporating Existing Functionality

* The Be *Well* Informed Guide demonstrated that relevant business logic can be extracted from an existing application and provided as web services for consumption by partner applications. Whether or not this was strictly possible was not necessarily in question.
* NHDES's Be *Well* Informed Guide was constructed in a way that re-engineers the existing functionality into component shared services with a reasonable amount of effort. It also demonstrated that it may be worth doing if the benefit to partners is sufficient.
* In the case of the Be *Well* Informed Guide, the existing application was re-engineered by the E-Enterprise Portal development team as described in the Technical Details contained in Appendix A.
  + If the initial application was services-based, then the Portal integration effort would have required only a UI development to call the underlying services and display back the returned results.
  + The initial application web services were not architected in a way that they could be configured or decoupled from the UI without development. The initial application contained some functionality and coding embedded on the front-end of UI, or web services that could be called to perform business logic. The logic embedded in the UI had to be de-coupled and incorporated into the web services. The E-Enterprise Portal application UI is now independent of the business logic and accesses the business logic through web service calls. Development to support integration of an existing application with a services oriented architecture or functionality into the E-Enterprise Portal may require only a custom UI component to call the source application's underlying services.

Note that a service adapter would likely be required to complete future integrations of front end and backend components.

## On General Observations

* The E-Enterprise Portal can support multiple co-development/collaboration models. The Be *Well* Informed Guide represents a model where custom Portal UI is created on top of services. This model is already demonstrated in the Portal through components such as the Trending Air component (sourced by [Village Green](http://villagegreen.airnowtech.org/) services) and the My Air component (sourced by [AirNow.gov](http://airnow.gov/) services). The E-Enterprise Portal also supports a plugin co-development/collaboration model with My Facility Manager. This component launches a self-contained widget, the Facility Widget, which is a JavaScript plugin that communicates cross-domain with .NET web services. These two co-development/collaboration models in current use within the Portal can be summarized as:
  + Self-contained (UI and logic) reusable applications/widgets - Example: My Facility Manager
  + Service-oriented Architecture (SOA)/Shared Services/Micro-services - Examples: Be *Well* Informed Guide, Missouri local government resource discovery, My Air/Village Green, Assistance Gateway
* Self-contained widgets provide the most autonomy for the feature developer.
* Shared services and Micro-services provide significant flexibility in how consumers of the services interact with the services and present the results.
* Further exploring whether joint governance would like for E-Enterprise to provide more self-contained widgets, base components to build from within something like GitHub, or a more formal application development framework should be explored. Building flexibility and multiple interoperability channels into a platform is not trivial. Assessing the desired need and expected adoption or usage of such tools will be an important factor in determining to what extent to explore these options more fully.
* Embedding business logic into configurable files, such as with Be *Well* Informed, is an excellent model for partner application developers to follow, if providing functionality within E-Enterprise.
* Partners will be constrained by any number of factors in their internal development efforts. Constraints could come in the form of limited expertise and standards imposed by centralized information technology (IT) management. To the extent possible, partners should consider architectures that employ self-contained widgets or are based on the development of shareable services when developing applications with functionality that they would like to see promoted in E-Enterprise or feel may provide benefit to other E-Enterprise partners. Following such practices increases the chances that E-Enterprise joint governance could agree to incorporate the functionality into E-Enterprise, or expose the underlying services, with a low level of partnership effort.
* During the initial conversations with stakeholders, the upfront and early exchange of the latest functional and technical documentation can help ensure a high level of clarity to better understand the design and requirements. It can also reduce the likelihood of re-work or issues later during development.
* Use of self-documenting Application Programming Interfaces (APIs), such as Swagger, can allow for easier and more robust integration between systems.
* When partners incorporate a version control process for the services that they make available for consumption, they allow for consumers to evaluate and provide feedback on enhancements to those services. This protects the existing functionality with minimal interruptions. This concept also allows partners to implement changes to their services at their discretion while giving flexibility to consumers when choosing to upgrade or maintain existing functionality.
* Partners who have access to the developer/author of the functionality are encouraged to provide contact information to the E-Enterprise Portal Coordinator, eportal [at] epa.gov, to communicate with that developer/author should any questions arise.
* Partners are encouraged to adhere to or provide API versioning for services to allow for flexibility in development and testing.
* Prospective partners are encouraged to edit files carefully and utilize an XML validator when modifying XML files or other files necessary for integration.
* A self-service configuration tool would likely help facilitate supporting future services in an efficient manner.
* There were challenges in loading and making accessible the Missouri's Gateway for Community Assistance source code and onboarding information into EPA's Reusable Component Services (RCS). RCS places restrictions on the maximum size of uploaded files. GitHub was offered as an alternative, but was ultimately rejected. Consideration should be made as to:
  + Where shared component source code/onboarding documentation will be housed for access by future partners
  + How subsequent updates to component source code/onboarding documentation will be made by the sharing partner
  + What minimum information should be included in each instance of component source code/onboarding documentation
* Concurrent enhancement development can be used to improve the user experience by standardizing the format and metadata associated with widgets identified, such as microservices approaches throughout the guidelines for the E-Enterprise Portal Assistance Gateway, Be *Well* Informed, and Federal Regulation Finder widgets. Additional guidance and specifications related to widget metadata and services will be developed including standardization around widget source information, disclaimer presentation, and informational modals.
* It is imperative to carefully introduce minimal content on the workbench when developing services that require new widgets or changes to existing widgets. An example would be to reduce the burden on performance to load the workbench page. This will ensure the stability of the workbench page and minimize any impacts to performance.

## On Policy/Guidance

* E-Enterprise governance may be required to develop a process for determining which features or applications desired by partners are included in the E-Enterprise Portal. This was a valuable lesson learned during initial conversations around E-Enterprise Be *Well* Informed. Without E-Enterprise governance processes, some informal vetting of the application and its approaches were required to be performed by EPA before inclusion. Well-defined governance processes for how functionality is to be included and reviewed are necessary for a successful complex partnership.
* In a case like Be *Well* Informed where the UI is recreated and "powered by" partner supplied services, joint testing and approval procedures may also be necessary prior to release of functionality within the E-Enterprise Portal.
* Future versions of this document will include production readiness and user acceptance test (UAT) procedures.
* Any guidance should align with the E-Enterprise Shared Services Strategy.