

## 5.2 Service Discovery and Governance

This section includes the following:

- [5.2.1 Service Discovery Functions](#)

Service discovery and governance allow service developers to leverage the rich service metadata for better discovery. Service discovery and governance help to accomplish the following:

**Promote service reuse:** The use of well defined service metadata promotes better discovery and reuse of services during design and run time. Service metadata includes information about service interactions and dependencies. It also includes a classification scheme for organizing services based on business objectives, domain, and usage. It links services to all the supporting artifacts in the specification and provides a placeholder for conformance statements. This enables better reuse across the enterprise and eliminates redundancy.

**Establish service policies:** Service policies help establish constraints on the service specifications and mandate an approach. Policies can be specified around governance, access control and other design and runtime constraints.

**Provide governance:** This includes predefined templates, workflows, and governance policies for governing the service lifecycle as well as an approval and review process for service specifications and the ability to promote services through the stages of the service lifecycle.

**Enable better discovery:** Complex search offers a natural and user-friendly way to find services by progressively refining search results using a variety of criteria including attributes, artifacts, classification, usage scenarios, and dependencies. This includes runtime contract discovery, a powerful query mechanism that allows either the service orchestrator or a program to find the services that best fit the requirements of a given process. This increases both runtime and design time flexibility by enabling selection of services based on computable metadata.

### 5.2.1 Service Discovery Functions

Service discovery functions include the ability to:

- Identify the service endpoint for analysis
- Identify the service directory endpoint for analysis
- Extract the service interface
- Annotate the service interface providing undiscovered features or behaviors
- Manage lifecycle, governance and versioning of the service interfaces

The requirements listed above are derived from the following use cases:

*Electronic Health Records:* The caBIG® Clinical Information Suite project is developing service specifications and lacks the infrastructure to govern these services. Vendors and external implementations are expected to leverage the caBIG® Clinical Information Suite service specifications and there is currently no infrastructure that allows easy discovery and consumption of this information.

*CBIIT Projects:* CBIIT has adopted SOA. Service lifecycle management and governance are industry best practices for all organizations adopting SOA. Better service discovery and reuse improves productivity, avoids redundancy and makes it easier for the CBIIT enterprise architecture governance team to manage NCI's enterprise services portfolio.

*Life Sciences:* Service discovery based on a rich metadata and semantics of the underlying data play a critical role in developing research pipelines. Research pipelines are developed by connecting data and analytical services together to achieve a research objective.

*Other national initiatives:* All EHR vendors and national initiatives rely on a services paradigm for integration and interoperability. A standardized services metamodel makes it easier for participating organizations to discover and reuse services.

*caGRID 2.0 Platform:* The caGRID 2.0 Platform provides a runtime registry for service discovery. This service registry relies on a small subset of information for discovery. The semantic infrastructure provides a mechanism to leverage rich service and artifact metadata to extend this capability.