Module 5 – NIEM Technical Architecture

**Slide 1 – Welcome Slide**

Welcome to Module 5: NIEM Model Technical Architecture

**Slide 2 – Module Overview**

Module Overview

In this module we will discuss NIEM’s Technical Architecture. The topics that we will cover are:

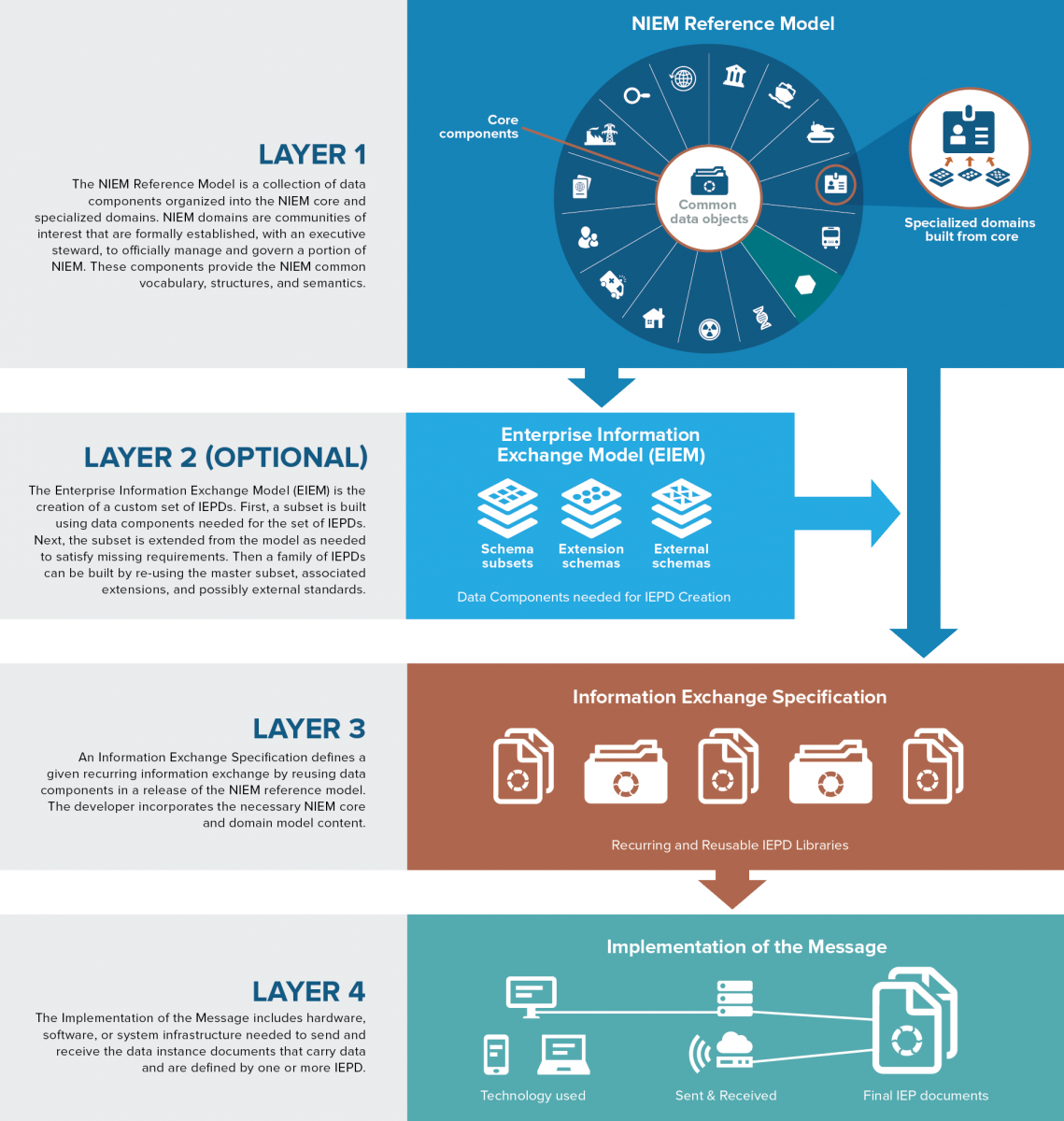
* Technical Architecture Overview
* NIEM’s Architectural Concepts and
* Additional Resources

**Slide 3 – Technical Architecture Overview**

Technical Architecture Overview

NIEM’s technical architecture is a set of reusable Extensible Markup Language (XML) Schema Documents (XSD). XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. An XSD describes an XML document—it has a set of constraints to define content structure imposed by XML itself. These schema documents contain commonly used data components and are grouped into abstraction layers. An abstraction layer hides implementation details to facilitate platform independence and eventual interoperability. Each abstraction layer reuses and extends data components from previous layers. The graphic below provides a description of NIEM’s abstraction layers.

* NIEM Domain Abstraction Layer: Used to provide mission-based/domain specific layer of data objects that specializes base objects from the NIEM Core and structures namespaces.
* NIEM Core Abstraction Layers: Contains commonly used definitions for generic objects like person, organization, and activity that are used across domains.
* External Standards Abstraction Layer: Contains definitions for objects used in standards defined external to NIEM.
* Support Abstraction Layer: Provides the underlying standardized structure for NIEM. Each of the data objects in the other abstraction layers reuse the basic data objects in this layer.



**Slide 4 – Introduction to NIEM Architectural Concepts**

Introduction to NIEM Architectural Concepts

NIEM is comprised of four main architecture concepts. These four concepts are Namespaces, Code Lists, Association Types, and Augmentation Points.

* Namespace: Used to avoid element name repetition. Each reference schema in NIEM is defined individually by a namespace. The namespace criteria must also be followed when developing exchange-specific schema documents.
* Code Lists: Most often used to limit possible values for an element. If code lists are reusable by many exchanges, they have the potential to be integrated into NIEM Core or a NIEM domain.
* Association Types: Link together related objects in an exchange. The use of association types prevents the need to redefine objects. Association types are implemented using references.
* Augmentation Point: An "augmentation point" is an abstract element that acts as a substitution group head, for which additional elements of any type may be substituted.

**Slide 5 – Additional Resources**

For an overview of additional NIEM resources related to this module, please visit the following links:

* INSERT <https://www.niem.gov/techhub/niem-model/technical-architecture> LINK
* INSERT <https://www.niem.gov/techhub> LINK
* INSERT <https://www.niem.gov/about-niem/niem-governance> LINK