HEALTH IT
STANDARDS TESTING INFRASTRUCTURE

NIST HIT Testing Infrastructure Overview

Robert Snelick

National Institute of Standards and Technology

October 27th 2009

Contact: rsnelick@nist.gov

The American Recovery and Reinvestment Act (ARRA)

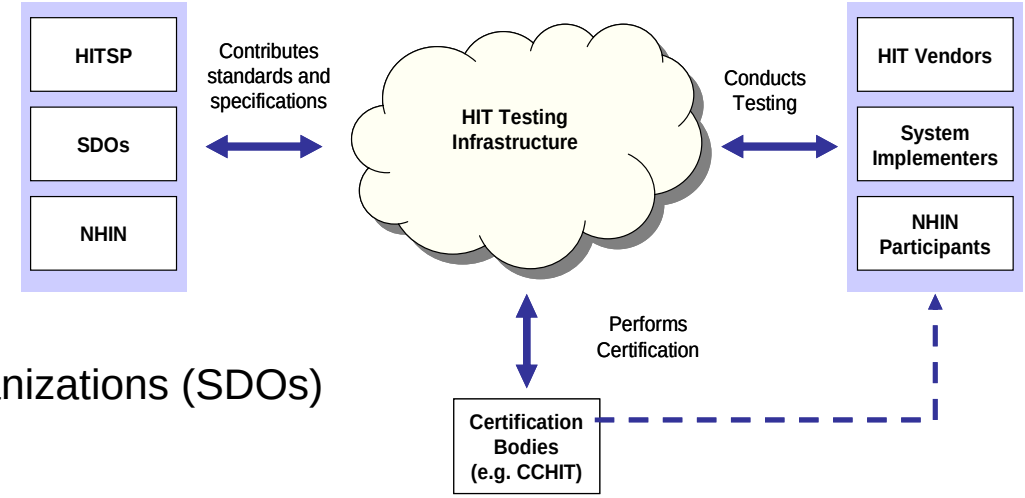
- Calls for NIST to lead the development of a health IT testing infrastructure
- The legislation calls for the Office of the National Coordinator (ONC) to work with the National Institute of Standards and Technology (NIST) to:
  - Ensure health IT standards are complete and robust
  - Establish a health IT standards testing infrastructure that supports industry consensus standards development and provides robust conformance and interoperability testing capabilities
  - Deploy those technologies to promote interoperable health IT adoption

NIST Role and Activities Overview

- Currently leads significant conformance and interoperability test development efforts
- Has well established collaborations with standard and tool development groups
- Neutrality is highly recognized among competing healthcare agendas
- NIST doesn't develop standards, perform certification, or conduct operational testing (in the Healthcare domain)
  - but provide the tools needed to test existing standards and/or specifications; tools that can be used by recognized certification bodies; tools that can be leveraged by the stakeholder organizations to integrate into their testing processes
- Recognized through ARRA responsibilities and funding to lead critical testing infrastructure piece of the national health IT efforts
- Already well positioned to take on this role given prior work and the plans we had in place
- ARRA funding will accelerate this effort significantly

NIST will lead the development of the health IT testing infrastructure

- Will provide a scalable, flexible, automated environment for current and future testing needs
- In collaboration with health IT stakeholders, NIST will work to harmonize the efforts of healthcare standards test development and delivery to meet the demands for conformance and interoperability within the healthcare domain.
- NIST will leverage existing tools and work with health IT stakeholders including:
  - CCHIT
  - HITSP
  - NHIN
  - Vendors
  - Implementers
  - Standards Development Organizations (SDOs)
  - Industry Consortia



In developing the health IT testing infrastructure, NIST will address several key objectives

- Provide a variety of testing services to health IT stakeholders that are implementing standards-based health IT solution
- Support a broad range of purposes such as conformance and interoperability testing
- Support numerous healthcare messaging and document data exchange standards including HL7 V2, HL7 V3, DICOM, NCPDP SCRIPT and CDA documents
- Provide a component based user interface and set of services so that the health IT stakeholders may use the infrastructure in different ways
  - *"A Framework for Building Test Systems—an SOA Approach"*
- Enable user customization so that the health IT stakeholders can establish their own individual test instances
- Support different delivery mechanisms including APIs, desktop applications, web applications, and web services
- Provide a feedback loop to enhance the healthcare standards
- Use and integrated existing tools
- Maintain strong collaboration with other tooling efforts (e.g., IHE Gazelle)
- Deliver and roll out tools and resources incrementally

Current Target Standards

- **CDA:** Clinical Document Architecture: Is an XML-based markup standard intended to specify the encoding, structure, and semantics of clinical documents for exchange.
- **HL7 V2:** Health Level Seven Version 2: Standard for moving clinical and administrative information between healthcare applications.
- **HL7 V3:** Health Level Seven Version 3: Standard for moving clinical and administrative information between healthcare applications.
- **NCPDP SCRIPT:** National Council for Prescription Drug Plans: Standard for ePrescribing.
- **DICOM:** Digital Imaging and Communication in Medicine: Standard for handling, storing, printing, and transmitting information in medical imaging systems.

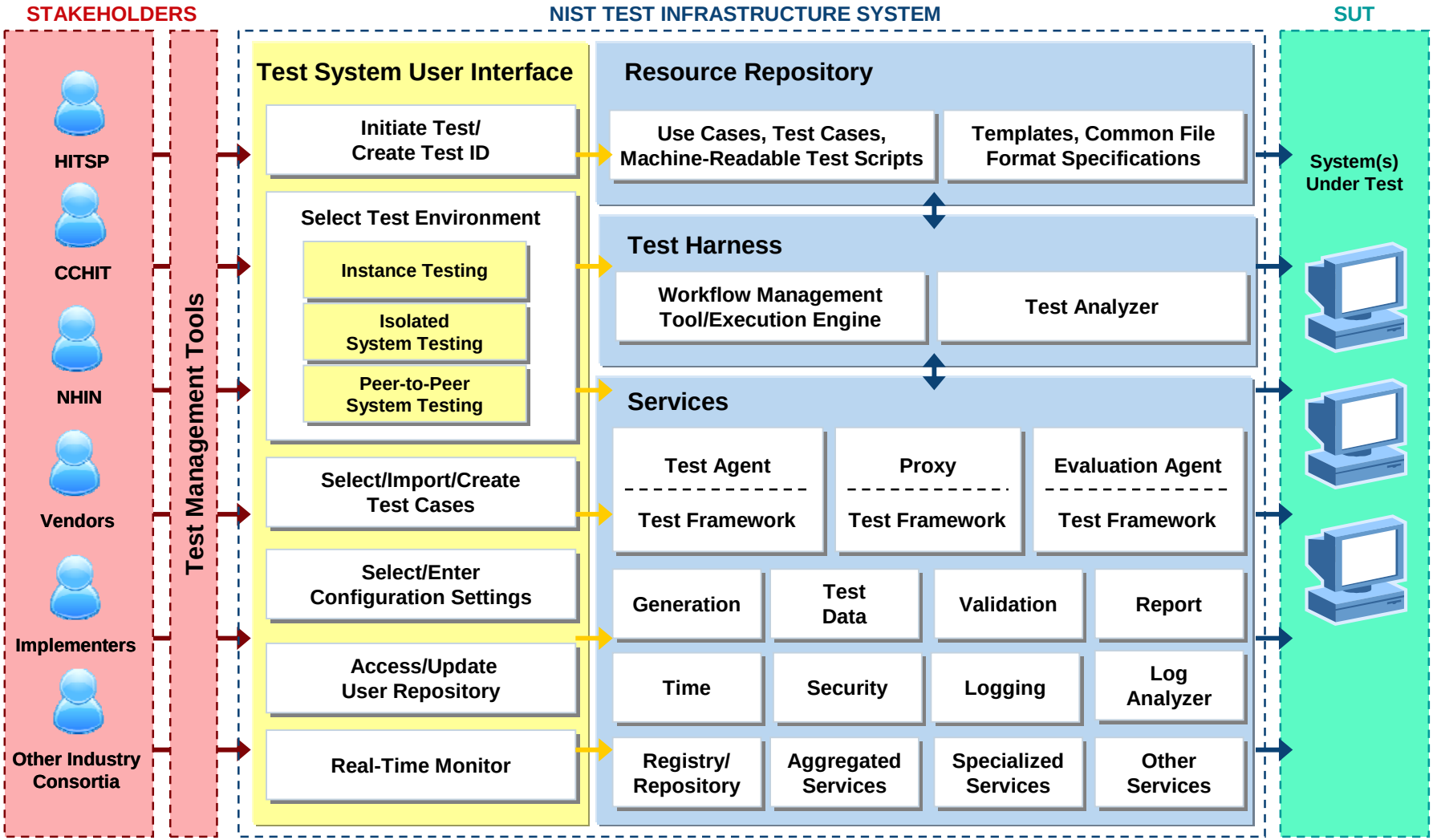
NIST Tool Set Overview I

- XDS-Cross Enterprise Document Sharing
  - Reference Implementation of the XDS document registry and document repository
  - Public registry server (Client Software)
  - XDS Toolkit (Downloadable toolkit for testing your own servers)
- HL7 V2
  - HL7 V2 Testing Toolkit
    - Testing Framework-supports test agents (simulators) and communication (v3 also)
    - Message Generation, Message Validation, Profile Validation
  - IHE PIX and PDQ Test Agents (Simulators)
  - IHE Pre-connectathon test tool (PIX and PDQ)
  - Generic HL7 V2 initiator and responder application
- HL7 V3
  - IHE PIX and PDQ Test Agents (Simulators)
  - IHE Pre-connectathon test tool (PIX and PDQ)

NIST Tool Set Overview II

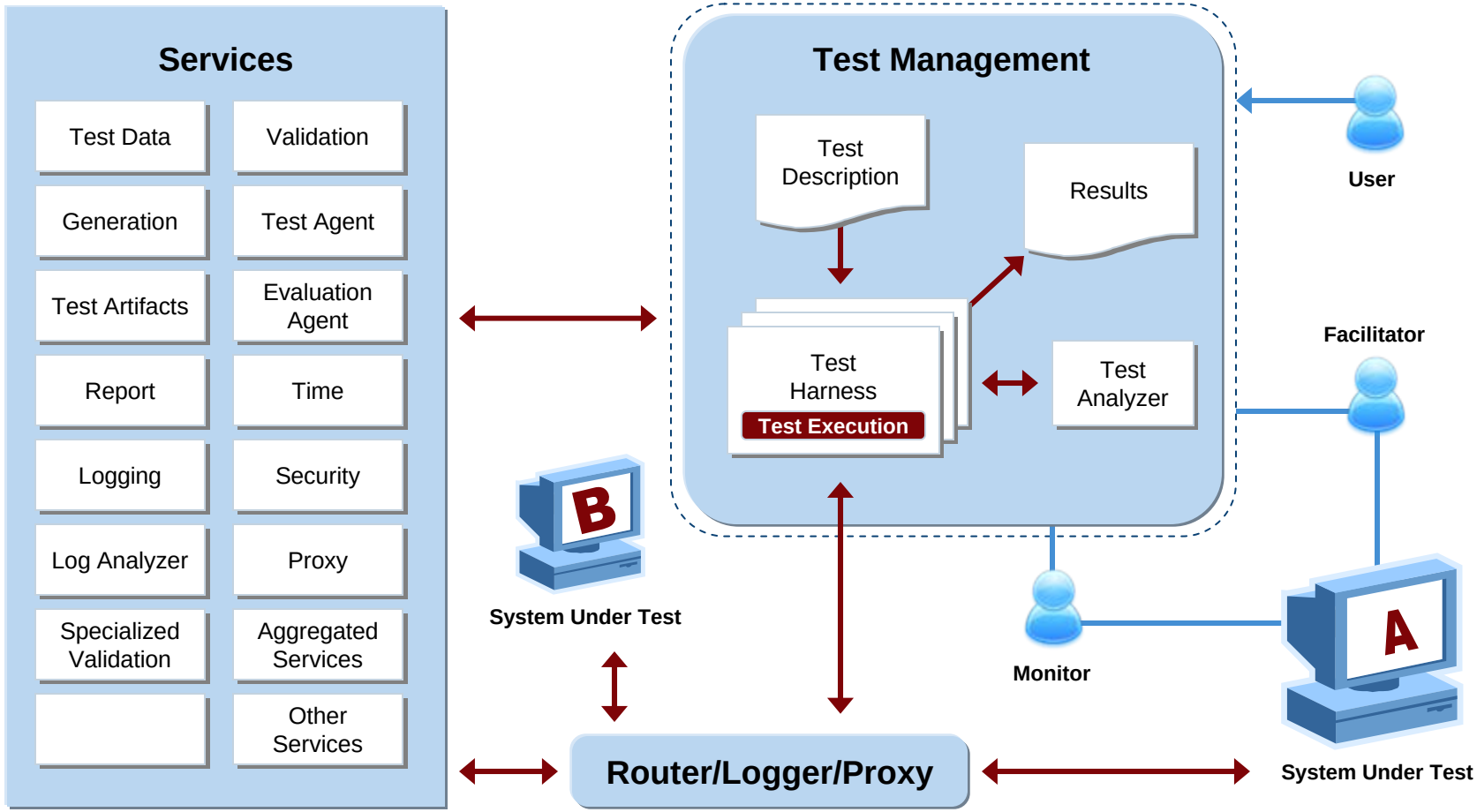
- CDA-Clinical Document Architecture
  - Validation of CDA/CCD documents based on HITSP, NHIN, and IHE specifications
  - Includes schematron rules and 28 document types
- Medical Device Communication
  - ICS Generator-build ISO/IEEE 11073 Compliant Medical Device Specializations
  - NIST HL7 V2 static message validation
    - validation of IHE-PCD integration profiles and associated HL7 V2 messages
  - Rosetta Terminology Mapping Management System (RTMMS)
    - repository of medical device nomenclature and associated co-constraints (e.g., unit of measurement)
  - XML Schema (Proposed Standard)
    - Electronic version of ISO/IEEE 11073 domain information model (DIM)

Conceptual View of Testing Infrastructure



NIST HIT Testing Infrastructure Design

A Framework for Building Test Systems—an SOA Approach



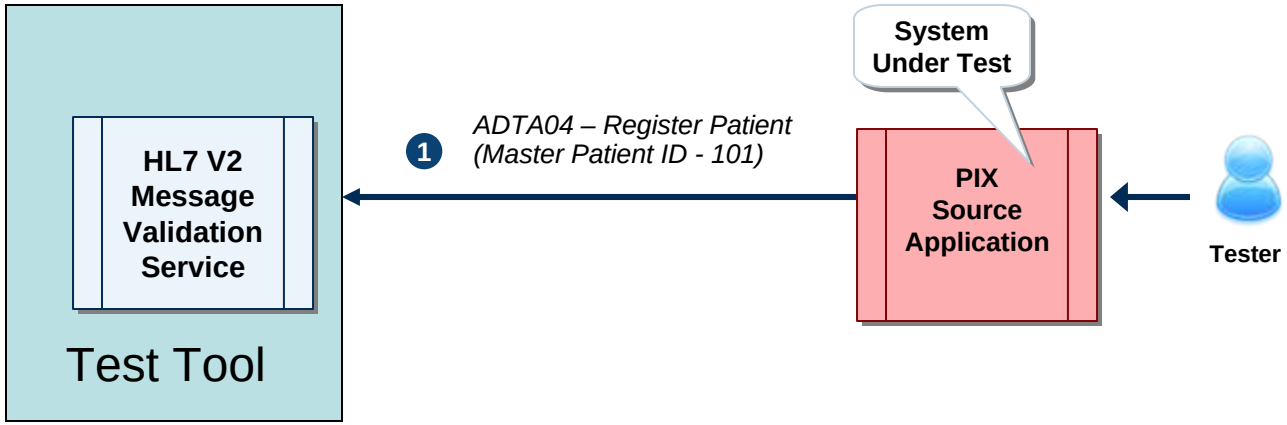
Types of Testing the Infrastructure will Support

- [A] Data Content Validation Testing
  - Validation of the data content contained in a test object. There is no context associated with the test object. That is, we are not concerned with the system that created the object or the system that will be receiving the object. In fact, the test object may have not been created by a system at all.
- [B] Data Content Conformance Testing
  - This type of testing is equivalent to Data Content Validation Testing with the difference that the test object is associated with the application that produced the object. Thus, an evaluation of the object can be translated into an evaluation of the application that produced it.
- [C] Communication Protocol Usage Conformance Testing
  - This type of testing asserts that an application correctly uses an allowed communications protocol. That is, on sending, the application correctly packages messages before sending them, and on receiving the application correctly extracts the message content from the package it was received in.
- [D] Testing for Conformance to the Requirements of the Data Exchange Standard
  - This type of testing evaluates if an application responds correctly to all messages, valid and invalid, that the application receives. This type of testing evaluates an application's reaction to variations in message structure and content. Messages are sent to the application with variations in the encoding characters and valid and invalid variations in content. The criterion for evaluation is receipt of a valid response from the application indicating that it processed a valid message or recognized an invalid message. No semantic evaluation of the response is made.
- [E] System Behavior Conformance Testing (from Application Functional Requirements)
  - This type of testing is designed to evaluate the behavior of an application. It generally consists of sending the application valid messages and evaluating the responses returned by the application for correct semantic content (when the SUT is a server). When the SUT is a client application it will be instructed to create a message or document, usually via a user interface. In order to conduct System Behavior Conformance Testing a test scenario is created in which a sequence of orchestrated transactions are composed to test adherence to specific functional requirements.
- [F] Syntactic interoperability Testing
  - This type of testing is designed primarily to establish that two applications are able to successfully exchange data. No evaluation of the application's processing of the data is made with this type of testing.
- [G] Semantic Interoperability Testing
  - This type of testing is the second phase of interoperability testing. If two applications establish that they are capable of exchanging data, this type of testing attempts to access if they also correctly process the data exchanged.

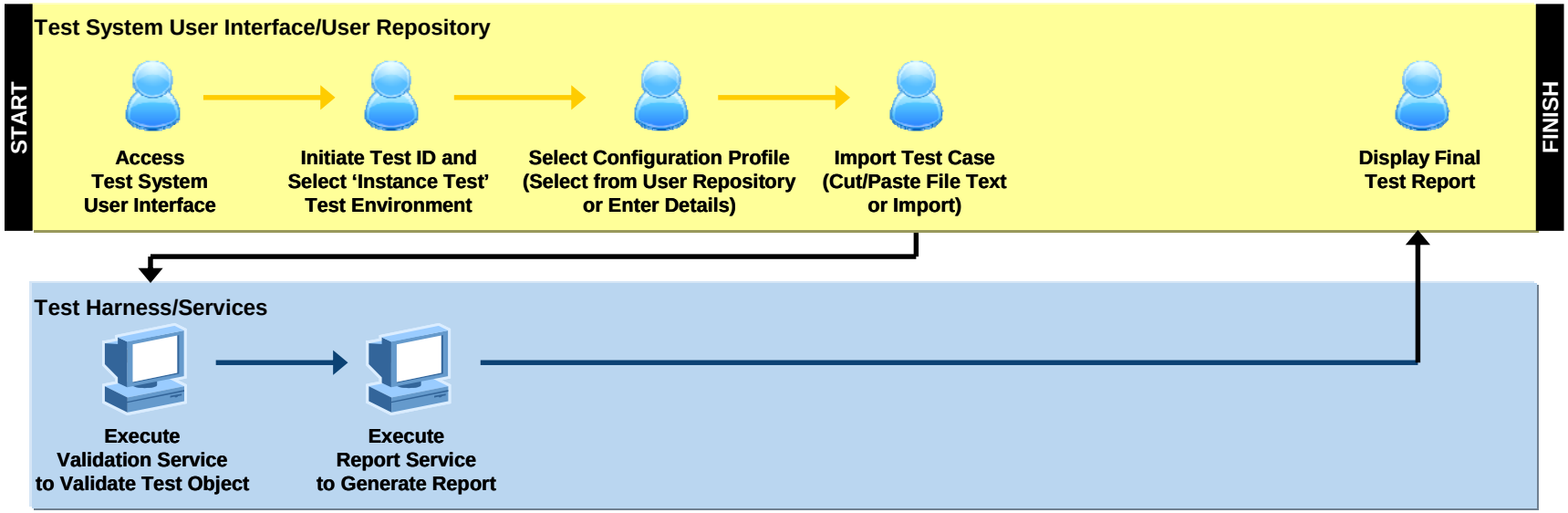
Test Environments

- ***Data Instance Test Environment:*** A test is conducted with a test object and a testing tool.
  - Evaluation of data content against a set of conformance rules
  - Data Content Validation Testing [A]
  - No Context; not associated with an application
  - Data Content Conformance Testing [B]
  - Test object is identified as having been produced by a specific application
- ***Isolated System Test Environment:*** A test is conducted with one vendor system and a test tool. The vendor system may interact with test agents and/or validation testing tools.
  - Includes Data Instance Testing Activities [B]
  - Protocol Usage Conformance Testing [C]
  - Testing for Conformance to the Requirements of the Data Exchange Standard [D]
  - Test range of conformance requirements
  - Multiple test cases conducted
  - System Behavior Conformance Testing from Application Functional Requirements [E]
  - e.g., Pre-connectathon Testing
- ***Peer-to-peer System Test Environment:*** A test is conducted among a group of vendor systems. A vendor system may interact with a test tool or other vendor systems.
  - Includes Isolated System Testing Activities
  - Syntactic Interoperability Testing [F]
  - Semantic Interoperability Testing [G]
  - e.g., Connectathon Testing

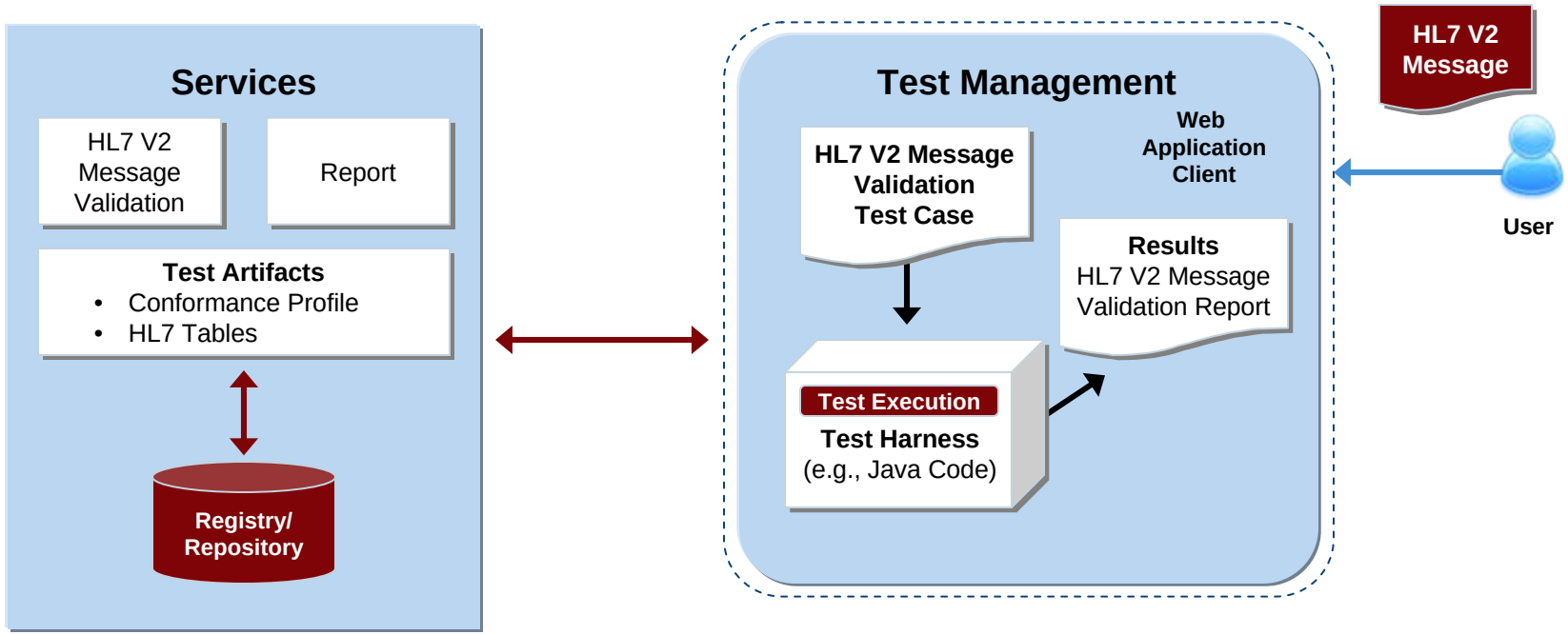
Data Content Conformance Testing
Message Validation



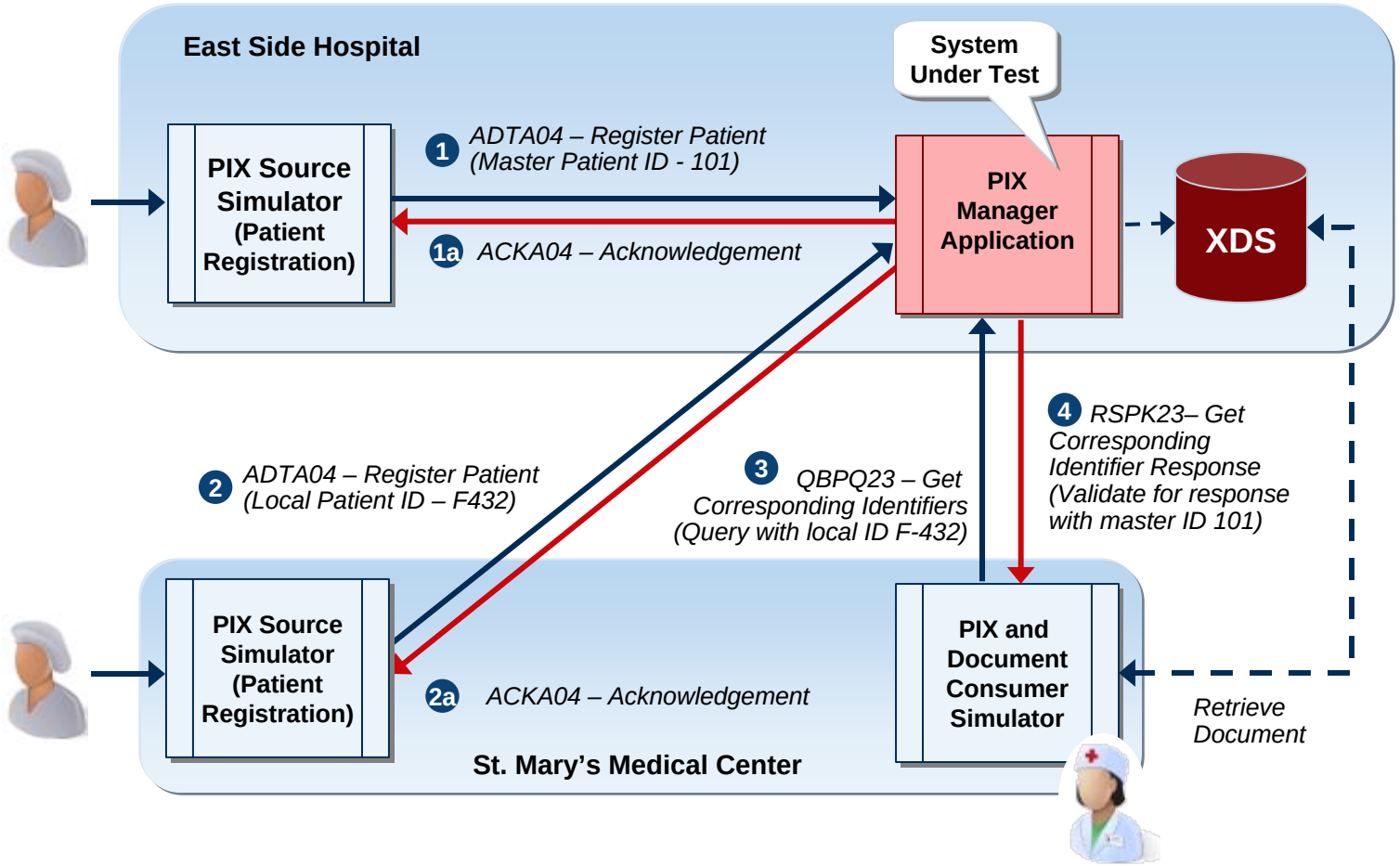
Data Instance Testing High-Level Use Case



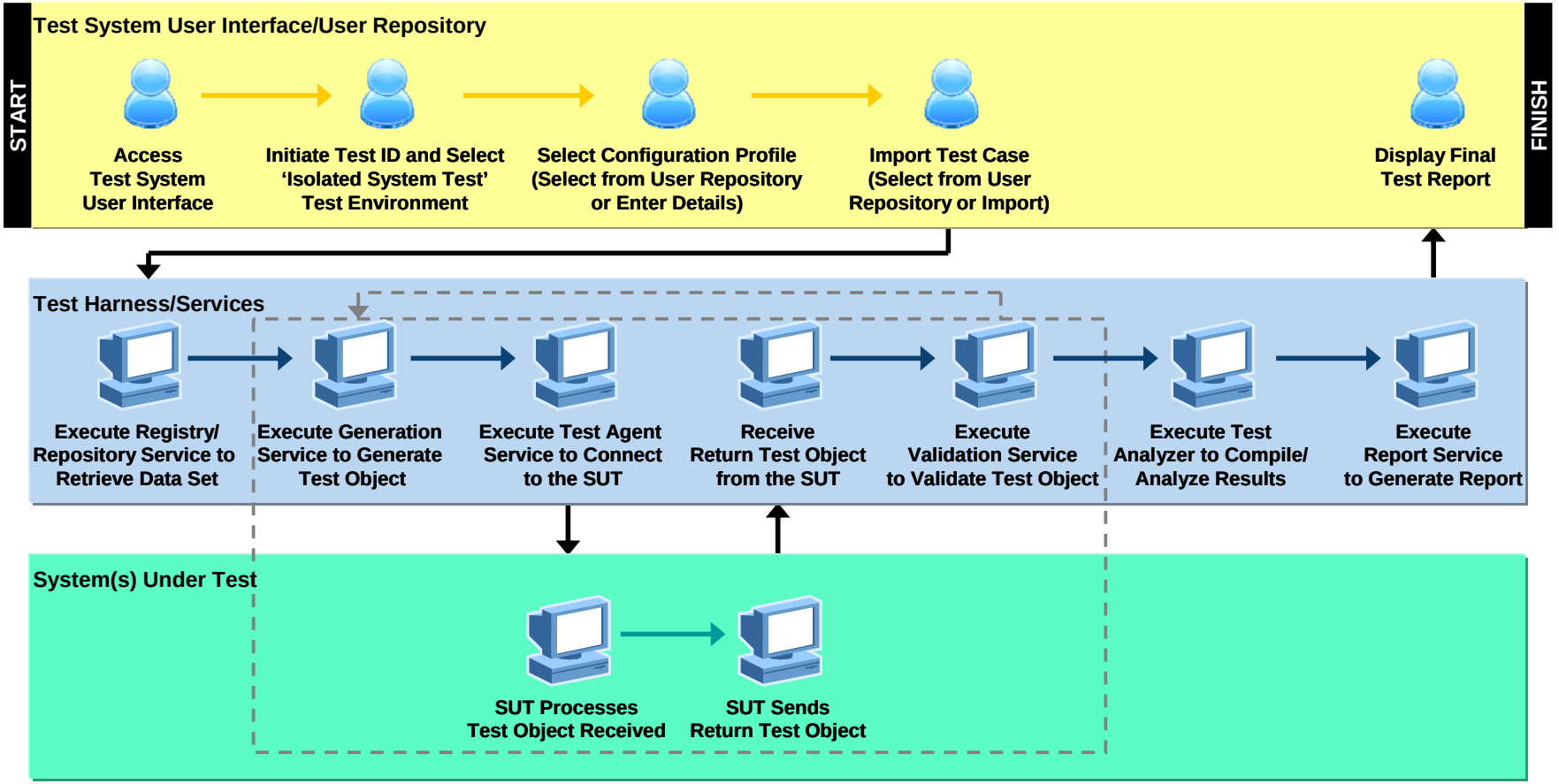
Data Instance Test Environment
E.g., *Conformance Testing of an HL7 V2 Message*



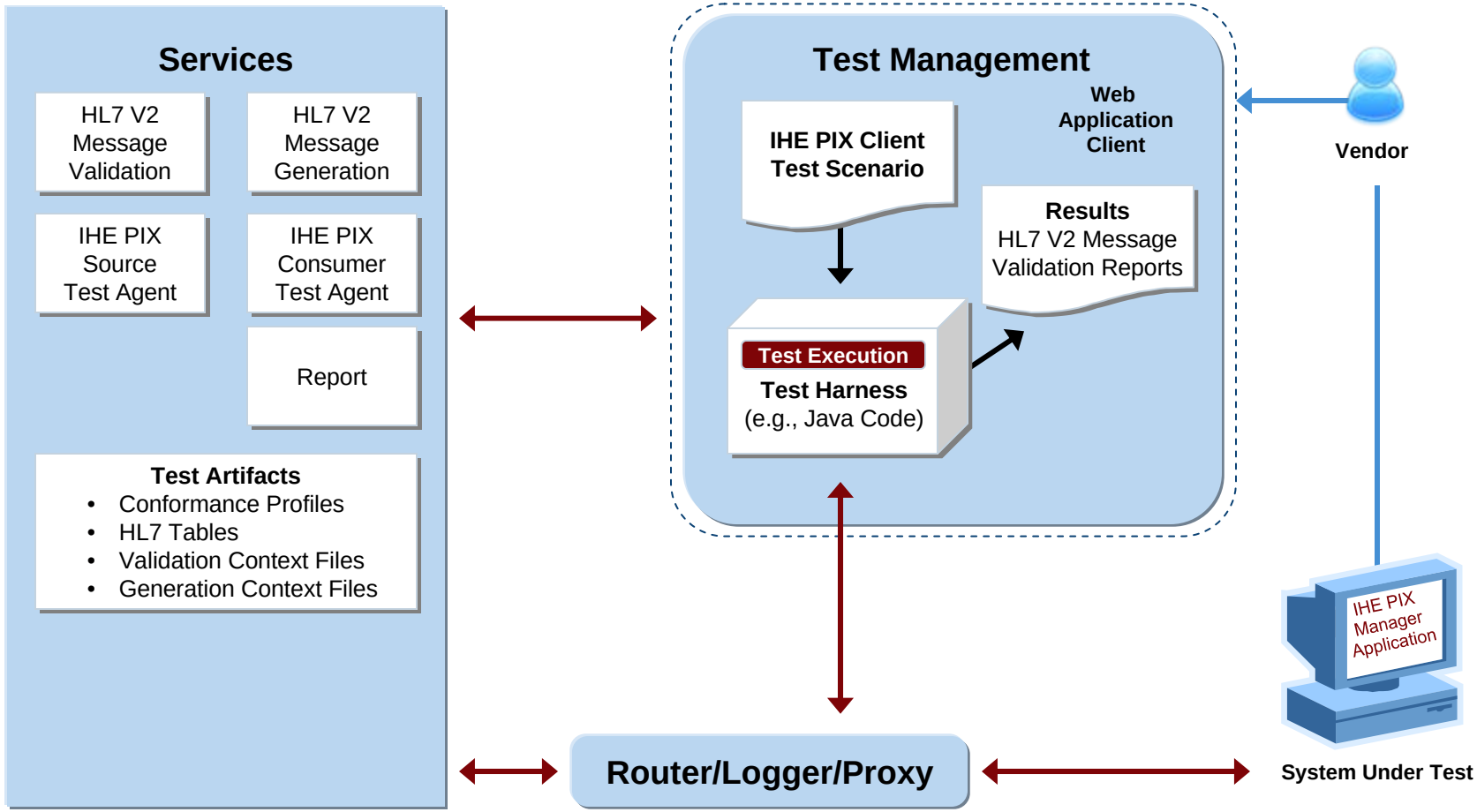
Isolated System Testing
Pre-Connectathon PIX Client Test Case



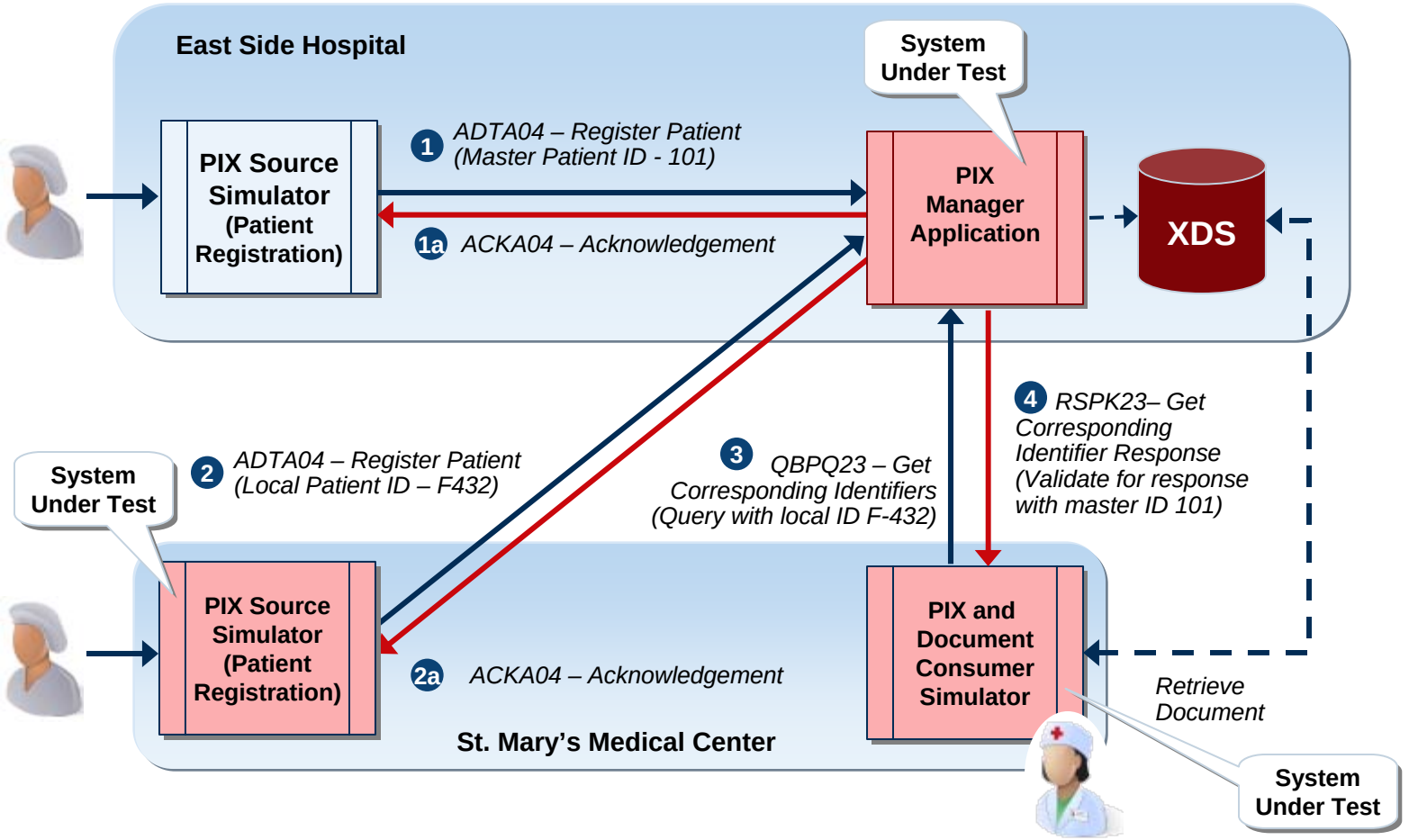
Isolated System Testing High-Level Use Case



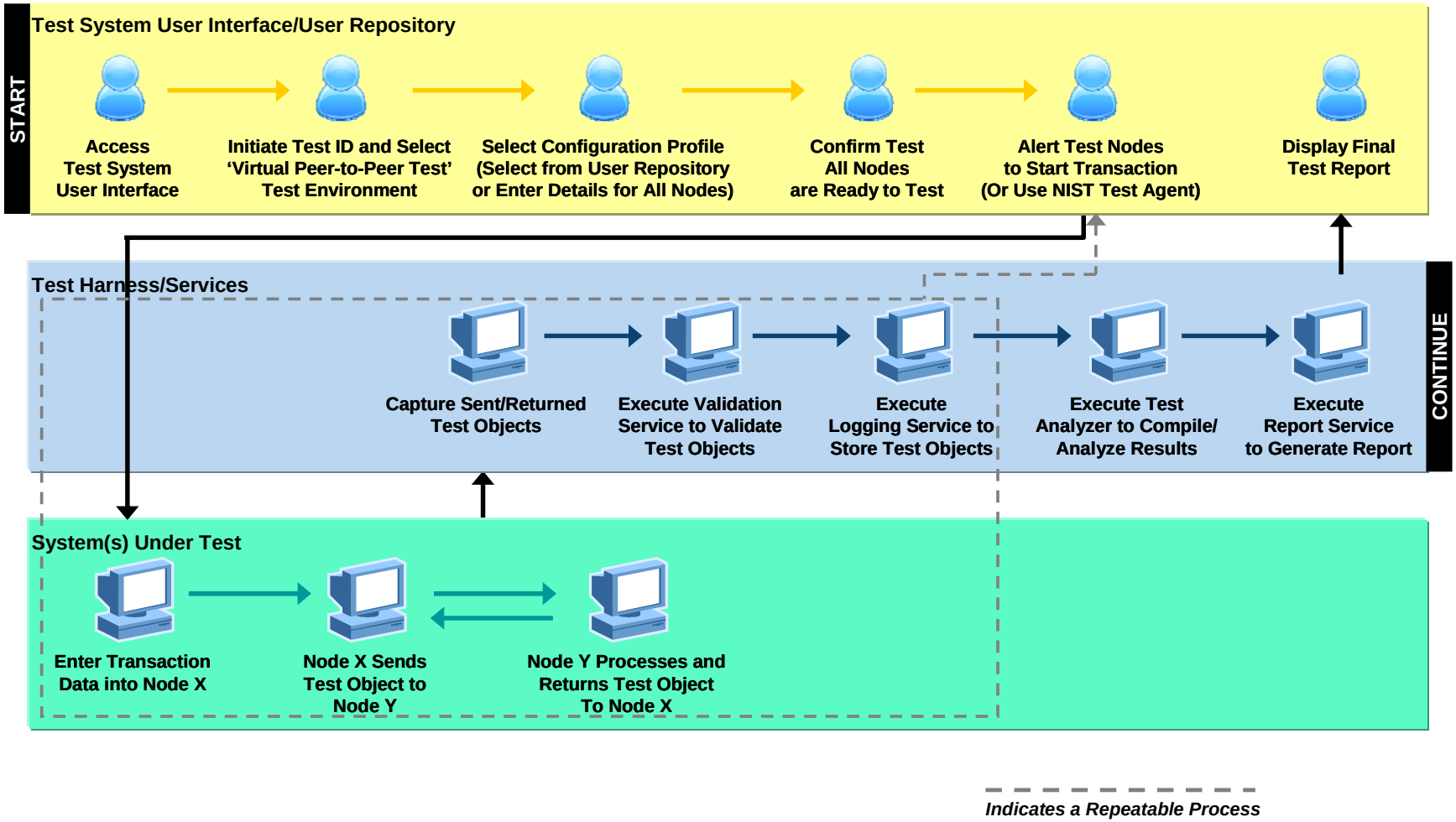
Isolated System Test Environment
E.g., *IHE PIX Testing using a Web Application Client*



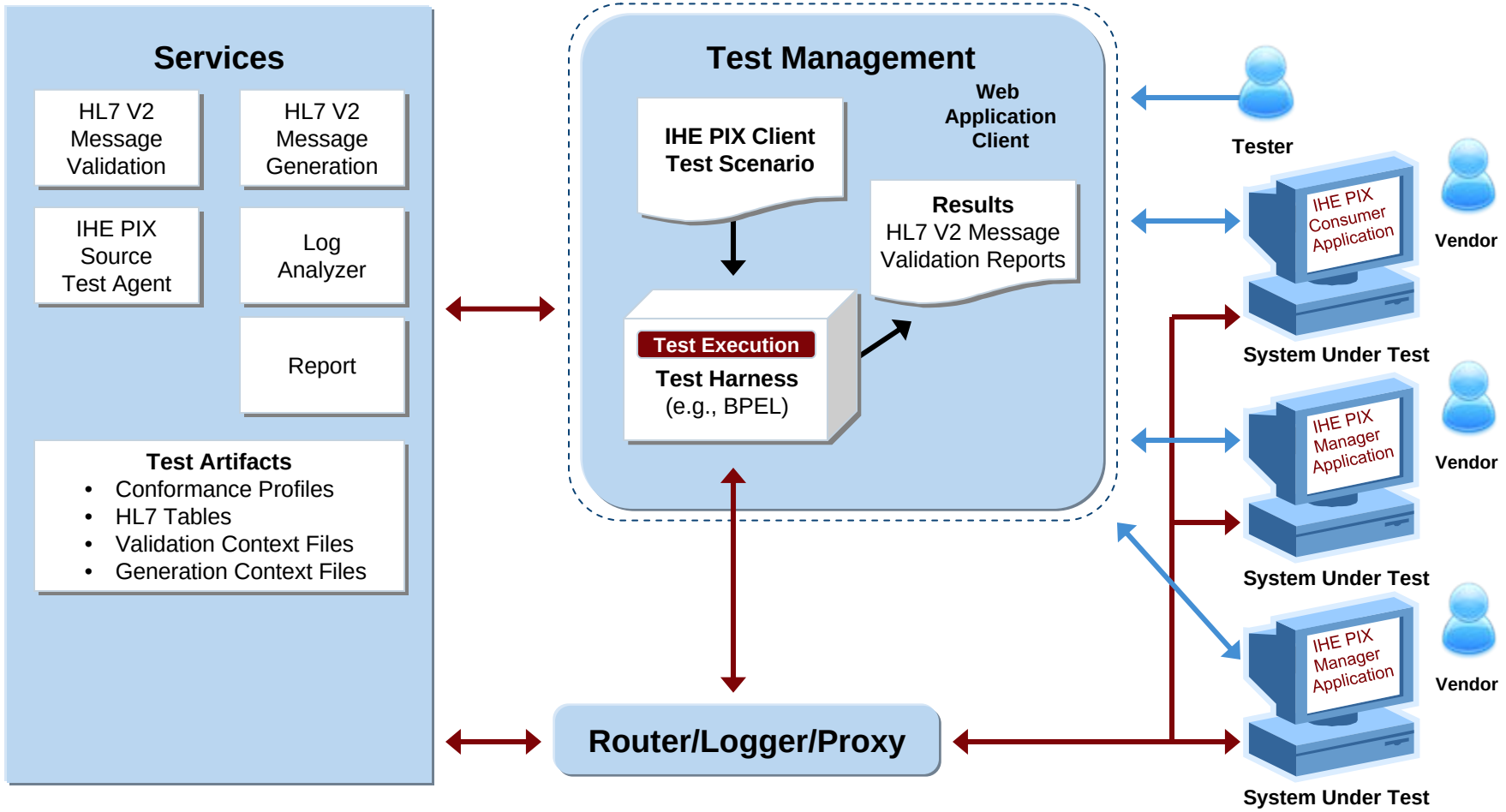
Peer-to-peer System Testing
Connectathon PIX Client Test Case



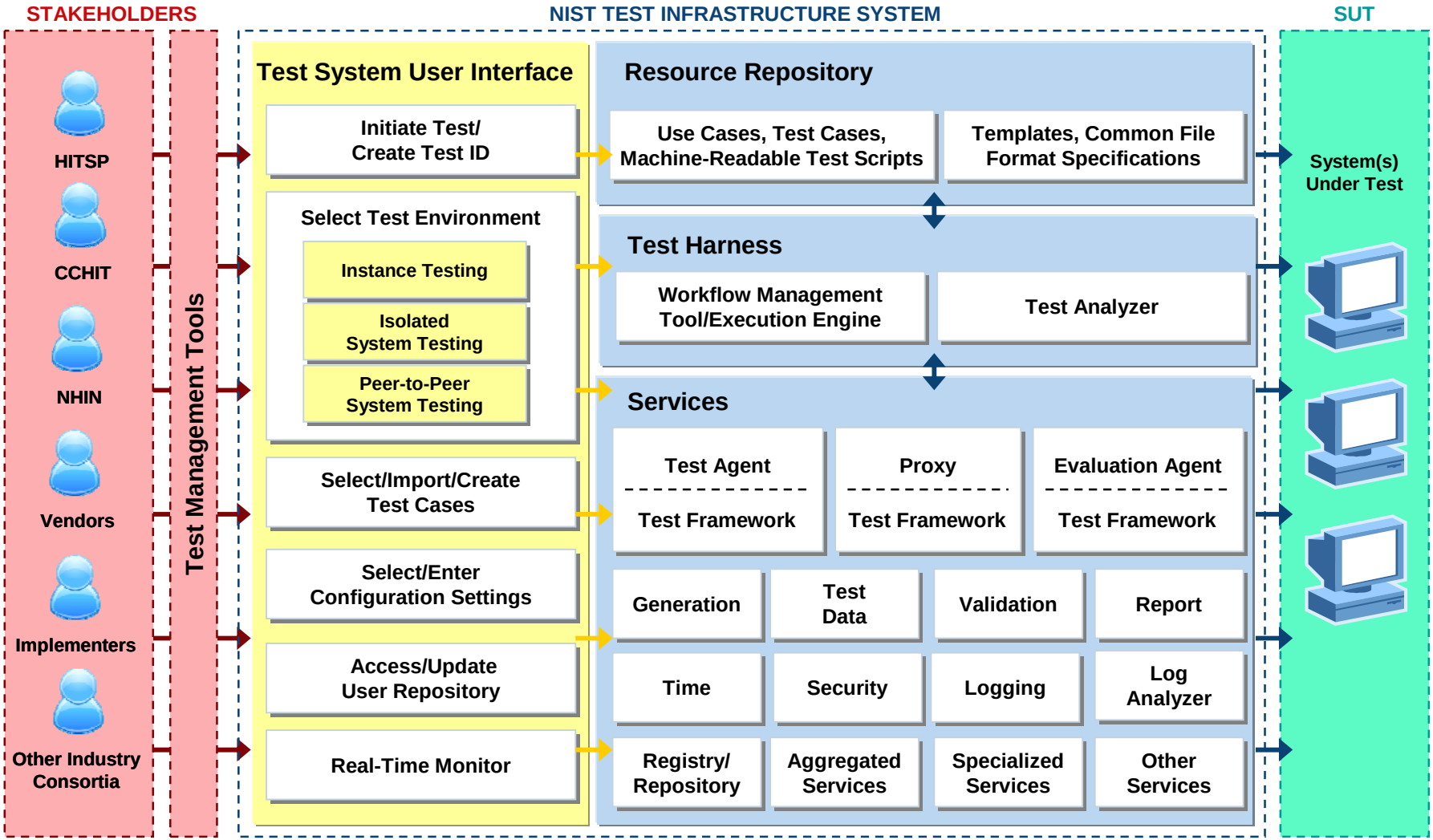
Peer-to-Peer System Testing High-Level Use Case



Peer-to-Peer System Test Environment
E.g., *IHE PIX Testing using BPEL/Proxy Model*



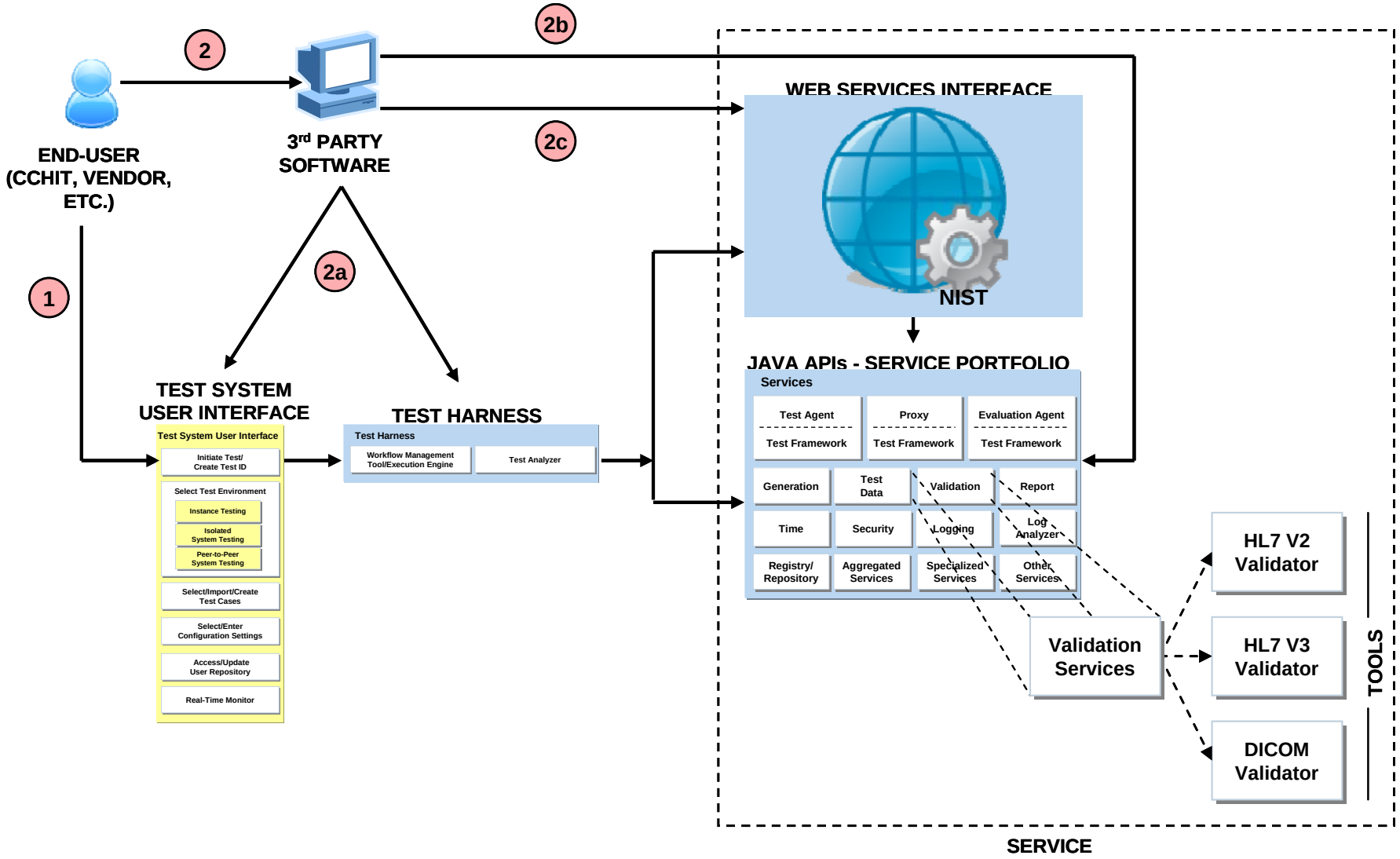
Conceptual View of Testing Infrastructure



Testing Infrastructure Key Components

- Test Harness
  - Portal into test system
  - Orchestration of services to execute tests
  - Test execution engine
  - Many test harnesses can be built (NIST and others)
- Services
  - Provides a testing function
  - Provides reusable components
  - Multiple delivery mechanisms
- Network Functions
  - Routing
  - Logging
  - Proxy
- "Glue" Components
  - Specifications of interface definitions
  - Common file formats
- Test Specification Methodology
  - Common template
  - Automated transformation to test scripting language
  - Tools to build test suites
- Design Templates
  - e.g., HL7 V2 test agents
  - e.g., Test Harness UI Templates
- Test Management
  - Ancillary system
  - Controls the setup, selection, and analysis of results
  - Not a main focus for NIST

Multiple Delivery Mechanisms



Summary

- Beyond a collection of tools
- Integrated testing system—Holistic View
  - Address the testing needs of entire process
  - From standards development to results analysis
  - Users can also make use of the framework to suit their needs
    - Independent testing components
    - Composition of services is at their discretion
    - Multiple delivery mechanisms
  - Formal methodology will live on beyond a collection of disparate tools
- Incremental Rollout
  - NIST will provide specific tools as needed to meet testing priorities as established by ONC for meaningful use certification
  - NIST will provide tools to support certification entities
  - NIST will provide a tools portal
- Community Effort