Treating Patients Right

The Federal Healthcare Information Exchange (FHIE) in the US Government

MDA™ Implementers' Workshop 2003

Tony Mallia
tmallia@ciber.com

ciber

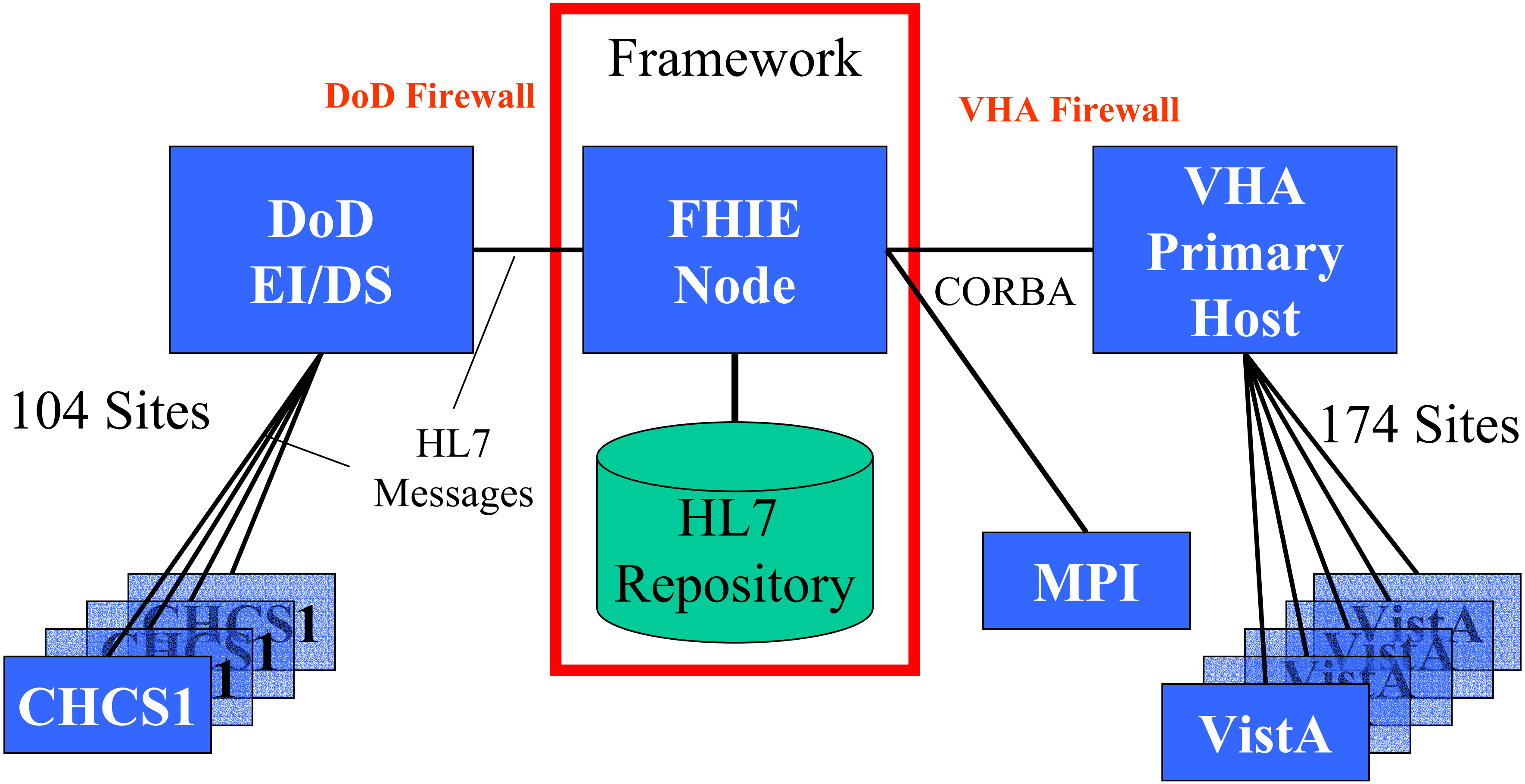
FHIE Background

- Presidential Review Directive-5 1998
- State of the project
  - Near Term Solution initiated April 2001
  - In Production July 2002
  - Enhancements under test
  - Developers:
    - VHA SD&D
    - Northrop Grumman IT (Framework)
    - SAIC for DOD EI/DS
    - ESI for VHA

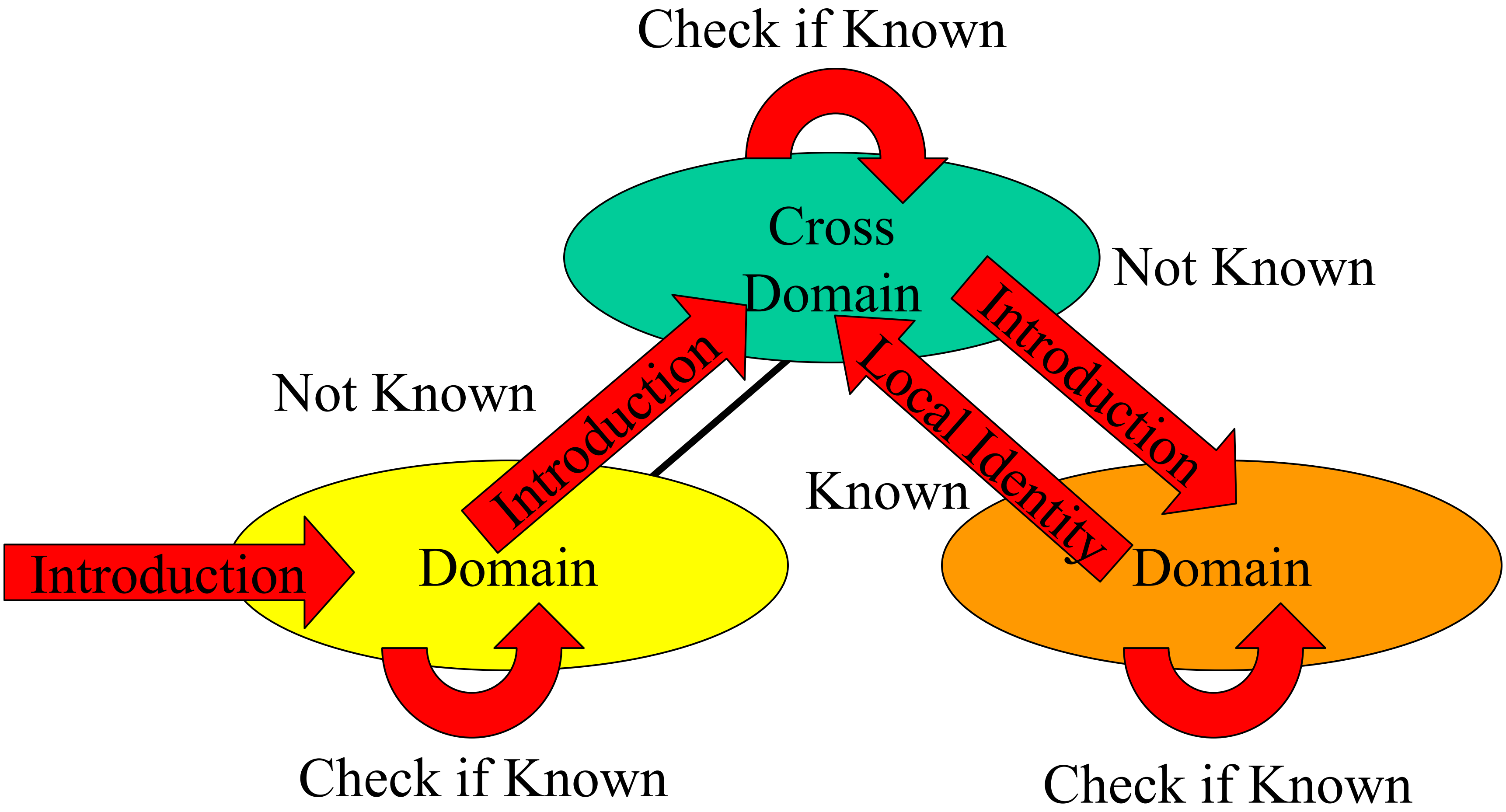
FHIE Near Term Overview

- Cooperating Healthcare organizations
  - 104 DoD Medical Treatment Facilities
  - 174 Veterans Health Administration Hospitals
- Provide VHA access to DoD health information under the necessary regulations
- Provide better healthcare
  - Previous history
  - Patient Safety

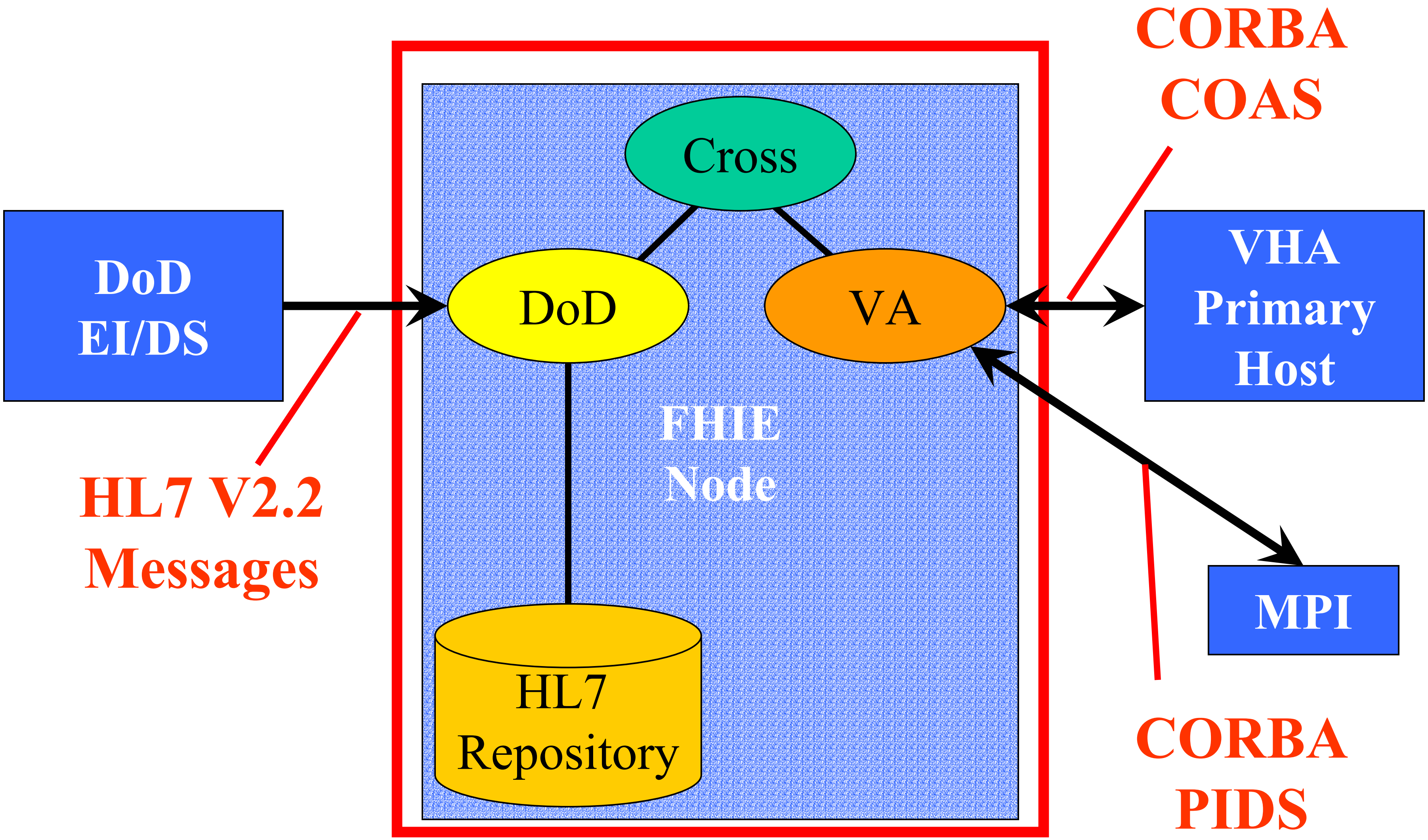
Configuration



Patient Introduction



FHIE Connector Protocols



HL7 in FHIE

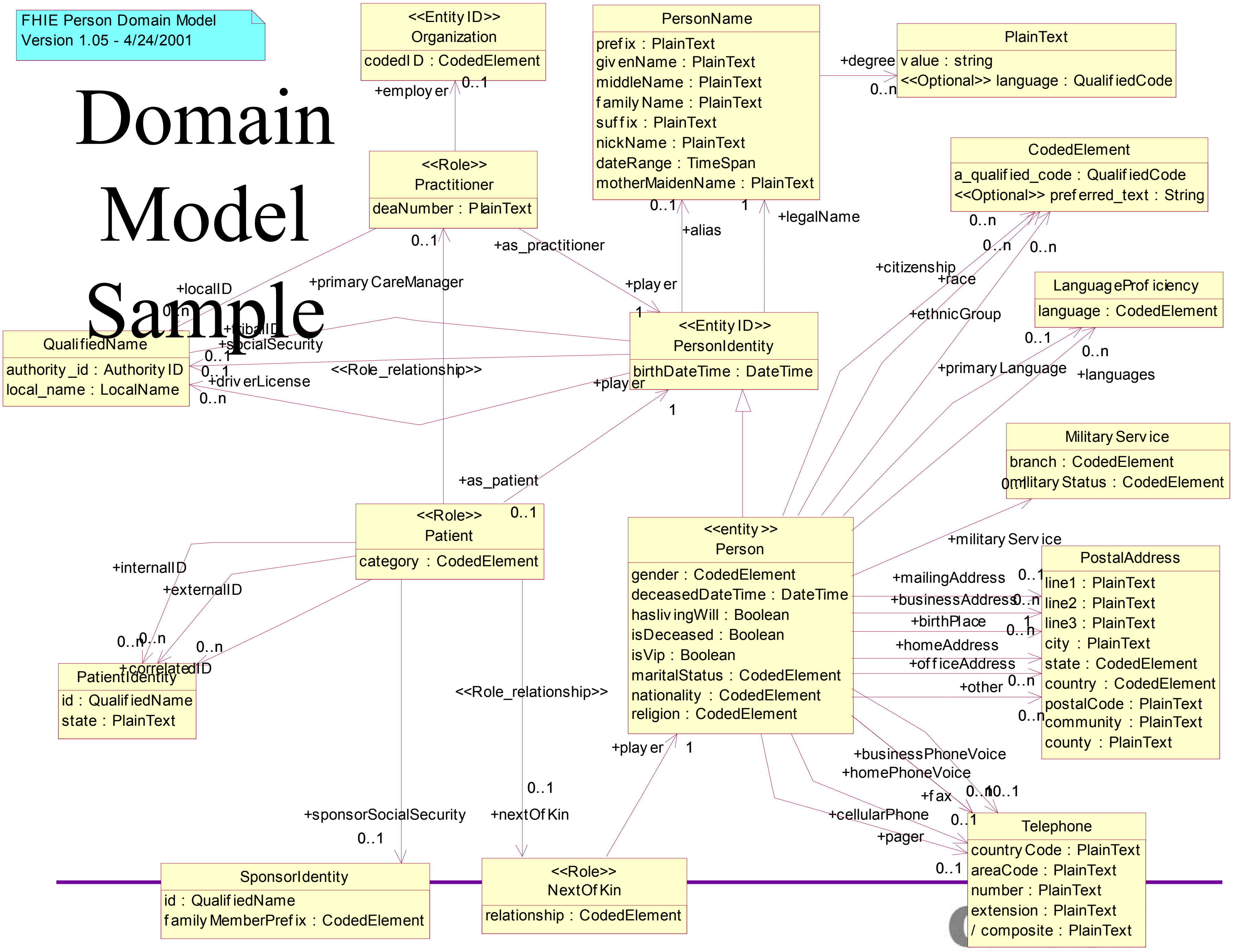
- Admission - Discharge - Transfer (ADT)
  - Patient Encounter with Diagnoses and Procedures
  - Discharge Summary
  - Allergies
- Laboratory Orders and results
  - Chemistry
  - Microbiology
  - Anatomic Pathology
- Medications (Outpatient Pharmacy)
  - Orders and Dispensing

Computational Independent Domain Model in UML

- Less than 200 classes
  - Person
  - Common Acts
  - Laboratory
  - Medications
  - Patient Encounter
  - Working Lists
  - Data Types

Domain Model Sample

FHIE Person Domain Model
Version 1.05 - 4/24/2001



<<Entity ID>> Organization
- codedID : CodedElement

PersonName
- prefix : PlainText
- givenName : PlainText
- middleName : PlainText
- familyName : PlainText
- suffix : PlainText
- nickName : PlainText
- dateRange : TimeSpan
- motherMaidenName : PlainText

PlainText
- value : string
- <<Optional>> language : QualifiedCode

CodedElement
- a_qualified_code : QualifiedCode
- <<Optional>> preferred_text : String

<<Role>> Practitioner
- deaNumber : PlainText

LanguageProficiency
- language : CodedElement

QualifiedName
- authority_id : AuthorityID
- local_name : LocalName

<<Entity ID>> PersonIdentity
- birthDateTime : DateTime

MilitaryService
- branch : CodedElement
- militaryStatus : CodedElement

<<Role>> Patient
- category : CodedElement

<<entity>> Person
- gender : CodedElement
- deceasedDateTime : DateTime
- haslivingWill : Boolean
- isDeceased : Boolean
- isVip : Boolean
- maritalStatus : CodedElement
- nationality : CodedElement
- religion : CodedElement

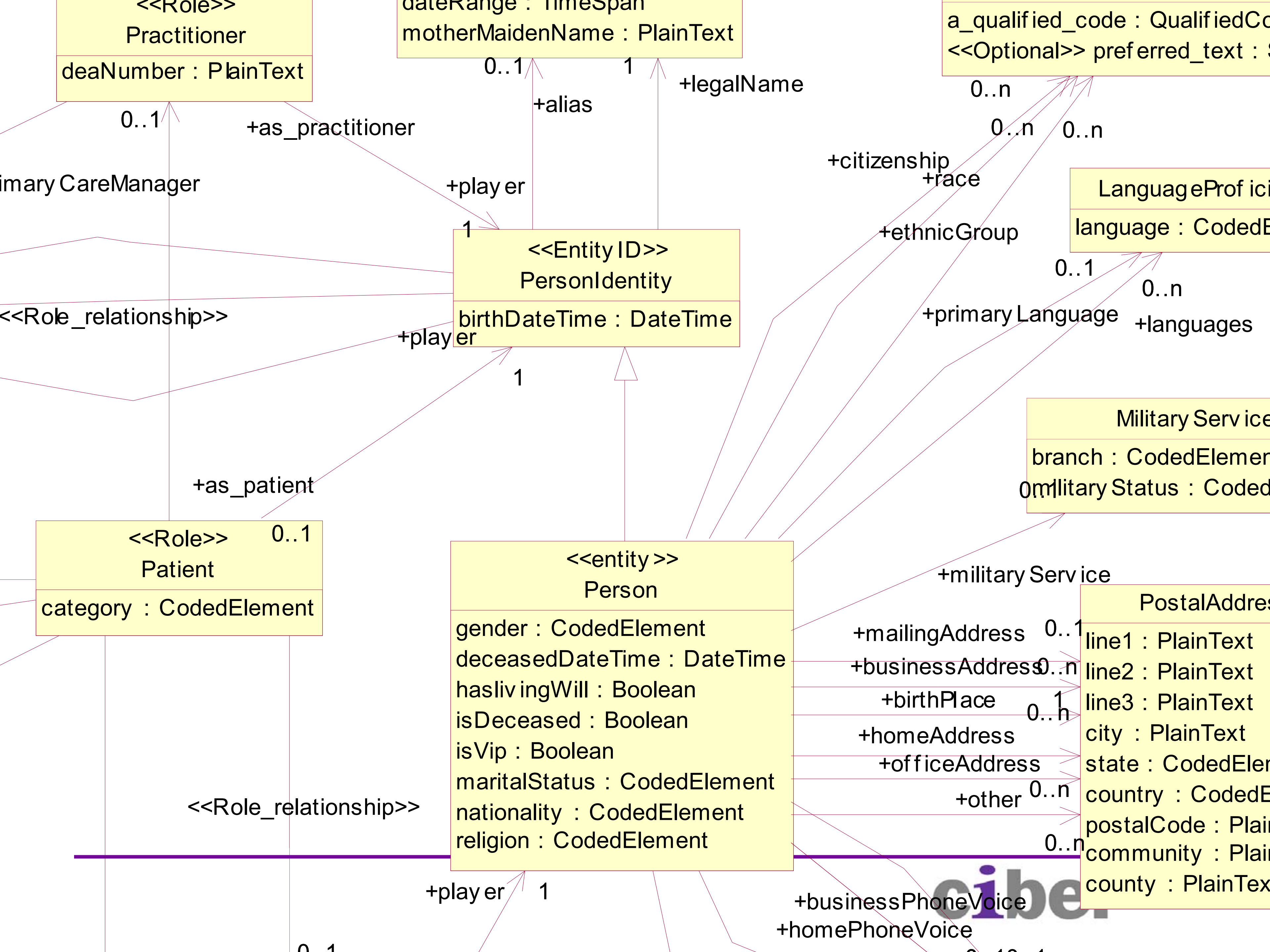
PostalAddress
- line1 : PlainText
- line2 : PlainText
- line3 : PlainText
- city : PlainText
- state : CodedElement
- country : CodedElement
- postalCode : PlainText
- community : PlainText
- county : PlainText

PatientIdentity
- id : QualifiedName
- state : PlainText

Telephone
- countryCode : PlainText
- areaCode : PlainText
- number : PlainText
- extension : PlainText
- / composite : PlainText

SponsorIdentity
- id : QualifiedName
- familyMemberPrefix : CodedElement

<<Role>> NextOfKin
- relationship : CodedElement



<<Role>>
Practitioner

deaNumber : PlainText

dateRange : TimeSpan
motherMaidenName : PlainText

a_qualified_code : QualifiedCo
<<Optional>> preferred_text : S

LanguageProfici
language : CodedE

<<Entity ID>>
PersonIdentity

birthDateTime : DateTime

Military Service
branch : CodedElemen
militaryStatus : Coded

<<Role>>
Patient

category : CodedElement

<<entity>>
Person

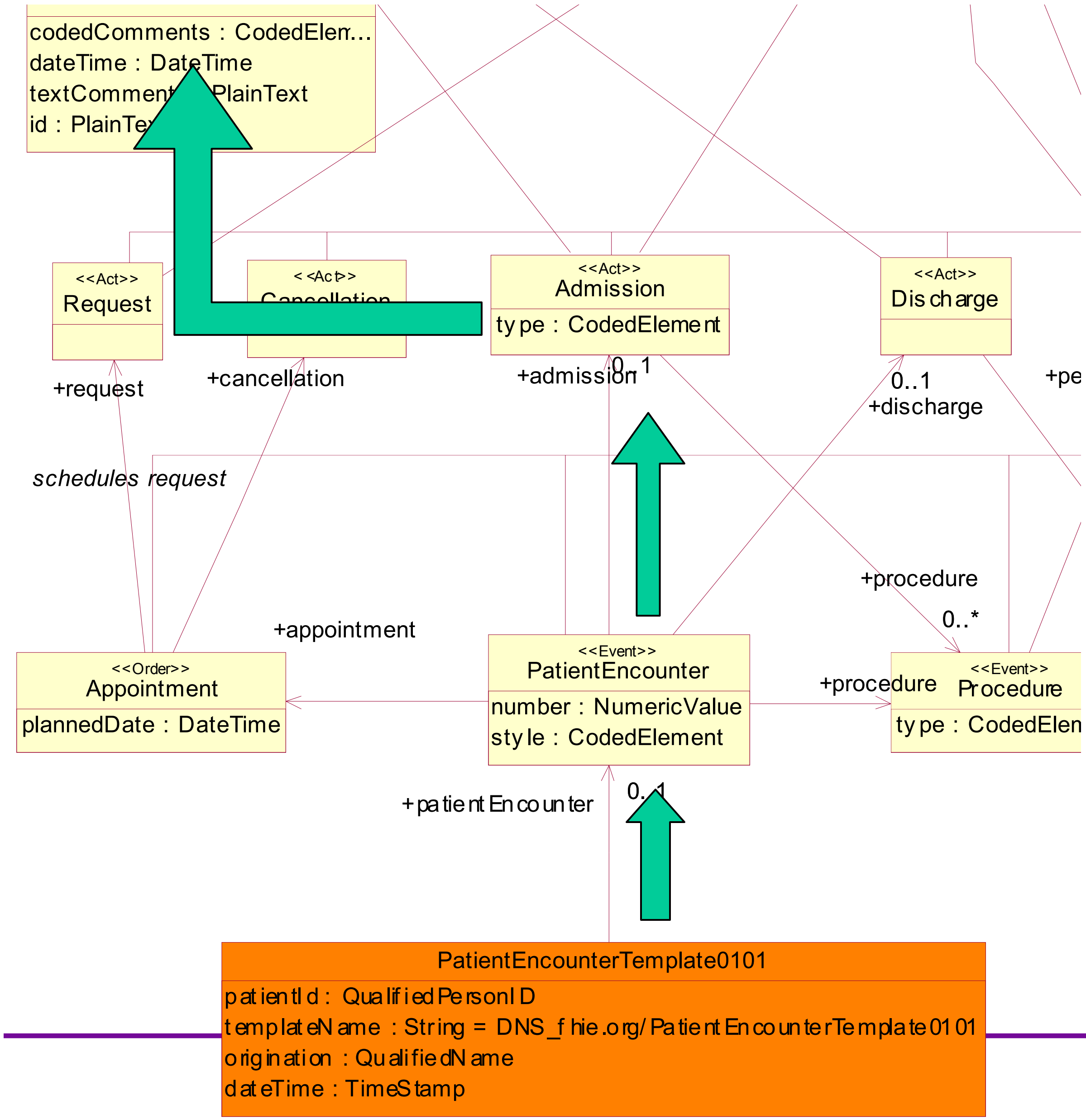
gender : CodedElement
deceasedDateTime : DateTime
haslivingWill : Boolean
isDeceased : Boolean
isVip : Boolean
maritalStatus : CodedElement
nationality : CodedElement
religion : CodedElement

PostalAddres
line1 : PlainText
line2 : PlainText
line3 : PlainText
city : PlainText
state : CodedElen
country : CodedE
postalCode : Plai
community : Plai
county : PlainTex

Record Templates

- Semantic Paths
  - UML Classes are types
  - UML Association End and Attribute names are roles
- Mapping semantics
  - Use UML dot notation for a single concept
- Relationships of Templates to Domain Model
  - Template is a Root class, Class Diagram and the first Association

patientEncounter.admission.dateTime



codedComments : CodedElem...
dateTime : DateTime
textComment : PlainText
id : PlainText

<<Act>>
Request

<<Act>>
Cancellation

<<Act>>
Admission
type : CodedElement

<<Act>>
Discharge

+request

+cancellation

+admission

0..1

0..1
+discharge

+pe

schedules request

+procedure

0..*

+appointment

<<Order>>
Appointment
plannedDate : DateTime

<<Event>>
PatientEncounter
number : NumericValue
style : CodedElement

+procedure

<<Event>>
Procedure
type : CodedElem

+patientEncounter

0..1

PatientEncounterTemplate0101
patientId : QualifiedPersonID
templateName : String = DNS_fhie.org/PatientEncounterTemplate0101
origination : QualifiedName
dateTime : TimeStamp

Template Example

Generation of the template data dictionary

patientEncounter.reason.a_qualified_code.coding_scheme_id.authority
patientEncounter.reason.a_qualified_code.coding_scheme_id.naming_entity
patientEncounter.reason.a_qualified_code.a_code
patientEncounter.reason.preferred_text
patientEncounter.comments.value
patientEncounter.comments.language.coding_scheme_id.authority
patientEncounter.comments.language.coding_scheme_id.naming_entity
patientEncounter.comments.language.a_code
patientEncounter.reasonComment.value
patientEncounter.reasonComment.language.coding_scheme_id.authority
patientEncounter.reasonComment.language.coding_scheme_id.naming_entity
patientEncounter.reasonComment.language.a_code

Platform Specific Formats

- Internal format - Java
- Connector Formats -
  - CORBA COAS Observation Data Structures
  - CORBA PIDS Traits
  - HL7 messages

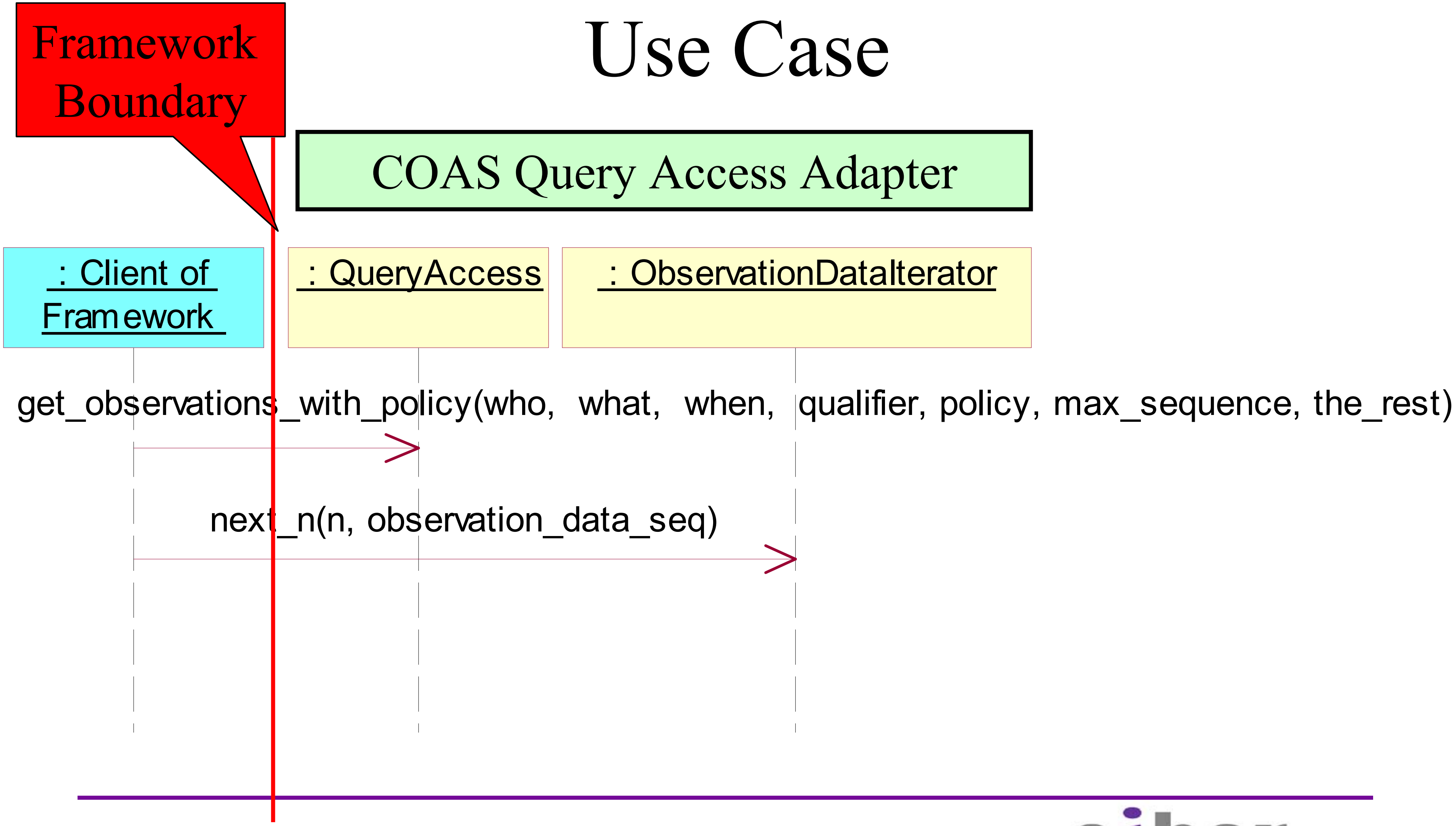
Layers of Design

- Enterprise collaboration
  - Only Connectors are visible
- Component Collaboration
  - Framework is visible
- Component internal design
  - Classes are visible
- Recursive inward refinement

Enterprise Collaboration

- Use Case driven
- Collaboration diagrams to system level
- Data Dictionaries and ICDs
- Reference Information Model
- Template definitions

FHIE006.I6(Deliver Patient Records) Use Case



Framework Boundary

COAS Query Access Adapter

: Client of Framework

: QueryAccess

: ObservationDataIterator

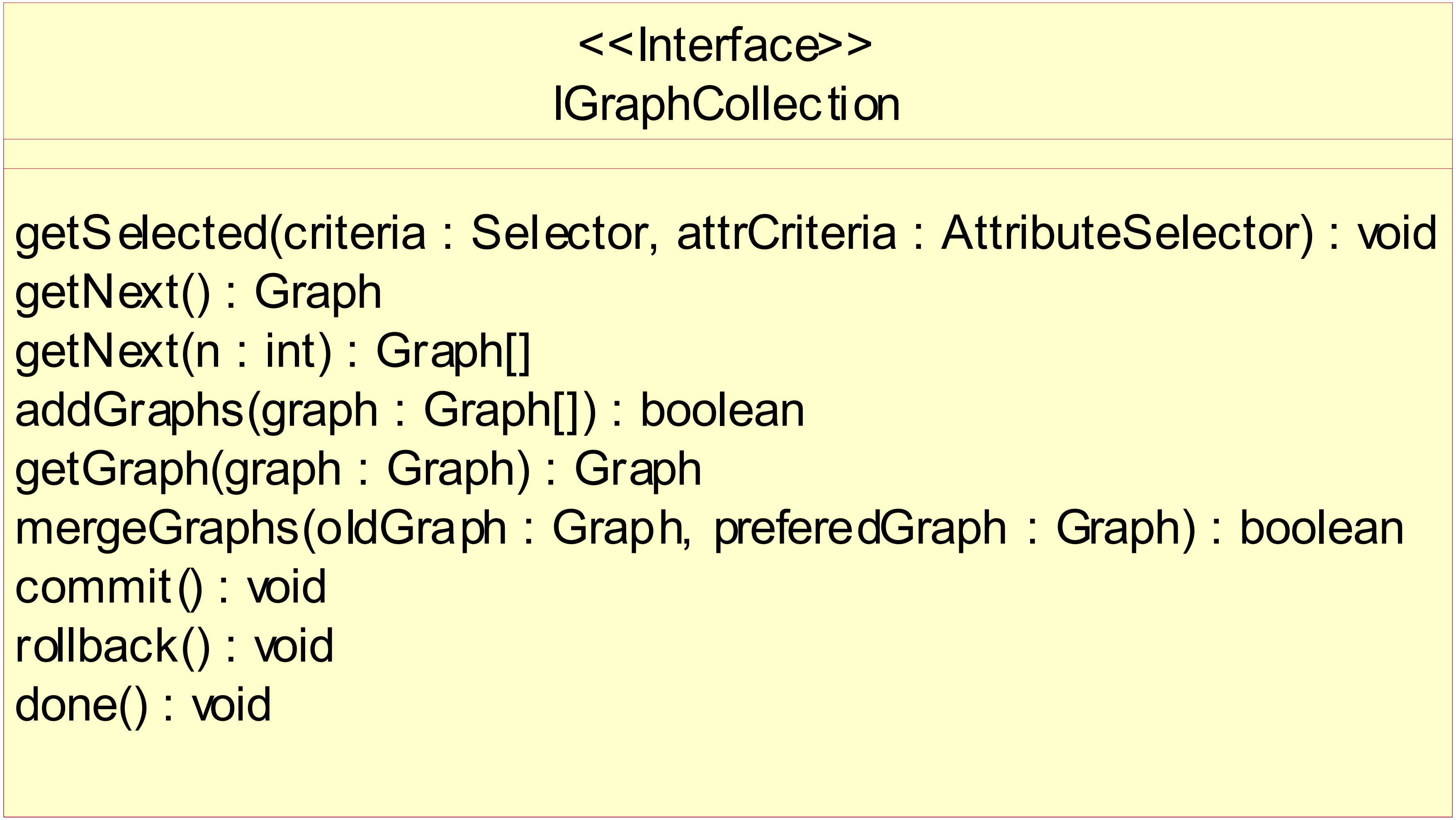
get_observations_with_policy(who, what, when, qualifier, policy, max_sequence, the_rest)

next_n(n, observation_data_seq)

FHIE Framework Architecture

- Graph Collections
  - Person Graph Collection
  - Record Graph Collection
- Templates and Graphs
  - Templates are the Record structure
  - Graphs are the Record instances
- Adapters convert from communication to/from GraphCollection operations
  - Push - Add
  - Pull - GetSelected

Graph Collection

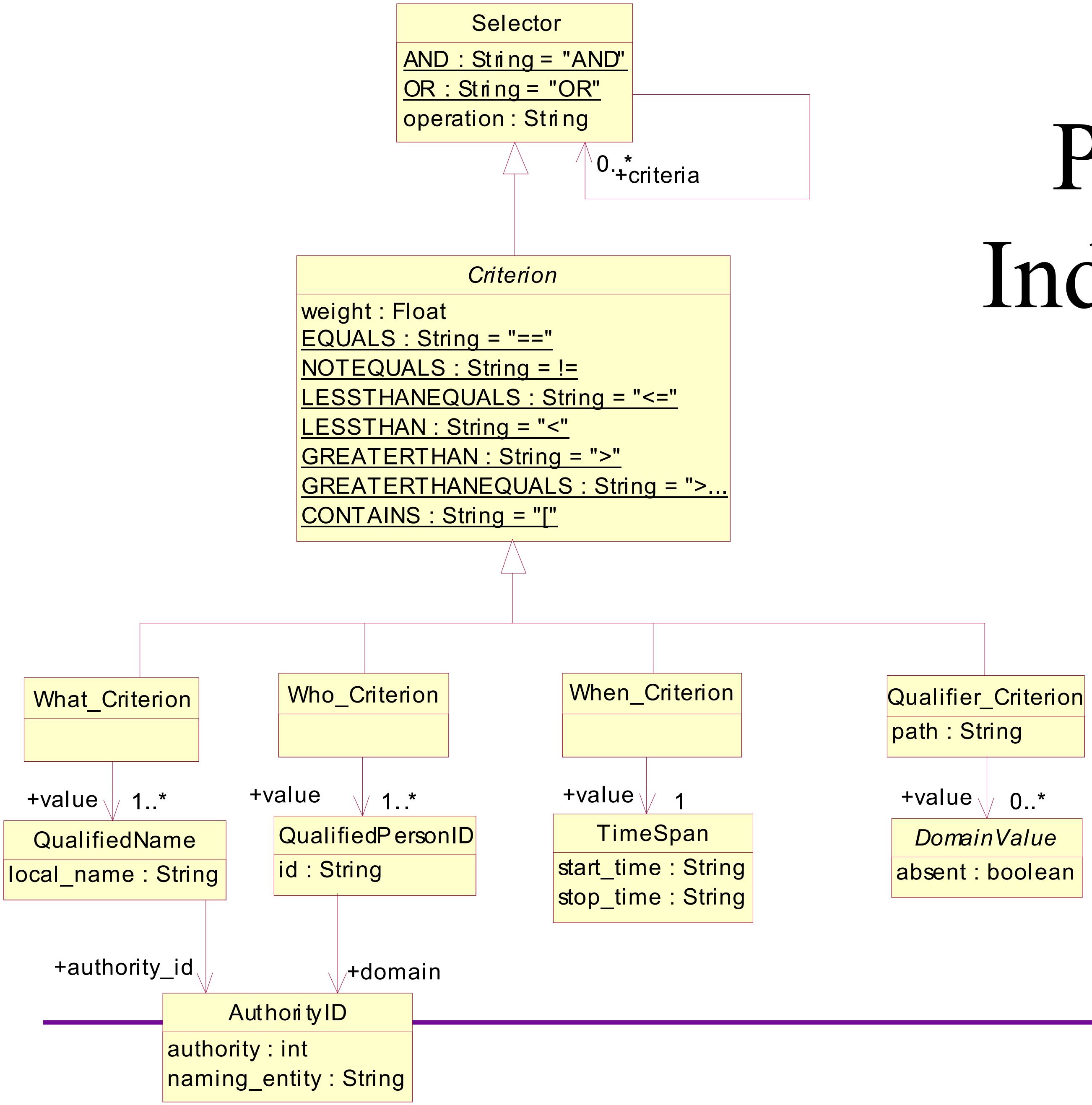


<<Interface>>
IGraphCollection

getSelected(criteria : Selector, attrCriteria : AttributeSelector) : void
getNext() : Graph
getNext(n : int) : Graph[]
addGraphs(graph : Graph[]) : boolean
getGraph(graph : Graph) : Graph
mergeGraphs(oldGraph : Graph, preferedGraph : Graph) : boolean
commit() : void
rollback() : void
done() : void

IGraphCollection.getSelected(criteria:Selector)

Platform Independent Query



Selector
AND : String = "AND"
OR : String = "OR"
operation : String

0..*
+criteria

Criterion
weight : Float
EQUALS : String = "=="
NOTEQUALS : String = !=
LESSTHANEQUALS : String = "<="
LESSTHAN : String = "<"
GREATERTHAN : String = ">"
GREATERTHANEQUALS : String = ">...
CONTAINS : String = "["

What_Criterion

+value
1..*

QualifiedName
local_name : String

Who_Criterion

+value
1..*

QualifiedPersonID
id : String

When_Criterion

+value
1

TimeSpan
start_time : String
stop_time : String

Qualifier_Criterion
path : String

+value
0..*

DomainValue
absent : boolean

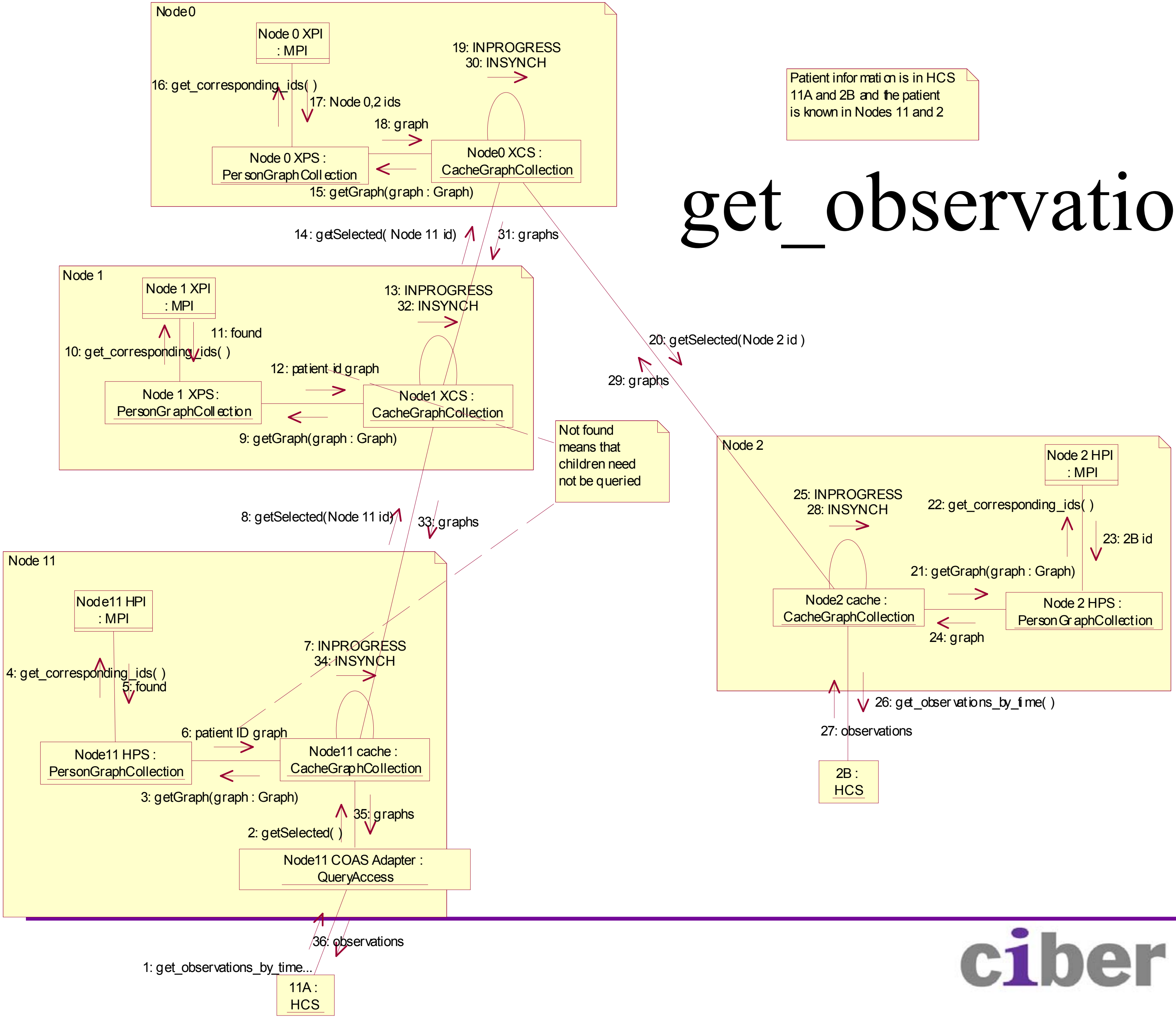
+authority_id
+domain

AuthorityID
authority : int
naming_entity : String

Platform Independent Computational Model

- Component Types
  - Adapters
  - Person Graph Collection
  - Clinical Graph Collection
  - Graph Collection Manager
- Adapter Operation driven
  - Collaboration diagrams to component level

get_observations



Node0

Node 0 XPI : MPI

16: get_corresponding_ids( )

17: Node 0,2 ids

18: graph

19: INPROGRESS
30: INSYNCH

Node 0 XPS : PersonGraphCollection

Node0 XCS : CacheGraphCollection

15: getGraph(graph : Graph)

14: getSelected(Node 11 id)

31: graphs

Patient information is in HCS 11A and 2B and the patient is known in Nodes 11 and 2

Node 1

Node 1 XPI : MPI

11: found

10: get_corresponding_ids( )

13: INPROGRESS
32: INSYNCH

12: patient id graph

Node 1 XPS : PersonGraphCollection

Node1 XCS : CacheGraphCollection

9: getGraph(graph : Graph)

20: getSelected(Node 2 id)

29: graphs

Not found means that children need not be queried

8: getSelected(Node 11 id)

33: graphs

Node 2

Node 2 HPI : MPI

25: INPROGRESS
28: INSYNCH

22: get_corresponding_ids( )

23: 2B id

21: getGraph(graph : Graph)

Node2 cache : CacheGraphCollection

Node 2 HPS : PersonGraphCollection

24: graph

26: get_observations_by_time( )

27: observations

2B : HCS

Node 11

Node11 HPI : MPI

4: get_corresponding_ids( )

5: found

7: INPROGRESS
34: INSYNCH

6: patient ID graph

Node11 HPS : PersonGraphCollection

Node11 cache : CacheGraphCollection

3: getGraph(graph : Graph)

2: getSelected( )

35: graphs

Node11 COAS Adapter : QueryAccess

36: observations

1: get_observations_by_time...

11A : HCS

Adapter Design

- Implicit Mapping
  - Template is the message structure
  - Formating of the message is to a programmatic rule
  - Adapter is simple
- Explicit Mapping
  - Message structure is the external Interface Control Document (ICD)
  - Elements must be converted to the Template
  - Elements are mapped or ignored
  - Mapping tables drive the adapter development

Component Internal Design

- Component façade operation driven
  - IGraphCollection operation definitions
- Class definitions
- Supporting class libraries
  - Datatypes
  - Coding schemes
  - Framework - other Components
- Sequence Diagrams to support Component operations

Testing

- Component test
  - tester always invokes IGraphCollection
- Subsystem test
  - tester invokes external connectors
- System test
  - user testing

Integrating Commercial Components

- Adapter wraps the commercial component
- Normalized to the Framework
  - All communication is normalized around
    - Graph Collection operations
    - Template record definitions
- Example of Commercial product integration
  - Commercial MPI wrapped at one stage of the project
  - Actual VHA MPI is wrapped in the VHA Domain Person Graph Collection
  - Commercial TQS wrapped during Proof of Concept

Wrapping Legacy systems

- Wrap the Legacy communication inside responding GraphCollection(s)
- Map the Templates to Legacy information
- Develop the adapter to transform the protocol

Wrapping Database access

- Databases are wrapped by the appropriate Graph Collection
- The Cache is wrapped with a SQLCacheGraphCollection and knows how to implement getSelected, getNext
- Graphs are serialized and persisted

Component Network Support

- Next steps
  - Component remote communication method hidden by GraphCollectionMgr
  - Protocol aware Proxy and Façade pattern buffers application form communication platform
  - Mix and Match protocols between Domains
    - RMI
    - CORBA
    - SOAP
    - JMS

Enterprise Integration

- Framework Architecture
- Domain Model and Templates
- Adapters and Components
- Component Model and Collaborations
- Distributed Development Process
- Enterprise Architecture Control