Open Immunize (OpenIZ) Solutions Architecture

BETTER IMMUNIZATION MANAGEMENT TOOLING FOR THE WORLD …

THE RIGHT THING, DONE RIGHT

Version: 0.4

Solutions Architecture & System Design

- Overview
- The Architecture
  - System / solution architecture
  - Deployment
- Master Data Model
- Privacy & Security
- Application Integration
- Roadmap
- Final Thoughts

Solutions Architecture

- OpenIZ & MEDIC follow Disciplined Agile Delivery (DAD)
  - Combines Unified Process (UP) and Scrum methodology
- Having a comprehensive solutions architecture assists in coherent system design & implementation:
  - *“Thus it is that in war the victorious strategist only seeks battle after the victory has been won, whereas he who is destined to defeat first fights and afterwards looks for victory” – Sun Tzu*

Overview

AN OVERVIEW OF THE OPENIZ COMMUNITY

Overview

- Immunization data leads to better immunization adherence
- There are no robust, open source, immunization platforms
  - Commercially available solutions (STC, IBM, etc.) are available
  - Many public health authorities “roll their own”
- Interoperability and open architecture are often not the focus of these projects
  - Immunization data is a key part of a health record, should be shared
  - Non-generic solutions are difficult to adapt in different jurisdictions

Challenges

- OpenIZ will operate globally
  - Variation of requirements
  - Variation of technical capacity
  - Variation of localization (forecasting, language, code sets, etc.)
- Security and privacy laws
  - Will vary widely between types of jurisdictions

Our Team

- MEDIC – mHealth & eHealth Development and Innovation Centre
  - Based at Mohawk College in Hamilton, ON Canada
  - 1 senior solutions architect – Justin Fyfe
  - 1 senior director – Duane Bender
  - 1 project manager – Paul Brown
  - 3 senior developers –
    - Mohammed Ibrahim
    - Nityan Khanna
    - Garrett Tyler
  - 10-20 junior developers in lab at any time

Our History

- Founded in 2007 as Mohawk Applied Research Centre (MARC)
  - Mission: Prototype standards based enterprise architecture designed for use in Canada.
- Completed over 50 projects with government and private industry
- Completed and deployed projects in Canada, USA, Africa and Asia
- Privacy & Security sensitive deployments in Canada
  - Some of the toughest privacy requirements in the world
- Currently working with Ministry of Health Long Term Care (MOHLTC) to develop consumer and clinician facing Immunization reporting systems.

Our Mission

To provide an innovative, secure, sustainable, enterprise ready platforms based on open architecture, and open standards.

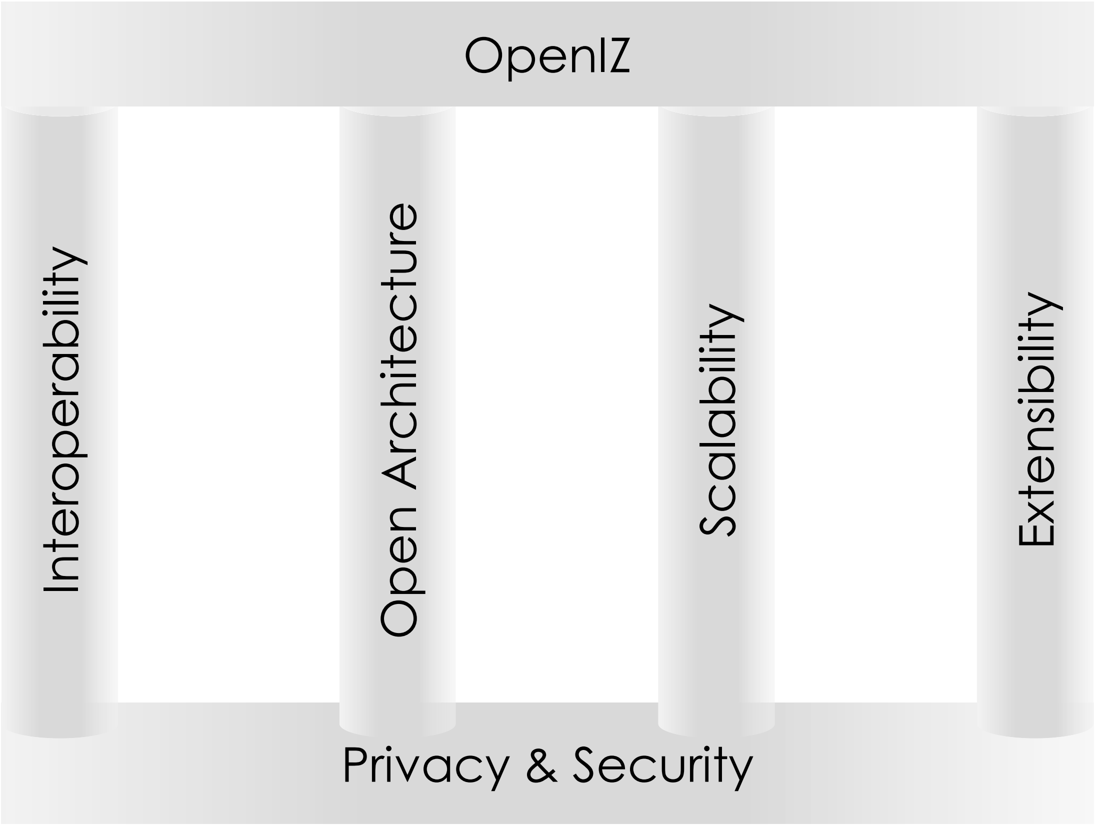
Our Capacity to build Capacity

- MEDIC is based in a College
  - Dedicated to knowledge and skills transfer in addition to software
  - Expertise and capacity to deliver post-secondary educational courseware
- Extend training to local developers
  - Have run detailed standards based training
  - Host knowledge transfer events at the college

How does OpenIZ meet this mission?

- Modular design
  - OpenIZ grows in scope and capacity as your deployment does
  - Extension points to allow easy customization of components
  - Mix and match best of breed open-source and commercial modules
- Open Architecture
  - All functionality is exposed via open interfaces
  - Leverages standards based interfaces
  - Integrates with HIE infrastructure

Pillars of OpenIZ



Architecture

A PLATFORM FOR:

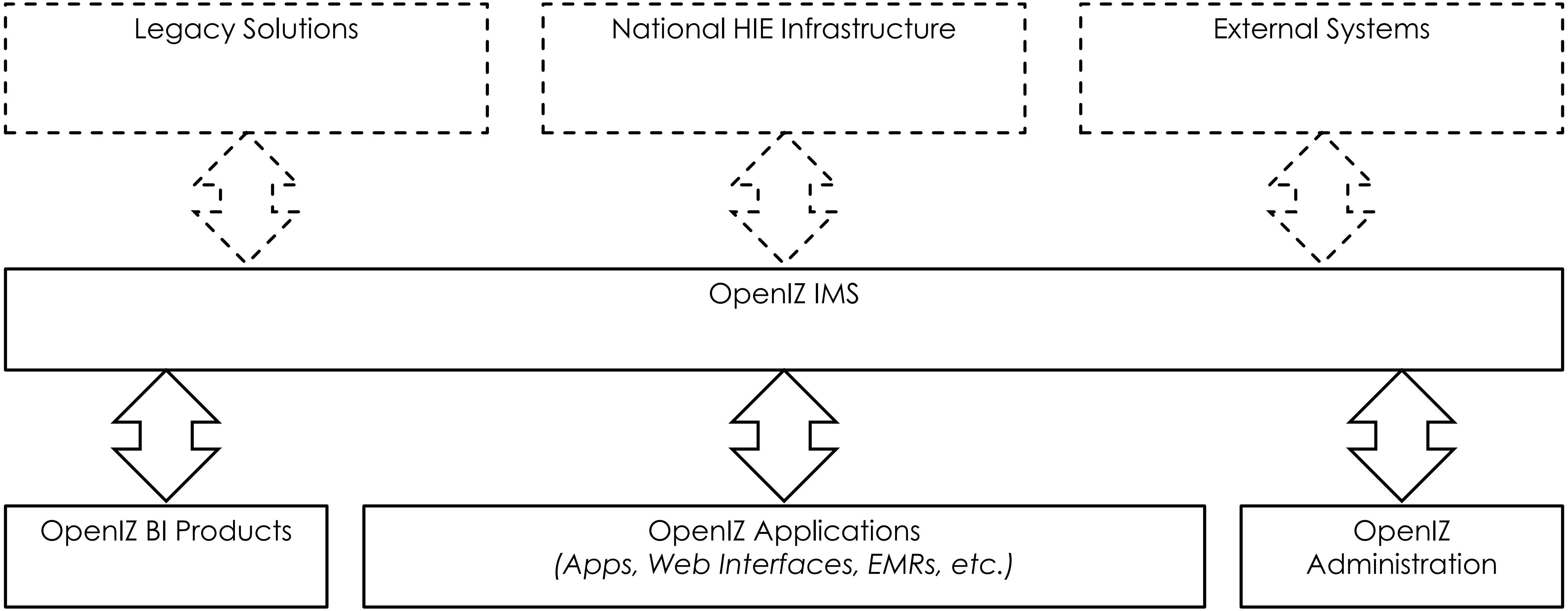
IMMUNIZATION

INNOVATION

INFORMATION

IMPLEMENTATION

The big picture



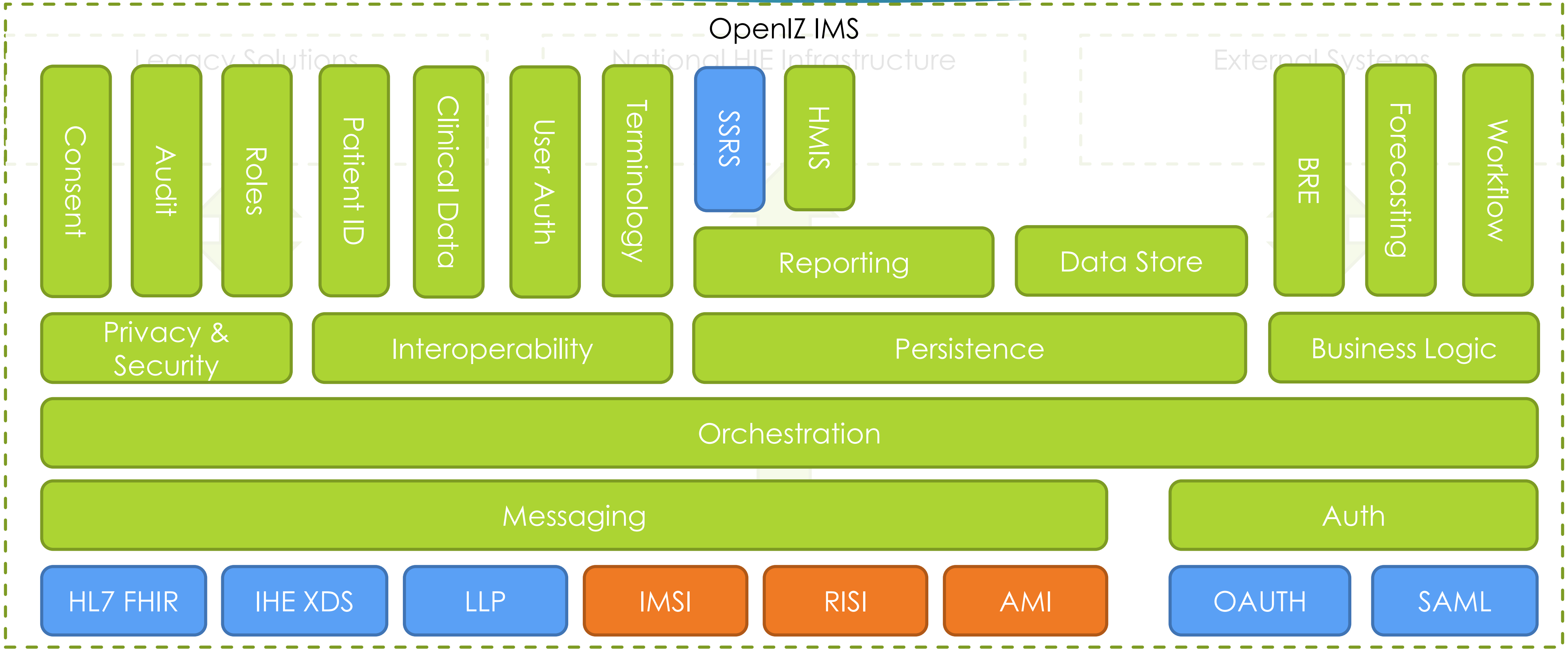
OpenIZ Applications

- The OpenIZ IMS has no user interface
  - Provides extension points “below the IMS”
  - OpenIZ tested apps can securely communicate with the backbone
- Control of Apps
  - Implementers may choose to allow OpenIZ tested apps are permitted to be used
    - By User Role, Action, Resource, etc.
- Can review all Point of Service applications prior to issuing certificates
- Examples:
  - A PHR connecting to retrieve forecasted immunizations
  - A mobile used by nurses to perform immunizations
  - A website used by patients to enter their immunizations

Immunization Management System

- Implemented as a services framework
  - Uses common micro-service pattern
  - Data model adheres to MVVM pattern
  - Extension events fired and can be subscribed by any module
- Highly customizable
  - Adapts to the environment
- Designed with implementers in mind
  - Extension points for:
    - Business rules based logic
    - Data lifecycle workflow
    - Interoperability to jurisdictional systems

Immunization Management System



OpenIZ IMS

Legacy Solutions

National HIE Infrastructure

External Systems

Consent

Audit

Roles

Patient ID

Clinical Data

User Auth

Terminology

SSRS

HMIS

Reporting

Data Store

BRE

Forecasting

Workflow

Privacy & Security

Interoperability

Persistence

Business Logic

Orchestration

Messaging

Auth

HL7 FHIR

IHE XDS

LLP

IMSI

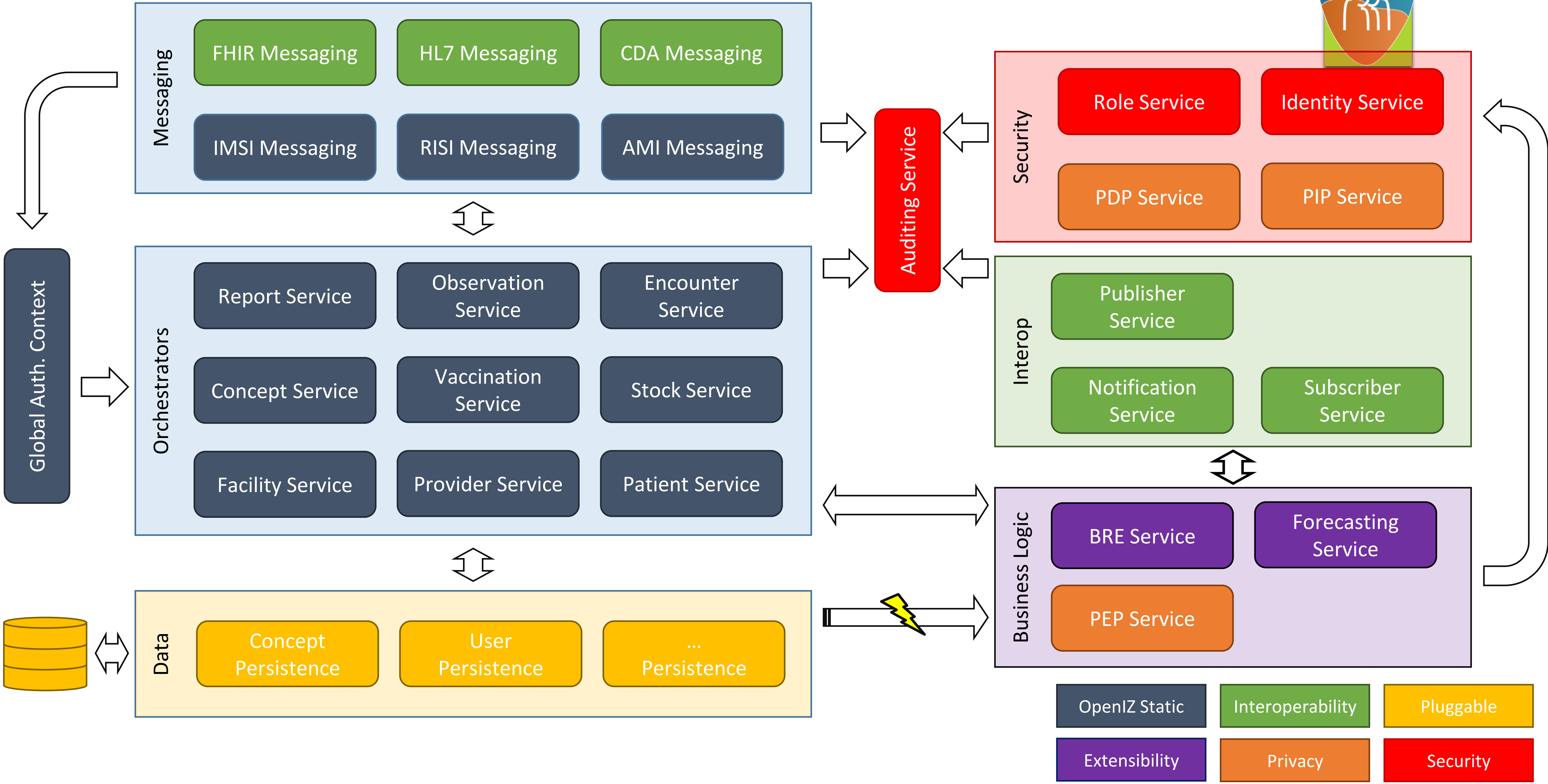
RISI

AMI

OAUTH

SAML

OpenIZ IMS Components



Messaging
- FHIR Messaging
- HL7 Messaging
- CDA Messaging
- IMSI Messaging
- RISI Messaging
- AMI Messaging

Global Auth. Context

Orchestrators
- Report Service
- Observation Service
- Encounter Service
- Concept Service
- Vaccination Service
- Stock Service
- Facility Service
- Provider Service
- Patient Service

Data
- Concept Persistence
- User Persistence
- ... Persistence

Auditing Service

Security
- Role Service
- Identity Service
- PDP Service
- PIP Service

Interop
- Publisher Service
- Notification Service
- Subscriber Service

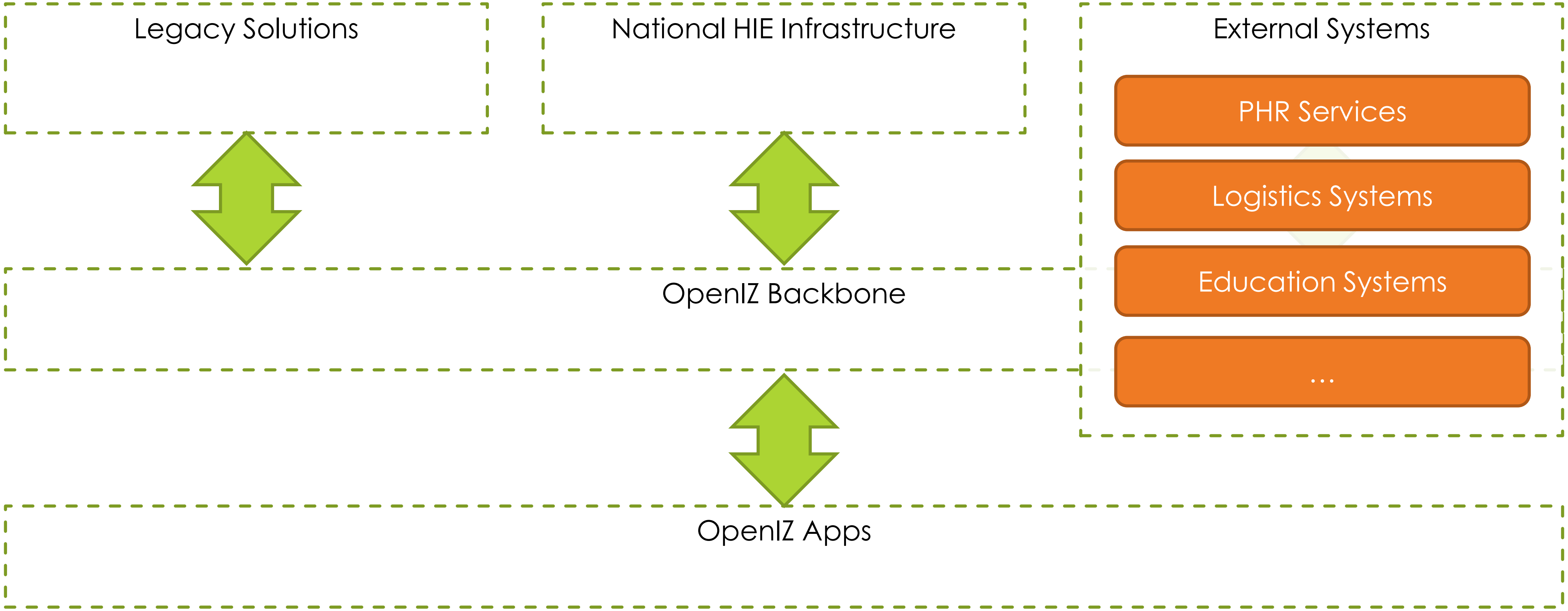
Business Logic
- BRE Service
- Forecasting Service
- PEP Service

OpenIZ Static	Interoperability	Pluggable
Extensibility	Privacy	Security

External Systems

- The Internet of Things
  - OpenIZ seeks to provide extension points to other medical and non-medical systems
    - Personal Health Services
    - Calendaring Systems (Outlook.com, Gmail, etc.)
    - Education Systems
    - Logistics Management Information Systems

External Systems



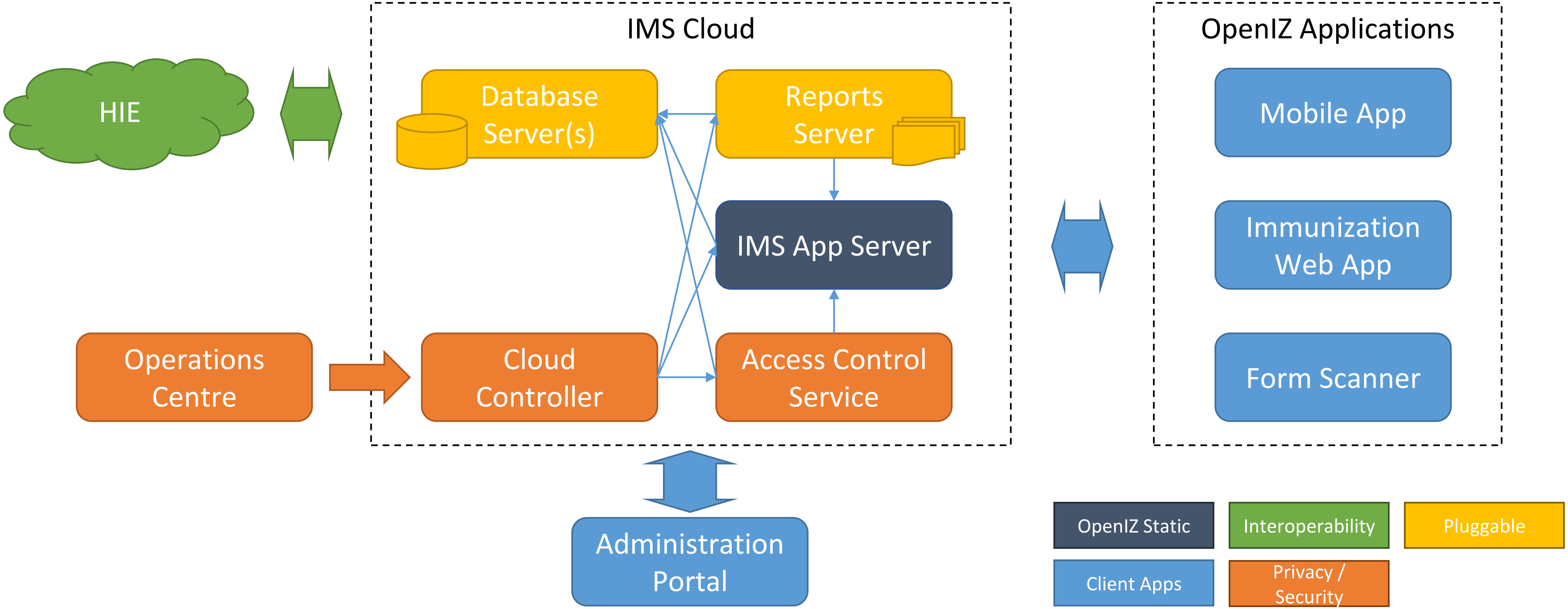
Deployment

DEPLOYMENT ARCHITECTURE

OpenIZ IMS Deployment

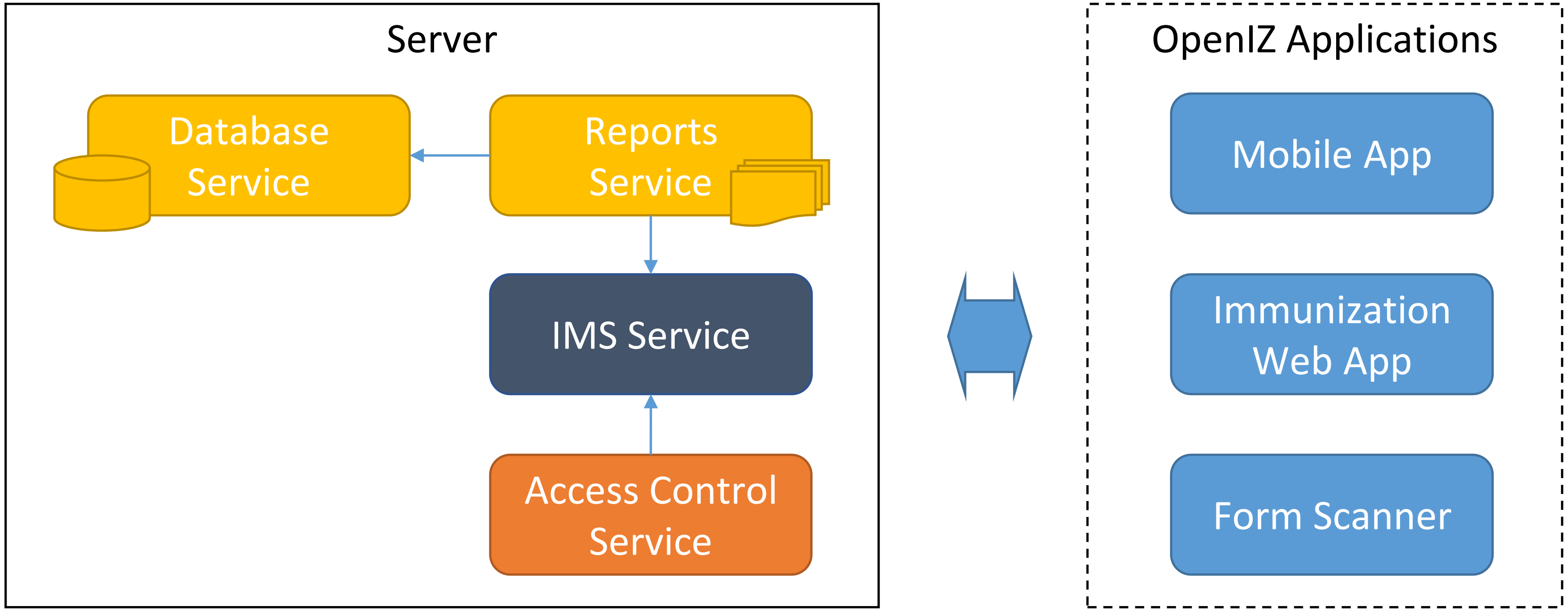
- A physically deployed IMS service may run all or a portion of services
  - Possible to balance the services over multiple servers
  - Specialize a server based on need
- Three types of deployment
  - Simple
    - All services run locally on one server
  - Cloud
    - Services run in a series of specialized host servers
    - Configuration & node authentication controlled by "cluster controller" node
  - Federated
    - Cluster with tiers for administrative subdivisions
    - Forecasted availability: OpenIZ Chippewa (Dec 2016) release

OpenIZ Deployment Pieces



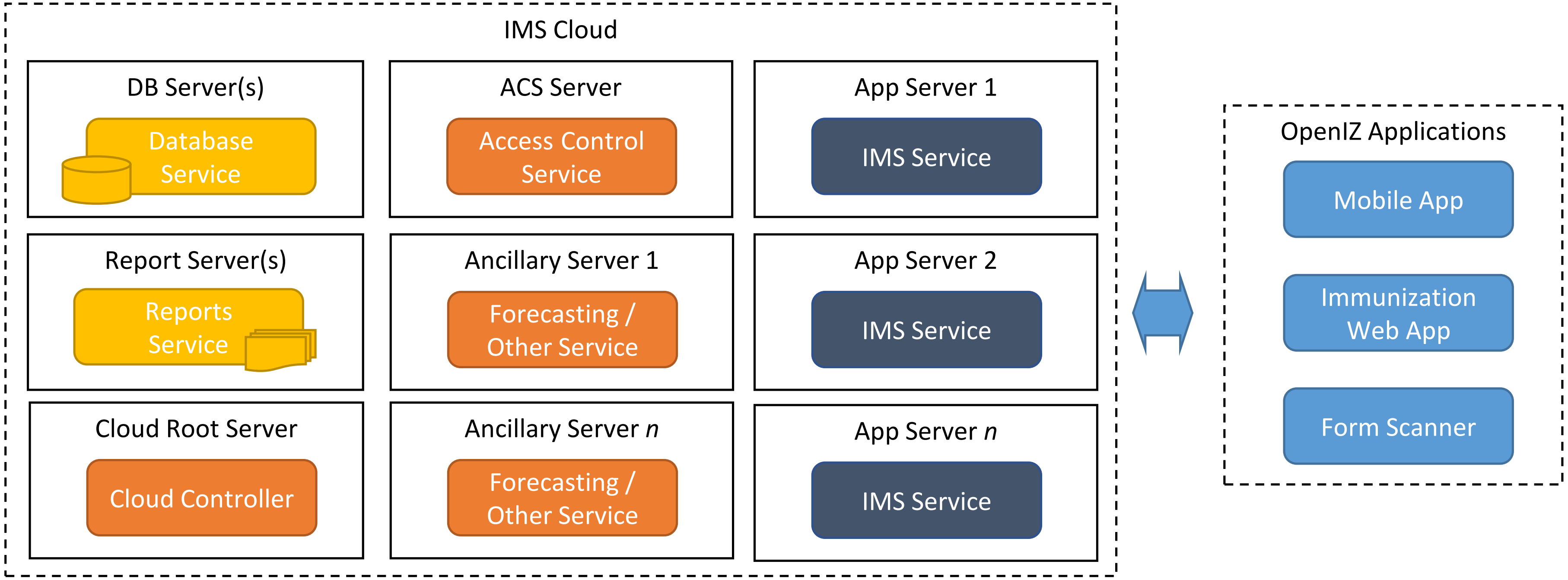
IMS Deployment : Simple

- Single server hosts all required IMS services

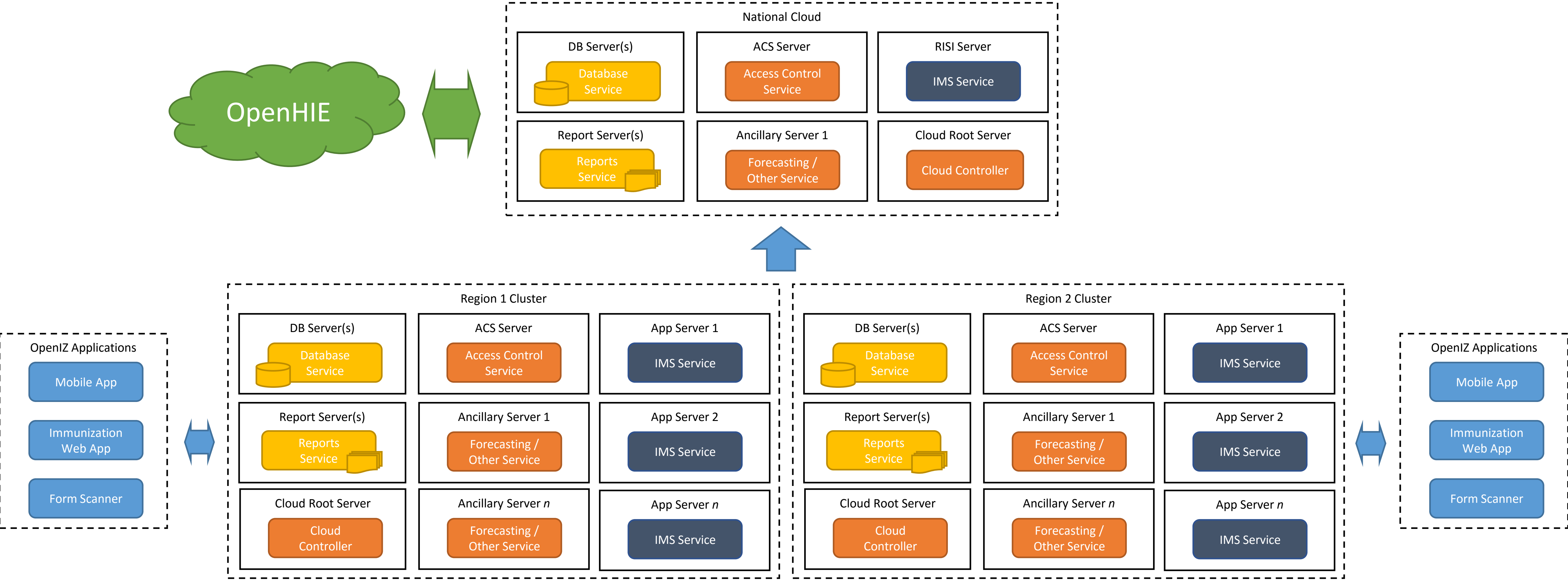


IMS Deployment : Cloud

- A series of servers are selected for particular roles
  - The role services must be independent and stateless



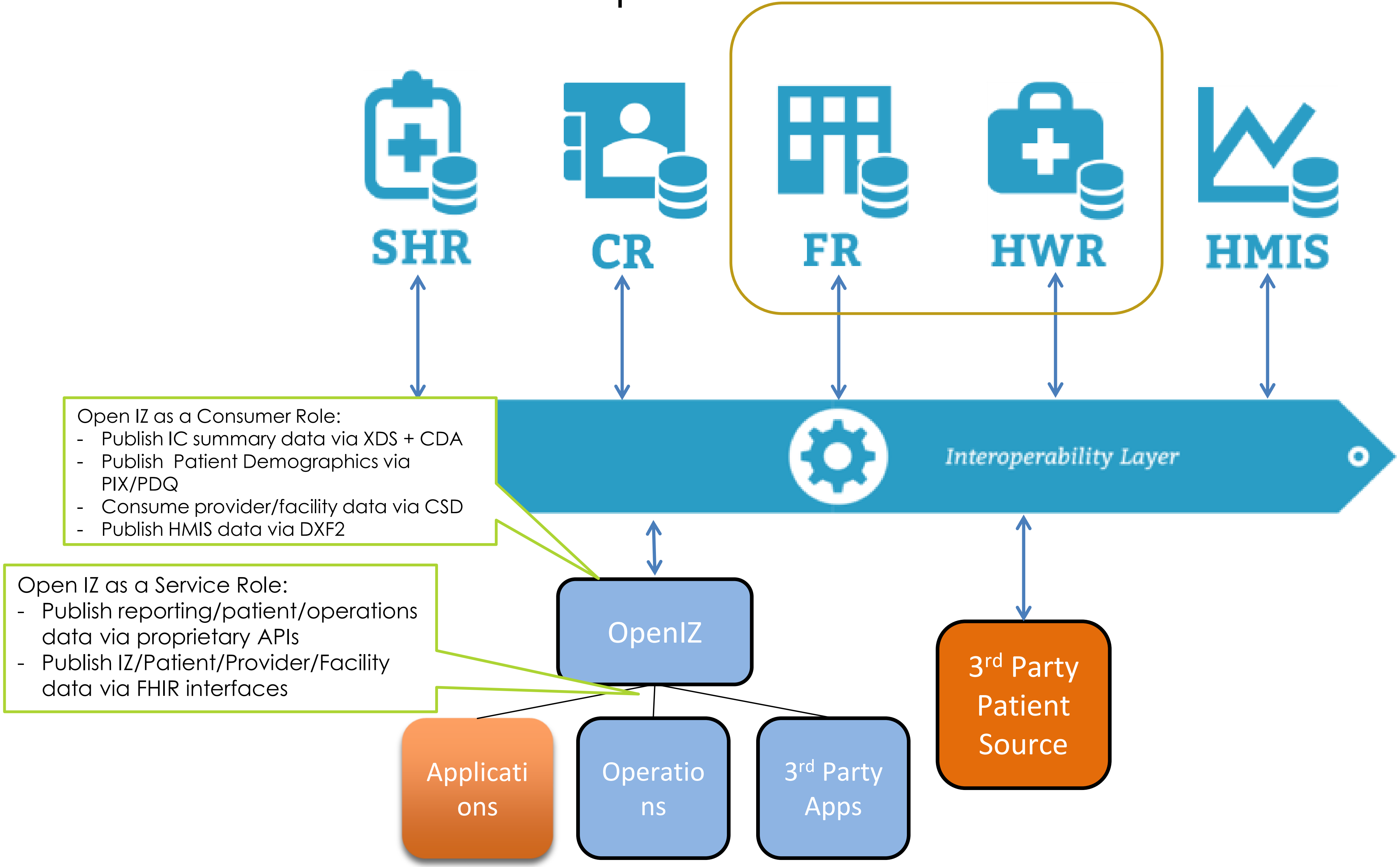
IMS Deployment : Federated



National Infrastructure

- OpenIZ integration with other infrastructure
  - OpenIZ as a consumer to HIEs (example: OpenHIE)
  - Standards based interfaces to regional or national infrastructure
- Clinician choice of sharing
  - OpenIZ will confirm and/or provide a confirmation of sharing prior to publishing PHI

National Infrastructure - OpenHIE



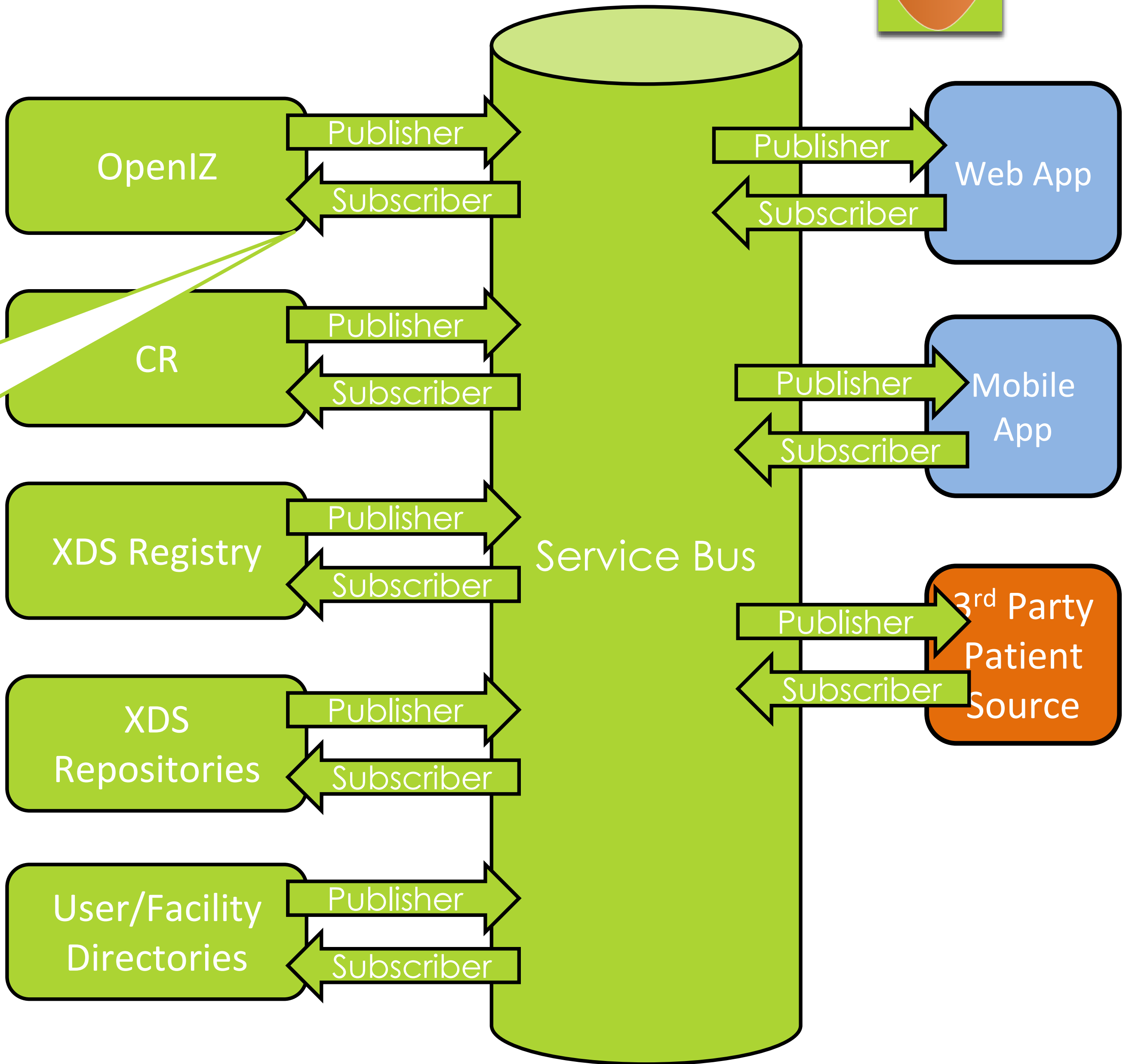
Open IZ as a Consumer Role:
- Publish IC summary data via XDS + CDA
- Publish Patient Demographics via PIX/PDQ
- Consume provider/facility data via CSD
- Publish HMIS data via DXF2

Open IZ as a Service Role:
- Publish reporting/patient/operations data via proprietary APIs
- Publish IZ/Patient/Provider/Facility data via FHIR interfaces

National Infrastructure – ESB HIE

- Open IZ has open architecture interfaces
  - Can act as a publisher of information from the LOB system
  - Can act as a subscriber of information from other LOB systems
- Mobile apps can integrate through the service bus as publisher/subscribers

Publisher: Patient Demographics, Immunization Summaries, Facility/Provider Information, HMIS, XDS Registry, Audit
Subscriber: Patient Demographics, Immunization Summaries, Facility/Provider Information



OpenIZ
Publisher
Subscriber

CR
Publisher
Subscriber

XDS Registry
Publisher
Subscriber

XDS Repositories
Publisher
Subscriber

User/Facility Directories
Publisher
Subscriber

Service Bus

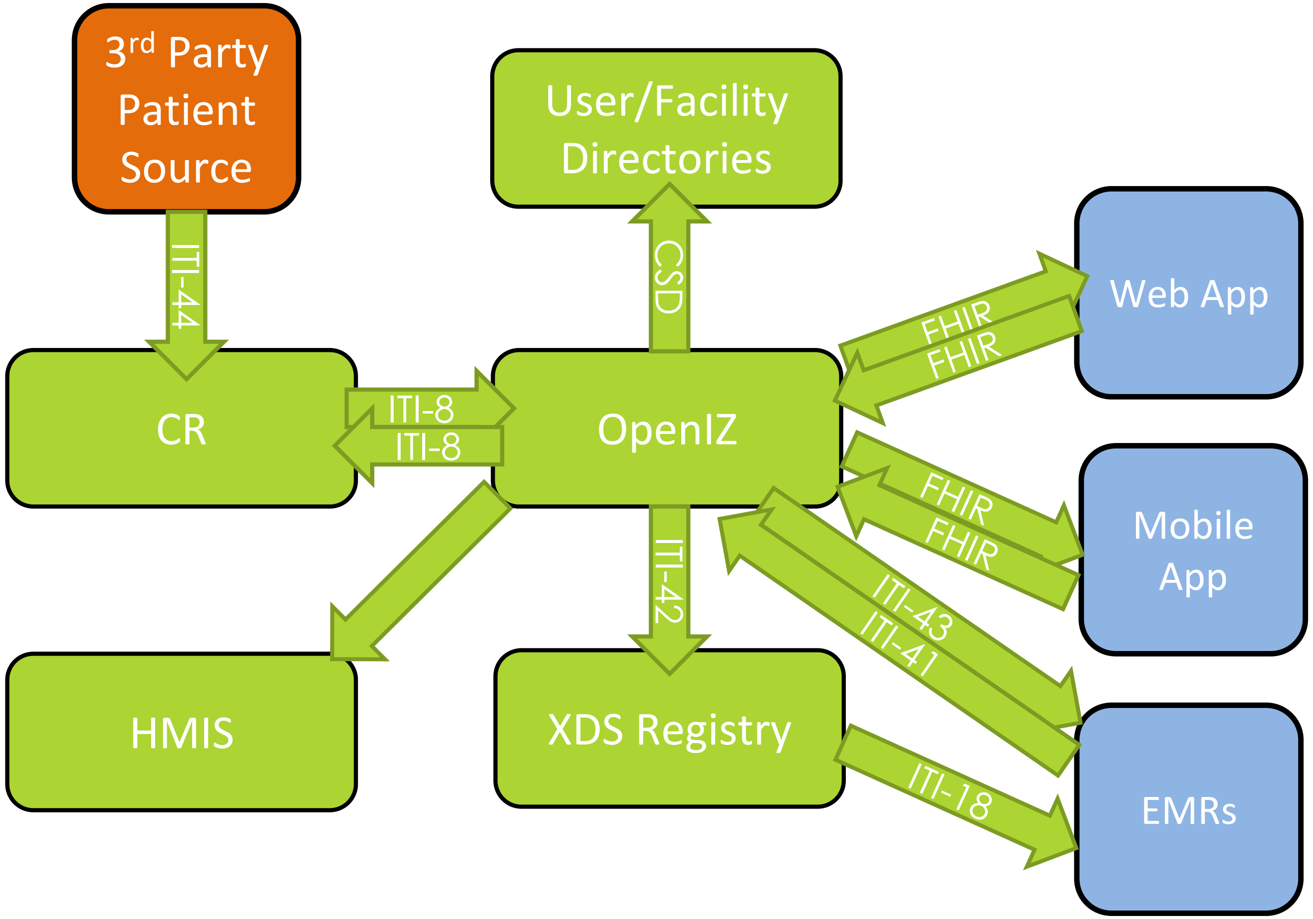
Web App
Publisher
Subscriber

Mobile App
Publisher
Subscriber

3rd Party Patient Source
Publisher
Subscriber

National Infrastructure – P2P HIE

- Open IZ has open architecture interfaces
  - Can act as an ODD query to/from third party EMR messages
  - FHIR interfaces and other proprietary interfaces still available for web/mobile apps
  - Interfaces to/from other supporting systems via standards based interfaces
- Same interfaces and code as used in OpenHIE and BUS based HIE



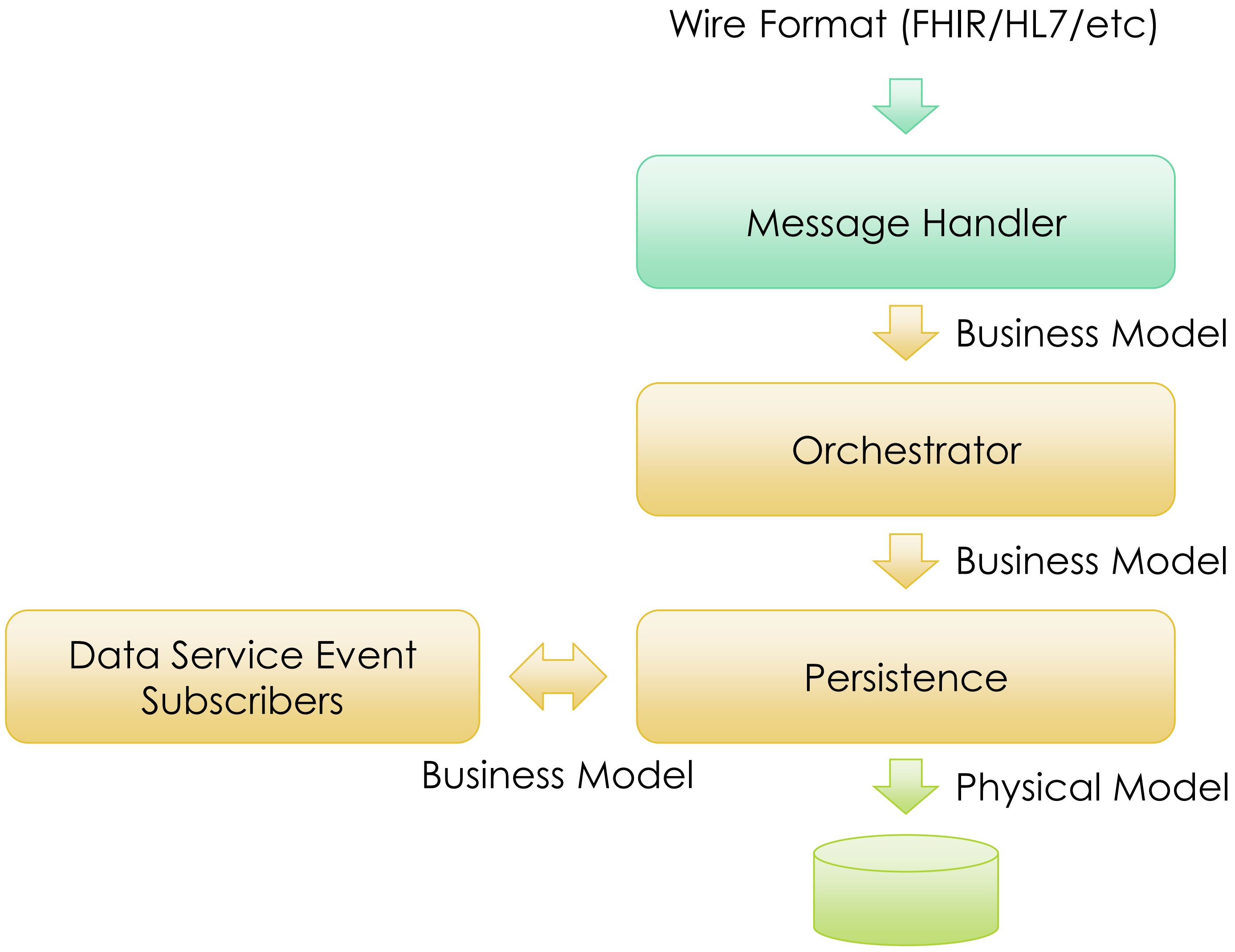
IMS Data Model

A MASTER DATA MODEL SUPPORTING A VARIETY OF JURISDICTIONAL REQUIREMENTS

Concepts of the Data Model

- OpenIZ sports a robust data model
- Consists of two layers:
  - Business Data Model – A model which uses classes to express concepts
  - Physical Data Model – Realization of the logical RDBMS model.
- Broken into 5 sub-categories
  - Privacy & Security
  - Clinical Data
  - Stock Management
  - Protocol / Workflow
  - Concept Dictionary

From Business Model to MDM



Wire Format (FHIR/HL7/etc)

Message Handler

Business Model

Orchestrator

Business Model

Data Service Event Subscribers

Persistence

Business Model

Physical Model

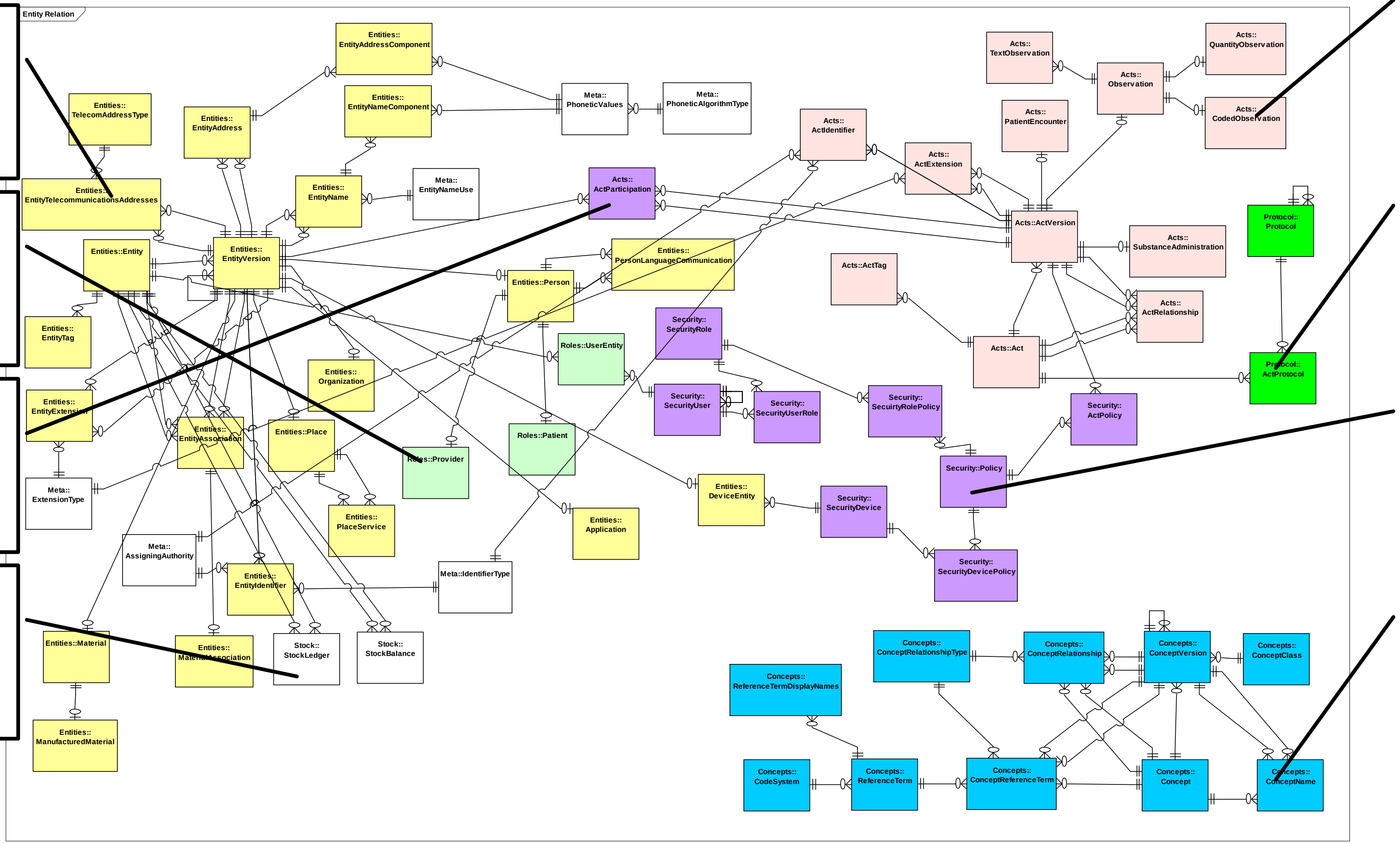
OpenIZ Master Data Model (MDM)

Clinical: Entities

Clinical: Roles

Clinical: Participation

Stock Management



Clinical: Acts

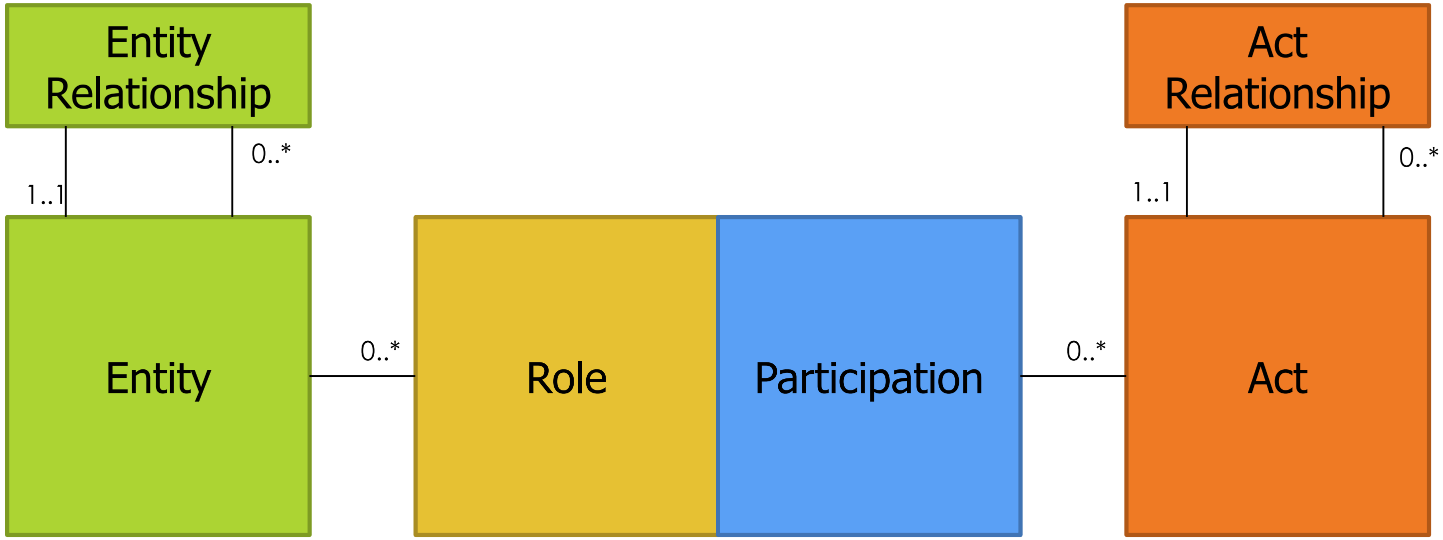
Protocol / Workflow

Security / Privacy

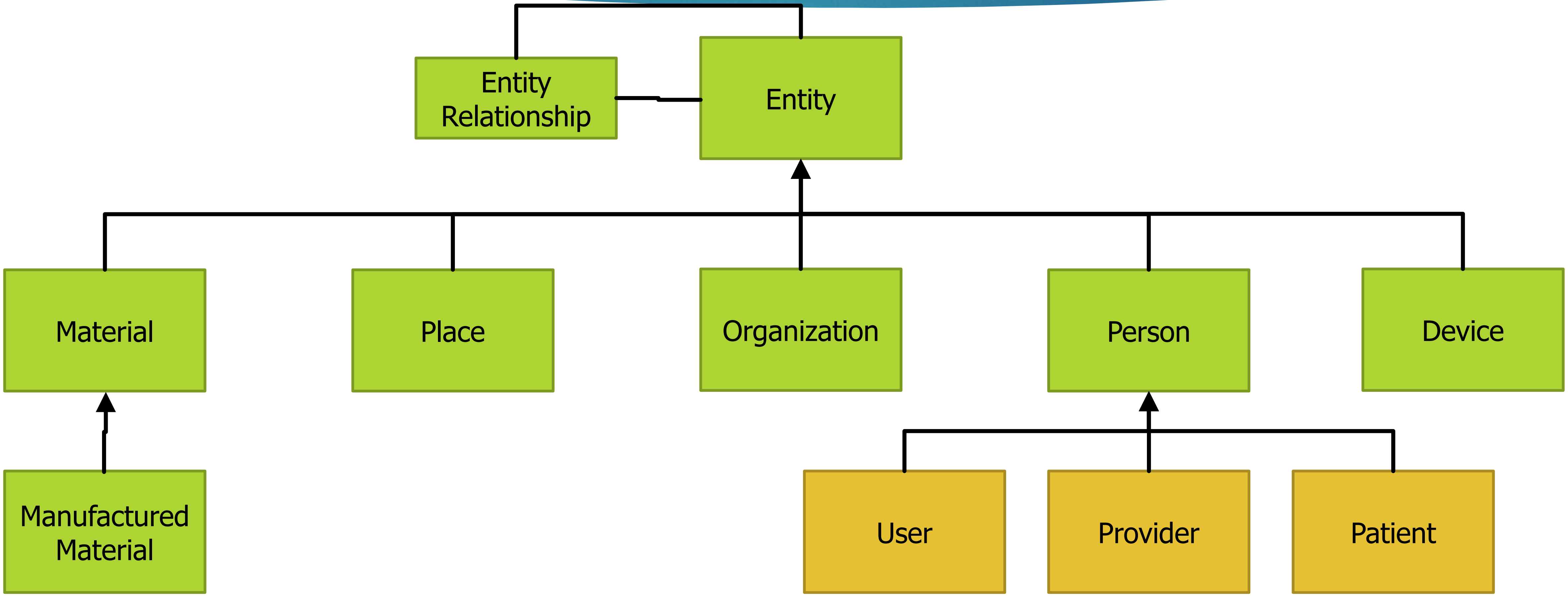
Concepts

Clinical Tables

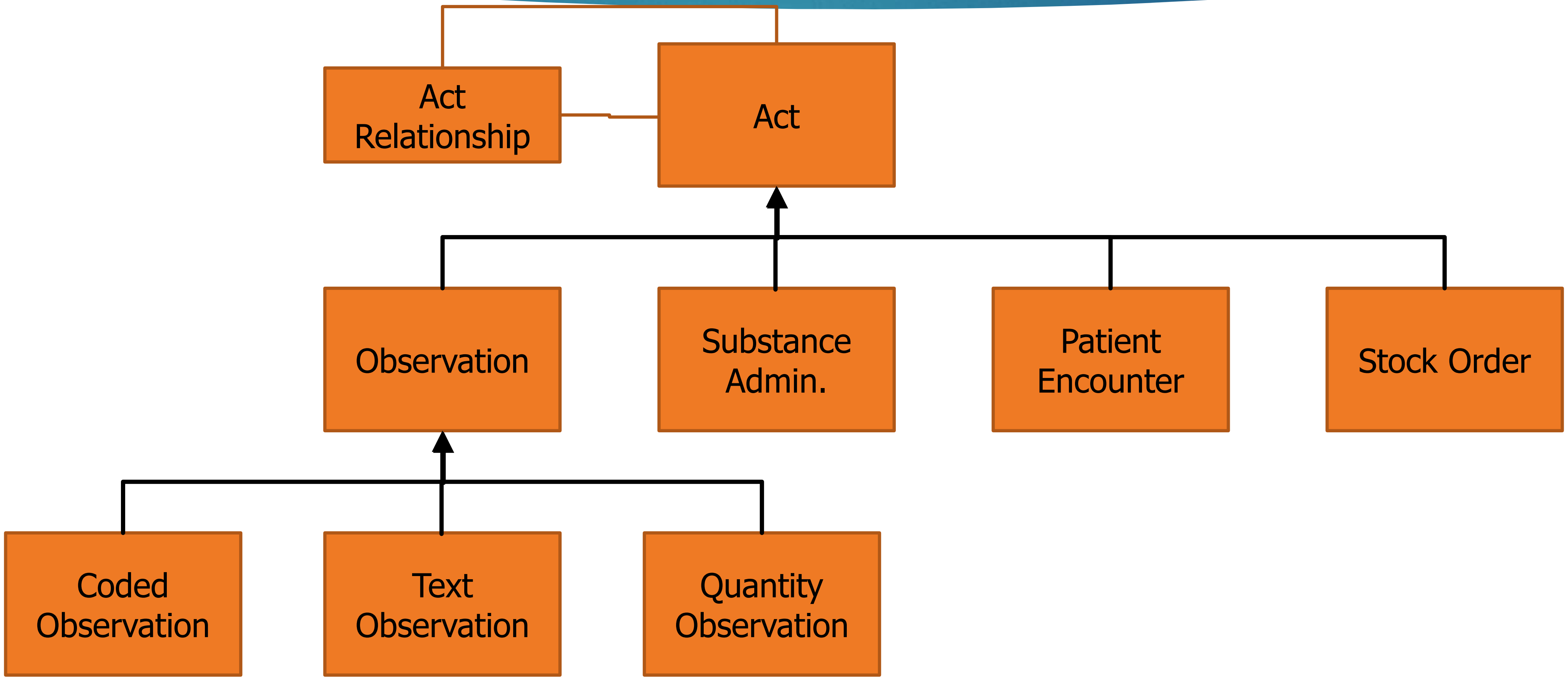
- Clinical data model is loosely based on the HL7 RIM
  - Entities playing roles participating in acts.
  - Complexities are handled by the API layer



Clinical Tables: Entities



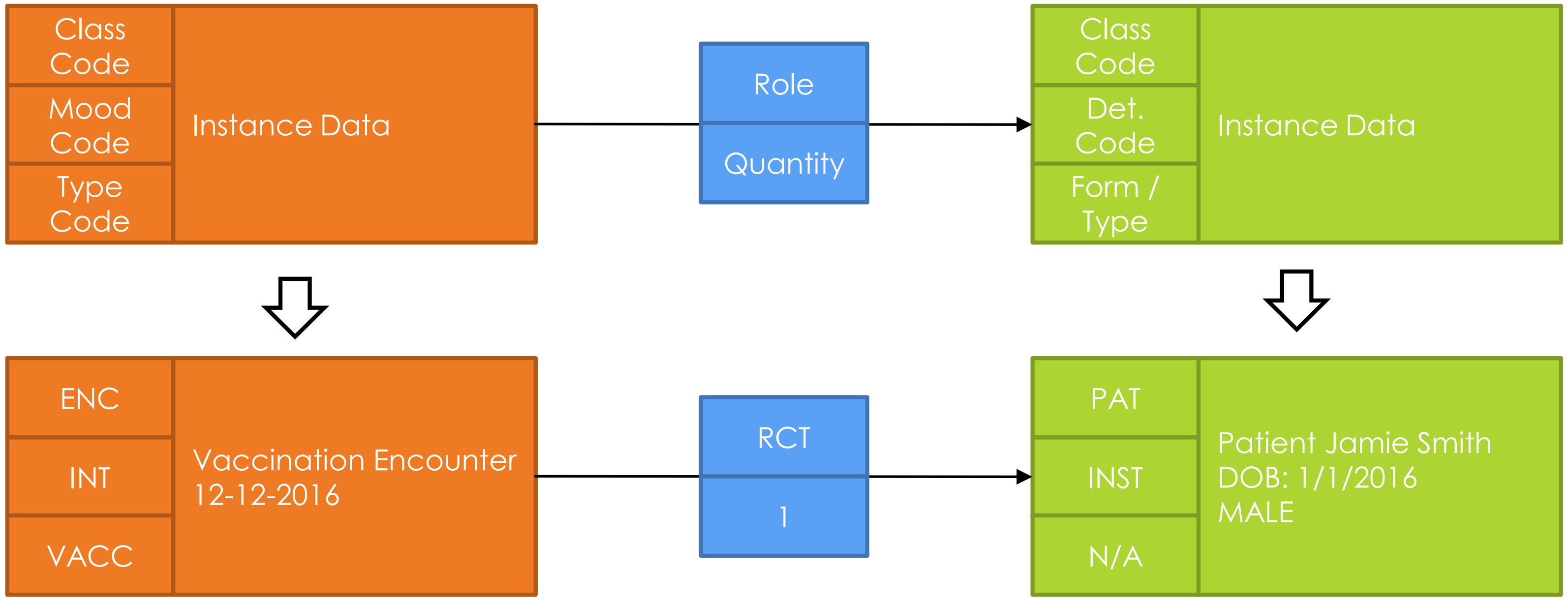
Clinical Tables: Acts



Example: Immunizing A Child

- A child *[entity]* presents to a clinic *[entity]* appointment *[act]* to receive their scheduled vaccinations. A clinician *[entity]* administers *[act]* each vaccine *[entity]*. The clinician then observes *[act]* a rash.
  - In this scenario an appointment *[Act: ENC, INT]* is setup between a clinic *[Entity: PLC, SDLOC]* and the patient (child) *[Entity: PAT, RCT]*.
  - At a later time an encounter *[Act: ENC, EVN]* is created which fulfills the appointment. The patient and clinic as well as clinician *[Entity: ASSIGNED, PRF]* are associated.
    - The act contains several substance administrations *[Act: SBADM, EVN, IMMUNIZE]* of vaccines *[Entity: MMAT, SUBJ]* are performed and linked to the encounter *[COMP]*
    - The coded observation *[Act: PROB, EVN, SYMPTOM]* of a rash is linked to the encounter *[CAUS]* as well.
      - If a severity is reported then it is recorded as an obs *[Act: OBS, EVN, SEVERITY]* and linked to the reaction *[SEV]*
    - If the clinician wishes to report an allergy, then a concern is created *[Act: OBS, EVN, DALG]* and the reaction is linked.

Information Modeling - Cards



Acts

Class Code

Mood Code

Type Code

Instance Data

Role

Quantity

Entities

Class Code

Det. Code

Form / Type

Instance Data

ENC

INT

VACC

Vaccination Encounter
12-12-2016

RCT

1

PAT

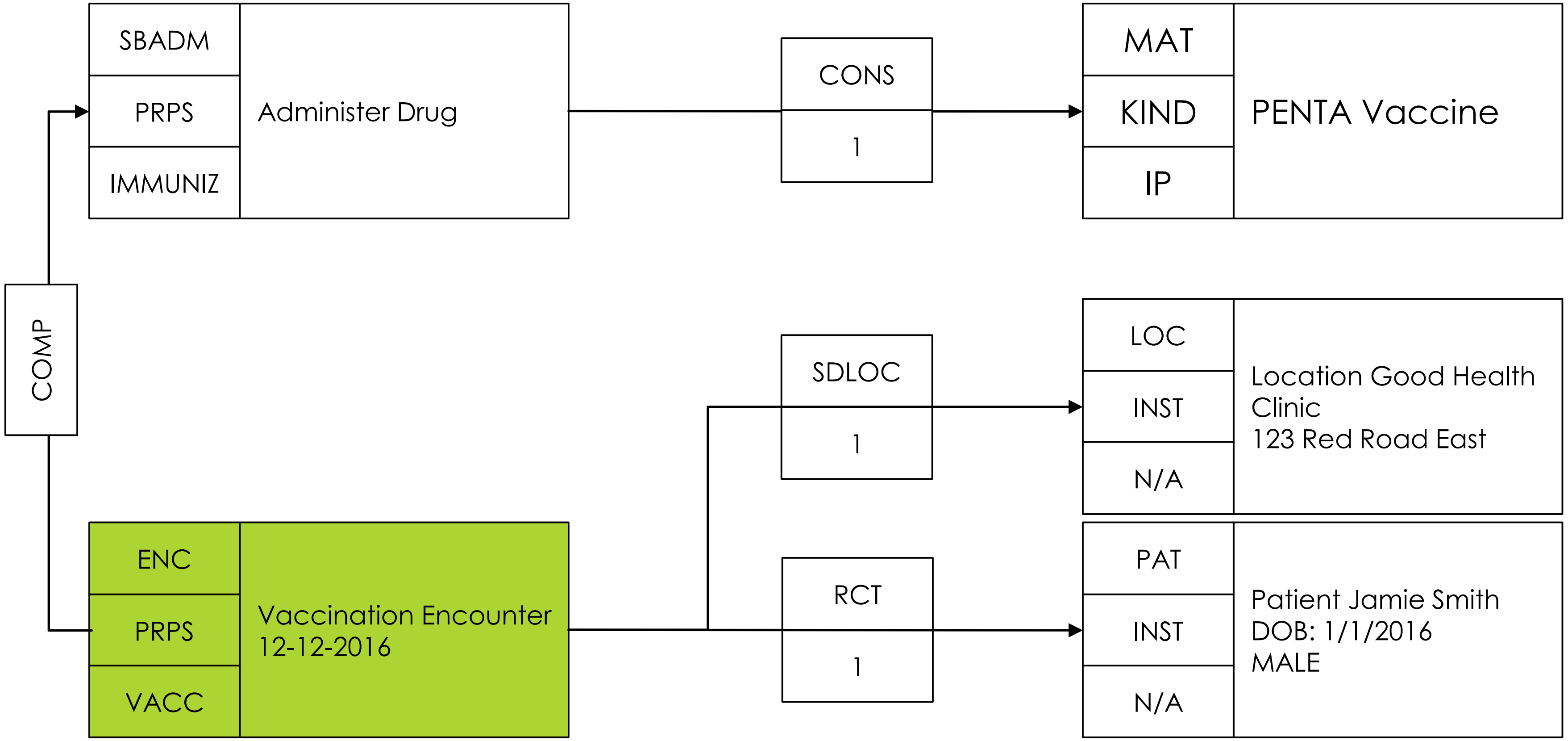
INST

N/A

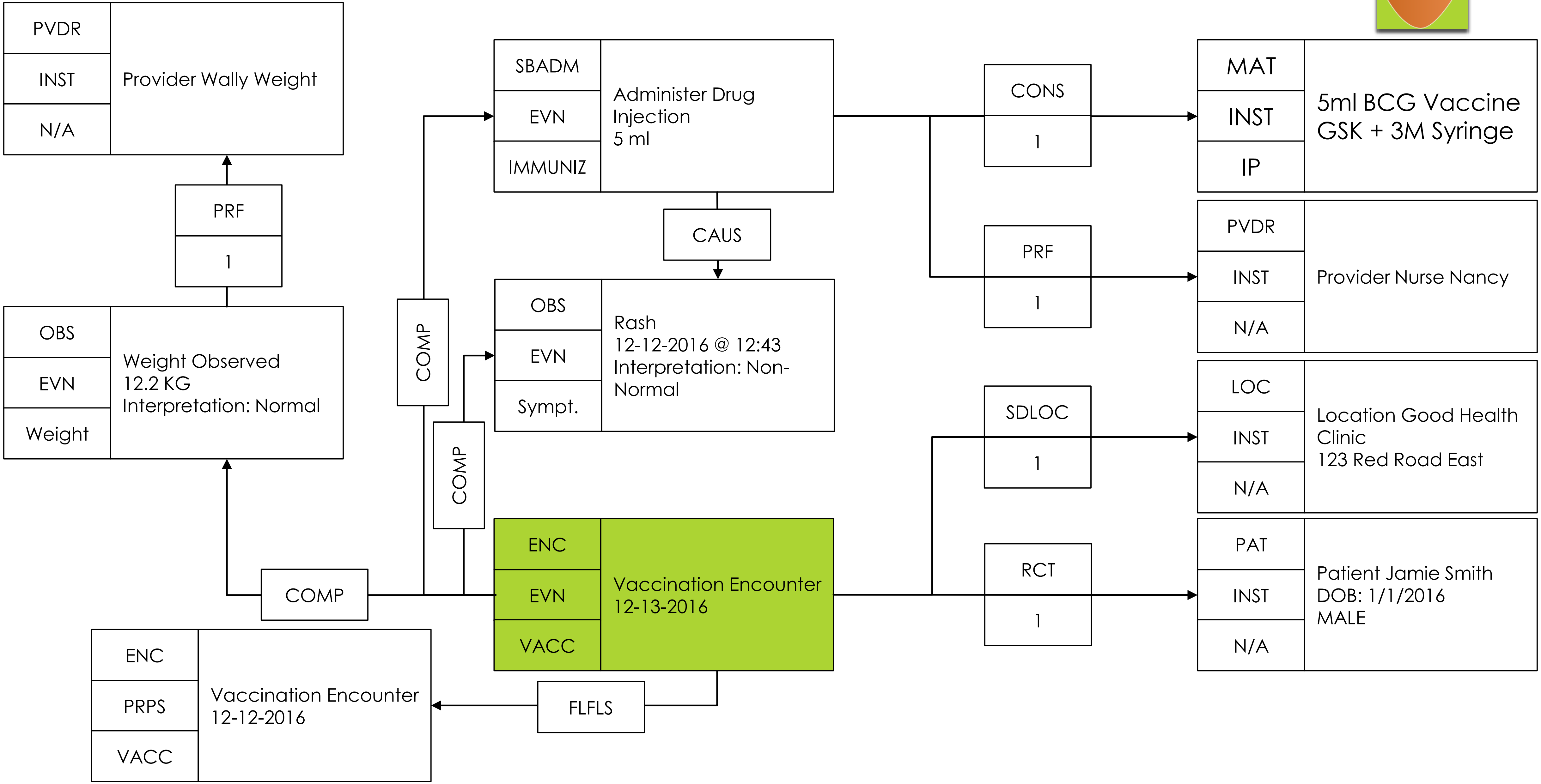
Patient Jamie Smith
DOB: 1/1/2016
MALE

Read: Intend to have an encounter with patient Jamie Smith

Modeling an Appointment from Forecaster



Modeling Vaccine + Weight + AEFI



PVDR
INST
N/A
Provider Wally Weight

PRF
1

OBS
EVN
Weight
Weight Observed
12.2 KG
Interpretation: Normal

COMP

ENC
PRPS
VACC
Vaccination Encounter
12-12-2016

FLFLS

COMP

COMP

SBADM
EVN
IMMUNIZ
Administer Drug
Injection
5 ml

CAUS

OBS
EVN
Sympt.
Rash
12-12-2016 @ 12:43
Interpretation: Non-Normal

ENC
EVN
VACC
Vaccination Encounter
12-13-2016

CONS
1

PRF
1

SDLOC
1

RCT
1

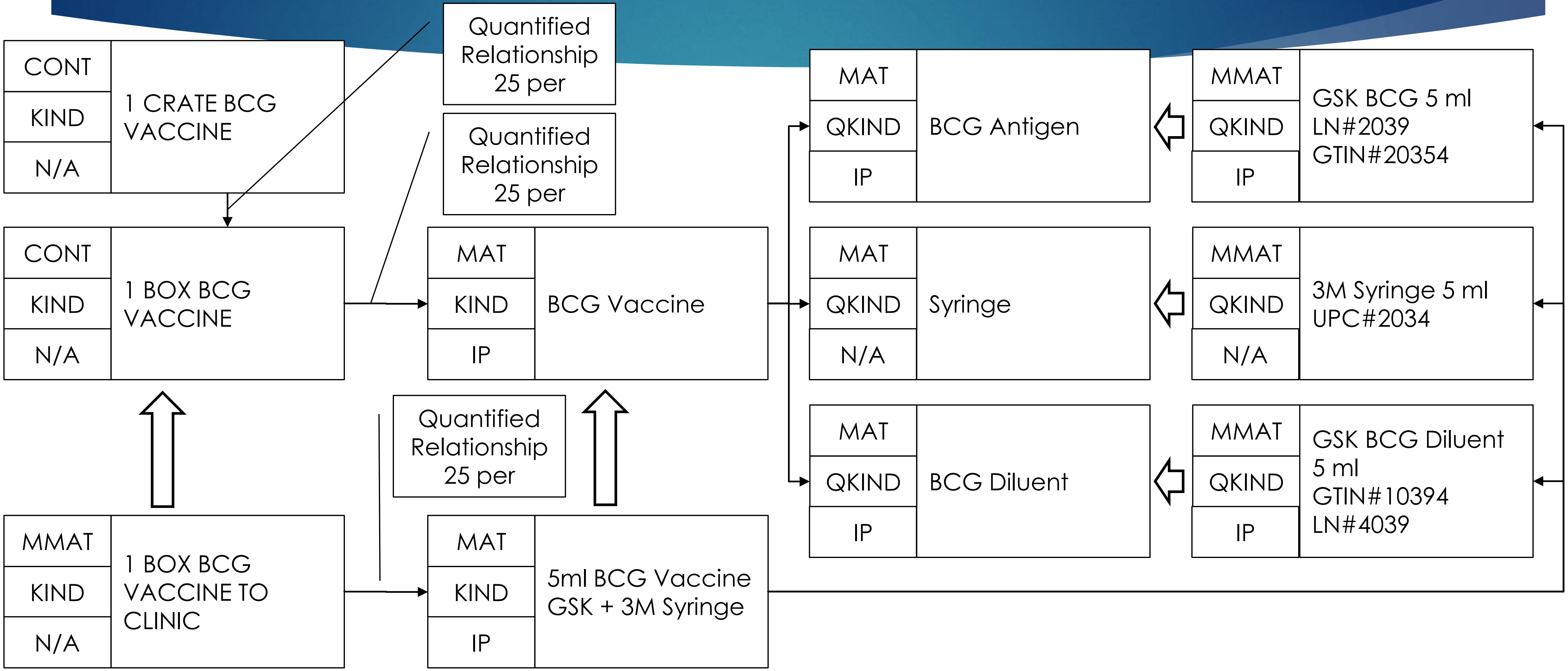
MAT
INST
IP
5ml BCG Vaccine
GSK + 3M Syringe

PVDR
INST
N/A
Provider Nurse Nancy

LOC
INST
N/A
Location Good Health
Clinic
123 Red Road East

PAT
INST
N/A
Patient Jamie Smith
DOB: 1/1/2016
MALE

Antigens/Manufacturers/Kits



CONT | KIND | N/A — 1 CRATE BCG VACCINE

Quantified Relationship 25 per

CONT | KIND | N/A — 1 BOX BCG VACCINE

Quantified Relationship 25 per

MAT | KIND | IP — BCG Vaccine

MMAT | KIND | N/A — 1 BOX BCG VACCINE TO CLINIC

Quantified Relationship 25 per

MAT | KIND | IP — 5ml BCG Vaccine GSK + 3M Syringe

MAT | QKIND | IP — BCG Antigen

MAT | QKIND | N/A — Syringe

MAT | QKIND | IP — BCG Diluent

MMAT | QKIND | IP — GSK BCG 5 ml LN#2039 GTIN#20354

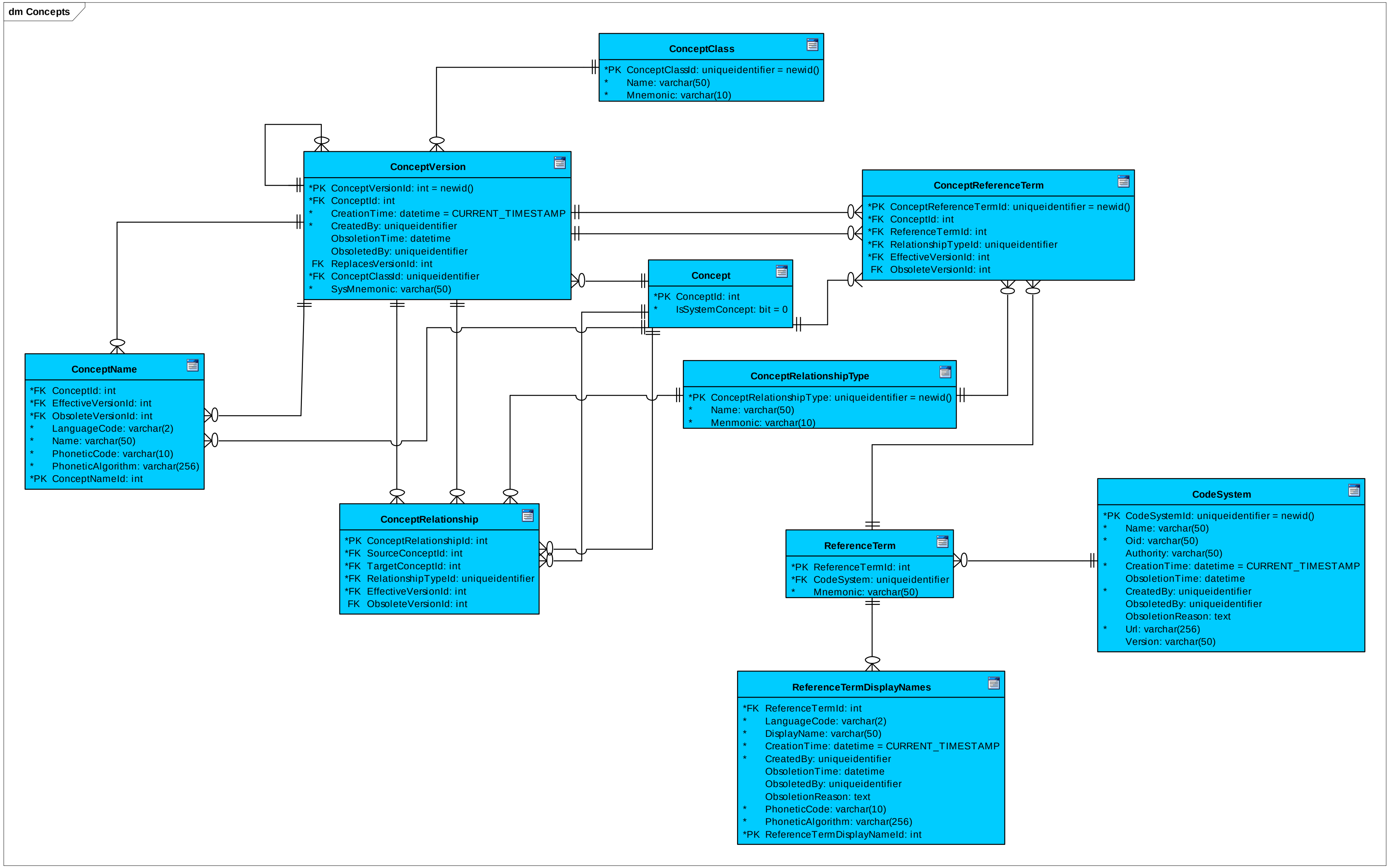
MMAT | QKIND | N/A — 3M Syringe 5 ml UPC#2034

MMAT | QKIND | IP — GSK BCG Diluent 5 ml GTIN#10394 LN#4039

Concept Dictionary

- OpenIZ uses a canonical concept dictionary
  - A unique list of concepts which can be used in the OpenIZ clinical tables
  - Curated by a jurisdiction prior to deployment or during deployment
- OpenIZ can translate
  - Reference Terms are wire-level representations of OpenIZ concepts.
  - Allows jurisdictions to select which code-systems are used
- OpenIZ can validate
  - Validates concepts based on concept classifications and concept set members
- Interoperability roadmap
  - Can integrate with external systems via ITerminologyService implementations
    - HL7 CTS 1.2
    - HL7 FHIR

Concept Dictionary



ConceptClass

*PK ConceptClassId: uniqueidentifier = newid()
* Name: varchar(50)
* Mnemonic: varchar(10)

ConceptVersion

*PK ConceptVersionId: int = newid()
*FK ConceptId: int
* CreationTime: datetime = CURRENT_TIMESTAMP
* CreatedBy: uniqueidentifier
ObsoletionTime: datetime
ObsoletedBy: uniqueidentifier
FK ReplacesVersionId: int
*FK ConceptClassId: uniqueidentifier
* SysMnemonic: varchar(50)

ConceptReferenceTerm

*PK ConceptReferenceTermId: uniqueidentifier = newid()
*FK ConceptId: int
*FK ReferenceTermId: int
*FK RelationshipTypeId: uniqueidentifier
*FK EffectiveVersionId: int
FK ObsoleteVersionId: int

Concept

*PK ConceptId: int
* IsSystemConcept: bit = 0

ConceptName

*FK ConceptId: int
*FK EffectiveVersionId: int
*FK ObsoleteVersionId: int
* LanguageCode: varchar(2)
* Name: varchar(50)
* PhoneticCode: varchar(10)
* PhoneticAlgorithm: varchar(256)
*PK ConceptNameId: int

ConceptRelationshipType

*PK ConceptRelationshipType: uniqueidentifier = newid()
* Name: varchar(50)
* Menmonic: varchar(10)

ConceptRelationship

*PK ConceptRelationshipId: int
*FK SourceConceptId: int
*FK TargetConceptId: int
*FK RelationshipTypeId: uniqueidentifier
*FK EffectiveVersionId: int
FK ObsoleteVersionId: int

ReferenceTerm

*PK ReferenceTermId: int
*FK CodeSystem: uniqueidentifier
* Mnemonic: varchar(50)

CodeSystem

*PK CodeSystemId: uniqueidentifier = newid()
* Name: varchar(50)
* Oid: varchar(50)
Authority: varchar(50)
* CreationTime: datetime = CURRENT_TIMESTAMP
ObsoletionTime: datetime
* CreatedBy: uniqueidentifier
ObsoletedBy: uniqueidentifier
ObsoletionReason: text
* Url: varchar(256)
Version: varchar(50)

ReferenceTermDisplayNames

*FK ReferenceTermId: int
* LanguageCode: varchar(2)
* DisplayName: varchar(50)
* CreationTime: datetime = CURRENT_TIMESTAMP
* CreatedBy: uniqueidentifier
ObsoletionTime: datetime
ObsoletedBy: uniqueidentifier
ObsoletionReason: text
* PhoneticCode: varchar(10)
* PhoneticAlgorithm: varchar(256)
*PK ReferenceTermDisplayNameId: int

Protocol & Workflow

- Business Rules Engine
  - Supports extensible business rules written in ECMA Script (IMS default is JScript)
  - Example business rule: Whenever a material is administered, update stock ledger.
  - JScript triggers interact with event model from the OpenIZ IMS event handlers.
- Forecasting Engine
  - Inference Engine: Rules are applied to Facts to make Conclusions
  - Example Rules:
    - Don't give live attenuated vaccines to immunosuppressed individuals
    - OPV must be given at birth, 6 weeks, …
  - Example Facts:
    - HIV is a immunosuppressed condition
    - OPV is a live attenuated virus
    - Baby Simba has HIV
  - Conclusions:
    - Baby Simba should not receive OPV doses

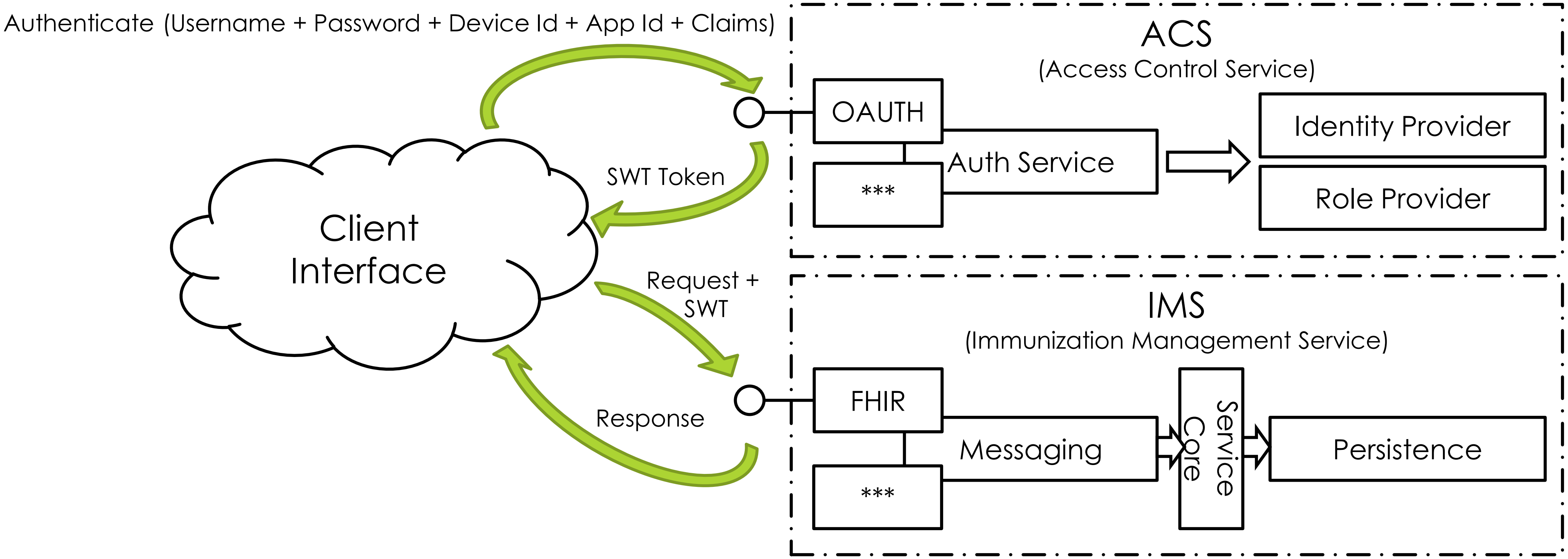
Security & Privacy

PRIVACY BY DESIGN – OPENIZ INTEGRATES PSA INTO THE FABRIC OF THE SOLUTION

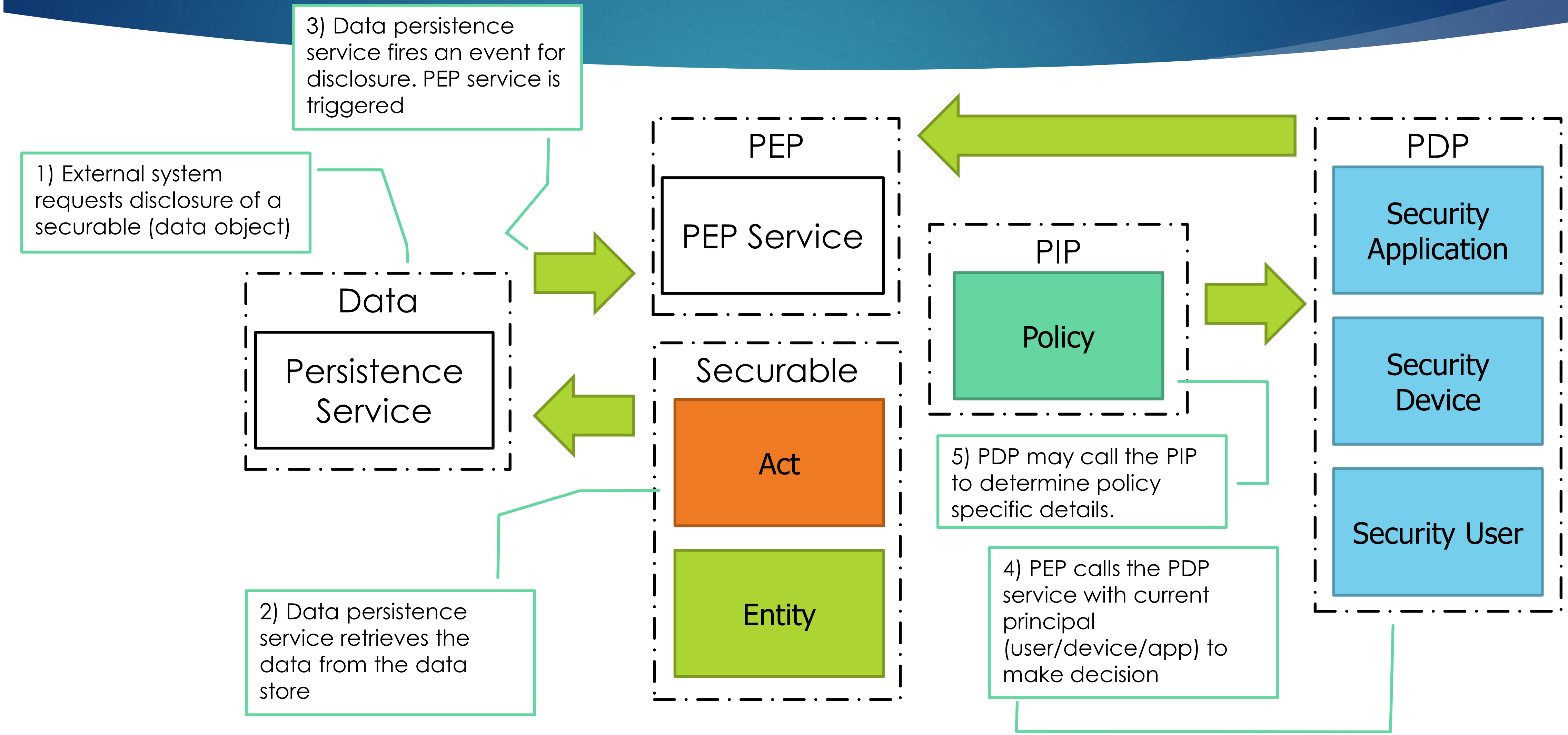
Security / Policy System

- Authentication / Authorization
  - Uses claims based authentication
- Policies
  - Policy Information – Stored in Policy classes based on Policy table
  - Policy Decision – Handled by querying associative entities to Role/Device/Application tables.
  - Policy Enforcement – Decision from built-in PDP is sent to IPolicyEnforcement service for action.
- Interoperability Profile roadmap
  - Security
    - OAUTH
    - BASIC Auth
  - Policy / ACL
    - XACML

Authentication Architecture



Security/Policy System



1) External system requests disclosure of a securable (data object)

2) Data persistence service retrieves the data from the data store

3) Data persistence service fires an event for disclosure. PEP service is triggered

4) PEP calls the PDP service with current principal (user/device/app) to make decision

5) PDP may call the PIP to determine policy specific details.

Data

Persistence Service

PEP

PEP Service

Securable

Act

Entity

PIP

Policy

PDP

Security Application

Security Device

Security User

Security Validation

- Embedded right into the .NET assemblies which make up OpenIZ
    - Uses CAS (Code Access Security) in both declarative and imperative ways
- Policies are customizable – you can extend policy behavior
- OpenIZ has built in policies for:
    - Login
    - Administrative functions – User management & password change
    - Clinical Functions – Write, Read, Delete, Query

IMS Applications

PUTTING A USER INTERFACE ON THE IMS

OpenIZ IMS vs. OpenIZ Apps

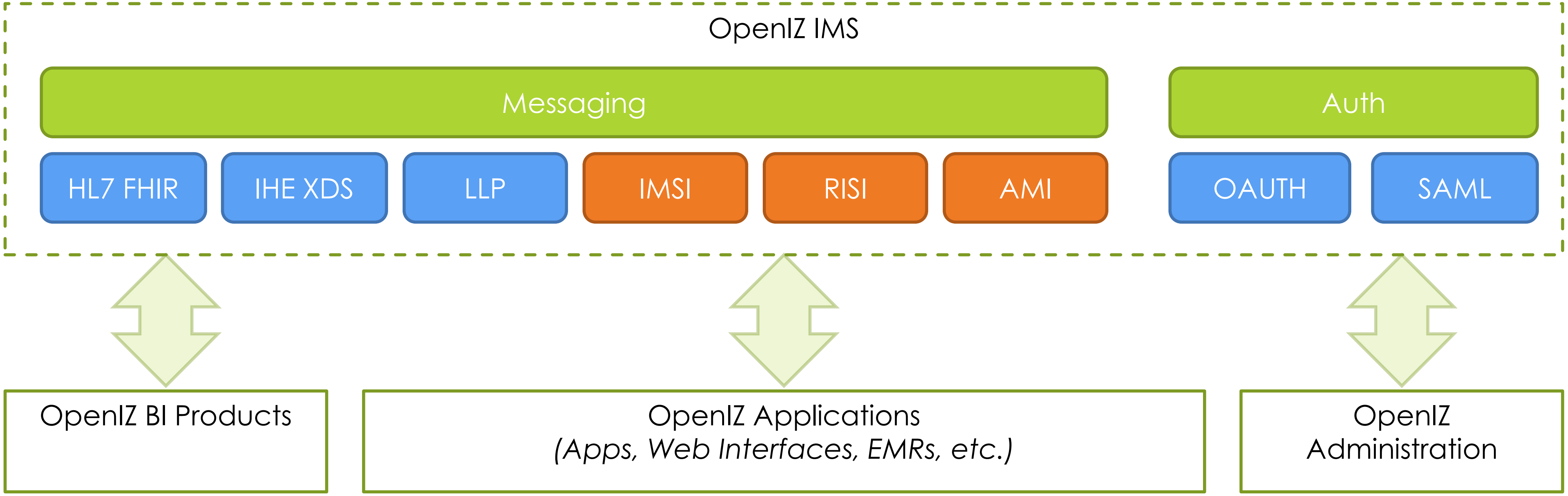
OpenIZ IMS

- The IMS is a headless service
- It performs all functions related to:
  - Integration with external parties,
  - Forecasting and appointment management,
  - Patient, Place, Vaccine, Act, management,
  - Stock management and reporting
- Open Architecture & Open Source

OpenIZ Apps

- User Interfaces to the IMS data
  - Web Sites
  - EMRs like OpenMRS or OSCAR
  - Mobile Applications
  - Business Intelligence Tools
- Consumers of IMS services
- Can be commercial applications

OpenIZ IMS Integration



OpenIZ IMS

Messaging

Auth

HL7 FHIR

IHE XDS

LLP

IMSI

RISI

AMI

OAUTH

SAML

OpenIZ BI Products

OpenIZ Applications
(Apps, Web Interfaces, EMRs, etc.)

OpenIZ
Administration

Messaging Handlers

- Proprietary Interfaces:
  - Immunization Management Service Interface (IMSI)
  - Report Integration Service Interface (RISI)
  - Administrative Management Interface (AMI)
- Standards Based Interfaces
  - HL7® Fast Health Interoperability Resources (FHIR™)
  - HL7® CDA™ over IHE XDS
  - HL7v2™ over LLP
  - SAML / XUA
  - OAUTH / IUA

IMSI (Immunization Management System Interface)

- RESTful API exposing the underlying IMS .NET API
  - 1-to-1 match for all immunization functions
  - Underlying system identifiers are exposed (UUIDs)
- Functions include:
  - Create/Update/Query/Get Entities
  - Create/Update/Query/Get Acts
  - Create/Update/Query/Get Concepts & Reference Terms
  - Create/Update/Query/Get Stock Ledger and Balances
- Useful for offline synchronization (exposes ALL clinical data from IMS)

RISI (Report Integration Services Interface)

- RESTful API used for business intelligence operations
  - Query only
  - All data is either aggregate or de-identified
  - Integrates with raw-table level data interface functions (i.e. bypasses the IMS .NET API)
- Integration with reporting engines
  - Report URL and Parameter Construction

AMI (Administration & Management Interface)

- SOAP Based Administrative Interface
  - Restarting/Stopping Nodes in IMS cluster
  - Installing/Removing handlers and service classes
  - Configuring AssigningAuthorities, Identifier Types, etc.
  - Configuring Reports and stored queries
- No access to clinical data
- Administrator access only

HL7® FHIR™

- RESTful, Standards-based API
- Exposes following resources in FHIR 1.0.2
  - Vaccination – Create/Update/Search/Get/History-Get
  - VaccinationRecommendation – Create/Update/Search/Get/History-Get
  - Patient – Create/Update/Search/Get/History-Get
  - Provider – Create/Update/Search/Get/History-Get
  - Organization – Create/Update/Search/Get/History-Get
  - RelatedPerson – Create/Update/Search/Get
  - Encounter – Create/Update/Search/Get
  - Observation – Create/Update/Search/Get
  - Problem – Create/Update/Search/Get
  - Allergy/Intolerance – Create/Update/Search/Get

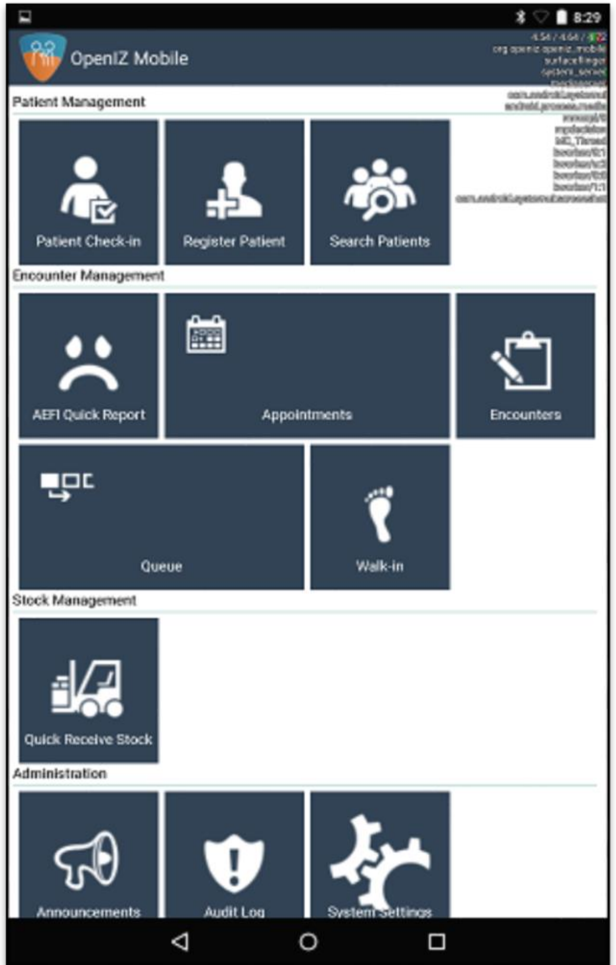
HL7® CDA™ over IHE XDS

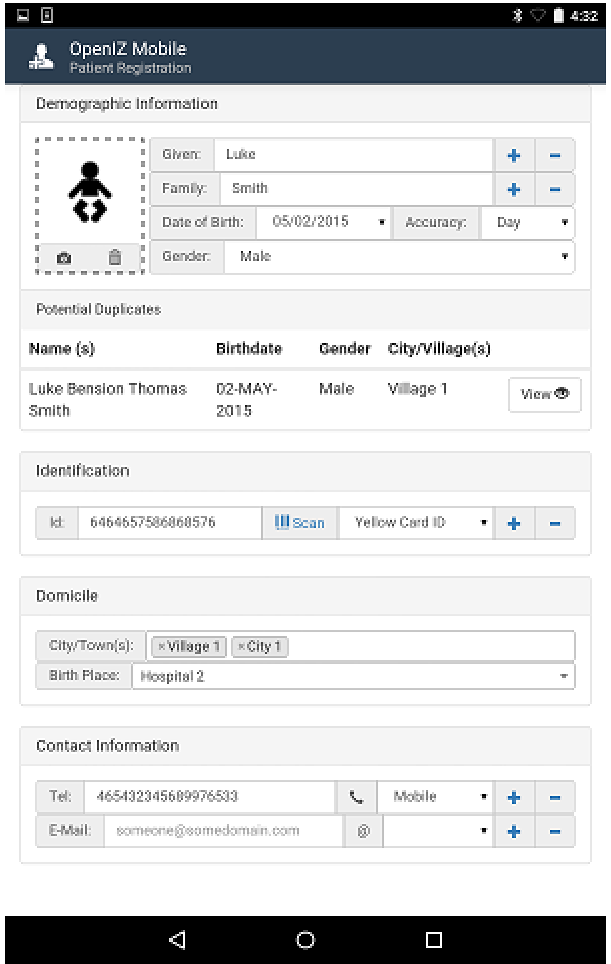
- IHE XDS
  - Static and On-Demand Documents
- CDA™
  - IHE PCC Immunization Content CDA (IC) Document Consumer and Document Source

HL7v2.x

- HL7 Version 2.5 interfaces over Lower Level Protocol (LLP)
  - ADT – A01 (Create), A08 (Update), A40 (Merge)
  - QBP – Q21 (Find Candidates)
  - VXQ – V01 (Vaccine Record Query)
  - VXU – V04 (Vaccine Record Update)
  - SRM – S01 (New Appointment), S02 (Appointment Revision), S04 (Appointment Cancellation), S06 (Appointment Delete/Obsolete)

Example: OpenIZ Disconnected Client





Demographic Information

Given: Luke

Family: Smith

Date of Birth: 05/02/2015 Accuracy: Day

Gender: Male

Potential Duplicates

Name (s)	Birthdate	Gender	City/Village(s)	
Luke Bension Thomas Smith	02-MAY-2015	Male	Village 1	View

Identification

Id: 6464657586868576 Scan Yellow Card ID

Domicile

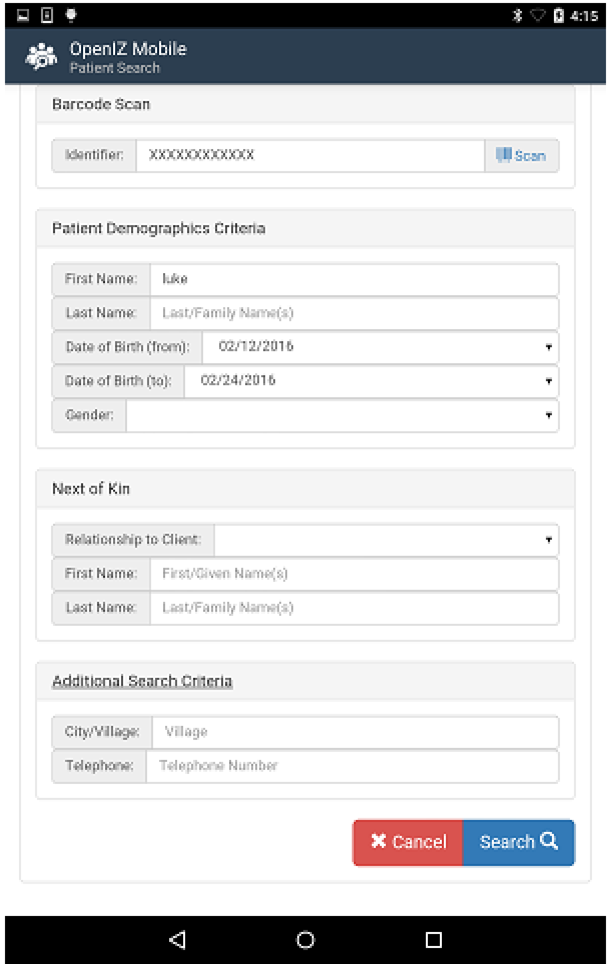
City/Town(s): ×Village 1 ×City 1

Birth Place: Hospital 2

Contact Information

Tel: 465432345689976533 Mobile

E-Mail: someone@somedomain.com



Barcode Scan

Identifier: XXXXXXXXXXXX Scan

Patient Demographics Criteria

First Name: luke

Last Name: Last/Family Name(s)

Date of Birth (from): 02/12/2016

Date of Birth (to): 02/24/2016

Gender:

Next of Kin

Relationship to Client:

First Name: First/Given Name(s)

Last Name: Last/Family Name(s)

Additional Search Criteria

City/Village: Village

Telephone: Telephone Number

Roadmap

WHERE WE’RE HEADING

Q4 – 2015 Roadmap

	OCT 2015	NOV 2015	DEC 2015
IMS	○ Solutions architecture commences ○ Initial development plan ○ Business plan completed	○ Registration of OSS project ○ Design of data model ○ SQL Server implementation of data model ○ Solutions architecture commences	○ Model view classes completed ○ View model classes designed and commence ○ Service core backbone commences ○ Solutions architecture completed.
Applications		○ Initial development plan	○ Public profile created ○ Application wireframes completed ○ Application technology platform selection commences.

Q1 – 2016 Roadmap

	JAN 2016	FEB 2016	MAR 2016 ◆
IMS	o IMSI interface completed o Service core classes / contracts completed o FHIR interface commences o View model classes completed o JavaScript BRE engine integrated.	o FHIR interface completed o LLP transport service integrated o ADT / QBP HL7 interfaces integrated o OAUTH provider integrated. o AMI commences o NxBRE forecaster integrated	o PostgreSQL Data Model implementation completed o RISI interface commences o SSRS RISI provider completed o Administrative UI completed o VXU/VXQ HL7 interfaces integrated.
Applications	o Disconnected client designed o Vaccination web user interface commences.	o Vaccination web user interface completed o Administrative/operations management console started.	o Administrative/operations console completed. o Business intelligence / analytics user interface commences.

Q2 – 2016 Roadmap

	Release: Algonquin		Release: Bluenose
	APR 2016	MAY 2016	JUN 2016
IMS	○ RISI interface completed ○ Participation at immunization connect-a-thon ○ Jasper RISI provider implemented ○ Basic JavaScript business rules implemented	○ Participation at HL7 FHIR connect-a-thon ○ Stock management BRE logic implemented. ○ IMSI sync protocol completed ○ PIX / PDQ client integration completed.	○ ICE forecaster implementation ○ XDS / CDA producer / consumer implemented ○ JUN-2016 CTP release package.
Applications	○ Demonstration at AppsForHealth 2016		

Final Thoughts

A FEW PARTING THOUGHTS