OVERVIEW OF HEALTH SERVICES PLATFORM

Rick Freeman

President & Founder

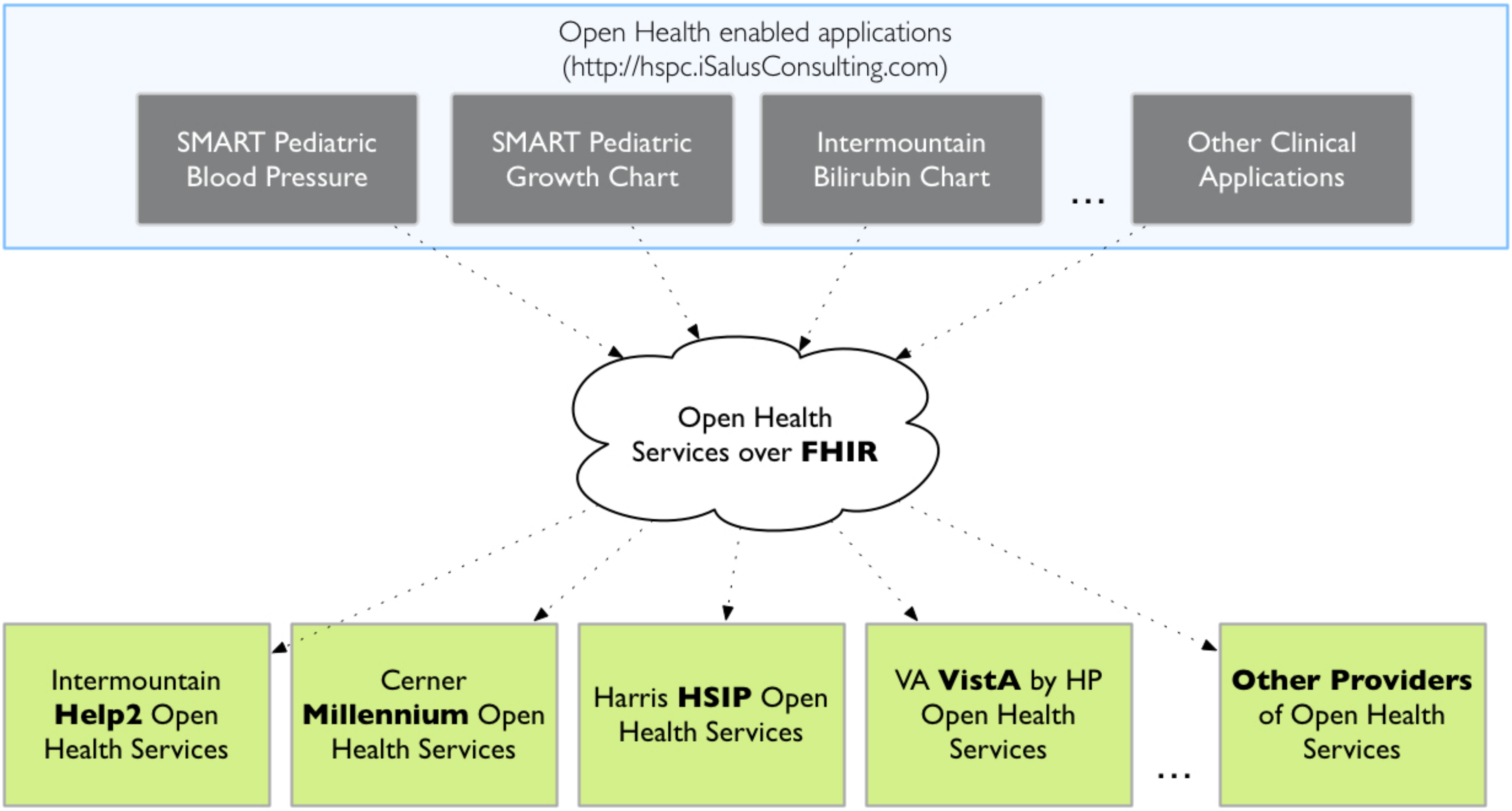
iSalus Consulting

May 9, 2014

Drop In Applications

- App Store
- App Certification
- EMR Agnostic
- Built on Data Standards
  - FHIR Profiles based on CEM
  - SNO Med CT, RxNorm, LOINC

Open Health Services Platform
HiMSS



Open Health enabled applications
(http://hspc.iSalusConsulting.com)

SMART Pediatric Blood Pressure

SMART Pediatric Growth Chart

Intermountain Bilirubin Chart

...

Other Clinical Applications

Open Health Services over **FHIR**

Intermountain **Help2** Open Health Services

Cerner **Millennium** Open Health Services

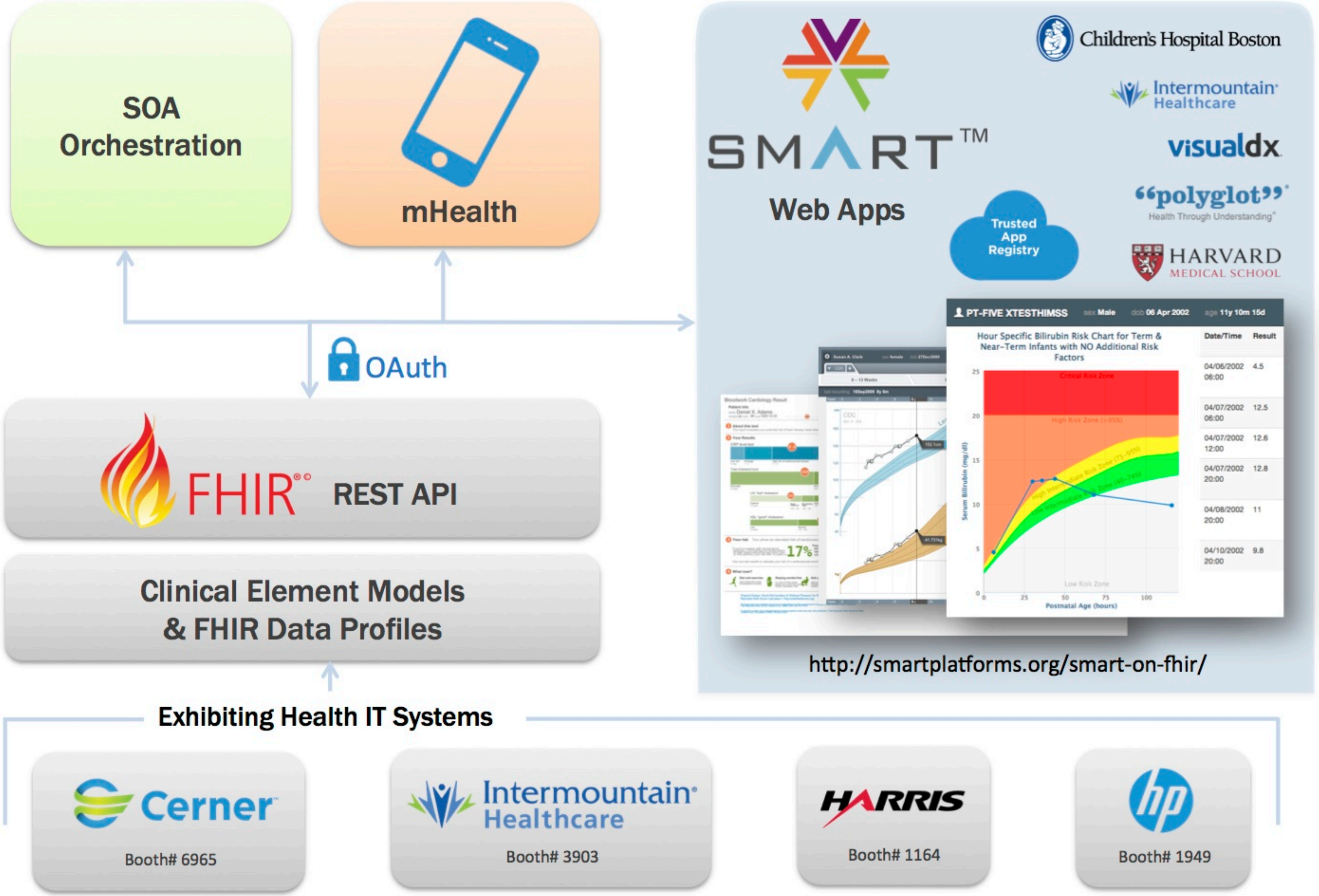
Harris **HSIP** Open Health Services

VA **VistA** by HP Open Health Services

...

Other Providers of Open Health Services

SMART on FHIR®© – Open Platform Architecture



SOA Orchestration

mHealth

OAuth

FHIR®© REST API

Clinical Element Models & FHIR Data Profiles

SMART™ Web Apps

Trusted App Registry

Children's Hospital Boston

Intermountain Healthcare

visualdx

"polyglot" Health Through Understanding

HARVARD MEDICAL SCHOOL

PT-FIVE XTESTHIMSS sex Male dob 06 Apr 2002 age 11y 10m 15d

Hour Specific Bilirubin Risk Chart for Term & Near-Term Infants with NO Additional Risk Factors

Date/Time	Result
04/06/2002 06:00	4.5
04/07/2002 06:00	12.5
04/07/2002 12:00	12.6
04/07/2002 20:00	12.8
04/08/2002 20:00	11
04/10/2002 20:00	9.8

http://smartplatforms.org/smart-on-fhir/

Exhibiting Health IT Systems

Cerner Booth# 6965

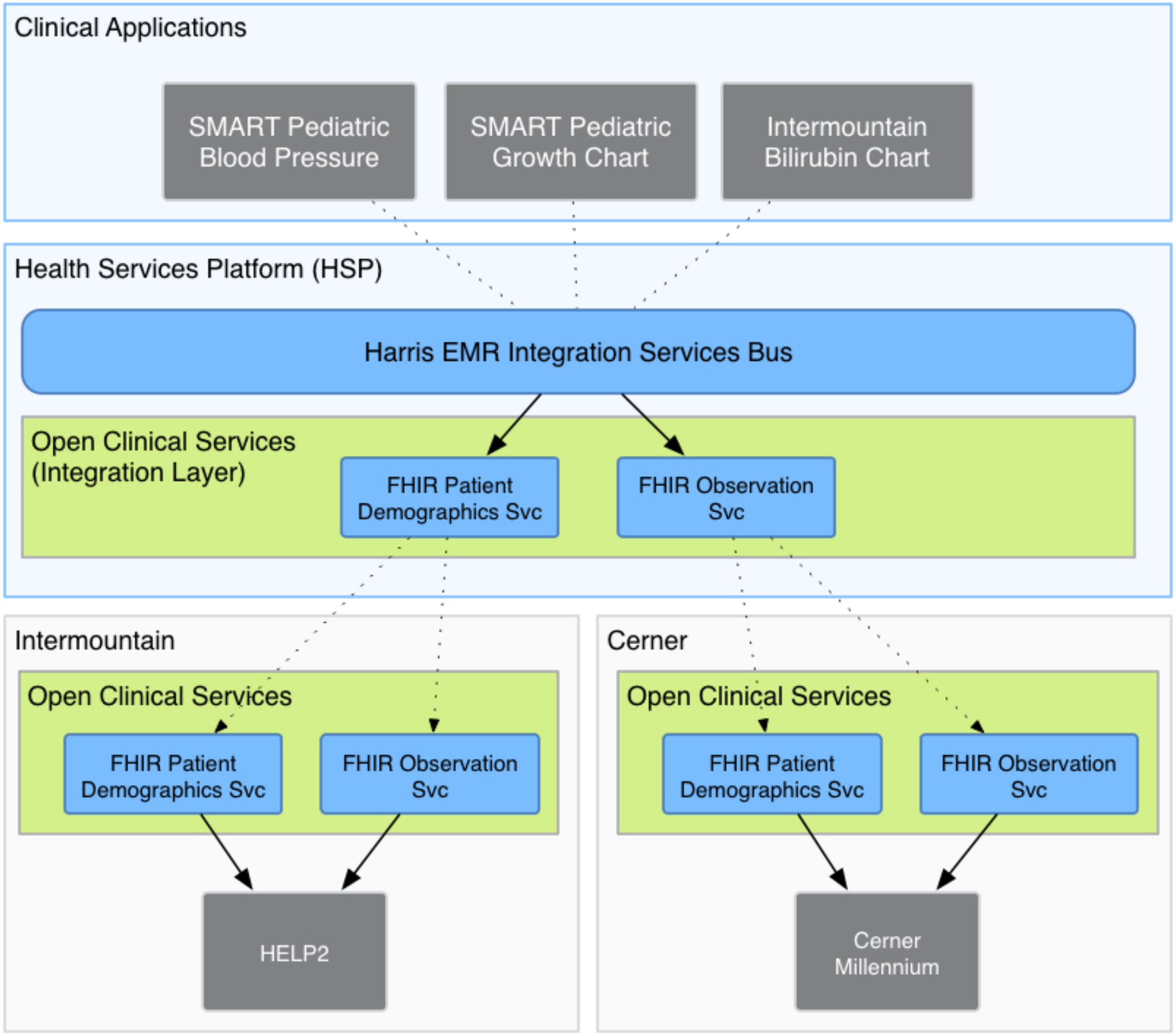
Intermountain Healthcare Booth# 3903

HARRIS Booth# 1164

hp Booth# 1949

HIMSS 2014 DEMO

HSPC HiMSS Demo



Clinical Applications

SMART Pediatric Blood Pressure

SMART Pediatric Growth Chart

Intermountain Bilirubin Chart

Health Services Platform (HSP)

Harris EMR Integration Services Bus

Open Clinical Services (Integration Layer)

FHIR Patient Demographics Svc

FHIR Observation Svc

Intermountain

Open Clinical Services

FHIR Patient Demographics Svc

FHIR Observation Svc

HELP2

Cerner

Open Clinical Services

FHIR Patient Demographics Svc

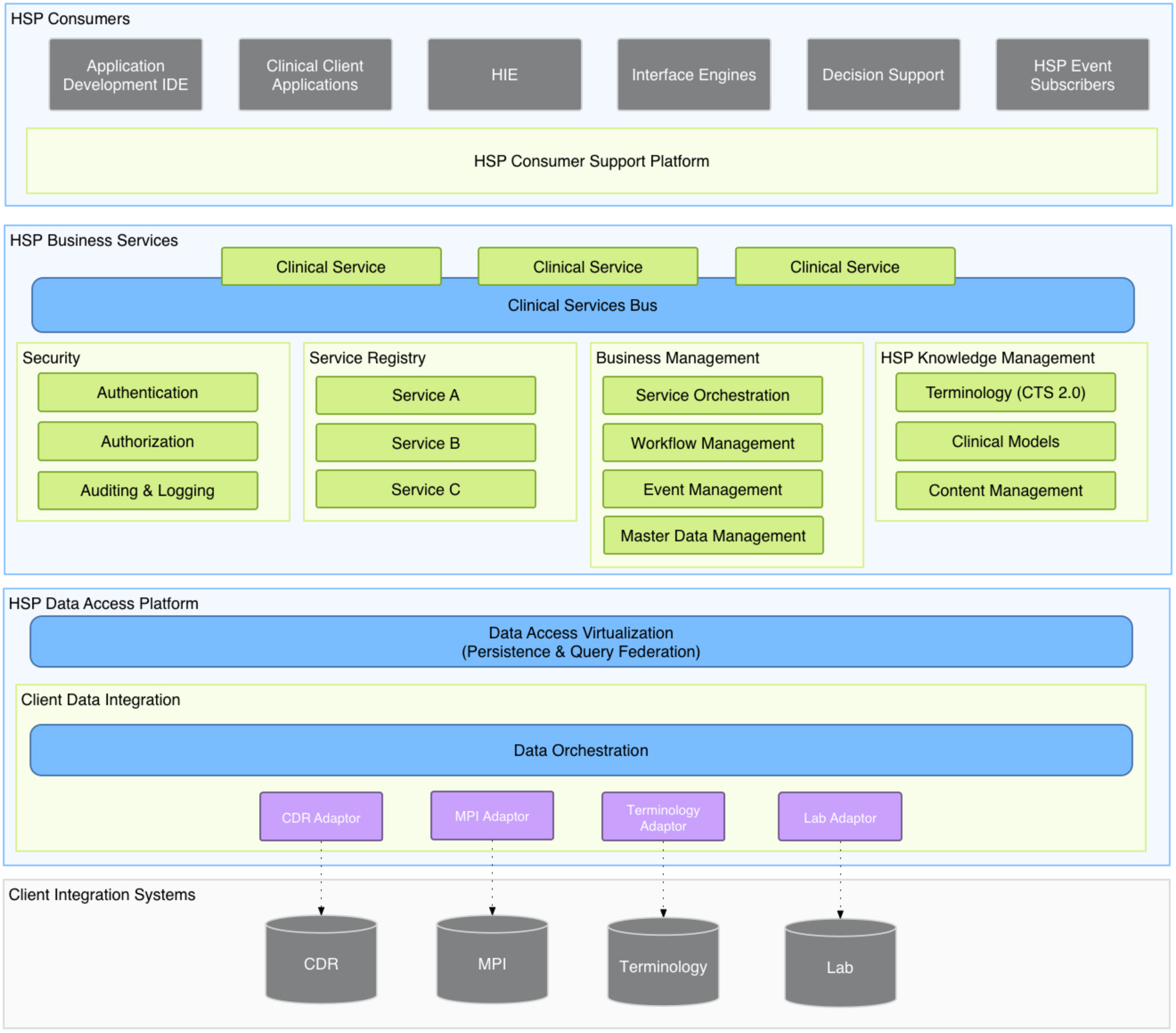
FHIR Observation Svc

Cerner Millennium

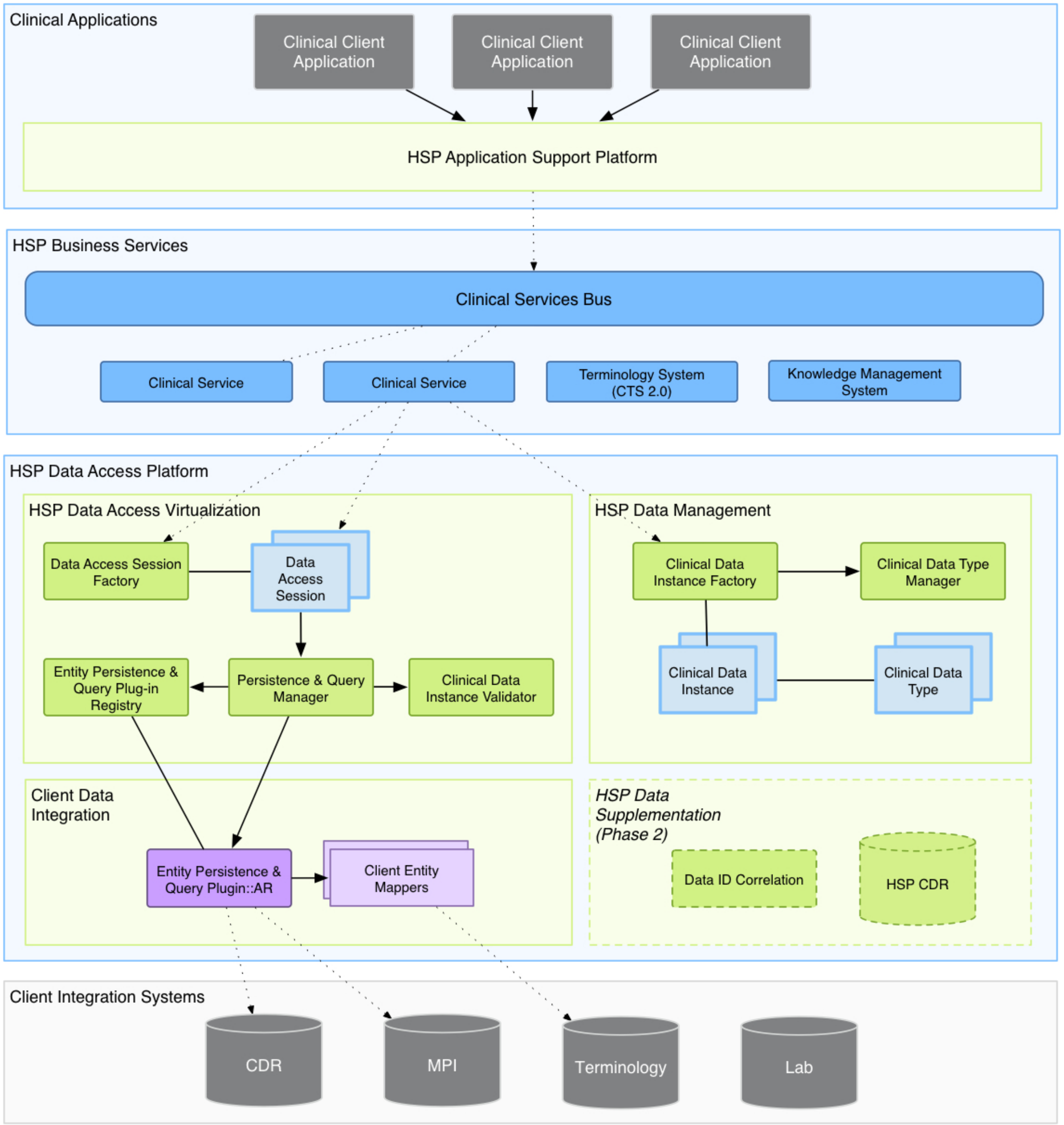
Existing Service Integration SDK

- Spring web app based framework for RESTful FHIR services
- Patient demographics and observations support for demo client apps.
- HTTP response handling (response streaming, headers for AJAX CORS etc)
- JSON serialization
- Data virtualization abstraction.
  - Data virtualization "plugins" were implemented by Intermountain, Harris and HP that provided backend data mappings to FHIR objects. The data mapping plugins were the only pieces that required provider specific implementation. Everything else was handled by the framework.
- Simple war file deployment (Tomcat, etc.)

HSP Functional Components



HSP Logical Components


Clinical Applications
Clinical Client Application
Clinical Client Application
Clinical Client Application
HSP Application Support Platform
HSP Business Services
Clinical Services Bus
Clinical Service
Clinical Service
Terminology System (CTS 2.0)
Knowledge Management System
HSP Data Access Platform
HSP Data Access Virtualization
Data Access Session Factory
Data Access Session
Entity Persistence & Query Plug-in Registry
Persistence & Query Manager
Clinical Data Instance Validator
HSP Data Management
Clinical Data Instance Factory
Clinical Data Type Manager
Clinical Data Instance
Clinical Data Type
Client Data Integration
Entity Persistence & Query Plugin::AR
Client Entity Mappers
HSP Data Supplementation (Phase 2)
Data ID Correlation
HSP CDR
Client Integration Systems
CDR
MPI
Terminology
Lab

Proposed Phase 1 Platform Capabilities

- FHIR based, RESTful using JSON read and write services
  - Supporting FHIR Resource & Profiles TBD ASAP
- Authorization, Authentication, and Auditing
- Patient context
- Provider/User management
- Simple eventing
- Basic terminology services
- Simple transaction management (not long-running)
- System management (logging and monitoring)

Proposed Future Platform Capabilities

- General Notification/Alerting/Messaging
- Advanced Terminology Services (subsumption, multiple representations)
- Event handling (pub/sub)
- Service Orchestration
- Long-running Transactions
- Decision Support Services

App Development Resources

For internal Distribution and Review Only

Not Yet Approved for Broad Release

Overview

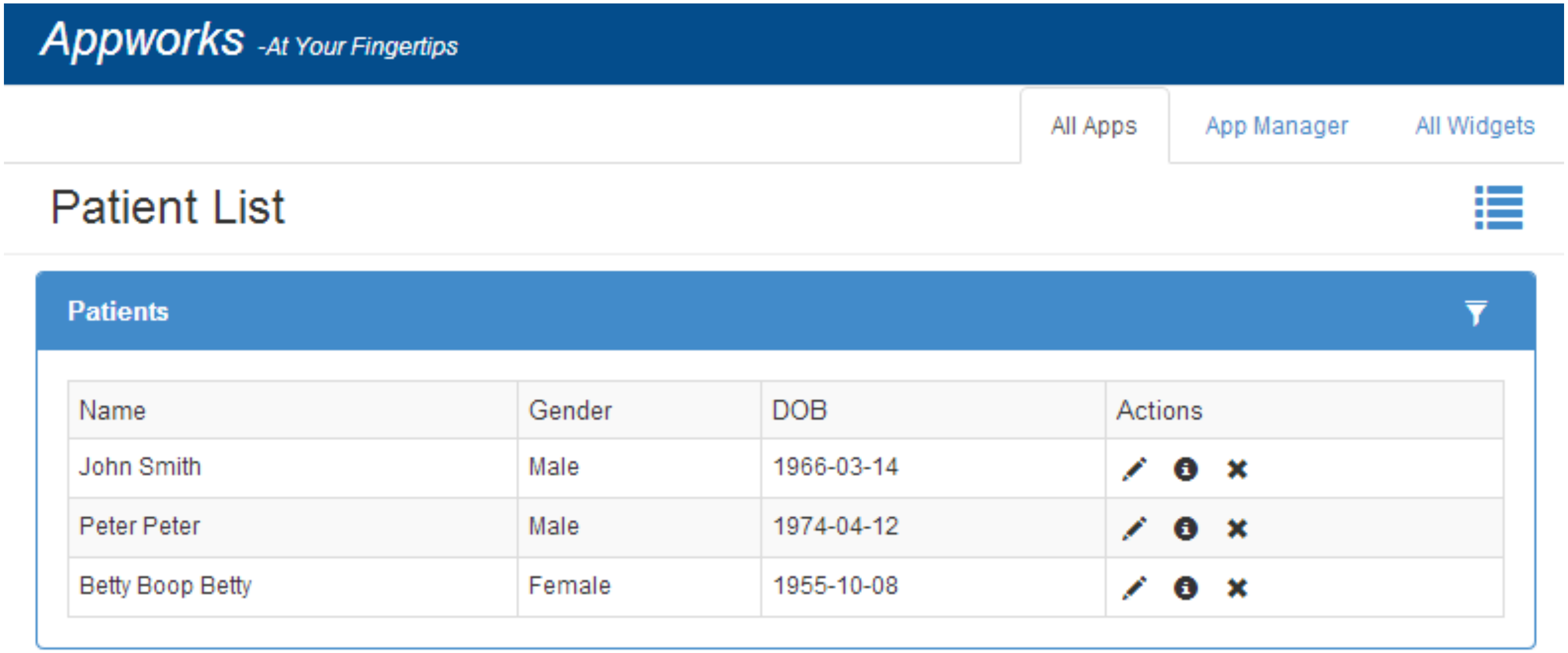
- Appworks is a framework to create a modular architecture for healthcare app
  - App = software that provides units of functionality and can be selectively installed in an environment
  - Interoperates with existing systems such as EHRs in the environment
- It provides an SDK for developers to make it easy to develop apps rapidly, using standards-based access to underlying data and operations
  - Collection of components that can be assembled into apps
    - Widgets: user-interface layer components
    - Service modules: business logic and data access components

What is an App?

- Provides unit of functionality
  - Used in a particular context
    - Clinical function, user, venue
  - Co-exists with and leverages other systems and services in the environment
  - May participate with other apps to coordinate series of activities
- Examples
  - Specialty apps
    - anticoagulation management
    - diabetes management
  - Consumer-oriented apps
    - Goal and parameter tracking
    - Insulin dosing
  - Status dashboards with drill-down
    - e.g., ICU, ED patients
  - Care coordination apps
    - Problem list management
    - Problem-based information assembly
      - Progress-note entry, problem list update, and order update facilitation
        - Post-op rounding and care plan management as a driving example
        - Chronic disease management

App Development

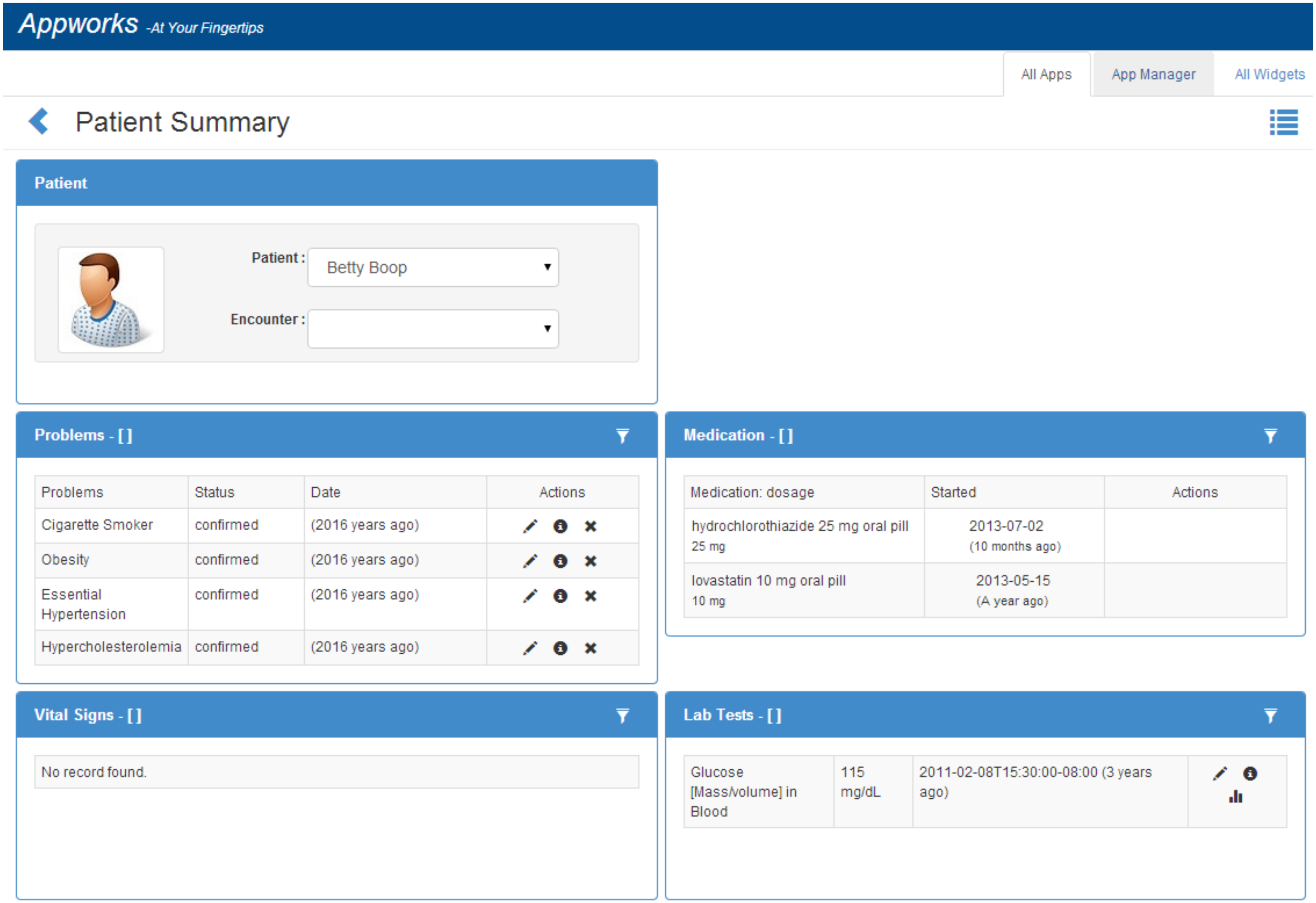
- Appworks' emphasis has been on creating apps easily
  - Using GUIs to compose apps by drag-n-drop of reusable widgets from a library or using a mark-up language
    - Widgets provide a view of data
  - Widgets can mostly be wired with each other and to back-end services through configuration
  - Widget library can be extended
  - Apps may be embeddable in containers (e.g., SMART, CareWeb)
    - Containers will provide services to the UI



Patient List

Patients

Name	Gender	DOB	Actions
John Smith	Male	1966-03-14	✎ ⓘ ✖
Peter Peter	Male	1974-04-12	✎ ⓘ ✖
Betty Boop Betty	Female	1955-10-08	✎ ⓘ ✖



Patient Summary

Patient

Patient : Betty Boop ▾

Encounter : ▾

Problems - [ ]

Problems	Status	Date	Actions
Cigarette Smoker	confirmed	(2016 years ago)	✎ ℹ ✖
Obesity	confirmed	(2016 years ago)	✎ ℹ ✖
Essential Hypertension	confirmed	(2016 years ago)	✎ ℹ ✖
Hypercholesterolemia	confirmed	(2016 years ago)	✎ ℹ ✖

Medication - [ ]

Medication: dosage	Started	Actions
hydrochlorothiazide 25 mg oral pill 25 mg	2013-07-02 (10 months ago)	
lovastatin 10 mg oral pill 10 mg	2013-05-15 (A year ago)	

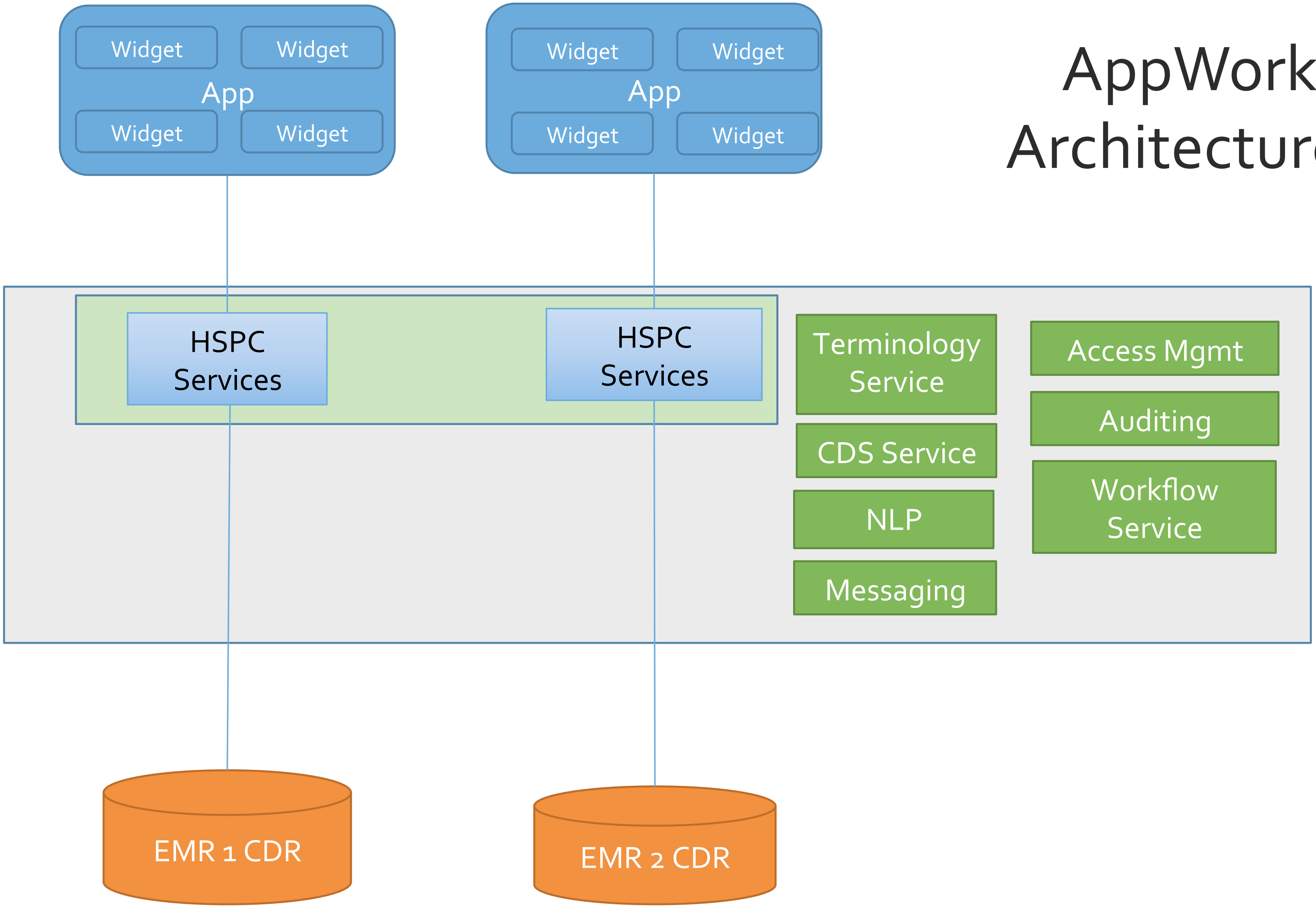
Vital Signs - [ ]

No record found.

Lab Tests - [ ]

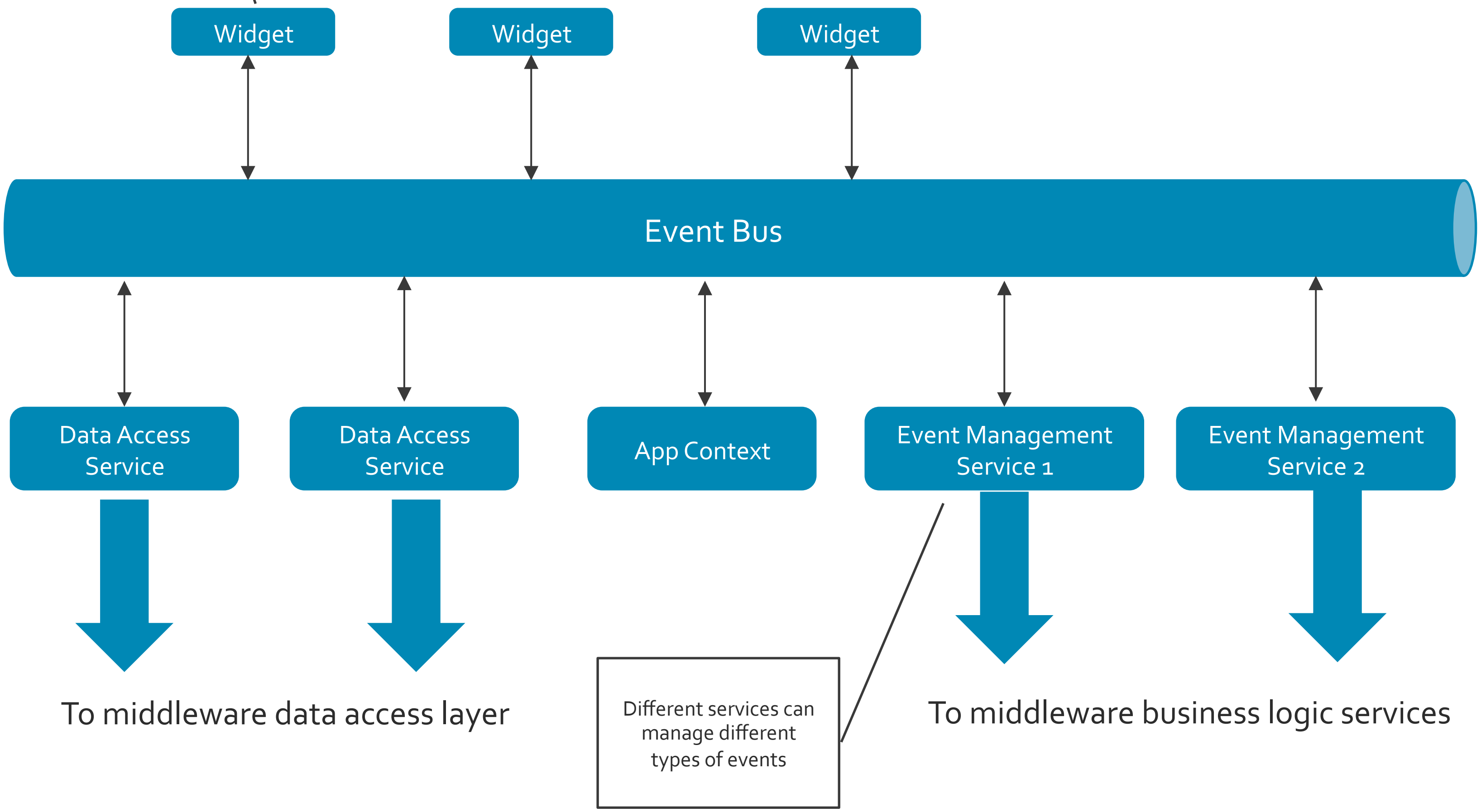
Glucose [Mass/volume] in Blood	115 mg/dL	2011-02-08T15:30:00-08:00 (3 years ago)	✎ ℹ 📊

AppWorks Architecture



App UI Architecture

Widget bound to specified data service



Widget

Widget

Widget

Event Bus

Data Access Service

Data Access Service

App Context

Event Management Service 1

Event Management Service 2

To middleware data access layer

Different services can manage different types of events

To middleware business logic services

Implementation

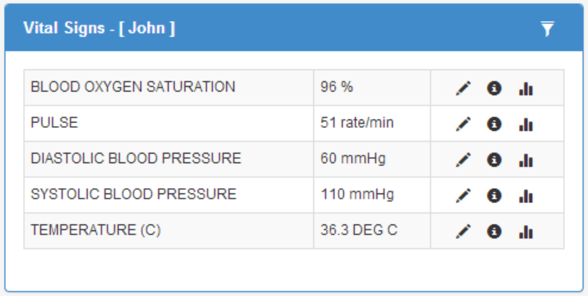
- Front-end: HTML(5) and AngularJS
  - AngularJS is a browser-side Javascript library
    - Enables MVC pattern for our widgets and creation of HTML “tags” for widgets
- Middleware will be an HSPC certified implementation
  - Appworks proposed implementation of HSPC will use Apache ServiceMix 4.x

Proposed HSPC Certified Middleware Implementation/Sandbox

- Modules implemented as OSGI services for inter-module collaboration within ServiceMix container
  - Use CAMEL for service orchestration optionally
  - Use Drools and jBPM for business process and logic execution
  - Goal to maintain compatibility with Websphere and other production stacks
- Services expose REST or other interfaces outside the container
  - Uses CXF (Apache library for SOA) and Jetty (web server)

Front End Implementation

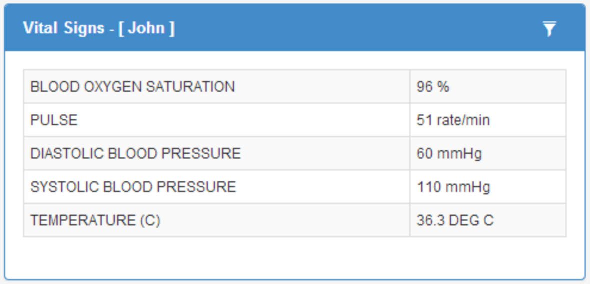
- Front-end implemented with AngularJS and HTML5
  - Implementing a set of widgets
    - A widget exposes modular functional capability to the user
      - Typically associated with a FHIR resource type, e.g., medication list viewer
    - An app is composed by assembling a set of widgets
    - Widgets decoupled, communicate with each other and with the services using Angular Services and publish subscribe approach
    - We're using AngularJS to make UI *components declarative* and reduce need for Javascript programming for app builders



Vital Signs - [John]

BLOOD OXYGEN SATURATION	96 %	✎ ℹ 📊
PULSE	51 rate/min	✎ ℹ 📊
DIASTOLIC BLOOD PRESSURE	60 mmHg	✎ ℹ 📊
SYSTOLIC BLOOD PRESSURE	110 mmHg	✎ ℹ 📊
TEMPERATURE (C)	36.3 DEG C	✎ ℹ 📊

<ayf-vitalsigns></ayf-vitalsigns>



Vital Signs - [John]

BLOOD OXYGEN SATURATION	96 %
PULSE	51 rate/min
DIASTOLIC BLOOD PRESSURE	60 mmHg
SYSTOLIC BLOOD PRESSURE	110 mmHg
TEMPERATURE (C)	36.3 DEG C

<ayf-vitalsigns *hide="actions"*></ayf-vitalsigns>

Clinical Data Model

- We will use HSPC FHIR profiles for clinical data models
- Data access layers implemented using HSPC services
  - RESTful FHIR based services
- Other middleware components use HSPC based FHIR as the data model
- Front-end widgets can display HSPC FHIR data

Developing Apps

- App composition
  - Create a complete application by wiring together a set of *existing* widgets and services
  - Requires HTML and minimal JS skills
  - Working on GUI for creating new apps
- Widget development
  - Create new widgets
  - Requires advanced HTML and JS expertise

- Service Orchestration
  - Need HSPC to define service composition strategyss
  - Need HSPC to support SOA

- Need HSPC to define method and roadmap to create additional services
  - HSPC to consider real-time deployment of new services (such as OSGi)

Appworks Roadmap for Remainder of 2014

- Expand functionality
  - FHIR resources
    - make more EHR data available to apps including add, update, delete
  - Widgets
    - To match additional FHIR resources
    - More functionality
      - Sorting, filtering, graphing
      - Binding to data and richer inter-widget interactions
  - Apps
    - Rounding
    - Problem list reconciliation
    - Mobile and desktop
  - Services
    - Terminology – integrate existing services
    - CDS
    - Workflow management
- App Designer GUI
  - Lets power user compose new apps by drag-and-drop and configuration
  - Library of skeletal apps
  - Use of BPML-like drawing tools to design app-widget/inter-app process flows
- Infrastructure
  - Security
  - Management console

Other goals for sandbox and related resources

- Integration of CareWeb functionality
- Ability to interact with and integrate SMART apps
- HCI/design services, usability testing lab
- Tech support, educational materials
- Reference implementation(s) testbed with various SOA stacks, EHRs
- Test database(s)
- Private workspaces for projects
- Hosting of app store
- Conformance testing

DEMO

- http://ec2-54-186-219-109.us-west-2.compute.amazonaws.com/ayf/index.html#/applist