VA

U.S. Department
of Veterans Affairs

2015 OSEHRA Open Source Summit
eHMP Overview

29/July/2015

Version: 1.0

Submitted by: Carl Gray, Christopher Edwards, Seth Gainey, Jeff Kopra

Administrative

- If there are questions that are left unanswered, you don't want to ask in a public forum, or time doesn't allow for: please direct your questions to:

  eHMPOpenSource@asmr.com

- We will ensure that your questions are answered by members of our development team.

Introduction

Key Objectives

- A key objective of the VistA Evolution Program is to enhance internal, cross-Agency (DoD/VA), and external partner interoperability by providing all clinically relevant data at the point of care for Veterans within the eHMP Application.
- eHMP is targeted to be the eventual replacement for the CPRS application.
- eHMP will accomplish this by integrating an enhanced graphical user interface, standards-based data, and integrating core clinical applications
- The development team works directly with VA stakeholders, which consists of on-the-ground doctors, technicians, SMEs, and leadership.

Problem to be Solved by eHMP

- There are ~ 130 non-integrated VistA systems in the VA
- Each holds local EHR patient data
- Patient data is not integrated between sites

eHMP Functional Goals

- View of patient record, showing several sources of information at once
- Advanced text search across patient record
- Medication Review
- Ability to document a clinical encounter (edit patient record, add allergies …)
- Order entry, order management
- Contextual information (Infobutton)
- Ability to recommend interventions based upon patient record derived from clinical practices

Future Goals or Enhancements

- Clinical Activities Management
- Enhanced Clinical Decision Support
- Additional Orders Management
- Support for Team Based Care
- Patient Education
- Secure Messaging
- eHMP as a Platform

Dependencies

License Dependencies

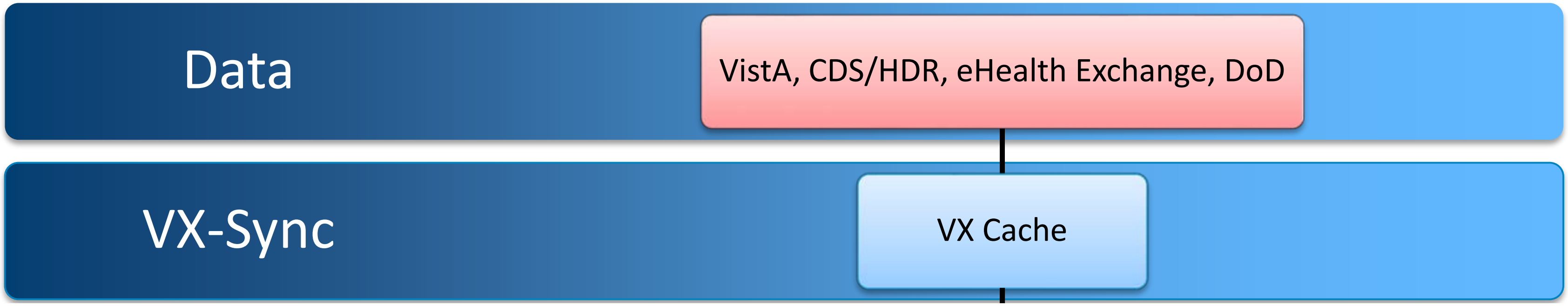
- Dependencies
  - Sentillion CCOW
    - Easily disabled
  - HighCharts
    - HighCharts has a non-commercial license
  - JDS is targeted for Cache
    - GT.M compatible with some minor modifications

VistA Access

- Access to VistA
  - The current code base and deployment plan does require that the user have access to VistA or to a VistA test instance.
  - eHMP should run on any current FOIA VistA instance
    - There are KIDS builds that are required to be installed
  - Currently uses the RPC Broker to communicate using a JavaScript implementation of the RPC Broker Client

Architecture

VX Cache/VX Sync



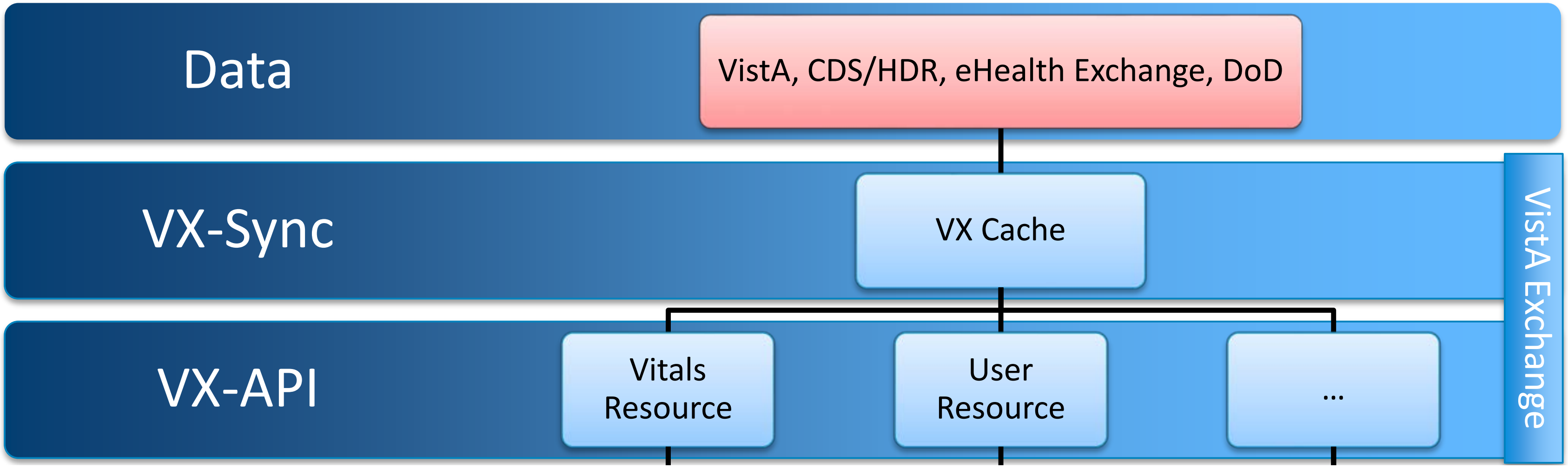
Architecture (VX Cache/VX Sync)

- Patient data is stored in VX Cache, which is a temporary store of patient data.
- Data is kept up-to-date using a publication/subscription from each VistA site.
- Other patient data is done using a request/response pattern.
- Other data sources include: VistA, CDS/HDR (Health Data Repository), eHealth Exchange, DoD (jMeadows), and planned support for patient generated data from Connected Health

Architecture (VX Cache/VX-Sync)

- VX Cache is implemented with a JSON document database (JSON Data Store or JDS), Solr and a rich-text document store.
- Patient Data in VX Cache is stored with standardized terminology codes (SCT, ICD, RxNorm, LOINC, CPT, UMLS) referred to as Record Enrichment.
- VX-Sync is the service in charge of syncing operations and putting updates into VX Cache to be made available to other layers of the application

VX-API


Data
VistA, CDS/HDR, eHealth Exchange, DoD
VX-Sync
VX Cache
VX-API
Vitals Resource
User Resource
…
VistA Exchange

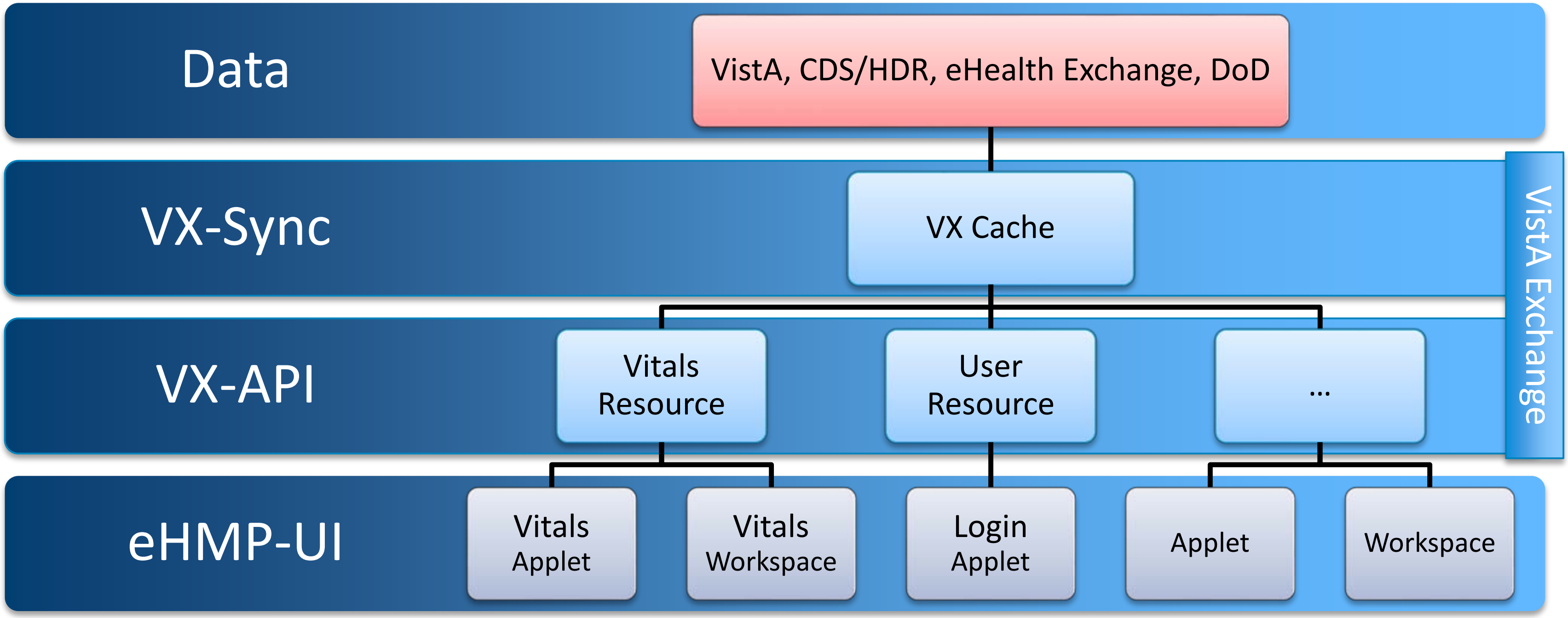
Architecture (RDK)

- Developed using JavaScript, express.js
- Deployed to node.js; relies on npm for package management
- The RDK is responsible for:
  - Providing server configuration and information about the request
  - Providing handles to common external systems, including VX Cache and VistA(s)

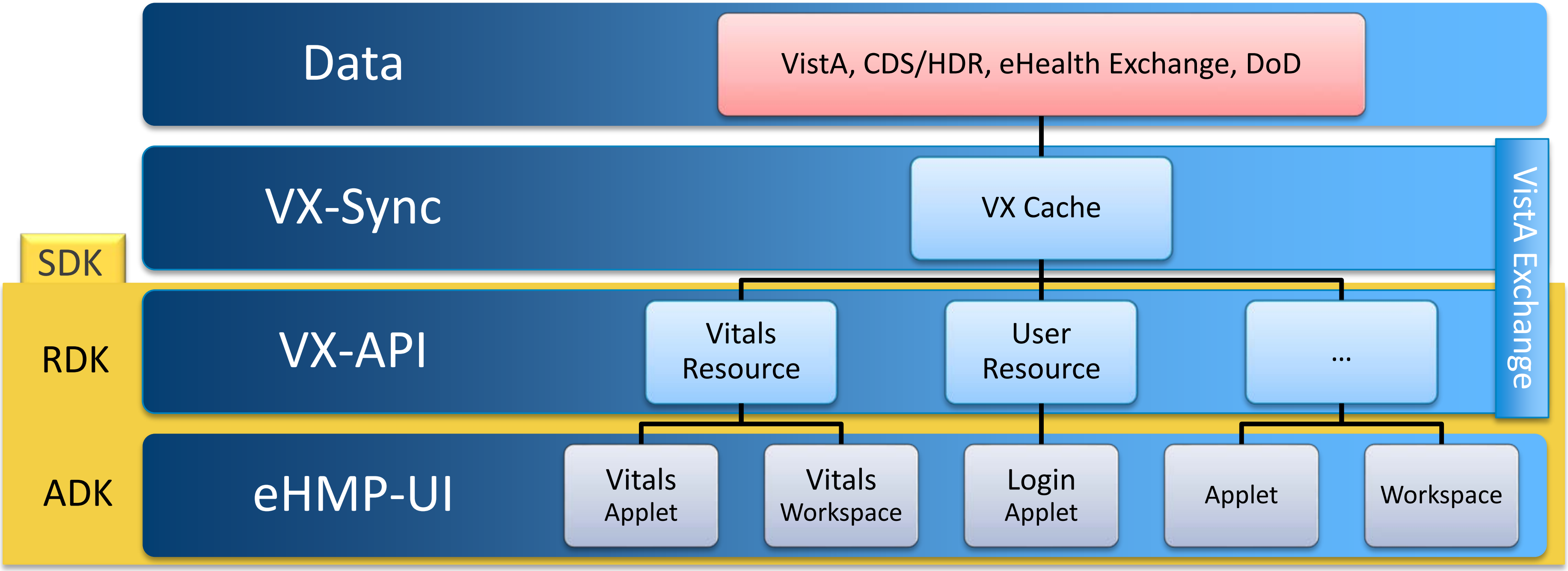
Architecture (VX-API)

- eHMP accesses VX Cache through the VX-API
- VX-API is a ReSTful web service API which provides patient data, search, authentication, access control, CCOW service, order entry, writeback, clinical decision support, and other operations needed for a clinical UI.
- VX-API is written in JavaScript and runs on node.js
- Can return data in both VPR and FHIR format

eHMP UI


Data
VistA, CDS/HDR, eHealth Exchange, DoD
VX-Sync
VX Cache
VX-API
Vitals Resource
User Resource
…
eHMP-UI
Vitals Applet
Vitals Workspace
Login Applet
Applet
Workspace
VistA Exchange

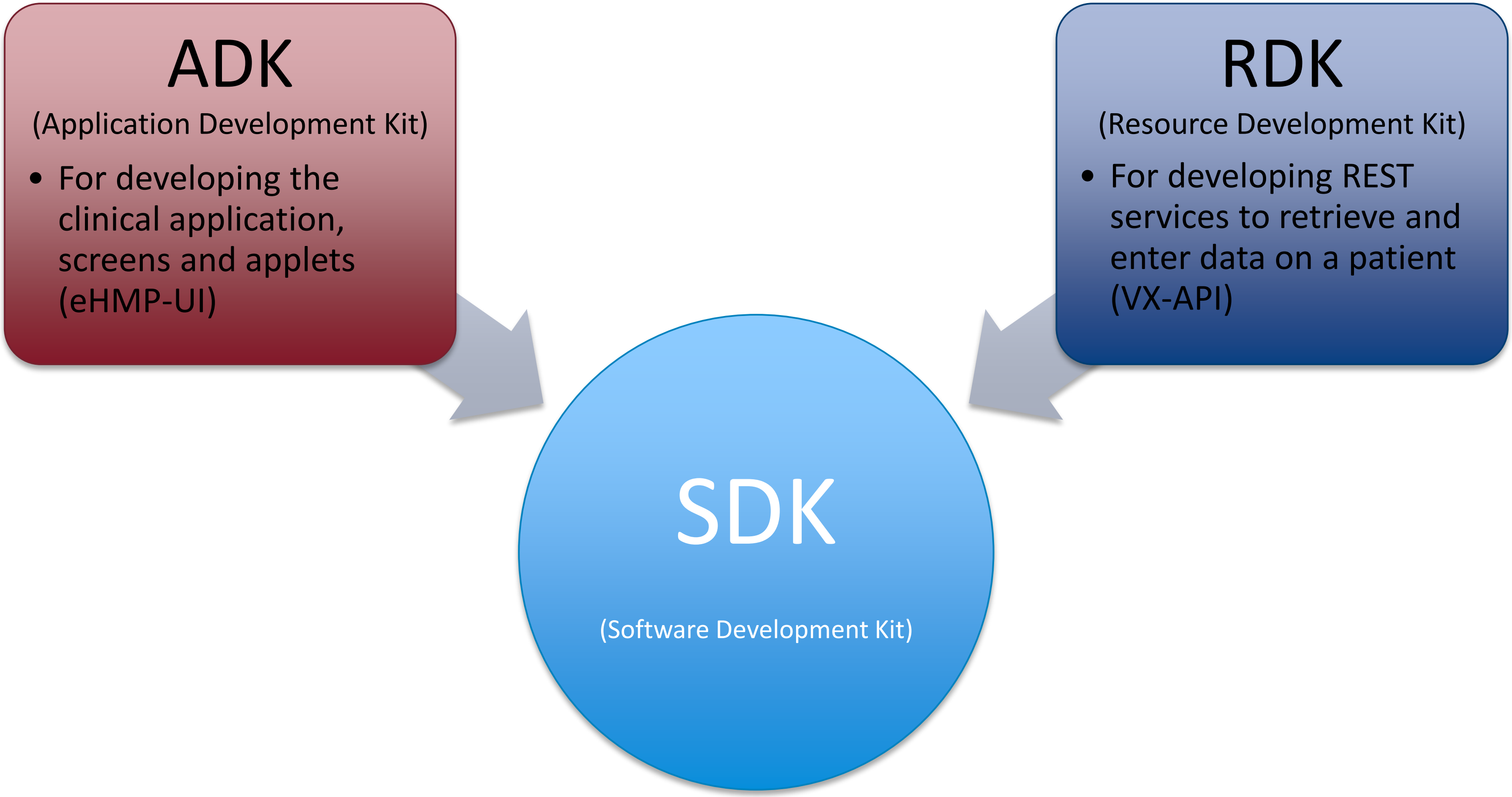
SDK


Data
VistA, CDS/HDR, eHealth Exchange, DoD
VX-Sync
VX Cache
SDK
RDK
VX-API
Vitals Resource
User Resource
…
VistA Exchange
ADK
eHMP-UI
Vitals Applet
Vitals Workspace
Login Applet
Applet
Workspace

Architecture (SDK)

- The SDK provides the ability to add incremental functionality to the web application through the development of applets (not to be confused JSR-286 portal or 1990's Java applets).
- The SDK allows for parallel development of applets by reusing a common UI library.

Architecture (SDK)



ADK

(Application Development Kit)

- For developing the clinical application, screens and applets (eHMP-UI)

RDK

(Resource Development Kit)

- For developing REST services to retrieve and enter data on a patient (VX-API)

SDK

(Software Development Kit)

Architecture (ADK)

- Developed using JavaScript, Backbone.js, Marionette.js, and Bootstrap
- eHMP UI is developed as a Single Page Application, which is static HTML served from the web server (not dynamic web pages such as JSP, PHP, etc. No J2EE web server or App Server)

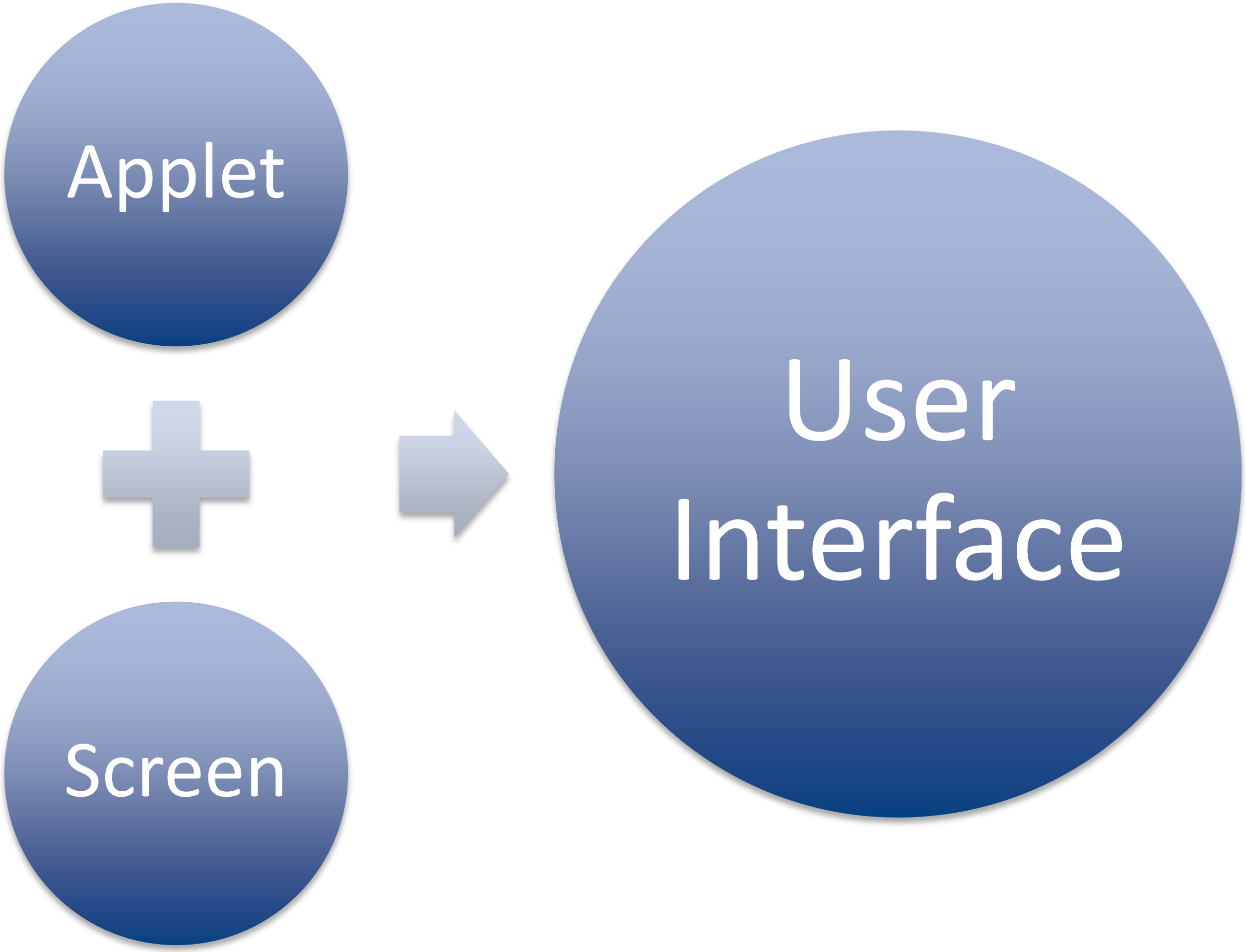
Architecture (ADK)

- A mechanism to arrange these applets through app configuration into layouts called screens
- Provides holistic application utilities and functions
- Includes common cross cutting concerns (authentication, authorization, etc…)
- Provides consistent visual themes
- 508 Support

Architecture (ADK)

- The ADK provides a mechanism for screen designers and application designers to:
  - Create a screen, choose from predefined layouts and assigning applets to regions
  - Provides the runtime UI shell for the application. Displays current patient, current user, provides navigation
  - Choose the screens that are part of the application

Architecture (ADK)



Applet

Screen

User Interface

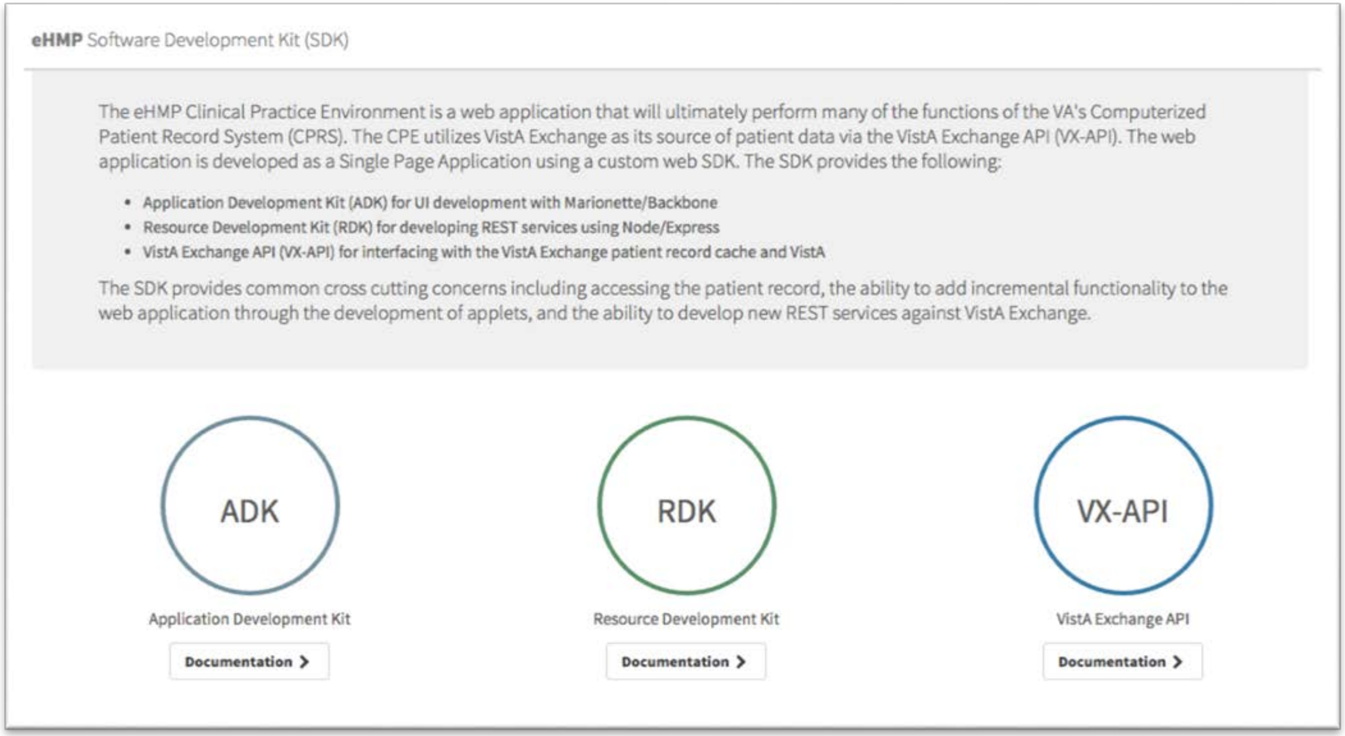
Live Demo

Building Applets & Resources

(using the SDK documentation)

Building Applets & Resources

https://ehmp.vistacore.us/documentation/



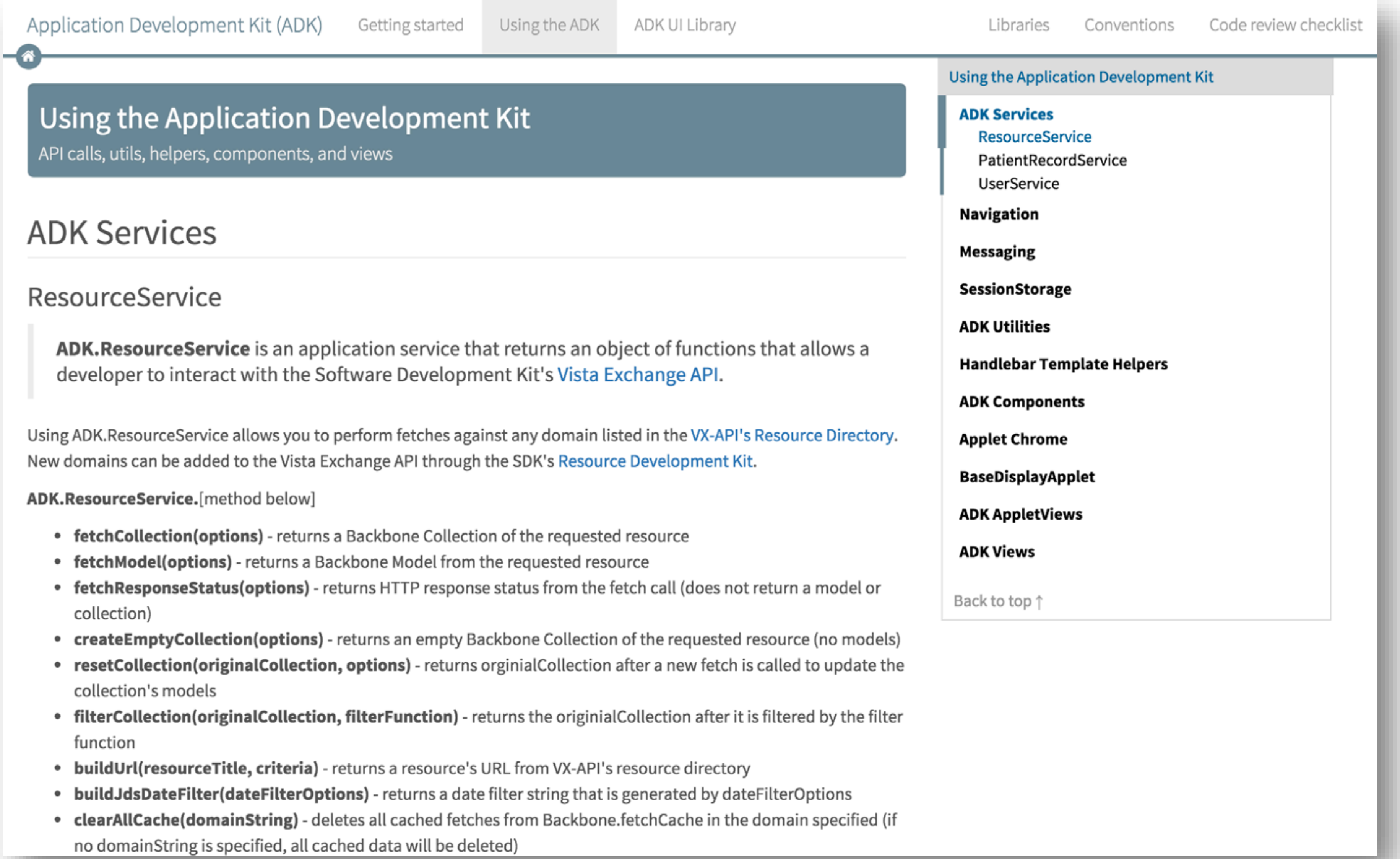
eHMP Software Development Kit (SDK)

The eHMP Clinical Practice Environment is a web application that will ultimately perform many of the functions of the VA's Computerized Patient Record System (CPRS). The CPE utilizes VistA Exchange as its source of patient data via the VistA Exchange API (VX-API). The web application is developed as a Single Page Application using a custom web SDK. The SDK provides the following:

- Application Development Kit (ADK) for UI development with Marionette/Backbone
- Resource Development Kit (RDK) for developing REST services using Node/Express
- VistA Exchange API (VX-API) for interfacing with the VistA Exchange patient record cache and VistA

The SDK provides common cross cutting concerns including accessing the patient record, the ability to add incremental functionality to the web application through the development of applets, and the ability to develop new REST services against VistA Exchange.

Building Applets & Resources



Using the Application Development Kit

API calls, utils, helpers, components, and views

ADK Services

ResourceService

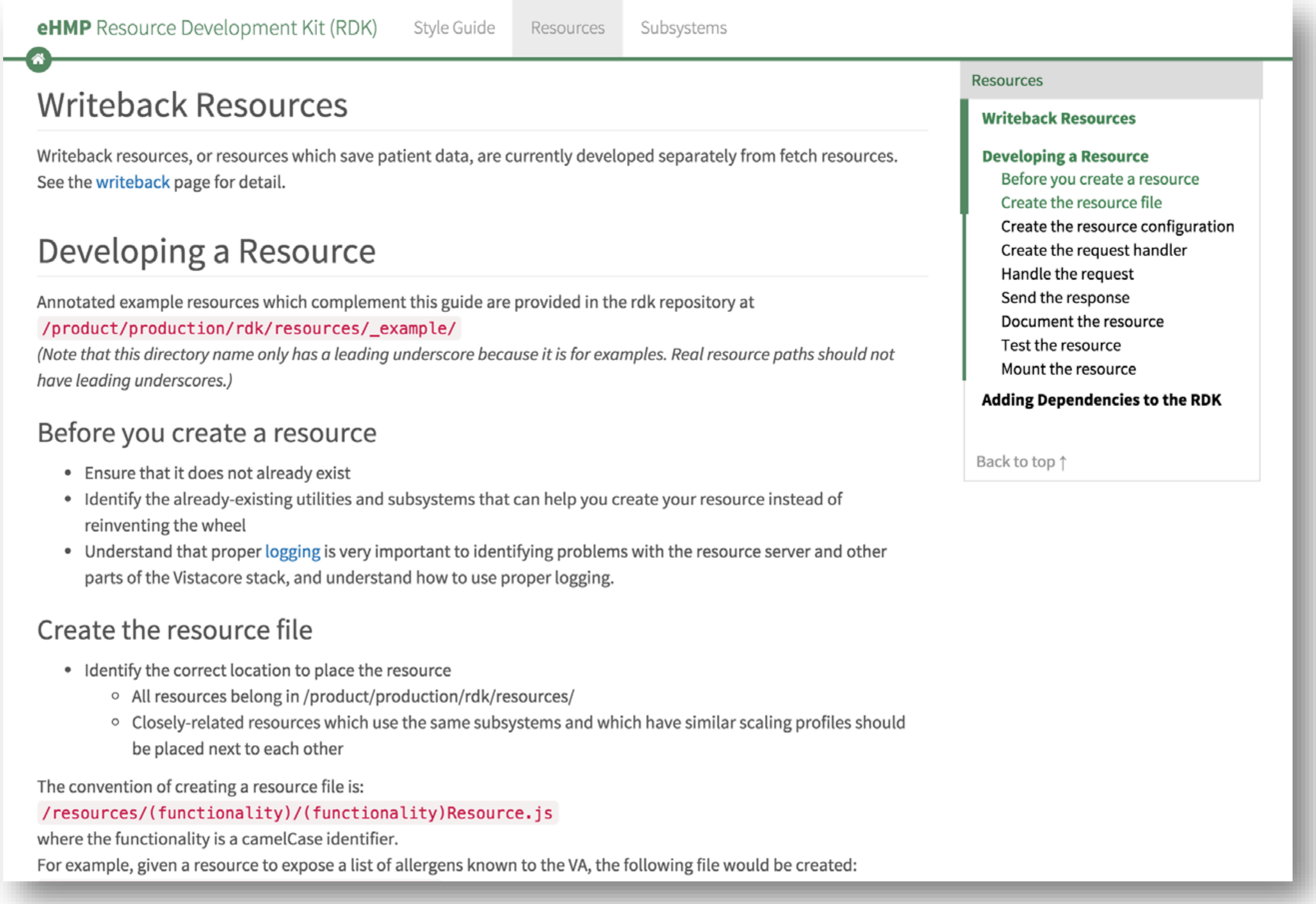
> **ADK.ResourceService** is an application service that returns an object of functions that allows a developer to interact with the Software Development Kit's Vista Exchange API.

Using ADK.ResourceService allows you to perform fetches against any domain listed in the VX-API's Resource Directory. New domains can be added to the Vista Exchange API through the SDK's Resource Development Kit.

ADK.ResourceService.[method below]

- **fetchCollection(options)** - returns a Backbone Collection of the requested resource
- **fetchModel(options)** - returns a Backbone Model from the requested resource
- **fetchResponseStatus(options)** - returns HTTP response status from the fetch call (does not return a model or collection)
- **createEmptyCollection(options)** - returns an empty Backbone Collection of the requested resource (no models)
- **resetCollection(originalCollection, options)** - returns orginialCollection after a new fetch is called to update the collection's models
- **filterCollection(originalCollection, filterFunction)** - returns the originialCollection after it is filtered by the filter function
- **buildUrl(resourceTitle, criteria)** - returns a resource's URL from VX-API's resource directory
- **buildJdsDateFilter(dateFilterOptions)** - returns a date filter string that is generated by dateFilterOptions
- **clearAllCache(domainString)** - deletes all cached fetches from Backbone.fetchCache in the domain specified (if no domainString is specified, all cached data will be deleted)

Building Applets & Resources



Writeback Resources

Writeback resources, or resources which save patient data, are currently developed separately from fetch resources. See the writeback page for detail.

Developing a Resource

Annotated example resources which complement this guide are provided in the rdk repository at `/product/production/rdk/resources/_example/`
(Note that this directory name only has a leading underscore because it is for examples. Real resource paths should not have leading underscores.)

Before you create a resource

- Ensure that it does not already exist
- Identify the already-existing utilities and subsystems that can help you create your resource instead of reinventing the wheel
- Understand that proper logging is very important to identifying problems with the resource server and other parts of the Vistacore stack, and understand how to use proper logging.

Create the resource file

- Identify the correct location to place the resource
  - All resources belong in /product/production/rdk/resources/
  - Closely-related resources which use the same subsystems and which have similar scaling profiles should be placed next to each other

The convention of creating a resource file is:
`/resources/(functionality)/(functionality)Resource.js`
where the functionality is a camelCase identifier.
For example, given a resource to expose a list of allergens known to the VA, the following file would be created:

Next Steps

Next Steps for OSEHRA

- Produce an Install Manual to be used by Open Source community
- Create more robust Open Source code drop
- Work with the VA Innovations team to create an Open Source development environment
- Work towards more Open Source specific documentation.

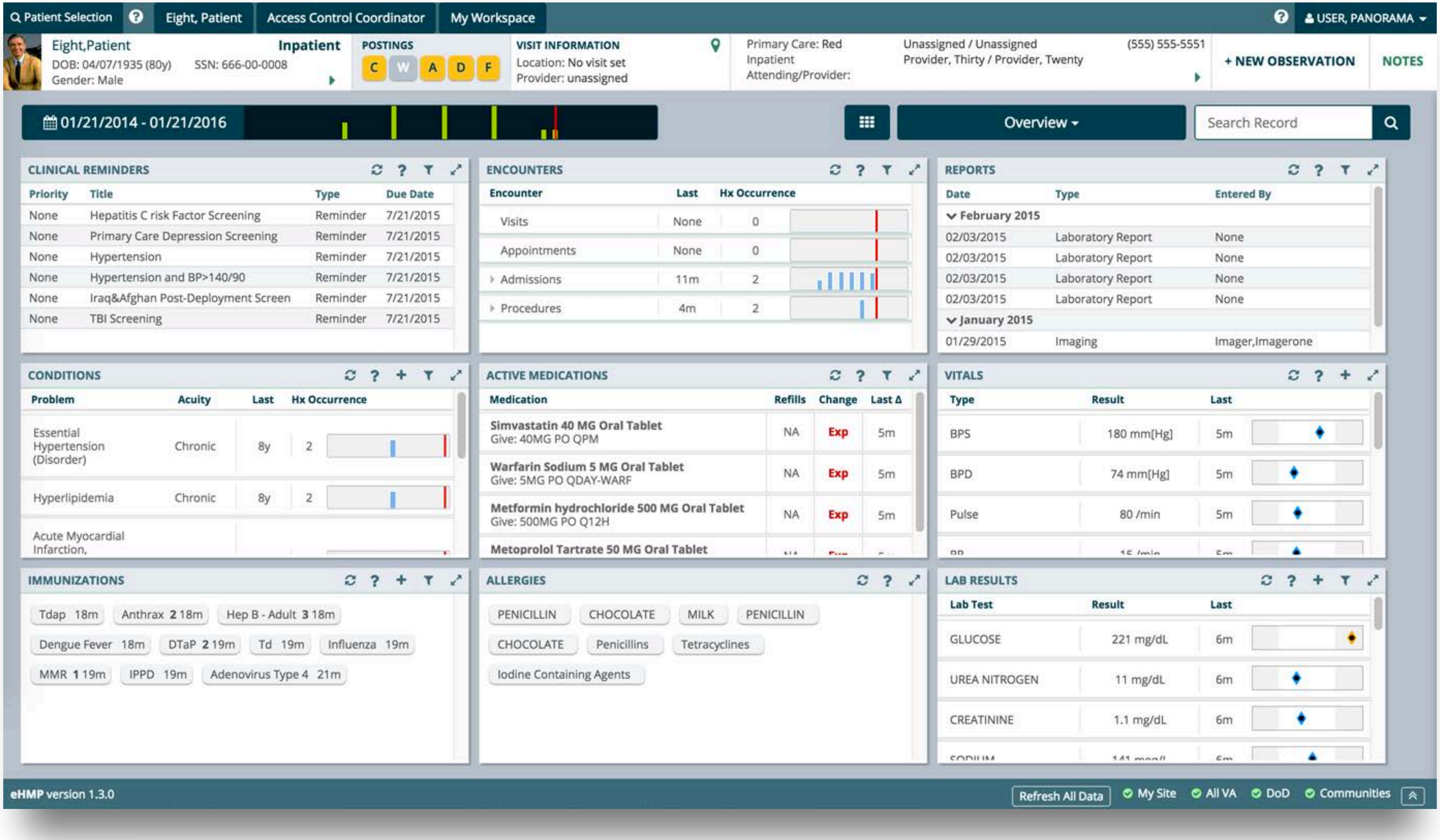
Questions

Follow Up Questions

Please pose all follow up questions to:

eHMPOpenSource@asmr.com

eHMP (Overview Workspace)



Eight,Patient Inpatient
DOB: 04/07/1935 (80y) SSN: 666-00-0008
Gender: Male

POSTINGS
C W A D F

VISIT INFORMATION
Location: No visit set
Provider: unassigned

Primary Care: Red
Inpatient
Attending/Provider:

Unassigned / Unassigned
Provider, Thirty / Provider, Twenty

(555) 555-5551

CLINICAL REMINDERS

Priority	Title	Type	Due Date
None	Hepatitis C risk Factor Screening	Reminder	7/21/2015
None	Primary Care Depression Screening	Reminder	7/21/2015
None	Hypertension	Reminder	7/21/2015
None	Hypertension and BP>140/90	Reminder	7/21/2015
None	Iraq&Afghan Post-Deployment Screen	Reminder	7/21/2015
None	TBI Screening	Reminder	7/21/2015

ENCOUNTERS

Encounter	Last	Hx Occurrence
Visits	None	0
Appointments	None	0
Admissions	11m	2
Procedures	4m	2

REPORTS

Date	Type	Entered By
February 2015		
02/03/2015	Laboratory Report	None
02/03/2015	Laboratory Report	None
02/03/2015	Laboratory Report	None
02/03/2015	Laboratory Report	None
January 2015		
01/29/2015	Imaging	Imager,Imagerone

CONDITIONS

Problem	Acuity	Last	Hx Occurrence
Essential Hypertension (Disorder)	Chronic	8y	2
Hyperlipidemia	Chronic	8y	2
Acute Myocardial Infarction,			

ACTIVE MEDICATIONS

Medication	Refills	Change	Last Δ
Simvastatin 40 MG Oral Tablet Give: 40MG PO QPM	NA	Exp	5m
Warfarin Sodium 5 MG Oral Tablet Give: 5MG PO QDAY-WARF	NA	Exp	5m
Metformin hydrochloride 500 MG Oral Tablet Give: 500MG PO Q12H	NA	Exp	5m
Metoprolol Tartrate 50 MG Oral Tablet			

VITALS

Type	Result	Last
BPS	180 mm[Hg]	5m
BPD	74 mm[Hg]	5m
Pulse	80 /min	5m
RR	16 /min	5m

IMMUNIZATIONS

Tdap 18m · Anthrax 2 18m · Hep B - Adult 3 18m · Dengue Fever 18m · DTaP 2 19m · Td 19m · Influenza 19m · MMR 1 19m · IPPD 19m · Adenovirus Type 4 21m

ALLERGIES

PENICILLIN · CHOCOLATE · MILK · PENICILLIN · CHOCOLATE · Penicillins · Tetracyclines · Iodine Containing Agents

LAB RESULTS

Lab Test	Result	Last
GLUCOSE	221 mg/dL	6m
UREA NITROGEN	11 mg/dL	6m
CREATININE	1.1 mg/dL	6m
SODIUM	141 mg/dL	6m

eHMP version 1.3.0