DRAFT

RxCLAIM Modernization

RxCLAIM Next Generation

OPTUMRx®

DRAFT

Overview

Five to Ten Year Project

Multi-Million Dollars

Long Term Strategic Initiative

Challenge:

Preserve Technology Leadership that OptumRx Currently has

Thoroughly Modernize All Aspects of RxCLAIM Over Time and In Place

Identify a Series of Projects that Work towards Overall Goals

RxCLAIM Current State

DRAFT

Considering RxCLAIM Ecosystem:

- RxCLAIM, RxBuilder, Eligibility, Accumulators, etc. etc.

Current Development Technologies In Use:

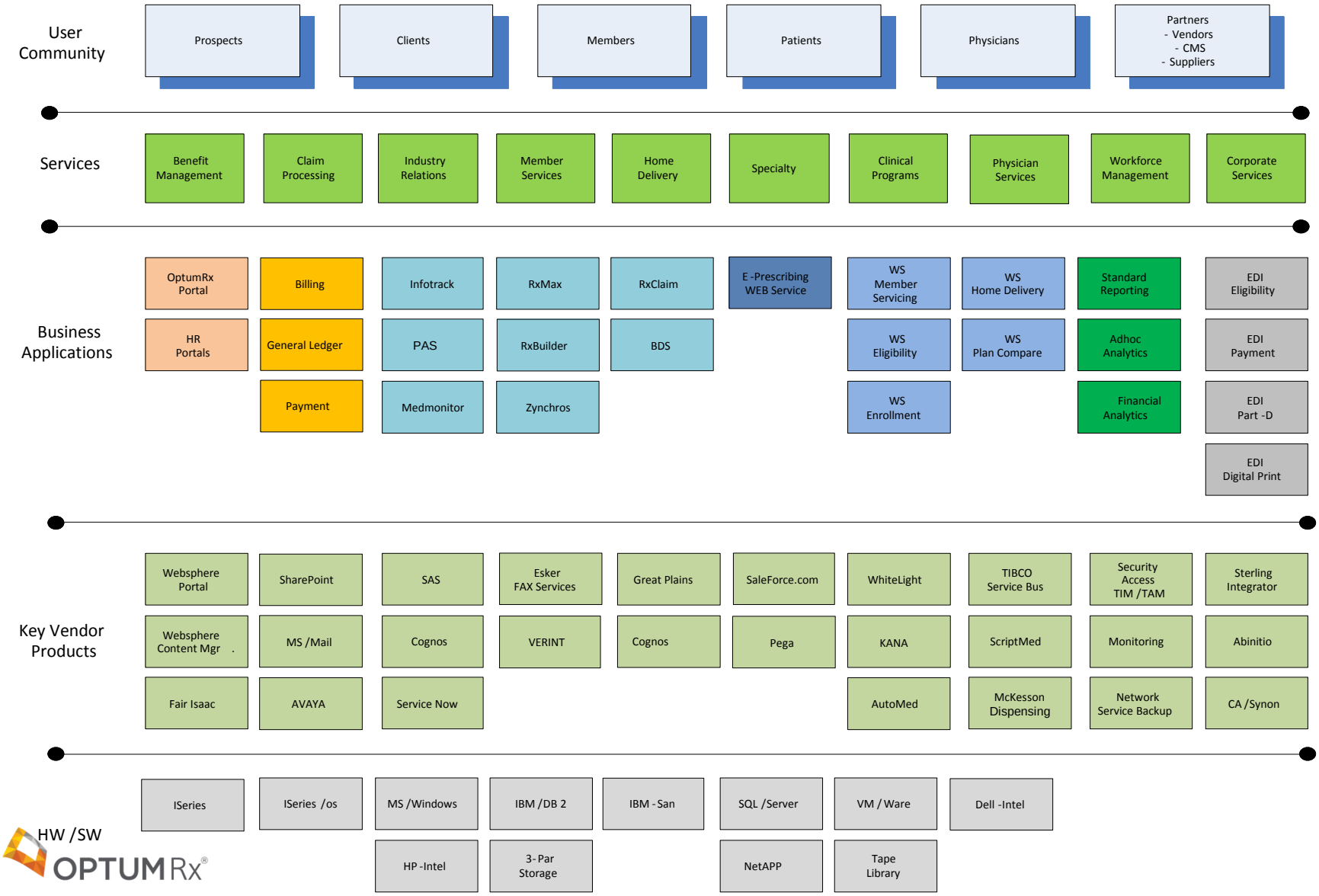
- Synon, COBOL, C, Java, RPG, Robot/TWS, SQL, DDS, Legasuite, MQ Series, Tibco

Core Application 30+ years old

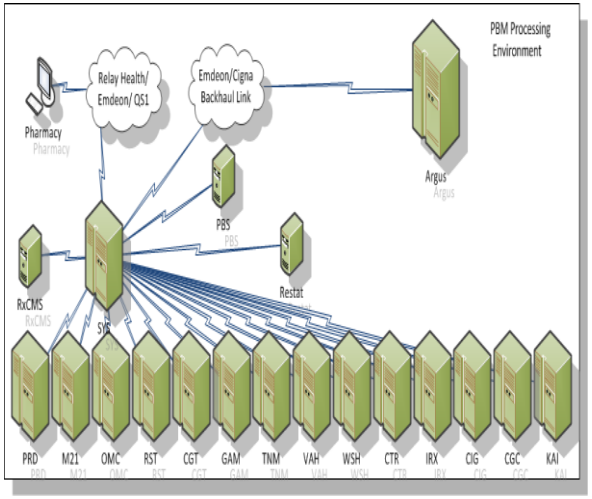
IBM I

Application Deployments and Upgrades – manual processes

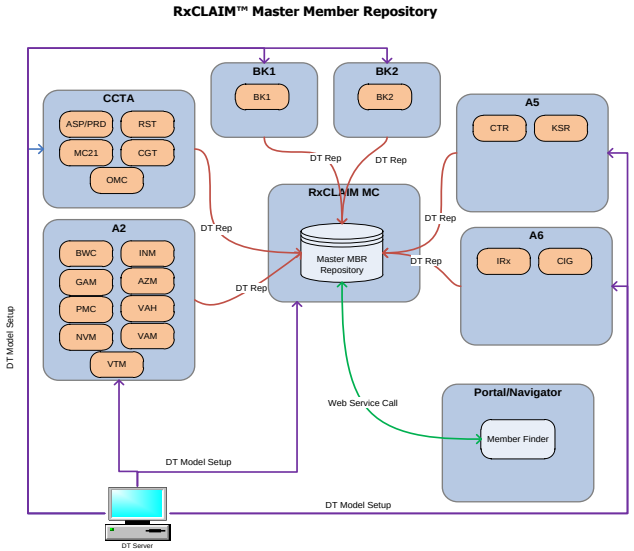
RxCLAIM Application Landscape DRAFT



Current Architecture



Future Infrastructure Architecture



Future Application Architecture

2017

2018

2019

2020

RxClaim Modern Phase 1
- Strategy & Roadmap
- Centralize Eligibility

Book1 BIN/PCN
Create Book 3 and 4
Archive / Purge
Migrate A7 to Minn DC

Data Center Migration
- Same HW, O/S
- Lift and Shift

RxClaim Modern Phase 2
- Centralize Ben Admin

RxClaim Application Modern
- Reengineer, Rewrite, Refactor

OPTUMRx®

RxCLAIM Modernization Goals and Baseline

Scalability

CPU - P9 architecture in 2019, CPU has a fair bit of headroom

Memory Usage - Could build out onto another frame if RAM becomes an issue, upper limit will be moving to 32 TB in the P8 lifecycle

Disk Space - We can only utilize 1/3 of the disk array due to DR/Flash copies using current iASP/SAN Replication strategy

- Batch system and generally environment size needs to be addressed
- Look at history, eliminate flash copies, purge/archive (project launched)

Table Size - 4B/1.7TB records per partition, Index Size - 1.7TB limit on spanning indexes

- Significant limitation compared with relevant competing platforms

Table Partitioning - 256 Partition Limitation

- SQL joins between multiple partitioned tables can impose further partitioning limitations due to query rewriting in from clause

RxCLAIM Modernization Goals and Baseline

Recoverability

Role Swap Timing - 10-20 Minutes for L-CTRx Role Swap

- MC/MAX - 20 Minutes, Book1/Batch 40-45 Minutes, Book2 - 20 Minutes
- RxMAX - 23 minutes
- Working with IBM to make Varyon and Varyoff faster
- Smaller LPARs to limit role swap timing

Active/Passive vs Active/Active - Currently evaluating dbMirror

- Many tradeoffs, especially storage space requirements, write commit latency

iASP and SAN level replication – Increases storage space requirements

- Secondary system unavailable for query access – requires “batch” system

RxCLAIM Modernization Goals and Baseline

Performance

Claim – current average response time - .1 - .4 seconds, varies by environment – critical to maintain current response time

Batch –Future evolution:

- Move more reporting to DW/BI and Big Data solutions
- ETL Tools – Ab Initio, Datastage
- SQL Enablement and Multi-threading
- Ongoing need for IBM I Base Batch Processes
- Look at replication implications for large batch processes

Web Services / API/ SOA – Need to have better control/management

- Long term SOA strategy

RxCLAIM Modernization Goals and Baseline

Code Organization and Modularity

Code and Tool Limitations - Synon

- Synon has size limitations on a module
- Generated COBOL is hitting limitations on module size
- PDM Source File Limitation
- Compiled Program Size Limitations
- Eliminated comments in order to stay within line limits
- Move to ILE, evaluate generating RPG instead of COBOL

Skill Sets

- Difficult to attract top talent with Synon/COBOL mix

SQL

- Limited adoption – mostly outside of OLTP
- Increase use of stored procedure, expand SQL usage

RxCLAIM Modernization Goals and Baseline

Future of Specific Dev Technologies

Synon

- Proprietary to CA
- Majority of Core RxCLAIM uses Synon
- 4GL assists in productivity and consistency
- Vast number of screens and other functionality
- Migration away from Synon - especially Synon/COBOL
- Move to ILE generation
- Look at Synon/RPG4
- Refactoring and SOA enablement

RxCLAIM Modernization Goals and Baseline

Future of Specific Dev Technologies

COBOL

- Substantial investment in COBOL for batch applications
- Skill sets are not there for COBOL out of college
- COBOL standard is remaining relatively static
- Migration away from COBOL
- ETL Tools, Spring Batch

RPG

- Substantial investment in RPG
- Majority is RPG 4 not Freeform
- RPG is not commonly taught in college, but uses familiar principles
- Proprietary to IBM I
- Future RPG programs should use Freeform
- Carefully evaluate options when considering new RPG programs

RxCLAIM Modernization Goals and Baseline

Future of Specific Dev Technologies

C/C++

- Small footprint in RxClaim
- RxBuilder and RxMAX utilize C extensively for core functions
- Skill sets are readily available, but not commonly used for business code
- Highly portable – recommend steady state

Java

- Batch and Reporting Java used for RxMAX
- External Web Interfaces (Max, Builder, etc.)
- Custom Applications (RTA, RxMACRO, ERA)
- Skill sets are readily available, commonly taught in college
- Probably growth area in RxCLAIM modernization

RxCLAIM Modernization Goals and Baseline

Future of Specific Dev Technologies

OPM

- RxCLAIM largely OPM for Synon
- Difficulty in mixing OPM and ILE
- Majority of RPG is ILE
- Majority of COBOL is OPM
- Single threaded
- Reaching source limits
- Mixing OPM with ILE tends to cause performance issues
- Resource scoping and activation groups are not manageable in OPM
- Recommend move to ILE for all applications

RxCLAIM Modernization Goals and Baseline

Future of Specific Dev Technologies

OPM

- RxCLAIM largely OPM for Synon
- Difficulty in mixing OPM and ILE
- Majority of RPG is ILE
- Majority of COBOL is OPM
- Single threaded
- Reaching source limits
- Mixing OPM with ILE tends to cause performance issues
- Resource scoping and activation groups are not manageable in OPM
- Recommend move to ILE for all applications

RxCLAIM Modernization Goals and Baseline

Modern Architecture and Development Concepts

Services Oriented Architecture

- ESB/MQ web services
- Minimal external services from inside adjudication - MBI, AARP
- RxBuilder – internal SOA for formulary,
- hosted on same IBM I environment
- Future growth area, part of modularity, modernization, scalability initiatives

Cloud

- No currently identified need for public cloud usage

Big Data

Distributed/Scale Out vs Scale Up

APIs

Development Contracts/Test Driven Development

Multi-threading vs Multi-Process

RxCLAIM Modernization Goals and Baseline

Modern Architecture and Development Concepts

Big Data

- Populated most important data Book1/Book2 to Data Lake (~130 files)
- L-CTRx environments populated to Data Lake from extract files
- Extract files not always up to date
- Data quality issues
- Currency of data can be an issue
- Still have many extract requests coming back to RxClaim dev - should be satisfied out of data lake
- Clean, consistent, comprehensive data warehouse environment to enable greater reporting and extract serviced by big data environment
- Better partnership with data lake

Distributed/Scale Out vs Scale Up

APIs

Development Contracts/Test Driven Development

Multi-threading vs Multi-Process

IBM I Platform Discussion

DRAFT

Platform Issues and Limitations

Table size limitations out of step with competitors, including DB2/Universal

Current static row length leads to either wasted space or block chaining

- would like to see dynamic row length, increased use of variable/null columns

Most DDL Most DDL changes result in data migration/conversion, which is challenging for large tables – need Catalog/Dictionary only updates

- would like to see adding nullable columns, increasing length of variable width columns not require a migration or conversion

No index hints – creates risk for OLTP implementations

- no optimizer is, or will be perfect, hints allow use of potentially sub-optimal, but highly predictable SQL execution

Continue work on SQL optimizer

- currently very sensitive to index definitions

IBM I Platform Discussion

DRAFT

Current State and Recommendations

Ongoing commitment for short and medium term

Excellent performance, resource manageability

Good support for standards and modern tools and techniques

Move to more open and portable technologies (e.g. Java, open source)

Mitigate long term risks, CA, IBM Power I Platform

Next Steps

DRAFT

Set Up Workstreams and Projects

- Disk space, Archive and Purge
- Eligibility Modernization
- Infrastructure Strategy Forum
- Centralized Benefit Administration
- Development Tools and Technology Evaluation and POC
- …