Modernizing VistA A Flexible Low Risk Approach

12575 U.S. Highway 1 | Suite 200 | Juno Beach, FL 33408 | OFFICE: 561.284.7000 | FAX: 561.227.0208

dssinc.com

Table of Contents

Executive Summary

"The Veteran that is receiving treatment at the Miami VAMC deserves the same quality care as the Veteran receiving care in Spokane VAMC." This statement demonstrates the need for a unified, consistent EHRM user experience across the VA enterprise. DSS is proposing bridge solutions to fulfill VistA modernization gaps in legacy technology for the next 10 years or until Cerner EHRM is fully implemented. VA envisioned a core benefit of the current EHRM project was to write off the technical debt of the VistA EHR by replacing it. In this sense, technical debt is the implied cost of software maintenance and enhancements that are meant to be eliminated by implementing a new COTS system with a "rip and replace" approach. Unfortunately, the reality is that EHRM is costing more and taking longer to implement while technical debt continues to accrue for the VistA EHR, which must remain coordinated with complex critical care clinical demands, time management optimizations due to workforce hiring challenges and most importantly, to optimize patient safety opportunities by using technology to assist busy clinicians.

Technical will increase as the federal government undertakes the most extensive EHRM transformation in the history of VA, which serves over 9.9 million patients and is utilized by over 350,000 clinical team members. History demonstrates that most commercial EHRs, including EHRM, carry decades of technical debt of their own. The current timeline necessitates investment in keeping VistA modernized and relevant to adequately serve patients while delivery of the new Cerner EHRM may take upwards of 10 years. The Cerner EHRM project is compounding VistA technical debt in multiple adverse ways:

- It is well known that unaddressed technical debt increases software entropy and the cost of maintenance and enhancements. Software entropy is the idea that software eventually rots over time if sufficient stewardship and care isn't taken to maintain coherence.
- Countless clinician requested enhancements have been sidelined for the legacy system because of the new EHRM which particularly impacts VA's scheduling, CPRS and pharmacy. Access to care is a top priority for patients. There are additional areas that have been deemed at the "end of life" that have lacked investment or another interim solution for the needed clinical functions.
- The VA will bear the cost of the product fragmentation (for example, technical debt) required by customizations of a commercial product to meet VA and federal government requirements.
- In a recent letter to Deputy Secretary Remy and Travis Dalton, former Cerner President, Representative Matthew Rosendale, ranking member of the House Subcommittee on VA Modernization, stated "I urge you to commence work immediately to fix the interface between Cerner's core electronic health record, PowerChart, and its pharmacy module, Medication Manager Retail (MMR). Until a true bidirectional interface is put in place, pharmacists…will struggle with a needlessly complicated, time-consuming, error-

> prone, double-entry process to prescribe medication that saps productivity and puts Veterans at risk."

In contrast, DSS proposes an alternative path to modernize VistA whereby we reduce and/or mitigate technical debt and patient safety risks though proper stewardship and targeted modernization of the current VistA EHR. A bridge can be created between VistA modernization and the Cerner EHRM implementation through this strategy. This flexible yet impactful VistA modernization approach addresses the fact that clinicians will be forced to use unmaintained VistA as technology advances over the next 10 years, thus impacting patient care during this time. We can accomplish this VistA modernization goal by separating the modernization effort into smaller, easier-to-digest and strategic milestones occurring in parallel to the Cerner EHRM roll-out. This helps ensure healthcare equity for all Veterans treated at VA facilities as Cerner is implemented in stages across the country.

These modernization solutions are contained within three primary focus areas:

1. Core VistA Modernization Solutions
2. VistA Module Modernization Solutions
3. CPRS Modernization Solutions.

A few of these solutions leverage DSS' suite of Juno ERH products, and all incorporate currently available VistA and CPRS modernization solutions. These focus areas are explained in more detail within Addendums 1-3 of this white paper.

Our approach to VistA modernization enables VA to select options from among the three primary solution groups as shown in following table and graphic:

Solution/Innovation (Core VistA)	Brief Description of Problem/Solution
DSS VSOA **(*Quick Win)**	DSS VistA API framework & integration tools for rapid integration of COTS systems to VistA; enhances VistA as an extensible platform for enhanced care without disrupting up/down stream dependent internal or external systems.
Time Zone Sensitivity for VistA **(*Quick Win)**	Each legacy VistA instance is limited to one time zone currently; VA has expressed a desire to continue to consolidate instances and even without consolidations, enterprise systems that read data from multiple VistA instances have to reconcile time differences.
RPC Broker Enhancements **(*Quick Win)**	Data flowing from client applications such as CPRS and BCMA that use RPC broker calls to view/save data from VistA are currently unencrypted. This is a significant security risk since the data flowing is PHI/PII. This change closes a large security hole in the legacy system.
Remediation of PCE Files to support evolving medical coding data sets	The legacy VistA Patient Care Encounter (PCE) Package connects to the VistA Clinical Lexicon for taxonomies and allows users to codify patient diagnoses, procedures, etc. The code sets used in HIT are changing and to meet industry standards/regulatory requirements and internal research initiatives, updates are needed.

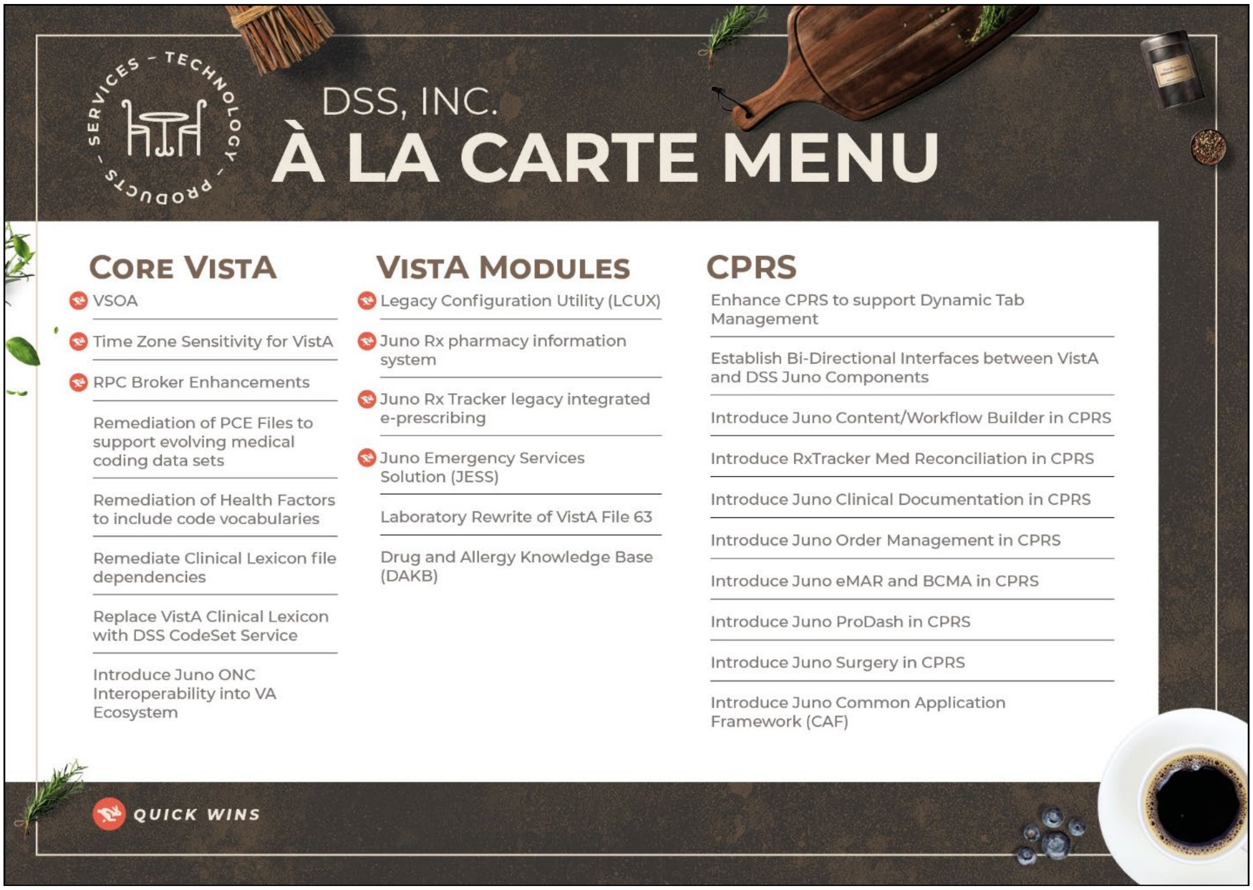
Solution/Innovation (Core VistA)	Brief Description of Problem/Solution
Remediation of Health Factors to include code vocabularies	This change is linked to the PCE changes and means that existing and extended code vocabularies could be used with the legacy VistA/CPRS health factors to store codified information that can be used for quality, reporting, and interoperability.
Remediate Clinical Lexicon file dependencies	This change is linked to the solutions listed above. Structural improvements are needed to sustain and remediate clinical lexicon file dependencies over the next 10 years.
Replace VistA Clinical Lexicon with DSS CodeSet Service	The current lexicon update utility could be replaced by flexible, extensible, and user friendly DSS CodeSet service. This is also linked to the three solutions listed above.
Introduce Juno ONC Interoperability into VA Ecosystem	VA has been exempted from ONC/MU but as a result, needed changes in support of industry standards for interoperability are falling further behind. This would bring VA to full industry standard interoperability compliance.

Solution/Innovation (VistA Module)	Brief Description of Problem/Solution
Legacy Configuration Utility (LCUX) **(*Quick Win)**	VistA is difficult to configure/maintain without years of specialty knowledge which is being lost as SMEs retire. DSS created a GUI for VistA module configuration that is user friendly and would be a significant improvement to the HIT staff at CO or sites who need to update VistA for 10 more years.
Juno Scheduling	VA has struggled with Enterprise scheduling for many years; DSS Juno scheduling is based on a successful VA pilot but is Cloud native and fully integrated to VistA.
DSS Juno Rx Tracker legacy integrated e-prescribing **(*Quick Win)**	DSS offers the only VistA-integrated e-prescribing solution for community pharmacy orders from CPRS.
DSS Juno Emergency Services Solution (JESS) **(*Quick Win)**	VA has long ED wait times and lacks a modern ED system across the enterprise; this DSS application can streamline ED processes and help decrease overutilization of non-VA ED/UC.
Laboratory Rewrite of VistA File 63	VistA lab package requires updates to continue to serve VA for the next 10 years.
DSS Juno Rx pharmacy information system **(*Quick Win)**	DSS created a GUI for VistA lab that provides Cloud-native UI to an outdated package; pharmacists in VA have extended scope of practice as care providers and need a modern pharmacy solution that is multi-formulary capable and does not disrupt ongoing daily operations.

Solution/Innovation (CPRS)	Brief Description of Problem/Solution
Enhance CPRS to support Dynamic Tab Management	This solution allows VA/DSS to introduce modern web applications to legacy CPRS without disrupting ongoing operations or making each change to the CPRS code base, which is time consuming.
Establish Bi-Directional Interfaces between VistA and DSS Juno Components	Allows VistA and DSS Juno components the ability to exchange data between systems. An interface engine facilitates messaging between Juno components, VistA instances and other applications producing or consuming HL7 messages.
Introduce Juno Content/Workflow Builder in CPRS	User interface to create and visualize custom workflows, note templates and Modules as well as enabling needed decision support/team coordination.
Introduce RxTracker Med Reconciliation in CPRS	Significantly improves efficiency and safety versus current process. Obtaining an accurate medication history is an essential part of the reconciliation process.
Introduce Juno Clinical Documentation in CPRS	A suite of web-based products (Clinical Action Center, Flowsheets and Notes) that enable clinicians to perform all aspects of CPRS documentation/notes function.
Introduce Juno ONC Interoperability into VA Ecosystem	The VA has been exempted from ONC standards and as a result, needed changes in support of industry standards for interoperability are falling further behind. This solution would bring the VA to full industry standard interoperability compliance.
Introduce Juno Surgery in CPRS	The current CPRS Surgery Package is cumbersome and outdated without a modern GUI. The Juno Surgery Module is a web-based comprehensive solution for perioperative care.
Drug and Allergy Knowledge Base (DAKB)	DAKB is linked to Juno Rx but can be implemented separately. The VA uses FDB for drug interaction checking using MOCHA but has not implemented FDB Drug-Allergy checks and maintains all the drug/allergy data manually. This is at best inefficient and at worst unsafe. DSS can update CPRS to use the FDB DAKB.
Introduce Juno ProDash in CPRS	Juno ProDash introduces much needed web-based patient case management capabilities for behavioral health and primary care. ProDash can effectively replace the current CPRS Consults and Discharge Summary tabs.
Introduce Juno Order Management in CPRS	Replaces CPRS Computerized Provider Order Entry and consolidates all aspects of the ordering process: configuration of orders, order sets, order viewer and ordering in one web-based system.
Introduce Juno eMAR and BCMA in CPRS	Replaces separate Barcode Medication Administration application to accurately administer, document, and track patient medication administration events to ensure patients take and receive their prescribed medications.
Introduce Juno Common Application Framework (CAF)	As a fully web-based interface it allows for easy integration of both existing Juno EHR developed content as well as nearly seamless integration with other web-based third-party applications. This will effectively allow for CPRS to be deprecated in favor of a fully web-based Juno EHR user interface.

This tri-pronged approach (Core VistA Modernization/VistA Module Modernization/CPRS Modernization) allows VA to pick solutions independent of each other from different solution groups, such as VSOA, RPC Broker Enhancements, and Juno Scheduling. In this example, these options represent incremental solutions from among our three solution groups that DSS can implement in parallel in the first phase on enhancements. This option to pick solutions in an a la carte fashion from our solution groups gives VA maximum flexibility while retaining awareness of budgetary and technical constraints.

DSS Bridge Solutions



SERVICES - TECHNOLOGY - PRODUCTS

DSS, INC.

À LA CARTE MENU

CORE VISTA

- VSOA
- Time Zone Sensitivity for VistA
- RPC Broker Enhancements
- Remediation of PCE Files to support evolving medical coding data sets
- Remediation of Health Factors to include code vocabularies
- Remediate Clinical Lexicon file dependencies
- Replace VistA Clinical Lexicon with DSS CodeSet Service
- Introduce Juno ONC Interoperability into VA Ecosystem

VISTA MODULES

- Legacy Configuration Utility (LCUX)
- Juno Rx pharmacy information system
- Juno Rx Tracker legacy integrated e-prescribing
- Juno Emergency Services Solution (JESS)
- Laboratory Rewrite of VistA File 63
- Drug and Allergy Knowledge Base (DAKB)

CPRS

- Enhance CPRS to support Dynamic Tab Management
- Establish Bi-Directional Interfaces between VistA and DSS Juno Components
- Introduce Juno Content/Workflow Builder in CPRS
- Introduce RxTracker Med Reconciliation in CPRS
- Introduce Juno Clinical Documentation in CPRS
- Introduce Juno Order Management in CPRS
- Introduce Juno eMAR and BCMA in CPRS
- Introduce Juno ProDash in CPRS
- Introduce Juno Surgery in CPRS
- Introduce Juno Common Application Framework (CAF)

QUICK WINS

Our flexible solution set allows the VA to make decisions about new initiatives for VistA enhancements and investment to run in parallel with the Cerner EHRM project and keeps the VistA platform viable even as parts of the legacy system are replaced. In the event that the EHRM project has significant delays or ultimately fails to achieve a national deployment, this approach gives VA a safety net with a modernized VistA platform that can scale up to a full replacement if necessary. Most importantly, this approach is forward thinking to put Veterans at the front of the solution for patient safety, access, quality, and clinician optimization so there is more time to spend with patients on their actual care versus working in antiquated, systems that are not adequately maintained for the next decade.

Modernizing VistA – A Flexible Low Risk Approach

Within the last five years, VA has assumed enormous and uncounted risk in the form of both patient safety as well as congressional budgetary implications associated with the "rip and replace" implementation of a new COTS based EHR into the Veterans Health Administration (VHA) ecosystem. The lack of a concurrent modernization backup plan for the existing VistA EHR ecosystem significantly compounds these risks. VistA must continue to operate indefinitely in parallel with EHRM replacement efforts, and as a contingency plan if EHRM replacement efforts do not meet all required functionality. Regulatory changes, security updates, and technology investments must be made to provide Veterans and VistA users across the nation with innovation and reliability for the next ten years. In an August 2021 article published within FCW, Roger Baker, former VA CIO, discussed "Why VA Must Keep VistA Healthy". This compelling article focuses on the need for risk mitigation as the following statements indicate:

- *'There is no "burning platform" that compels VA to take risks in its program to replace VistA'*
- *'It is unclear what plans VA has for dealing with mandatory VistA upgrades needed during that timeframe, such as new national medical mandates (e.g., completed ICD-10 upgrades) or medically urgent system changes (e.g., tracking and controlling opioid prescriptions)'*
- *'…the moratorium on further Vista investment…creates at least four major strategic issues for VA, each of which creates the potential for a major impact on Veteran medical care if not addressed.'*

We commend the VA for enabling the establishment of a modernization platform by moving to the Cloud to leverage an abundance of cloud services. This is the beginning of an enterprise-level SOA-based approach. However, the VA must acknowledge past implementation challenges on large VA "rip and replace" projects such as:

- **IFCAP**: Core Financial and Logistics System/CoreFLS (VA Award $472M/$620M or 1.1B in total)
- **IFCAP**: RISE (VA Award $114M)
- **Supply Chain**: Financial and Logistics Integrated Technology Enterprise/FLITE (VA Award $400M)
- **Supply Chain**: RTLS (VA Award $543M)
- **Lab**: Cerner Lab/LSRP (VA Award $20M)
- **Revenue Cycle**: Patient Financial Services System/PFSS (VA Award $62M)
- **Scheduling**: Southwest Research Scheduling/RSA (VA Award $127M)
- **Scheduling**: Medical Appointment Scheduling System/MASS (VA Award $623M+32M)
- **Surgery**: Surgery Quality Workflow Management/SQWM (VA Award $55M)
- **CPRS/VistA**: iEHR/eHMP/VistA4 (VA Award $286M)
- **Interoperability**: VistA Service Assembler/VSA (VA Award $52M)
- **Interoperability**: Enterprise Messaging Infrastructure/EMI (VA Award $16M)
- Healthcare Associated Infections and Influenza Surveillance/HAIISS (VA Award $40M).

None of these projects were as large and as ambitious as the current Cerner EHRM rollout (VA Budgeted $16B).

The problems, challenges, and failures associated with the programs listed above foreshadow a potential derailment of the latest “rip and replace” initiative – the Cerner EHRM project. This is demonstrated by several key patient safety concerns including: lack of ability to flag a patient record, medication ordering and dispensing errors, delayed and erroneous prescriptions, missing data associated with the data migration process, bottlenecks in referrals to specialists or community care, and system outages, just to name a few. Based on past VA experience, these are serious issues, which are systemic both vertically and horizontally across the enterprise and have the potential to seriously impact the current EHRM project.

In sharp contrast to large “rip and replace” projects with a demonstrated history of high-risk projects as shown above, strategic, smaller scale initiatives have been reliably more successful. We recommend a more measured approach to modernizing VistA in parallel while the Cerner EHRM is underway. Many of these were initiatives led by DSS – a company who has partnered with VA for years and understands the complexity of the legacy system and operational needs. In addition to self-developed solutions for VA, DSS has assisted more than thirty-five commercial vendors to integrate with VA systems. DSS has supported VistA in non-VA deployments since 2006, and our helpdesk for our current VA applications has a 4.89 out of 5 satisfaction rating. We have a 99.94% first response compliant rate and a 99.89% compliant rate for service delivery windows, proving that DSS can support VA users as these parallel VistA enhancements are deployed. Our product implementations have been highly successful and have had unprecedented positive impact on improving care for Veterans.

Underscoring VA’s unique business model, no commercial healthcare applications have been integrated without “unique-to-VA” modifications. Furthermore, it is evident that VHA policies and procedures will continue to be non-conformant to many healthcare practices supported by a typical commercial EHR. These established VA workflows have meaning and purpose to the Federal Government and need to be honored to ensure legislative requirements are met for (but not limited to) Veterans Equitable Resource Allocation (VERA), Veterans' disability compensation, budget preparation, billing, planning, determinations of access and market coverage, quality reviews, and other multiple performance drivers. VA Integrated Business Teams must have flexibility regarding infrastructure requirements and business rule establishment to meet directives for care coordination and care/disease management, to comply with requirements of standard, intermediate, and complex surgical procedures, as well as services provided in a Veterans Affairs Domiciliary, Community Living Centers, or Multi-Specialty Community Based Outpatient Clinics.

The work performed by DSS, Inc. over the past three decades as a loyal vendor partner of the VA has proven that it is feasible to modernize VistA components at all levels of the VA technology stack; maintain the best elements of VistA; and augment VA workflow with fully integrated solutions that meets VA business and regulatory requirements. The remainder of this assessment will reflect what type of low-risk components can be introduced in timeframes that are comparable to, or an improvement on, current VA EHRM standards. DSS will also describe how these VistA modernization components can be introduced into the highly complex VA technology environment without disrupting VA clinical and/or financial workflows.

Conclusion – Bringing Multiple Solutions Together

To summarize, the VA has low risk choices from among our three solution groups when considering modernization of their legacy VistA EHR. These groups allow the VA to select multiple options for Core VistA Modernization Solutions, VistA Module Modernization Solutions, and CPRS Modernization Solutions. The VA has already embraced cloud platforms, and they have rolled out DevSecOps and Product Line Management plus many other new technologies and practices. VistA Modernization has arrived in a way that meets VA regulatory requirements and preserves VA workflows. The current EHRM “rip and replace” project has been beset with issues, such as at Spokane and Walla Walla, which implemented the full Cerner Millennium solution, and also at Columbus where only Cerner Scheduling was implemented.

DSS provides a logical path toward VistA modernization that is not sole source or locked into one vendor but rather enables the VA to continue selecting the best commercial applications for use within a standards based EHR platform across the VA. VistA should also be modernized to meet auditing and other security needs (that rightfully drew criticism) and pushed the EHRM project forward as if no alternative approaches existed. The DSS modernization plan ensures VistA works well for all VA sites no matter where they are in the EHRM implementation timeline while also providing a viable contingency option for sustainment if Cerner deployment continues to be delayed or ultimately fails to meet the VA’s needs.

DSS can introduce any or all of the above-mentioned VistA modernization solutions to best meet VA’s needs for the next ten years without losing any of the VA-specific data elements that VistA has been built to capture and store. This is imperative as a robust VA and wider Federal ecosystem exists that encapsulates VistA both upstream and downstream which is highly dependent upon VistA data. These systems contain 30 years of business and regulatory rules specific to how government and the VA work. Failing to provide a single dependent data point, such as appointment wait times for oversight reports, means VA lacks key data and cannot provide current levels of transparency to Congress or the Veterans they serve regarding scheduling. This example was noted by OIG in its report about EHRM Scheduling, but it begs the question “what about the many other data points that aren’t as well known?”

While many of the VistA modernization solutions that DSS proposes fit within legacy VistA packages and data structures, our approach also involves superimposing the Cloud native DSS Juno EHR framework around VistA to initially wrap and renew parts of the VistA EHR, and to gradually replace the thick client user applications such as CPRS and BCMA with Cloud native enterprise EHR capabilities. This allows new functionality to be introduced, helps meet changing regulatory needs, and promotes codifying and tagging data elements in a manner that is compliant with 21st Century Cures Act regulations for data sharing, security, privacy, and analytics. It also allows the system to move from current data silos within VA data centers to a scalable, enterprise, multi-tenant VISN or VHA wide single EHR instance. In this scenario, workflows are preserved or enhanced without wholesale disruption.

The new modernized VistA-aware Juno EHR platform would ensure key clinical data sets exist while leveraging the existing and highly VA specific VistA data elements to properly continue to share data with all upstream and downstream government systems that are dependent upon VistA's VA specific data which are used to meet Congressional reporting mandates, determine Veteran eligibility, and impact VA funding in addition to direct Veteran medical care. A phased VistA modernization implementation across the VA means changes to the backend data base and frontend user facing applications are applied in a systematic, low risk manner. Operational continuity is a paramount need for the VA while training has been another significant source of frustration and employee dissatisfaction with the current EHRM deployment in part because of the disruption the "rip and replace" strategy introduces. With this approach to modernization, institutional change management is accomplished in a logical way so that training is minimized since change is gradual and data integrity is preserved.

The VA must invest strategically in VistA for at least another decade, and perhaps longer, depending on the revised timeline and final outcome of the Cerner EHRM project. The VA and DSS have a low-risk option for quick wins and to modernize parts of VistA without negatively impacting patient care. DSS can provide the needed bridge for enhancing legacy VistA VA sites to ensure VA staff can continue to safely provide the "best care anywhere" while protecting Veteran data, meeting changing regulatory requirements, engaging in interoperability with community providers, and offering the latest innovative health IT advancements for the foreseeable future.

Appendix I

Core VistA Modernization Solutions – the 'What' and the 'How'

*{*Quick Wins}*

Initial modernization efforts should focus on "quick wins" that can rapidly improve existing technology and operations for much needed enhancements and usability gains for both the system, end users, improved Veteran care coordination, and/or to meet regulatory requirements. The following DSS solutions can be deployed in the near term, providing low risk and high reward across the VA Enterprise:

- *VSOA
- *Time Zone Sensitivity for VistA
- *RPC Broker Enhancements
- Remediation of PCE Files to support evolving medical coding data sets
- Remediation of Health Factors to include code vocabularies
- Remediate Clinical Lexicon file dependencies
- Replace VistA Clinical Lexicon with DSS CodeSet Service

Each of the above noted "quick win" solutions bring rapid modernization and enhancement, at the system level, without negatively impacting current operations and represent a much-needed step toward ensuring VA sites using legacy VistA do not fall behind. While the remaining optional core enhancements bring much needed change to critical VistA infrastructure that will remain provide VistA the flexibility needed for future modernization.

***What is VistA SOA Suite (VSOA)?** In 2014, DSS developed VSOA, a unique, flexible, high-performance API engine built specifically for VistA that enables rapid development of modern and innovative web-based solutions. VSOA is a flexible, lightweight yet powerful middle tier "scaffolding" that supports numerous industry-standard service-layer technologies for communicating with thick and thin client applications. Originally developed to provide organic, VistA-savvy SOA application services for DSS products, VSOA has evolved and adapted to recent and ever-changing VA standards, VA architectures, and development needs. The VSOA API engine is complemented by VSOA Cube – an API development console designed to simplify the "construction" of application specific APIs using a point-and-click interface. A VSOA API is composed of simple (YAML formatted) VistA RPC definition files (RPCD) which quickly expose/transform legacy VistA RPC logic into standard RESTful web services in addition to RPC orchestration handlers that simplify existing VA business rule knowledge necessary to safely file data to VistA. VSOA also (automatically) generates standardized OpenAPI documentation. In short, VSOA both modernizes and standardizes the web application interface for VistA applications.

VSOA was ready to deploy before VA awarded a contract for VistA web services which resulted in the failed VistA Service Assembler (VSA) project. While admirable in concept, the VSA creation of a VistA-based SOA layer using a combination of a Federating Services Platform and 'Assembler Wizards' did nothing to address the fragility introduced into the system by using consumer-facing contracts. Worse, it made it easier to create and introduce such fragilities while simultaneously and dangerously bringing consumers closer to the implementation details of VistA itself. VSA required VistA M routines to act as adapters; VSOA requires no VistA patching. VSA produced faux JSON where array numbers were the “name” and the up-careted data was the “value;” VSOA produces true name-value output that fully describes VistA API output. VA Medical Domain Web Services (MDWS) also failed as an enterprise SOA solution. Encapsulation and the arbitrary introduction of Medical Domain Objects shifted the dependency problem from the service platform to service domains specific to the data while the domain objects were not fully (and with an ever-changing VistA could never be fully) representative of the actual data.

VSOA is approved by VA TRM, currently deployed in the VA Enterprise Cloud (VAEC), and is used by multiple DSS applications in production today. VSOA standardizes development based on open and accessible technologies and fosters innovation by removing roadblocks between modern solutions and legacy VistA. DSS has converted hundreds of VistA RPCs to web services including all the most used CPRS, TIU, Registration, Vitals, Scheduling, etc. APIs. VSOA has allowed us to rapidly modernize our products and innovate with new web-based applications that were not possible in the past.

> **How will VSOA be introduced into the VA ecosystem?** Since the current implementations of VSOA are licensed for use by specific VA business owners running DSS products, a separate enterprise installation would be necessary to provide web services to VistA for new applications. DSS can provide a VSOA Production Binary for Windows, macOS, or Linux installation plus the vsoaSettings.yaml, associated RPCD, and handlers to be copied to a VSOA Workspace folder. Steps to set up VSOA within a Docker container can also be provided, as well as additional configuration for certificates, etc. The separate VSOA Cube installer for developer workstations is also available for Windows, macOS, or Linux-Ubuntu.
>
> A single VSOA server can host communications for multiple VistA instances. On startup, VSOA establishes connections to configured VistA servers, gathers each VistA server sign-on message, and proceeds to build a reservoir of pooled connections for that VistA server. Pooled Connections are an important aspect of working with VistA through the XWBTCP protocol because each user requires their own VistA connection. Having a reservoir of pooled connections speeds up the inherently lengthy VistA connection and login process. Each connection is identified, maintained, and secured through an encrypted token delivered on a successful login from VSOA to the application on behalf of the user. In turn, VSOA manages a user's VistA session and all subsequent transactions through VSOA require this encrypted token.
>
> The Security Framework for VSOA is based on the concept of schemes and strategies. Out of the box, VSOA implements the JsonWebToken (JWT) strategy. Per HSPD-12, VA applications must be PIV-enabled and integrate with VA Access Services. Accordingly, VSOA implements the requisite VistA sign on protocol that requires a VA STS or WebSSO issued SAML token. This SAML token is digitally signed by a trusted Service through PKI and provided to VistA during the sign on process. Following the transmission of the SAML token, VistA then pairs the identity defined in this token with a provisioned VistA account.

***What is the Time Zone Sensitivity Framework for VistA?** The time zone sensitivity framework is a prebuilt set of DSS-namespaced M routines and Cache (IRIS) Objects custom made for a VA VistA database. The lack of support for multiple time zones within a single VistA instance has been a known issue for a long time. The architecture and early evolution of VistA enabled years of development to occur without consideration of the issue of time zone sensitivity. The decentralized nature of the VA's original hospital information system meant that client applications only needed to provide the local date/time; the database stored that information and generated timestamps based on local server date/time. However, the VA is moving toward "cloud first" architectures and enterprise applications. Remote access to VistA is increasing and has been in place a long time, for example, VA Clinical Contact Center users in Dayton, OH service Veterans all over the West Coast. The VA has also sought the consolidation of its existing 130+ VistA instances into a more manageable number to lower support costs and activities. These changes, particularly consolidation, either benefit from, or require, VistA time zone support.

Time zone support allows the database to be hosted anywhere while keeping timestamps relevant to the time zone where user input is occurring. In VistA, support for multiple time zones would allow a single VistA instance to support hospitals in multiple time zones. It would also render the time zone of the database's host server irrelevant. In the current paradigm, many VA facilities are geographically close in proximity but reside in different time zones requiring VistA to be hosted on different servers to support local date/time. This adds complexity and cost for administrating these systems. Therefore, support for multiple time zones represents a cost and complexity savings which is part of our proposal for a clear, and relatively inexpensive, path to achieving this modification.

How will the Time Zone Sensitivity Framework be Implemented within VistA: DSS has developed a KIDS build, VFDTMZ, which contains all components necessary to implement the VistA-side changes for this enhancement, but this is only part of a comprehensive time zone solution. VA applications now utilize several different types of user interfaces (for example, web, GUI, and roll-and-scroll) with hundreds of remote procedures (RPCs) to support modern applications. If the VA adopted the standard model of Coordinated Universal Time (UTC) conversion which occurs on the client application with the database only handling date/time storage, all these components would be impacted. VistA is currently backwards in its current time zone support infrastructure – the database is performing conversions rather than the clients.

In the release of FileMan 22.2, the VA partially addressed the issue of time-zone sensitivity by introducing the Universal Time data type. This new data type provided all three of the basic components needed for time-zone sensitivity:

1. Converts a local date/time to UTC using an input transform connected to the data type.
2. The input transform has access to the time zone rule database built specifically for the data type. The time zone used is primarily based on the users DUZ(2), a variable identifying the facility the user logged in to.
3. Stores the UTC formatted date/time in a FileMan date compatible format.

By encapsulating the conversion, application of Daylight-Saving Time rules, and storage in one data type, the VA has enabled VistA applications to be time-zone sensitive while avoiding having to remediate the UI's, RPCs, and the supporting code. The Universal Time data type relies on an "in-house" time-zone database which is defined by VA staff.

Most technologies needing access to time zone definitions and rules utilize one of two time-zone databases: the Time Zone (tz) Database or the Microsoft Time Zone Database. The tz Database, managed by the Internet Assigned Numbers Authority (IANA), is the most used because it is in the public domain, updated regularly, and is relatively easy to incorporate into other technologies. Technologies such as Delphi, SQL Server, the .NET Framework, and others, have adopted the tz database as the standard source for time zone definitions and rules. DSS developed an alternative to the Universal Time data type to address the weakness of the VA's solution (particularly using the in-house time zone database) and provide flexibility in terms of moving to a more common architecture. Like VA, DSS utilizes a data type in its VFDTMZ KIDS build to handle the conversion, rules, and storage and it can be set up in the same way as the Universal Time data type.

However, there are two key differences of the DSS solution:

- DSS uses the tz database as the source of time-zone definitions and rules, making the DSS solution more compatible with standard development technologies.
- The input transform is designed to recognize a UTC formatted date/time allowing the conversion process to be bypassed, if necessary.

Initially, the DSS data type can be used in the same way as the FileMan Universal Time data type, accepting a local time and then handling the conversion, rules, and storage. As VistA applications evolve to handle time zone in the client, the DSS data type is flexible enough to handle these changes. The DSS data type can take the local time and time zone from the UI and handle the conversion and the storage. Or if the UI converts the local time to UTC, the DSS data type can handle just the storage. Time-zone sensitivity, or converting a local date/time to Coordinated Universal Time (UTC), requires three basic components:

- A conversion utility to convert the local date/time to UTC.
- A set of rules for the conversion utility to use, defining what the time zones are and what rules to use for each (e.g., default offsets, when daylight saving time (DST) transitions occur, etc.).
- A data type within the database allowing a date/time (including seconds and the time zone's offset) to be stored.

The inputs into the conversion utility are the local date/time and the time zone of the event being recorded. Ideally, both inputs would be coming from the client application, as the client can accurately get both data items, and the conversion would be done on either the client side or the database side. On most non-FileMan databases, the client is responsible for both acquiring the input and doing the conversation, while the database is only responsible for storing the UTC formatted date/time. Once a data type for implementing Universal Time has been introduced to VistA, the next step in creating a multiple time zone solution is to analyze which existing date/time fields should be converted to Universal Time. This can be approached incrementally by VistA package (I.e., CPRS, Pharmacy, Lab, Radiology, etc.). Once the date/time fields have been identified for conversion, additional VistA modifications will be required to include any client application, routine, API, RPC, Data Object, or report that stores, displays, edits, or prints so that the new date time displays in the desired format. These modifications will be the single largest effort in updating the data to Universal Time.

What is RPC Broker Logging? The RPC Broker Logging/Audit enhancement improves upon VistA FileMan audit capabilities (which is limited to the database) by logging events where users might view, query, or print but not change the database data itself. Once logged, the site administrators can review and report on HIPAA-sensitive issues. The improved RPC Broker and patient-related RPC modifications are coupled together to provide seamless and thorough audit logging, which can be used to meet HIPAA and Meaningful Use (MU) requirements to protect Protected Health Information (PHI) and Personally Identifiable Information (PII) sensitive data.

How will RPC Logging be Introduced into VistA: Two VistA KIDS builds are needed to implement RPC logging/auditing: VFDOXB to implement logging, and VFDAUD to provide auditing reports. Also, the XWBDLOG routine within the VistA legacy RPC BROKER package needs to be edited to add a single line of code to call an entry point in the VFD namespace. RPC Broker logging/auditing is necessary to achieve and maintain MU certification. The purpose of RPC logging supported by the VFDOXB KIDS build is to put context around PHI and PII by capturing the details of RPC calls and logging those details in an audit log. The details required for capture include:

- Who made the RPC call (based on the user’s DUZ), and the name and purpose of the call
- What application initiated the call, and what data is included in the call (the file, field number, and data)
- Where the RPC was executed (the server, namespace, etc.)
- When the call was executed

RPC logging is not a replacement for FileMan’s intrinsic auditing features. It instead augments FileMan auditing by filling in the gaps in FileMan’s auditing functionality. While FileMan accurately captures writes to the database, it does not capture reads and cannot address applications not using FileMan. RPC logging provides a way to increase the VistA’s auditing capabilities by logging RPCs passing PHI and PII from an UI to the FileMan database. RPC auditing consists of three steps, (1) Capture the RPC data as it is sent between the application UI and the M database and store it in a temporary global, (2) Process the data stored in the temporary global and prepare it for storage, and (3) Store the RPC data in an auditing file.

The initial capture of the RPC data is done by introducing a hook in the RPC Broker’s debugging routine, XWBDLOG, to capture the raw RPC data and store it in a temporary global. In addition to the RPC data, basic call information such as the name of the RPC, the date and time of execution, as well as the server, namespace, IP address, and name of the calling application are recorded. This initial step is done without any data processing and is performed in the background to avoid creating any latency in the execution of the RPC. A second background process is used to prepare the RPC data for storage in an audit file. The processing utility is initiated by TaskMan and can be scheduled to run as frequently as deemed necessary. The purpose of this utility is to parse the raw data and organize it in a standard format. Because the content of an RPC varies from RPC to RPC, the processing utility relies on a processing routine written specially for the RPC to organize the data. Processing routines can be attached to an RPC either by adding a field in the REMOTE PROCEDURE file or creating a separate file specifically designed to store this relationship. The processing utility also handles the saving of RPC data. A FileMan based file is created specifically for the storage of the RPC data. Maintenance of the library of processing routines will need to become part of the VistA development process. As new RPCs are written and existing RPCs are changed, new processing routines will be created, or existing processing routines will be modified. The capture and processing utilities, as well as many of the VistA RPC-specific processing routines, have already been developed by DSS.

Identified Modifications to Core VistA: VA must accept that core VistA routines must be modified with this enhanced RPC broker framework. For logging to occur, a change was implemented within a core RPC broker transaction routine: LOG^XWBDLOG was modified to include one line of foreign code to “hook” into a DSS routine (RPCLOG^VFDXWBA) to allow for additional broker auditing.

What is RPC Broker Encryption? A long time, and lesser known, security risk associated with executing M-based APIs within VistA has existed since the inception of the RPC broker in the form of unencrypted clear text being sent (via TCP/IP) to/from RPC broker-based applications, including but not limited to VA CPRS. VA mitigates this risk by having client applications and VistA within the VA network, but VA’s digital transformation will include Zero Trust which dissolves the traditional enterprise boundaries and requires secure transmissions regardless of location inside/outside the network. To mitigate this security risk, DSS proposes an RPC Broker Encryption enhancement that consists of a custom M routine and Delphi component available for immediate VA implementation. Encrypted RPC Broker calls are also available through the DSS VSOA engine which is an RPC broker API web service implementation.

How will RPC Encryption be Introduced into VistA centric legacy applications: The DSS RPC encryption enhancement consists of the following:

1. A modification of the VA’s Delphi RPC Broker code to support TLS encryption.
2. Implementation of the modified RPC Broker component into existing Delphi applications such as CPRS, etc.
3. Configuration of the Cache server to open a port for encrypted RPC Broker communication while applications still using the non-encrypted RPC Broker are phased out.
4. Deployment of an M routine (KIDS build name is TBD) to support encryption/decryption on the VistA side.

To properly secure the RPC Broker communication, DSS utilized a commercial Delphi TLS encryption component that captures the RPC call data as a string before it is transmitted through the TCP/IP stream. Within the RPC Broker Delphi component, Wsockc.pas needs to be modified to support encryption of the TCP/IP stream to/from VistA. The broker is also modified to attempt to use TLS encryption first and then default back to a non-encrypted state for legacy compatibility. The rest of the RPC Broker functionality remains unchanged.

Upon the recommendation of Intersystems to guarantee backward compatibility, DSS modified the native %ZISTCP routine to accept additional optional parameters to ensure that encryption occurs at the Cache port that feeds the RPC broker. This kernel routine is used by numerous M routines to open socket connections. DSS identified all these routines and tested to ensure encryption occurs between the VistA service and the desktop client via TCP/IP. The enabling of encryption is parameterized so that it can be installed prior to deployment and disabled if necessary while communication with VistA client applications is tested out.

Identified Modifications to Core VistA: VA must accept that core VistA routines must be modified with this enhanced RPC broker enhancement. For messaging encryption to occur across the enterprise, DSS recommends changing the %ZISTCP routine to accept an additional parameter used for encryption. Additionally, all broker-based applications will eventually need to be recompiled with the modified Delphi RPC Broker component. The VistA environments also require installation of a specific XWB version of the associated RPC Broker M routine (another legacy package routine change) to ensure encryption compatibility. VA can mitigate implementation timelines for this change, and not commit to a big-bang cutover, by configuring a second broker port within VistA to receive encrypted RPC calls. Any client that has been compiled with the encrypted RPC Broker would be configured to connect to the new (encrypted) port. Any legacy broker client would continue using the original RPC Broker port until they are recompiled and ready to support encryption.

What Is the Updated VistA PCE (Patient Care Encounter) File Structure? The PCE enhancements allow for greater interoperability, allow VistA to easily support new code vocabularies, and provide support for the concept of a FHIR Episode of Care. New PCE file structures will be introduced to extend and enhance the current PCE framework; these changes are completely backwards compatible with all existing VA upstream or downstream technology components and local VistA PCE data dependencies. The new files bring native support for any coding system introduced to the LEXICON package, e.g., DSM 5, Surgery AORN, Nutrition ANDUID or Nursing NANDA or NOC, plus the new files support multiple different coding systems within a single coding category, e.g., the coding category of Diagnosis must be able to support ICD-10, DSM 5, and SNOMED-CT codes. Furthermore, these enhancements support ONC Clinical Quality Measure Concepts (CQM) such as Procedure Not Done or Intervention Not Preformed. As a file-based enhancement no new M development is required to implement additional coding systems. If a new coding system is introduced in the LEXICON package, only a configuration change in the new PCE file(s) is required to support the new coding system. An important design constraint of the PCE enhancement is that pointers to the code files, for example, ICD-10 codes pointing to File 80, will no longer be used, thereby facilitating the eventual refactoring of the LEXICON package for simpler coding system updates. To support Episode of Care, existing PCE database structures need to be updated so that one or more outpatient VistA visits can be linked together. These VistA visits can be on the same day or across multiple days for series patients such as chemo-therapy patients, physical therapy patients, or mental health therapy patients.

DSS is recommending these changes to the current PCE structures to future proof VistA as new coding vocabularies are created and required for federal reporting purposes. For example, as part of ONC Meaningful Use Stage 1, SNOMED was introduced as a required interoperability coding vocabulary for national reporting, VA responded by creating a new V file with new M routines to support the new file structure. With the proposed modifications to PCE the introduction of a new required interoperability coding vocabulary would not require any new M development for PCE and would instead only require new configurations be added or existing configurations updated. Another feature of these modifications is the ability to document and codify procedures that were "not done" and why they were not done, a requirement for Clinical Quality Measure reporting that links directly to evidence-based care for the patient.

How will VistA PCE Evolve within VistA: A new KIDS build, VFDENC, includes new PCE files, menus, options, and APIs to support this change. Additionally, a minimal amount of legacy Order Entry and PCE code needs to be modified: remote procedures ORWPCE SAVE and ORWPCE PCE4NOTE, and APIs DATA2PCE and PXKENC APIs need to be modified with optional data inputs to support the new V ENCOUNTER CODE and V EPISODE OF CARE structures; these modifications will be *backward compatible* for all legacy calling applications.

The VFDENC KIDS build introduces four new FileMan database structures to VistA. The first is the V ENCOUNTER CODE file that will maintain all the various coding entries for the patient. The second and third are configuration control files so that code vocabularies can be stored in the V ENCOUNTER CODE file for each category of coding, e.g., Diagnosis, Procedures, etc. For example, Diagnosis could be configured to allow ICD-10, SNOMED, and DSM-5 as coding vocabularies. The last new file structure is the V EPISODE OF CARE file which allows the linking of one or more VistA visits to a single Episode of Care entry. To maintain both forward and backward compatibility, when data is filed directly into the existing V files it will also be automatically propagated into the new structures as appropriate and when data is filed directly into the new structure and there is a V file that supports the coding vocabulary it will automatically be filed into the existing V file structure.

DSS recognizes that current downstream systems such as CDW, FMS, AITC, VERA, etc. will not initially be able to use these new structures. Therefore, when possible, data is replicated in the traditional PCE V files that these systems currently use. These modifications will allow the downstream systems to make use of the richer coding vocabularies when VA chooses to update these systems as the data will already be in VistA. These new PCE structures, APIs, RPCs, and Menu options are currently deployed and used in DSS' commercial implementation of VistA and have been in place since 2018. Furthermore, these modifications allowed DSS to obtain several ONC 2015 meaningful use certifications.

Identified Modifications to Core VistA: VA must accept that core VistA RPCs and routines must be modified with this enhanced PCE framework. For full forward and backward compatibility, the remote procedures ORWPCE SAVE and ORWPCE PCE4NOTE as well as the APIs DATA2PCE and PXKENC APIs must be modified.

What is the Updated VistA Health Factors Framework? DSS has turned VistA Health Factors into their own coding vocabulary to better support workload reporting, national goal reporting, national monitoring, reminder resolution and other areas where coded data is desirable. Without coded HF data it is difficult for non-VistA systems to be interoperable with VistA using this important clinical dataset. It is also difficult to maintain uncoded data, so this type of data is simply left off the integration discussion. If HF data is not captured in non-VistA systems used for patient treatment, then it cannot be transmitted to VistA for reporting which in turn could result in VA under reporting workload to Congress. However, when HF entries are treated as a "local code set" as defined by VA then non-VistA systems could treat HF in the same manner as any other coding vocabulary and integrate the capture of HF codes into their standardized workflows. For example, Juno EHR's Clinical Content/Workflow Builder allows multiple ways to utilize standardized and local coding vocabularies so they are captured and then transmitted to downstream systems such as VistA. Also, once HF are converted to a local coding vocabulary, the codes can be used natively within any FHIR representation that supports FHIR CodeableConcept or Coding elements for data interoperability.

While Health Factors (HF) within VistA do maintain codes from multiple coding vocabularies such as SNOMED, ICD-10, CPT-4 that are linked to individual HF, this is not required. Specifically, most VA reporting functions do not have external coding vocabularies, so there are limited opportunities for external systems to send codified data to VistA to fulfil the VA reporting needs. DSS has solved this issue by making HF themselves into their own coding vocabulary with the HF code automatically assigned when the HF is created. This is *completely backwards compatible* with all existing VA upstream or downstream technology components and local VistA HF data dependencies.

How Health Factors Will Evolve as a Code Vocabulary within VistA: A KIDS build, VFDHF, includes the VistA components necessary to implement this change. New fields will be added to the VistA HEALTH FACTOR (#9999999.64) file plus new input transform routines are installed to automatically create the new HF codes and to codify all existing HF and HF Categories. Automatic coding makes use of the Station Number of the system creating the HF and the Station ID of the system that created the HF Category. Both national and local HF can be added to any HF Category and the HF code will still be able to be created automatically.

When transforming HF into a local coding vocabulary, we were careful to maintain full backward compatibility, including the ability for each VistA system to create a new HF which is globally unique code across VA; just relying on the HF internal entry number (IEN) assigned by FileMan is not sufficient to maintain uniqueness. VA has set aside various HF IENs for special use – this business rule is maintained. Additionally, the assigning of specific codes to each HF should be handled automatically at the local level as new HF are created. All HF modifications are backward compatible for the legacy calling applications.

As part of DSS' commercial VistA implementation (vxVistA), Health Factors were enhanced to be a local code vocabulary in 2018 and have been functioning as such at all our clients since 2018 with no reported issues. Several design steps were necessary to create this flexible solution. The first step was to add two new fields to the HEALTH FACTOR (#9999999.64) file: CAT CODE and HF CODE. CAT CODE is the Health Factor Category code and has the following form, YYY.XXX where YYY is the Station Number of the VistA system where the HF Category is initially created and XXX is just an incremental counter. While the creation of a new HF Category will automatically populate the CAT CODE field, the user can override the assigned values. Note that the pieces YYY and XXX of the CAT CODE are not limited to three numeric values the expand to the number of digits needed. The option to create HF Categories has been modified to allow the overriding of the automatically assigned CAT CODE value. HF CODE is the Health Factor code and has the following form, CAT CODE.ZZZ where CAT CODE is the Health Factor Category code of the HF Category that was assigned when this HF was created and ZZZ is the additional code that uniquely identifies this HF within the assigned HF Category. Since HF can be assigned to any HF Category, and a HF can be locally created, the format of ZZZ is more complicated to guarantee uniqueness across all VistA systems. The algorithm to create the ZZZ portion of the HF CODE is as follows:

> *If the Station Number of the CAT CODE is the same as the Station Number where the current HF is being created, then the ZZZ value is the next open integer value less than 100,000. If the Station Number of the CAT CODE is different than the Station Number of where the current HF is being created, then the then ZZZ will be the next open integer value of all the health factors in that category between the values of YYY000000 and YYY999999 where YYY is the editing systems station number*.

A post install routine will automatically assign both CAT NUM and HF CODE for all currently defined Categories and Health Factors that do not have these values defined.

Finally, the PXRM Code Set File used for clinical reminders is updated with a new entry for Health Factors with both the short name and full name set to HF. Additional modifications to the PXRM code base will allow HF to be used within VA generated Value Sets.

What is the Updated VistA V-File Structure? The implementation would enhance the existing V-Files (V POV, V CPT, V HEALTH FACTOR, etc.) to remove the dependence on internal Lexicon and coding files (ICD DIAGNOSIS and CPT for example) to allow for more flexible codification of elements.

> **How will VistA V-Files be Updated to Remove VistA's Clinical Lexicon Dependency:** A VistA KIDS build, VFDO, contains necessary routines and Data Dictionary updates, including new fields added to selected V files to maintain the Code System, Code, and Clinical Description. The existing pointers to the coding files will be repurposed and Output Transforms will be created to pull the data from the newly created fields instead of the existing pointers. The APIs used to store and retrieve this data would also be updated to make use of these new structures appropriately. This would help to remove the current limitations that exist in these files because they are directly tied to the Lexicon and Code System-specific files. The ICD Diagnosis and CPT files cannot be referenced directly per VA directive, so all access is coming through well documented API entry points that can be modified to store and pull data to the newly created fields, without much remediation.

What are the Juno EHR Code Set Services? The Juno EHR Code Set Service is a cloud based clinical terminology management system that provides users with common utilities to access codes and value sets needed to meet regulatory requirements for documentation of patient care. This service also provides functionality for maintaining and updating code systems and value sets, including allowing for the creation of custom entries. The service will be able to import codes from the nationally recognized and industry standard code systems, such as SNOMED-CT, ICD-10, and LOINC.

> **How will the Juno EHR Code Set Services be Introduced into the VA Technology Ecosystem:** Juno Code Set is part of Juno EHR Common Services. The web server components and enterprise MSSQL Server database are installed in the cloud. A VistA KIDS, VFDO, will include modification of current LEX, ICD, and ICPT package APIs to route the calls to the Juno EHR Code Set Service via VistA web service calls. The Juno Code Set Service Server and Service will be defined using an entry in the VistA WEB SERVER (#18.12) and WEB SERVICE (18.02) files to be created as part of the VFDO post-installation routine. These entries allow the endpoint to change without requiring source code changes. The installation guide will step the installer through additional manual steps to configure the web service. Existing code lookup functionality will remain the same, such as being able to limit searches by subsets (like CPRS uses for the Problem List). As with the V-File remediation, Data Dictionary updates would be made in the VFDO KIDS to remove and replace direct pointers to ICD and ICPT files to be replaced by a minimum of three fields: CODE, CODE SYSTEM, and DESCRIPTION. Output transforms will also be added to the altered pointer fields so that the proper data is still returned by any legacy application using FileMan APIs to retrieve the data.

Appendix II

VistA Module Modernization Solutions – the 'What' and the 'How'

*{*Quick Wins}*

This phase allows for the much-needed modernization of critical VistA modules/clinical domains areas necessary to treat the Veteran patient in a modernized environment while transforming many of the legacy workflows. Initial modernization efforts should focus on "quick wins" that can rapidly improve existing VistA module technology and operations for much needed enhancements and usability gains for both the VistA modules, end users, improved Veteran care coordination, and/or to meet module regulatory requirements.

The following DSS solutions can be deployed in the near term, providing low risk and high reward across the VA Enterprise:

- *Legacy Configuration Utility (LCUX)
- Juno Scheduling
- *Rx Tracker legacy integrated e-prescribing
- *Juno Emergency Services Solution (JESS)
- Laboratory Rewrite of VistA File 63
- *Juno Rx pharmacy information system

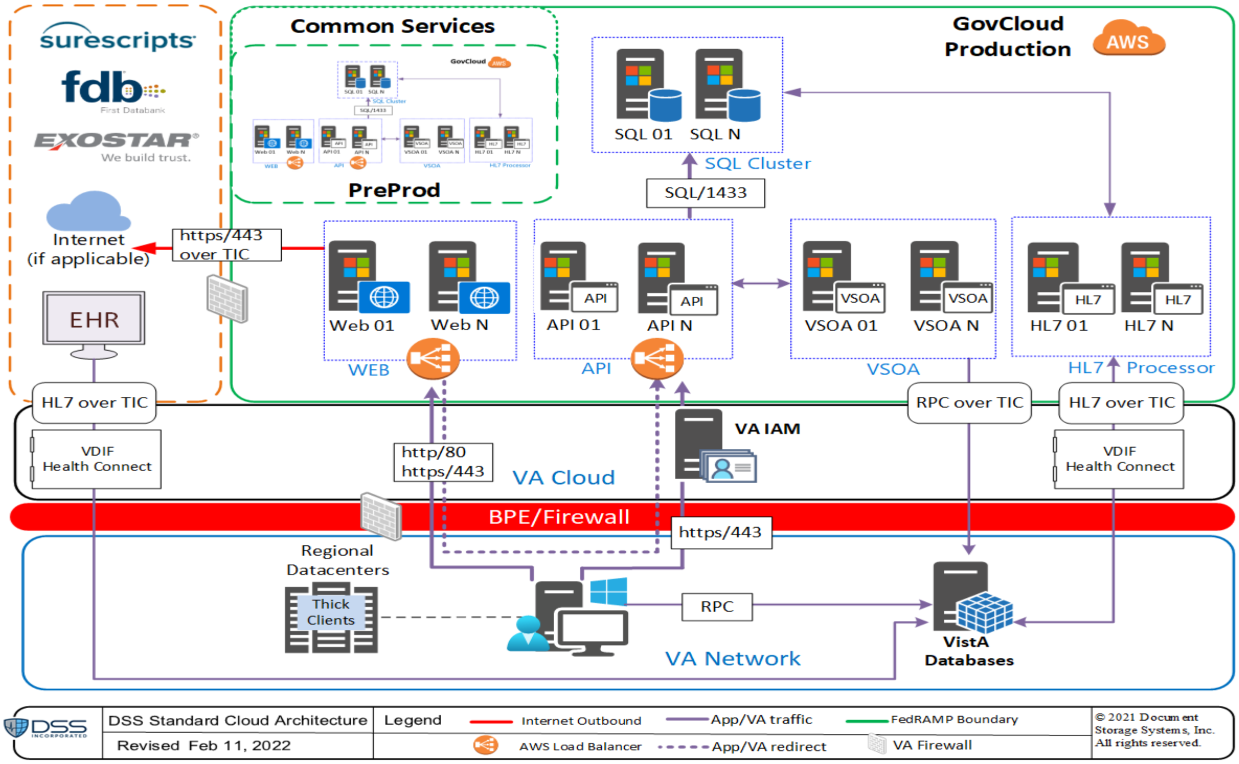
These VistA module modernization solutions provide a way to close gaps in the legacy system modules and opens new pathways to innovation.

What is Legacy Configuration Utility? Legacy Configuration Utility (LCUX) is a configuration tool in Juno EHR that provides convenient modern UX experience to maintain VistA configuration files currently edited via roll-and-scroll options, for example: User setup, Pharmacy setup, Ethnicity, Race, Sexual Orientation, Gender Identity, Language, Relationship. User Management is streamlined and self-contained, so users do not have to jump around to many different options to setup demographic data, keys, options, person class, user class, NPI, Authorization to write Med orders, DEA#, etc. The same is true in Pharmacy. LCUX replaces many roll-and-scroll Pharmacy set up screens including Drug Enter/Edit, Medication Route Enter/Edit, Standard Schedule Edit, and many other site parameter and configuration options. Users are presented with modern, consistent views and edit actions to replace FileMan input template, ScreenMan and List Manager edits which utilize and require cumbersome function key knowledge and 'jump' capabilities to skip fields. While the presentation within LCUX is new, all VA business rules for editing data remain in place. Common business practices are also supported, for example, LCUX User Management can clone an existing user or create a new user from scratch.

> **How will Legacy Configuration Utility be Introduced into VistA:** LCUX is a web server application that runs on a client's browser. The LCUX web server components and DSS VistA SOA Suite (VSOA) are installed in the cloud, while the application VFDBLC KIDS build is installed on each site's VistA instance. VSOA is the lightweight middle tier used to communicate with VistA using JSON APIs. There is no external (non-VistA) database for LCUX. Both the LCUX Web App Server and VSOA Server are compatible with Linux, macOS (OS/X) and Windows operating systems. The following diagram illustrates the

DSS Standard Cloud Architecture used by many of the products being discussed in this document:

DSS Standard Cloud Architecture



What is Juno Scheduling? A successor and continuation of the ForSite 2020 (F2020) application which is a VistA centric registration and scheduling solution being used by VHA within the Indianapolis VAMC. **Juno Scheduling** is an enterprise wide, rules-based, resource scheduling module that provides "conflict free" scheduling for any department throughout the enterprise. Rules to ensure conflict free scheduling include scheduling rules, clinical rules as well as billing rules. To provide the patient with a most convenient scheduled time, functionality includes searching for appointments across site boundaries for a conflict free first available time slot. For speed when scheduling multiple procedures, a feature known as "Auto Search" automatically coordinates conflict free times both within a department as well as across multiple departments. Inbound orders and faxes automatically update scheduling work queues for the scheduling process to begin. The **Juno Registration module,** a complementary module of Juno Scheduling, is an enterprise-wide registration module to allow for a "**quick registration**" to ensure demographic data integrity that integrates with VA's Master Patient Index for all sites to share the most current demographics, contacts, and billing information. To assist with financial clearance, real time insurance verification may be launched updating the patient's insurance data within seconds. To alleviate bed assignment issues, a consolidated Bed Board supports both inpatient and observation patients bed assignments, transfers, and discharges. The configuration of the registration module includes system as well as site specific tables to meet unique requirements throughout the organization.

How will Juno Scheduling be implemented: Juno Scheduling is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud, while the application DSIS KIDS build is installed on each site's VistA instance to support HL7 integration. Juno Scheduling sends and receives a variety of HL7 message types: ADT, SIU, SRM, and REF. The DSIS KIDS build and VistA's native HL7 package are leveraged to send inbound and outbound scheduling and patient registration data between the two systems. A Juno Scheduling specific set of HL7 transactions will be defined along with event driver protocols, subscriber and client protocols, logical links and other necessary components supporting VistA inbound and outbound messages. Each protocol will include VA SAC-compliant processing routines to assemble and/or parse messages. The DSIS KIDS build was successfully deployed to VA production VistA instances to support the Faster Care mobile app pilot project and the legacy Juno Scheduling product, ForSite2020, a Best in KLAS system. ForSite2020 has been used by the Indianapolis VA Medical Center since 2000. HL7 messaging will be transmitted over VA's internal network and/or internet. An interface engine, such as InterSystems HealthShare, will be required to facilitate messaging between the cloud hosted Juno Scheduling solution and the site VistA instances.

What is RxTracker? RxTracker is a CPRS integrated ePrescribing solution currently being piloted by VHA VISN 4 supporting full provider clinical workflows and allowing Veterans a choice to fill prescriptions at VA or community retail pharmacies. The RxTracker application is more than a stand-alone application to route orders and provide medication care guidelines, it also serves as an integration platform providing robust RESTful web service APIs as well as a variety of standardized HL7 messages that passively seed pre-identified data sources for future ePrescribing workflow as determined and integrated by an EHR. At its core, the RxTracker system is a comprehensive solution that gives medical providers the ability to effectively perform a full prescribing workflow for all inpatient, outpatient, and ambulatory care settings. RxTracker consolidates and automates the ePrescribing process for all medications, including controlled substances to support the Drug Enforcement Agency (DEA) Electronic Prescribing of Controlled Substances (EPCS). For example, if a patient is taking a combination of controlled and non-controlled substances, providers can create one medication list and RxTracker will manage the details by accessing the relevant state's Prescription Drug Monitoring Program (PDMP) database through an integrated PDMP interface.

All RxTracker functionality is fully compliant with Meaningful Use (MU) criteria and all state and federal regulations providing flexible two-factor authentication and identity proofing. Examples of MU criteria met by RxTracker are as follows: (b)(3): Electronic Prescribing, (d)(1): Authentication, Access Control, Authorization, (d)(2): Auditable Events and Tamper-Resistance, (d)(3): Audit Report(s), (d)(4): Amendments, (d)(5): Automatic Access Time-out, (d)(6): Emergency Access, (d)(7): End-User Device Encryption, (d)(8): Integrity, (g)(3): Safety-Enhanced Design, (g)(4): Quality Management System, (g)(5): Accessibility-Centered Design.

RxTracker leverages the First Databank (FDB) State and Federal Controlled Substances Module™ (including the most current State and Federal DEA schedules). With these features in-place, the RxTracker web service APIs and HL7 messaging framework can offer comprehensive ePrescribing workflow for an EHR. Since the EHR's medication ordering and results reporting workflow differs, the final integration details will depend on the capabilities of the EHR development and configuration staff, and well as the available EHR integration entry and exit points. An integration can be approached as a simple HL7 unidirectional or bidirectional implementation or may evolve into a more complex API-driven integration which allows users to view real-time RxTracker data, FDB data, and access Surescripts verification processes to name a few features.

RxTracker provides all high-level needs identified in a recent VA ePrescribing RFI, plus additional capabilities:

- Create and send eligible Veteran's prescriptions to a non-VA community pharmacy of the Veteran's choice
- Bidirectional integration with VistA/CPRS
- Distinguish prescriptions that originate in a VA clinic from those originating from a community care clinic for appropriate workflow consideration
- Validate data against the medical health record in accordance with VHA patient safety rules and to enhance quality of care and comply with state prescribing regulations

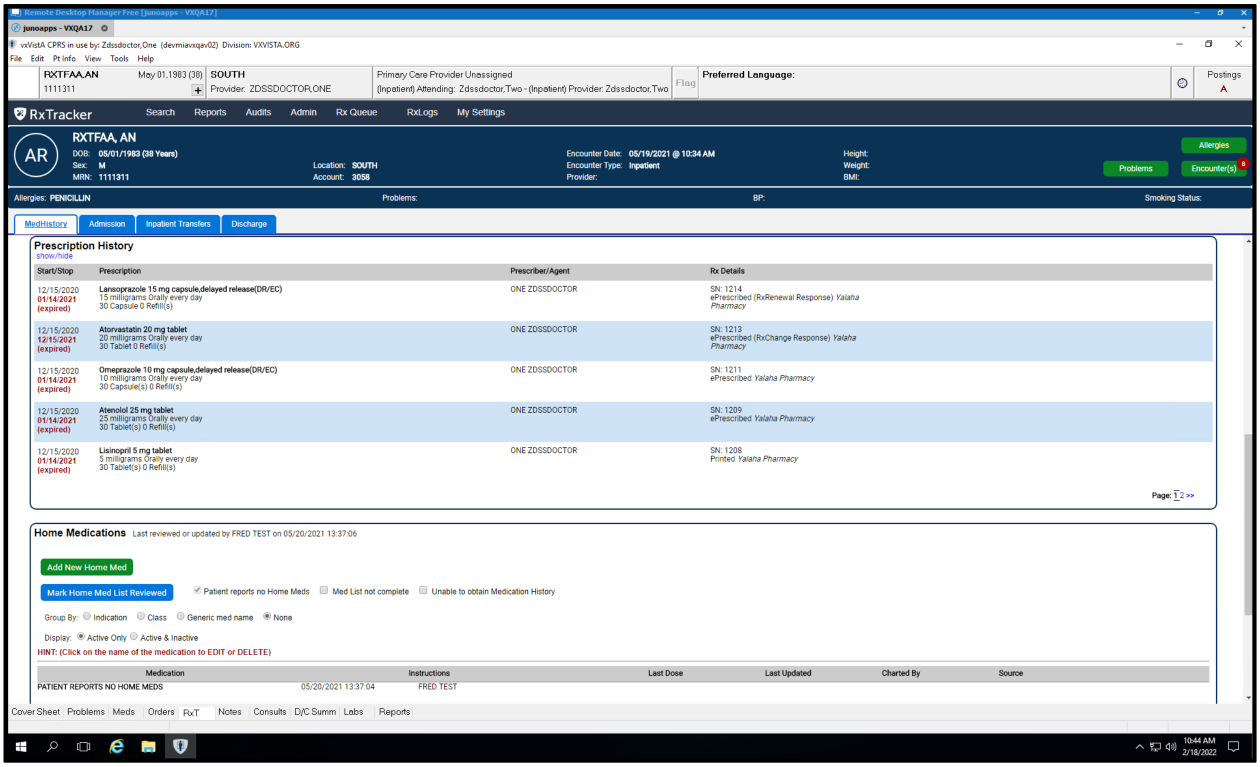
RxTracker's RxQueue module displays incoming messages from community/retail pharmacies. This dashboard allows VA to instantly receive confirmation that the retail pharmacy has received the order, conveniently review, respond to questions, and receive all prescription information (change request, renewal request). As part of the available integration, providers can be alerted to the presence of RxQueue messages as part of their normal CPRS workflow, distinguishing prescriptions that originated in a VA clinic from those originating from a community care clinic. RxTracker's RxFill module uses the bi-directional VistA/CPRS integration to provide the patient's medication history under prescription history. This feature reports Veteran retail prescription fill statuses and provides more insight into the Veteran's history of medication adherence than is currently available to VA today.

How will RxTracker be implemented: RxTracker is a web server application that runs on a client's browser. The web server components, enterprise Commercial MySQL database, and DSS VistA SOA Suite (VSOA) are installed in the cloud, while the application DSHX KIDS build is installed on each site's VistA instance. VSOA is the lightweight middle tier used to communicate with VistA using JSON APIs. Both the RxTracker Web App Server and VSOA Server are compatible with Linux, macOS (OS/X) and Windows operating systems. An interface engine, such as InterSystems HealthShare, will be required to facilitate messaging between the cloud hosted RxTracker solution and the site VistA instances.

Initially to be launched from the CPRS Tools menu, RxTracker could be implemented as a CPRS tab using the **Dynamic CPRS Tab Management** enhancement discussed in Phase II.

In this example, DSS RxTracker e-prescribing capability is embedded within CPRS leveraging DSS' Dynamic CPRS Tab Management approach. CPRS users gain new e-prescribing functionality without significant retraining or workflow disruption.

Dynamic CPRS Tab Management with DSS RxTracker e-Prescribing Integration



Remote Desktop Manager Free [junoapps - VXQA17]
junoapps - VXQA17
vxVistA CPRS in use by: Zdssdoctor,One (devmiavxqav02) Division: VXVISTA.ORG
File Edit Pt Info View Tools Help

RXTFAA,AN May 01,1983 (38)
1111311
SOUTH
Provider: ZDSSDOCTOR,ONE
Primary Care Provider Unassigned
(Inpatient) Attending: Zdssdoctor,Two - (Inpatient) Provider: Zdssdoctor,Two
Flag
Preferred Language:
Postings
A

RxTracker Search Reports Audits Admin Rx Queue RxLogs My Settings

AR
RXTFAA, AN
DOB: 05/01/1983 (38 Years)
Sex: M
MRN: 1111311
Location: SOUTH
Account: 3058
Encounter Date: 05/19/2021 @ 10:34 AM
Encounter Type: Inpatient
Provider:
Height:
Weight:
BMI:

Allergies: PENICILLIN
Problems:
BP:
Smoking Status:

Prescription History
show/hide

Start/Stop	Prescription	Prescriber/Agent	Rx Details
12/15/2020 01/14/2021 (expired)	Lansoprazole 15 mg capsule,delayed release(DR/EC) 15 milligrams Orally every day 30 Capsule 0 Refill(s)	ONE ZDSSDOCTOR	SN: 1214 ePrescribed (RxRenewal Response) *Yalaha Pharmacy*
12/15/2020 12/15/2021 (expired)	Atorvastatin 20 mg tablet 20 milligrams Orally every day 30 Tablet 0 Refill(s)	ONE ZDSSDOCTOR	SN: 1213 ePrescribed (RxChange Response) *Yalaha Pharmacy*
12/15/2020 01/14/2021 (expired)	Omeprazole 10 mg capsule,delayed release(DR/EC) 10 milligrams Orally every day 30 Capsule(s) 0 Refill(s)	ONE ZDSSDOCTOR	SN: 1211 ePrescribed *Yalaha Pharmacy*
12/15/2020 01/14/2021 (expired)	Atenolol 25 mg tablet 25 milligrams Orally every day 30 Tablet(s) 0 Refill(s)	ONE ZDSSDOCTOR	SN: 1209 ePrescribed *Yalaha Pharmacy*
12/15/2020 01/14/2021 (expired)	Lisinopril 5 mg tablet 5 milligrams Orally every day 30 Tablet(s) 0 Refill(s)	ONE ZDSSDOCTOR	SN: 1208 Printed *Yalaha Pharmacy*

Page: 1 2 >>

Home Medications Last reviewed or updated by FRED TEST on 05/20/2021 13:37:06

☑ Patient reports no Home Meds ☐ Med List not complete ☐ Unable to obtain Medication History

Group By: ○ Indication ○ Class ○ Generic med name ● None

Display: ● Active Only ○ Active & Inactive

HINT: (Click on the name of the medication to EDIT or DELETE)

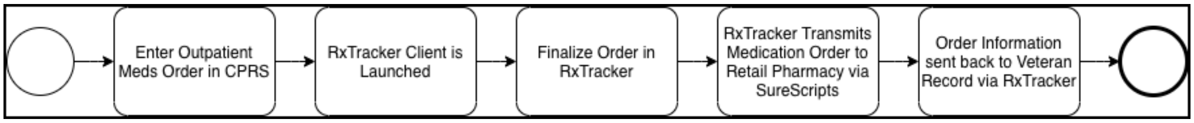
Medication	Instructions	Last Dose	Last Updated	Charted By	Source
PATIENT REPORTS NO HOME MEDS	05/20/2021 13:37:04 FRED TEST				

Cover Sheet Problems Meds Orders RxT Notes Consults D/C Summ Labs Reports
10:44 AM
2/18/2022

RxTracker integrates with VistA via standard APIs (OR*, VADPT*, XLF*, XU*, etc.) and HL7 messages (ADTs, OMP, RDE and MFN). The DSHX KIDS file produced for VISN4 would be installed at each site to utilize and subscribe to VistA API and HL7 components. The RxTracker framework ensures CPRS data sharing and integrity while offering a seamless CPRS workflow and a continuous and fluid ordering process. CPRS integration includes both single sign-on and initial patient context and preserves VA Medication Order Check Healthcare Application (MOCHA) medication order safety checks. RxTracker also supports updating and creation of new orders within the RxTracker order finalizing screens. Orders can be sent to both internal VA pharmacies as well as external community pharmacies. Any new or modified orders placed in RxTracker are sent back to VistA and are available within both the VistA pharmacy modules and CPRS, ensuring VistA remains the single system of record for all medication orders. RxTracker transmits prescriptions to external retail pharmacies and receives order updates from these pharmacies in a secure manner using VA-approved technology.

Besides VistA/CPRS integration, there are three external system interfaces used by RxTracker: (1) Surescripts – connects to retail pharmacy to exchange patient, provider, and order data, (2) First Data Bank – checks drug-drug and drug-allergies interactions (order knowledge), and maintains the Drug File, and (3) Exostar for EPCS – provides ID proofing, two-factor authentication, and digital signature of providers when prescribing controlled substances. Typical bi-directional workflow is as follows:

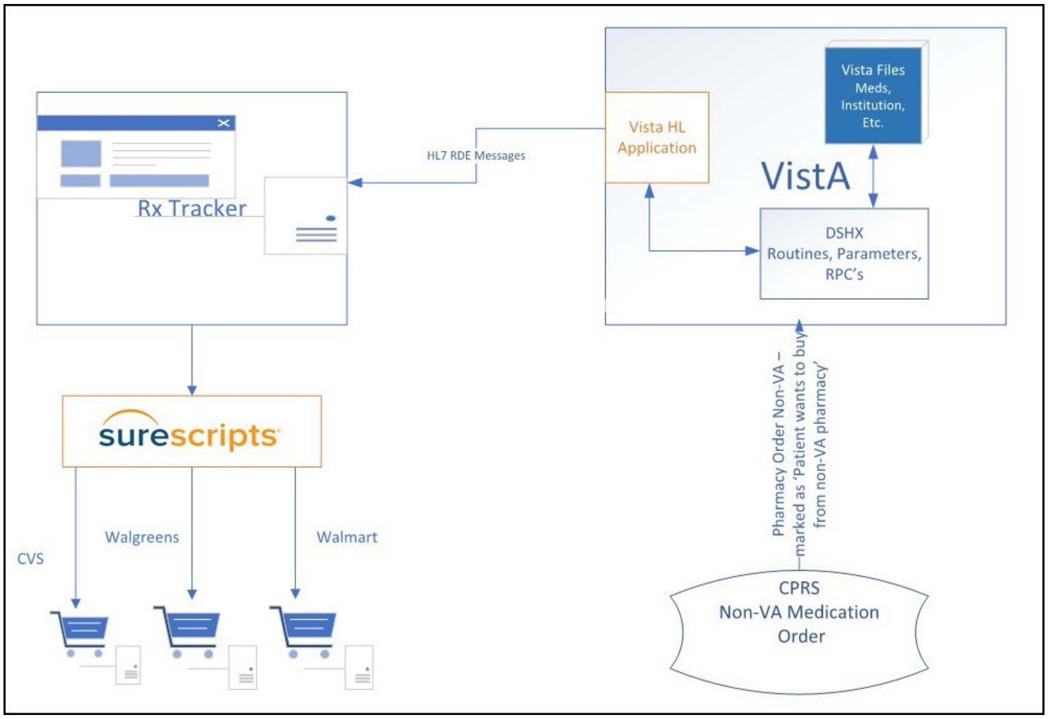
Typical Bi-directional Workflow


Enter Outpatient Meds Order in CPRS
RxTracker Client is Launched
Finalize Order in RxTracker
RxTracker Transmits Medication Order to Retail Pharmacy via SureScripts
Order Information sent back to Veteran Record via RxTracker

The provider initiates an Outpatient Medication Order from within CPRS and designates as an external pharmacy fill. When the order is signed by the provider after normal VA Medication Order Check Healthcare Application (MOCHA) medication order safety checks are invoked, an event is triggered that is detected by the VistA-side DSHX KID components to transmit order data via HL7 to RxTracker. RxTracker client is launched (automatically depending on integration) and the provider finalizes the prescription in RxTracker which then transmits the order to the retail pharmacy via Surescripts. In addition, RxTracker allows for authorizations info, renewal responses and change requests. ePrescription information (status/order changes) flows from the retail pharmacy back to VistA/CPRS allowing VistA to be the true System of Record.

The workflow diagram below summarizes the data exchanged between the VA Patient Record and Surescripts via RxTracker.

Data Exchanged Between the VA Patient Record and Surescripts via RxTracker


Rx Tracker
HL7 RDE Messages
Vista HL Application
Vista Files Meds, Institution, Etc.
VistA
DSHX Routines, Parameters, RPC's
surescripts
CVS
Walgreens
Walmart
Pharmacy Order Non-VA – marked as 'Patient wants to buy from non-VA pharmacy'
CPRS Non-VA Medication Order

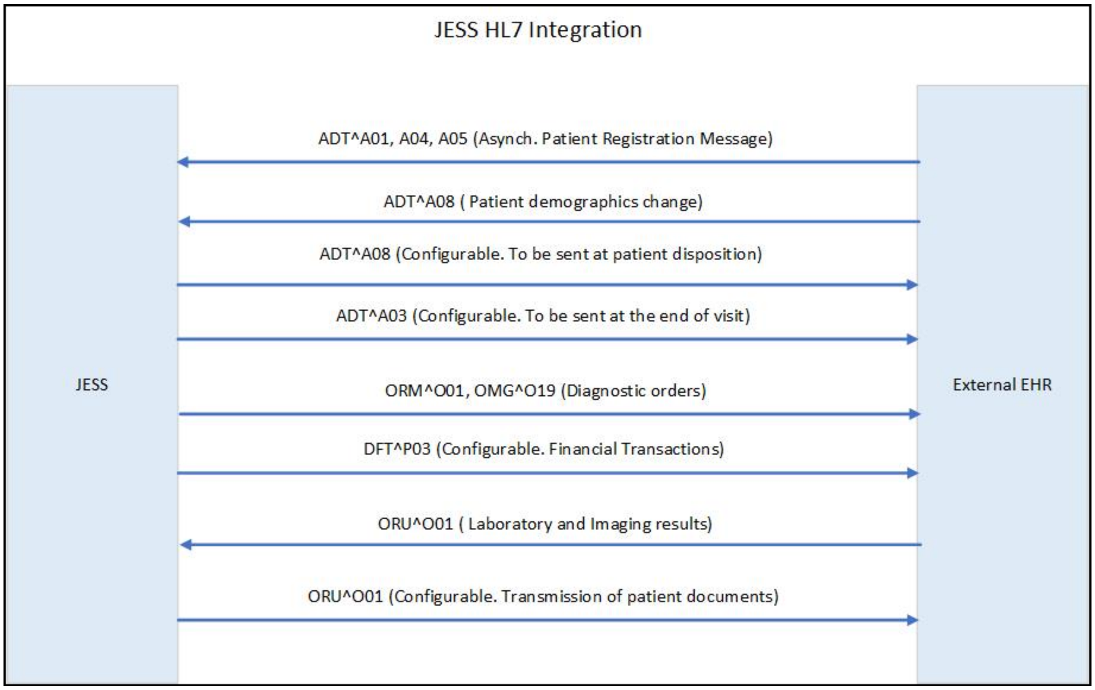
What is JESS? The Juno Emergency Services Solution (JESS) is a best-of-breed, VistA ready, Emergency Department (ED) solution. JESS is a cloud-based web application for Emergency Department Information Systems, that tracks patients from presentation to the emergency department through disposition to the next appropriate level of care. The cloud-based solution is extremely intuitive and provides a dynamic documentation interface that fits seamlessly into a fast-paced environment like the ED. Data can be accessed from any connected device at any time. JESS is content complete and includes patient tracking, triage documentation, nurse documentation, physician documentation, order entry, RxTracker (using Surescripts) integrated into the workflow, and a discharge instruction module. Rapid Charts assists providers with completing notes using complaint-specific content, streamlining data entry, and saving valuable time. Quality of care monitoring and clinical decision support is accomplished with the CareTRAK^{TM} module using evidence-based algorithms specific to the patient's problems. While JESS strategically prompts clinicians to capture thorough documentation, CareTRAK^{TM} identifies potential quality and risk concerns specific to each patient's needs by monitoring the medical record and generating real-time, meaningful alerts. JESS provides extensive reporting to measure activities in the emergency department to include regulatory requirements, quality monitoring and identification of high-risk patients, and process management to help identify patient flow metrics.

How will JESS be implemented: JESS is a web server application that runs on a client's browser. The web server components and enterprise Commercial MySQL database are installed in the cloud, while the application DSIHJUNO KIDS build is installed on each site's VistA instance to support HL7 integration. An interface engine, such as InterSystems HealthShare, will be required to facilitate messaging between the cloud hosted JESS solution and the site VistA instances. JESS will be implemented as a stand-alone application; a shortcut with the URL for the cloud hosted solution will be made available to VA Emergency Department staff.

Following the natural emergency department workflow, JESS receives patient registration information from the hospital EHR in the form of HL7 ADT A01, A04 or A05 messages. JESS also receives and processes ADT A08 messages for patient updates. With patient account information synchronized with the facilities' EHR, JESS provides the ability to place diagnostic and treatment orders (HL7 ORM O01, OMG O19), financial transactions (HL7 DFT P03) as well as receiving laboratory and imaging CPOE (computerized order entry) results in the form of HL7 ORU O01 message types. The CPOE order cycle: ordered, received, reviewed, is continuously updated, and presented in real time to the end users in the form of an application status board. This information includes configurable timeout periods to alert the users when orders are taking an inordinate amount of time. Optionally, JESS can also be configured to send patient updates to VistA using outgoing HL7 ADT A08 messages and disposition information at the end of the visit using ADT A03 message types. A Digital Archival System (DAS) is part and parcel of the JESS offering. Through configurable printing, all documents created during the JESS patient visit (Physician and nurse notes, orders, and discharge documents) are sent to VistA/TIU via an ORU^R01 containing a Base 64 encoded document to ensure patient clinical documentation is satisfied in a VA expected format.

As required by ONC, JESS produces CCDA documents like CCD, referral notes or discharge summary for the ER visit, which are also available for transmission via Direct protocol. The diagram below illustrates the mapping between JESS triggering events and VistA-supported HL7 message types.

Mapping Between JESS Triggering Events and VistA-supported HL7 Message Types


JESS HL7 Integration
JESS
External EHR
ADT^A01, A04, A05 (Asynch. Patient Registration Message)
ADT^A08 (Patient demographics change)
ADT^A08 (Configurable. To be sent at patient disposition)
ADT^A03 (Configurable. To be sent at the end of visit)
ORM^O01, OMG^O19 (Diagnostic orders)
DFT^P03 (Configurable. Financial Transactions)
ORU^O01 (Laboratory and Imaging results)
ORU^O01 (Configurable. Transmission of patient documents)

DSS has been integrating HL7 solutions with VistA since 2004; our integration technologies are standards-based with a focus on reusability. The HL7 subset of our Integration Framework core package utilizes standard HL7 messaging and communication protocols that are approved by VA's HL7 Message Administrator. We are well positioned to implement JESS, a best-of-breed Emergency Department product, at every VA Medical Center to improve quality of care.

What is the VistA Lab Rewrite of File #63? One of FileMan's best attributes is its role as database modeler. Unfortunately, not every VistA file is documented using FileMan. In the case of the VistA Laboratory information system package, this occurred because the lab code pre-dates FileMan itself. While the VA Laboratory development team has, over the years, been converting hard sets in the code to FM calls, gaps remain. For example, the ^LA and ^LAH globals used in the Laboratory package do not have Data Dictionaries and are not accessible via FileMan. DSS worked to eliminate FileMan incompatibility issues beginning with the LAB DATA (#63) file. Our VistA Lab remediation focuses on improvements which includes Ask on Entry (AOE) to capture additional test information from within CPRS. This data is passed to the lab package and then included in outbound HL7 order message OBX segments. We have also included Multiple Reference Range functionality that provides more complex age and sex reference ranges than are supported in the existing field in the LABORATORY TEST (#60) file. We implemented an Auto-accept process of inbound results for the LEDI HL7 interface. We have also created a better framework for the LAB DATA (#63) file to make it better able to store results and comments as well as all other data being brought in through the LEDI HL7 interface. These changes make it much easier to create NIST-compliant HL7 Lab result messages.

How will the newly designed VistA VA FileMan File #63 be Introduced into VistA: DSS initially presented the proposed solution during a VistA Expo community conference in Seattle in 2011, and DSS delivered copies of the fully defined files with field descriptions, but adoption within the national legacy Laboratory package was never completed. DSS proposes completing this work which can be implemented using a VistA KIDS build.

Most globals can be identified by cross-referencing the direct global read/write instances against ^DIC(file#,0,"GL") which holds the global storage for a file as documented by FileMan. To support standardized access to VistA data via FileMan, information regarding non-compliant globals and the data they contain, such as in ^LA and ^LAH, must be included in the Dictionary of Files, Data Dictionary, Data Type Descriptor, and other resources maintained by FileMan. The DSS development team took on this challenge and remediated many of the problematic areas for Laboratory. We created a model for data definitions of laboratory files with the following suggested file numbers to bring them up to full FileMan compliance:

- Lab Data Instance (#63.5)
- Lab Data Definition (#63.6)
- Lab Data Definition Types (#63.61)

We also recommend refactoring other outliers in a similar manner. A complete review of "unregistered" files within the Laboratory, Pharmacy, Immunizations, Lab AP Order Dialogue, Auto Verification and Microbiology applications should be accomplished to include documenting the structure and content of the files and creating a FileMan Data Dictionary containing all appropriate structural, and descriptive information that can be used read/write data to fully remediate these VistA packages.

What is Juno RX? Juno RX is both an upgrade and a replacement for VistA Pharmacy. Juno RX is written in modern web-based technologies using DevSecOps procedures with an intuitive multi-screen workflow designed by pharmacists. The current Pharmacy component of VistA utilizes a roll-and-scroll interface and lacks key components required by standard practices found in other Rx information systems. Every day throughout VA, thousands of pharmacists and pharmacy technicians interact with the legacy application. Unfortunately, roll-and-scroll has multiple limitations: it slows adoption, contributes to inefficiencies, and limits the display of critical data necessary to treat the patient. Juno Rx increases adoption and productivity by presenting users with current and new orders in a customizable worklist, and then allows users to process those orders on a dual monitor using a web/browser-based interface that presents all necessary information without having to "jump" around various options or applications to view data. Retention and recruitment of staff is a key success factor of all healthcare organizations, particularly in the current employment environment due to COVID. While Pharmacy has always been a leader in technology adoption, both new pharmacy talent and experienced pharmacy staff expect information systems to do more than capture data through data entry screens. Juno RX was created using Human Centered Design principles and meets all

VA Digital Standards beginning with understanding people and their needs through designing the whole experience and measuring performance. Juno Rx enhances clinical decisions, and enables improvements in patient care, an important consideration since Pharmacy impacts thousands of clinicians and hundreds of thousands of the 9 million Veterans enrolled in VA healthcare every day.

VA Pharmacy is unlike its private sector counterparts. VA Pharmacy integrates all parts of inpatient, outpatient, nursing home, and mail-order workflows plus has access to allergy, adverse reaction tracking and laboratory results that work seamlessly to increase patient safety through clinical monitoring and customizable alerts. The Juno RX system has been developed with patient safety, pharmacist workflows, and regulating bodies in mind. The system leverages in-depth order checks for patient safety and displays pertinent information clearly, facilitating a pharmacist's fiduciary responsibility to their patients and ensuring the appropriateness of medication therapy. Pharmacists can generate required labels and send alerts to nurses using the native closed-loop bar code medication administration (BCMA) system. Home/self-medication programs are supported within the system with indicators that pass to nurses within BCMA. BCMA supports sliding scale administration in addition to PRN criteria for pain score. The Juno Rx system leverages a modern web user interface and cloud-based infrastructure that meets the needs of today with room for expansion into the future while providing a seamless flow of medication data from electronic order entry through dispense and administration.

How will Juno RX be Implemented: Juno RX is a web server application that runs on a client's browser. The web server components, enterprise MSSQL Server database, and DSS VistA SOA Suite (VSOA) are installed in the cloud, while the application DSSR KIDS build is installed on each site's VistA instance. VSOA is the lightweight middle tier used to communicate with VistA using JSON APIs. Juno RX integrates with VistA via standard APIs (OR*, DI*, DSIC*, PS*, VAPDT*, etc.) and other necessary Pharmacy package (PS*) entry points. Both the Web App Server and VSOA Server are compatible with Linux, macOS (OS/X) and Windows operating systems. Juno RX requires certain VistA Pharmacy configuration settings to function properly. These settings, such as default med routes, IV room configuration, user security keys, etc. are described in the DSSR KIDS build installation guide. Juno RX will be implemented as a stand-alone application; a shortcut with the URL for the cloud hosted solution will be made available to VA Pharmacy staff.

DSS Implementation Support Team, which includes technical subject matter experts, will work with VA OI&T counterparts to facilitate successful implementation and maintenance of Juno RX. Installation of the cloud components will precede local site implementation of the Juno RX DSSR KIDS build and VistA configuration modifications. A DSS Installation Specialist will install, configure, and test the software to a state where the product is ready for subsequent training and go-live activities. DSS will also provide on-site training to educate future users on features, functionality, and optimal usage of Juno RX and underlying system settings and features. Our professional training specialists engage class participants in instructional content and utilize active learning techniques, promoting retention of the material. Training is reinforced via feedback from participants, hands-on practice exercises, over the shoulder coaching/mentoring, and proficiency evaluations. Proficiency evaluations include testing and evaluating the user's understanding of the software capabilities and functions before the training team departs.

DSS thoroughly analyzed the VA's unique needs for Pharmacy. While Juno RX provides immediate improvement for pharmacy staff using legacy roll-and-scroll options, the application will require some VA-centric enhancements such as Consolidated Mail Outpatient Pharmacy (CMOP) and Clozapine order checks, for example, to be completed within a phased implementation. DSS has experience on both the CMOP and Clozapine contracts at VA; these enhancements, and other VA-specific features, have already been designed within Juno RX and will be developed upon DSS receiving sponsorship from the VA.

Appendix III

CPRS Modernization Solutions – the 'What' and the 'How'

{Transforming CPRS to accept Web Based Applications and into a Platform}

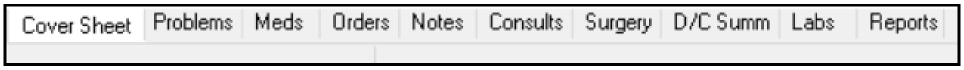
CPRS modernization establishes CPRS as an extensible shell so that new functionality can be added, which also brings substantial new interoperability improvements and functional capabilities. CPRS Modernization introduces a series of DSS solutions, or any other COTS best of breed web solutions, into a new CPRS extensible shell from a client-server model that brings additional platform enhancements to improve Veteran care. These sample DSS solutions are as follows:

- Enhance CPRS to support Dynamic Tab Management
- Establish Bi-Directional Interfaces between VistA and Juno Components
- Introduce Juno Content/Workflow Builder in CPRS
- Introduce RxTracker Med Reconciliation in CPRS
- Introduce Juno Clinical Documentation in CPRS
- Introduce Juno ONC Interoperability into VA Ecosystem
- Introduce Juno Surgery in CPRS
- Drug and Allergy Knowledge Base (DAKB)
- Introduce Juno ProDash in CPRS
- Introduce Juno Order Management in CPRS
- Introduce Juno eMAR and BCMA in CPRS
- Introduce Juno Common Application Framework (CAF)

This CPRS modernization effort will provide VA with the components to modernize CPRS in a gradual manner using proven approaches for VA modernization and sustainment of CPRS and VistA.

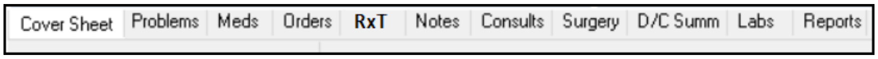
What is Dynamic CPRS Tab Management? Dynamic CPRS Tabs Management makes it possible to run modern web applications within CPRS rather than simply launching them via the CPRS Tools menu. To safely and effectively integrate CPRS with externally built web applications chosen by the VA, such as the various Juno EHR web applications discussed here, a tightly woven integration will be essential. Tight integration ensures patient context is maintained and provides modern OAuth 2 authentication. The enhancement allows CPRS tabs to be dynamically built and displayed for the selected application and user base. If a solution is not purchased by all facilities, or no longer desired, the tabs can be quickly hidden without requiring CPRS source code changes. This diagram shows a typical CPRS tab configuration:

Typical CPRS Tab Configuration



The CPRS integration for DSS RxTracker is an example of adding a custom module using Dynamic CPRS Tabs Management. While RxTracker can be run from the CPRS Tools menu as a standalone application, our commercial implementation includes tight CPRS integration using a tab. This diagram shows the CPRS tabs with the added RxTracker application:

RxTracker Application is added to the CPRS Tab Configuration



Any number of new tabs may be created by the site(s); hence the ‘dynamic’ tab creation capability. If a new module replaces legacy CPRS functionality, the historical tab may be hidden and not used. For example, implementing a new Order Entry application may make the existing CPRS Orders tab obsolete. VA IT can now hide the Orders tab without requiring CPRS source code changes, testing, and release. CPRS users who are defined as ‘Reports tab only’ will remain as such; they are not impacted by this enhancement.

How will the Dynamic CPRS Tab Management Components be Introduced into CPRS: A VistA KIDS build, VFDO, contains two new parameters needed to implement CPRS tab management capabilities. VFD CPRS TABS SHOW allows VA IT personnel to show/hide any existing or new tab, and VFD URL stores the root location for the new web-based application module. A one-time modification to CPRS source code is needed to check these parameters to show/hide the tabs and to launch an encapsulated application when the user clicks a tab with an associated URL. DSS can perform the CPRS source code changes or deliver a document with the changes to VA for the CPRS development team or other VA service contractor to complete, test, and release. The KIDS build can be implemented prior to the CPRS GUI changes without impacting the sites.

When CPRS initializes, all visible tabs are created with their associated forms. For dynamic tabs, the form will contain required interfacing components such as SOAP, XML, and/or Chromium components plus any necessary Timers for timeout configuration. Synchrony objects are created to assure real-time information flow between VistA and the web application. Full or partial patient refresh actions will occur when users navigate to or from the Dynamic Tab(s). If no patient is ever selected and CPRS is exited, the form and its components are simply destroyed without ever logging into or accessing the web module in any way.

Web application initialization begins with the construction of a contextual OAuth2 authentication token with is shared in subsequent communications between the web application and VistA/CPRS. With the communications token in place, CPRS constructs any direct synch API calls by gathering relevant information, constructing an information object, and transmitting the object to the receiving endpoint. Any failures are indicated within CPRS.

When the provider finishes with the current patient, whether they exit the application or choose to switch to a different patient, the Dynamic Tab will call the web application with a ‘LogOut’ request, clear any currently assigned token for web module, and reload an empty form. This assures there is no accidental carryover information between patients which would pose a patient safety issue. Additionally, on exiting, CPRS will close and destroy all web module connections and used components to appropriately clean up threads prior to performing CPRS's own closing procedures.

> **Identified Modifications to Core CPRS Delphi Code:** VA must accept that core CPRS Delphi code must be modified with this enhanced CPRS framework.

What are the Flexible Bidirectional Interfaces between VistA and Juno EHR? Juno Integration has developed robust HL7 2.5.1 messages that include but are not limited to ADT's, Orders, Observations, Results, Detailed Financial Transactions, and Scheduling messages. HL7 integration from Juno to VistA processes patient admissions, transfers, discharges, and updates utilizing the DSS ADT Filer, a customized HL7 integration library used to better communicate between VistA and a foreign system. As an extension of the DSS Integration Framework, which is nationally installed at VA, DSS will introduce additional HL7 integration through the Juno Integration HL7 Processor that sends bi-directional order messages between Juno EHR and VistA, for example, Laboratory and Pharmacy for JunoRX and other selected applications.

The HL7 outgoing (from Juno to VistA) message processor has the following characteristics:

- Outgoing HL7 Message Processor will respond to certain events in the Juno system depending on what is implemented for a customer. Events include such things as patient registration, patient admission, patient discharge, patient demographic updates, lab orders, imaging orders, medication orders, lab results, imaging results, medication administrations. The event will generate the appropriate HL7 2.x message
- The message will be placed on the outgoing message queue that is connected to an integration engine (e.g., HealthShare Health Connect). The Juno Integration HL7 processor will also temporarily store the message in an internal database table.
- The selected HL7 integration engine will be responsible for both transmitting the message to all downstream systems, including but not limited to internal Juno subsystems such as JESS and RxTracker that have been configured to receive the specific message, as well as to transform said message into the format required by the EHR. Integration Engine configuration and transformations are outside of Juno EHR design.
- The internal database table that holds the messages in Juno will be used as a temporary auditing of sent messages and as the source of messages that require re-transmission. Due to the expected volume of messages generated by Juno EHR each day it is expected that this database table will be purged regularly.

The Juno incoming (from external source to Juno) message processor contains the following:

- Incoming messages in HL7 2.x, FHIR, API, X12, CDA, or flat file formats are sent to VA's HealthShare integration engine.
- HealthShare transforms the messages into a HL7 2.5.1 format and places them in a RabbitMQ queue
- RabbitMQ notifies the Juno HL7 Processor that messages are in the queue via its "push API".

- The Juno HL7 Processor reads the messages off the queue and using a pre-configured HL7 Library, de-serializes the message, creates HL7 Data Objects and Domain Specific Data Objects.
- Using Juno API's, data from the Domain Specific Data Objects are written to the Juno DB.
- Juno APIs are also used to place messages into the HL7Message table in the database. This table maintains a list of the messages received that can be used by the Juno Interface Utilities for reporting or to reprocess messages if necessary.

How will the Juno EHR Bidirectional Interfaces be Introduced into VA: HL7 integration requires multiple components to support integration between VistA and Juno EHR components. Juno EHR HL7 components to be installed in the cloud include Juno HL7 Processors, HL7 Library components, Data Objects, Message Logs, a MSSQL Server database, and RabbitMQ to provide message queue capabilities. An interface engine, such as InterSystems HealthShare, will be required to facilitate messaging as necessary between the cloud hosted Juno HL7 pieces, the site VistA instances, and other systems or applications producing or consuming messages for this effort. A VistA KIDS build, SISIADT, contains the ADT Filer and will be installed on each site's VistA instance if ADT integration is desirable. The ADT Filer does not merely assist with VistA consumption of ADT messaging; it contains a preprocessor that can route inbound messages to another system and wait for responses before proceeding with internal message flow and error reporting. The preprocessor can be useful, for example, to call VA's Master Veteran Index to generate national Veteran identifiers for new patients as part of local registration. API integration to VistA requires VSOA to be installed in the cloud. VSOA components are compatible with Linux, macOS (OS/X) and Windows operating systems.

HL7 integration between healthcare entities is not new. Where Juno EHR differs is the proprietary namespaced M routines that allows for VistA to properly consume HL7 messages produced by Juno EHR. The Juno HL7 Interface Utility provides a visual way to configure and monitor Juno inbound and outbound HL7 messages using a dashboard. The Interface Utility displays Juno outbound HL7 messages with an easy parser for technicians in the IT role. The dashboard allows users to check message status, start/stop an interface, check configuration details, view queued messages, and view message errors. As an example of the uniqueness of a customization integration approach for VistA, DSS processing routines call specific medication pharmacy module APIs to ensure home medication triggered by the current RxTracker system implemented within VISN4 are properly stored within the VistA pharmacy package.

The DSS extended integration tool set is a TRM-compliant solution that interfaces Juno EHR modules with VA systems including VistA. **No massive bulk data migrations from VistA to Juno EHR are required**. Data remains in the system of record (VistA) and is used on an as-needed basis.

What is the Juno EHR Clinical Content/Workflow Builder? Juno's Clinical Content Builder is a configuration tool used to create custom Flowsheets, Note Templates, and Modules using user configured Items, Topics, Templates, Subjects, and Groupings. Flowsheets allow users to document in a patient's chart in a grid-like format consisting of columns with a date/time where each row is an Item (e.g., a vital sign observation). The rows are part of a Topic, that Topic is part of a Subject, and that Subject is part of a Grouping. A Note Template is essentially a questionnaire with a list of questions and patient information created using Items, Topics, and Objects with formatting, condition, and narrative configuration options. Once a template is created in Template Builder, and published, it is available for use in the Notes module, or any other module integrated with Notes. To clarify how these elements work together:

- ***Items*** are questions and answers used to capture codified responses (observations and/or questionnaire responses). When creating an Item, administrators have a list of answer “value types” to choose from to meet a variety of documentation needs: Numeric, Text, List of Value Single-select, List of Value multi-select, Calculation, Bundle, Date, and Date/Time. Objects exist in the template builder as a ‘widget’ created by a developer.
- ***Topics*** are a collection of items. Items that share the same Topic would typically be documented together (i.e., Vital Signs would be a Topic that includes Items of Temperature, Blood Pressure, Pulse, etc.). Topics can also be assigned codes via the Codes tab.
- ***Subjects*** are a collection of topics and/or templates.
- ***Groupings*** are a collection of subjects and/or templates. Once a grouping is created, roles can be assigned under the permissions tab. Only the roles selected will see these Groupings. The Groupings can then be assigned to specific modules, so only the Groupings selected will display in the appropriate module.
- ***Objects*** include patient information objects and template objects. Template objects are things such as section headers and read only text that can help to format a template. Patient information objects will display patient information in the ‘note’ and allow for edit and add, when applicable (example: there is a need to display the patient’s preferred language in a note and allow the user to edit that field. An object would be created so the information can feed back and forth). Objects help achieve what items cannot; they can also be used for more complex data entry or functionality that an item capturing a simple question/answer cannot accomplish.

The purpose of the Clinical Content Builder is to provide a user interface to build and *visualize* the relationships between Groups, Subjects, Topics, Items, and Objects. Visualization of how these components are related will assist in understanding the areas an Item or an editable Object that can be documented. This allows the administrative users to understand if a new Flowsheet (Grouping), Subject or Topic really needs to be created or if an existing entry can be amended to meet the needs of many. Template builder provides user friendly drag and drop functionality to create custom note templates using prebuilt configuration Items and predefined Objects. Template builder offers over 15 preset layouts, advanced font formatting, item response narratives, and conditional logic. Voice recognition capabilities are built into the Template element.

User-configured Flowsheets and Note Templates can be placed into various Juno EHR clinical and administrative modules, or a custom module, so that workflows can be created based on the user role. Overall, the Clinical Content Builder helps achieve a common business goal of creating a highly configurable EHR that is user friendly not only to the clinical end user but also to the facility administrator.

> **How will the Juno EHR Content/Workflow Builder be Introduced into CPRS:** Clinical Content/Workflow Builder is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud. No VistA integration is necessary. The Clinical Content/Workflow Builder can be implemented as a standalone web application, or within a CPRS tab using the **Dynamic CPRS Tab Management** enhancement discussed in this document. The new tab can be made available for selected administrative users. The type of content being built, for example, a note template or flowsheet, will depend on what other Juno modules are selected by VA. If VA purchases Juno Flowsheets, then administrators will configure the flowsheet items, topics, groups, etc. using the Clinical Content Builder. Any site(s) utilizing the complete Juno EHR platform will need the Clinical Content Builder to create all necessary content items.

What are the RxTracker/Med Reconciliation Components? Every organization should mandate that medication reconciliation be performed for each outpatient encounter, each inpatient admission and at each discharge. Obtaining an accurate medication history is an essential part of the reconciliation process. The RxTracker Med History page includes multiple sections, each playing a part to support efficient and safer prescribing practices. The Medication Reconciliation process begins by documenting home medications for a patient. Providers/clinicians are required to document medications that patients state they are currently safely taking, including medication, dose, route, and frequency. If something is unknown by the patient, for example, dosage, the medication can still be recorded. The documented medication will display with an indicator to let others know the information is incomplete. Home medications propagate throughout the outpatient and inpatient medication reconciliation workflow. Home Meds is a longitudinal list and should be updated with the most current information obtained from patients. Retail Pharmacy history is obtained from Surescripts. It is a combination of data reported to Surescripts by Pharmacy Benefit Management systems and community/retail pharmacies for medication fill history for the patient. Retail pharmacy history is a tool that clinicians use to analyze previously dispensed medications allowing providers the visibility into adherence patterns and medication compliance which supports safer medication practices. To obtain Retail Pharmacy History for patients, it is necessary to click on the Patient Consent checkbox. This is a required step. The organization, not the RxTracker vendor, is responsible for updating general consent forms to include language for obtaining retail pharmacy history. If a patient states they are still taking a medication on the Retail pharmacy history list, providers can use the COPY function to copy the medication to the Home Med section. This supports efficient documentation processes.

Copying a medication is a two-step process allowing providers to edit the med, if applicable. Up to 300 medications will be displayed for a 12-month period with the most current medications displayed first. Retail pharmacy history should be obtained once per OP encounter or once per IP admission per patient. Accurate patient demographic data is important. Surescripts uses patient demographic data to match patients to their Master Patient Index. Medication Reconciliation contains a history to document all entry/edit actions by users.

This addition to CPRS will effectively replace the current CPRS Meds tab with new and enhanced Juno EHR web- based functionality while maintaining all the current functionality of the current CPRS Meds tab. Both VistA files and Juno EHR tables will be updated as needed by using the previously defined flexible bidirectional interfaces between VistA and Juno EHR.

> **How will the RxTracker/Med Reconciliation Components be Introduced into CPRS:** Med Reconciliation is a component of the **RxTracker** application discussed in this document. Med Reconciliation can be implemented as a CPRS tab using the **Dynamic CPRS Tab Management** enhancement discussed in this document. This provides a seamless integration with VistA/CPRS making medication reconciliation easy to perform. VA can deprecate the existing CPRS Meds tab with a simple configuration when the site is ready to fully cutover to RxTracker/Med Reconciliation.

What is Juno EHR Clinical Documentation? The Juno EHR Clinical Documentation consists of a suite of modules (Clinical Action Center, Flowsheets and Notes) that enable clinicians to perform all aspects of documentation. The Clinical Action Center (CAC) is the main landing page for many clinicals and offers patient search, assignment, and a snapshot of the patient's record. The Flowsheet module is a core module that provides clinical users the ability to create accurate and timely documentation detailing medical treatment, assessments, vital signs, Input and Output fluids recording, etc. Flowsheets allow the clinician to view observations over time in a grid format. The Notes module addresses the need for one-time documentation of patient observations and other clinical data. Notes are highly configurable and allow for the collection of nursing documentation but also physician notes such as History and Physicals (H&P), Consultations, and Discharge Summaries. The Juno EHR Clinical Documentation suite of modules addresses typical challenges faced by users – for example, nurses being unable to locate/view/edit patient records in one place, spending a lot of time recording large amounts of information, or having to document the same information in different areas of the chart. Clinical Documentation provides the ability to:

- View and document patient data on a single, unified, interactive workspace.
- Monitor and analyze historical and real-time trends.
- Customize the display of data based on clinical workflow.
- Enhance communication between clinicians with patient snapshot and bedside bulletin or handoff communication.
- Configurable using Juno's Clinical Content Builder tools.

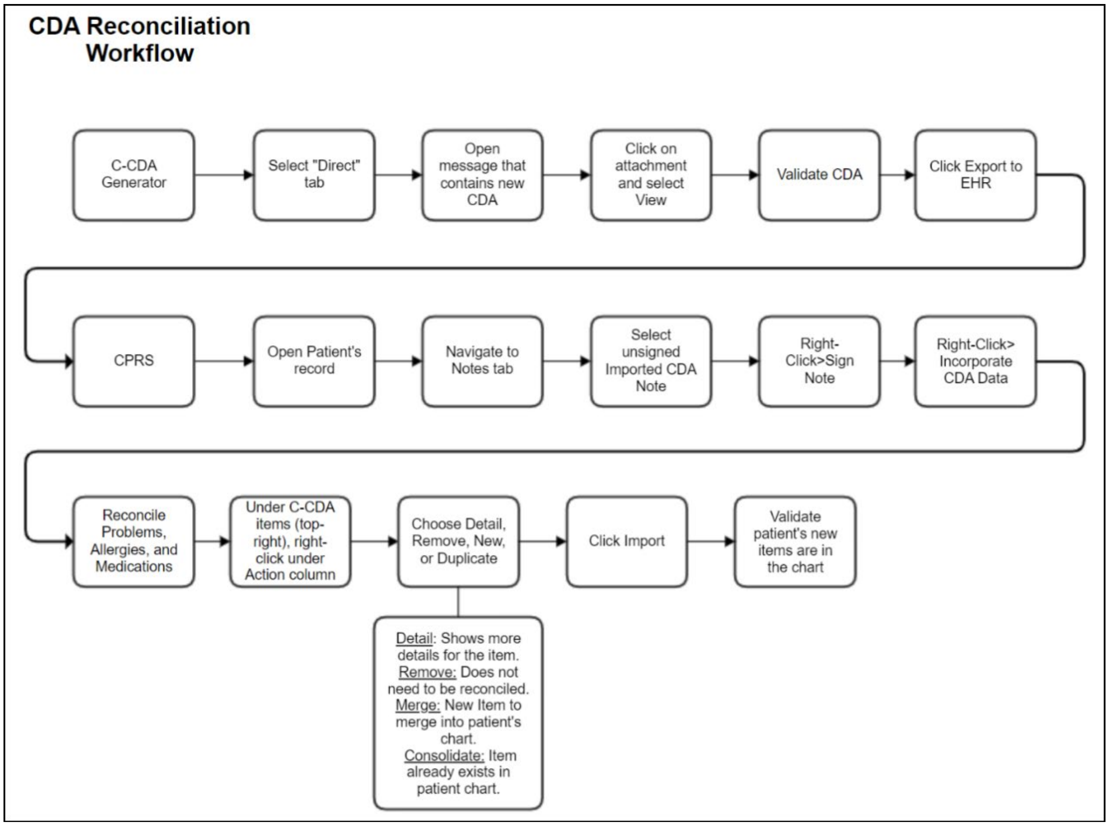
This addition to CPRS will replace many documentation functions such as Vitals input functionality and eventually Text Integrated Utility notes function (TIU) with new and enhanced Juno EHR web-based functionality while maintaining all the current functionality of the current Labs tab, Notes and Vitals input functionality. Both VistA files and Juno EHR tables will be updated as needed by using the previously defined flexible bidirectional interfaces between VistA and Juno EHR.

> **How will Juno EHR Clinical Documentation be Introduced into CPRS:** Juno Clinical Documentation is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud. Clinical Documentation is dependent on the **Clinical Content/Workflow Builder**. Integration to VistA would require the **Juno EHR Bidirectional Interfaces** enhancement. Juno Clinical Documentation can be implemented as a CPRS tab using the **Dynamic CPRS Tab Management** enhancement discussed in this document. This provides a seamless integration with VistA/CPRS making data view/entry easy to perform. Juno Clinical Documentation requires the Clinical Content Builder module to define what data and how data is presented to the user as well as how the user documents information for the patient.

What is the Juno EHR ONC Interoperability and Information Blocking Support Components? Under the 21st Century Cures Act, the Office of the National Coordinator for Health Information Technology (ONC) released a rule requiring EHRs to be interoperable and permitting monetary fines for information blocking. EHRs can share information using a variety of standardized content, methods, and measures such as Fast Healthcare Interoperability Resources (FHIR), Continuity of Care Document (CCD), Consolidated Clinical Document Architecture (C-CDA), Clinical Quality Measures (CQM), and Direct Messaging, to name a few. To be fully interoperable, EHRs need to be able to generate standard clinical content and receive/ incorporate content into their system. The **Juno Reconciliation Tool** allows a provider to reconcile a patient's active medication list, medication allergy list, and problem list from a received document like a CCD, D/C summary, or Referral and incorporate the information by adding or reconciling the received data with what is in Juno EHR. The Reconciliation Tool does this by simultaneously displaying (i.e., in a single view) the data from the record sent and comparing it with the information displayed in Juno EHR in a manner that allows a user to view the data and their attributes. This enables a user to create a single reconciled list of each of the following: Medications, medication allergies, and problems. The user can receive and validate the accuracy of a final set of data, update the list, and incorporate the data chosen into Juno EHR. **Juno Reports Generator** provides users the ability to pull relevant Meaningful Use data from VistA. Within Juno Reports Generator KIDS build, a VFD REPORT GENERATION LOOKUP file specifies relevant value sets that are 'watched' for certification and/or reporting purposes and indicates the endpoints to grab the relevant information. The reports can be used for Inpatient, Outpatient, and meaningful use automated measure calculations. The reports can then be exported to an XML file.

How will the Juno EHR ONC Interoperability and Information Blocking Support components be Introduced into VistA: The Juno Reconciliation Tool relies on Direct messaging to receive documents to be reconciled. The tool imports external documents as unsigned TIU notes. New features within CPRS signature functionality launch the reconciliation tool to perform actions on the data.

CDA Reconciliation Workflow


CDA Reconciliation Workflow
C-CDA Generator
Select "Direct" tab
Open message that contains new CDA
Click on attachment and select View
Validate CDA
Click Export to EHR
CPRS
Open Patient's record
Navigate to Notes tab
Select unsigned Imported CDA Note
Right-Click>Sign Note
Right-Click> Incorporate CDA Data
Reconcile Problems, Allergies, and Medications
Under C-CDA items (top-right), right-click under Action column
Choose Detail, Remove, New, or Duplicate
Click Import
Validate patient's new items are in the chart
Detail: Shows more details for the item.
Remove: Does not need to be reconciled.
Merge: New Item to merge into patient's chart.
Consolidate: Item already exists in patient chart.

Juno Reports Generator contains a VistA KIDS build in the VFD namespace. This KIDS build contains all the files, routines, and options necessary to define and capture relevant data for exporting out of VistA for meaningful use and quality measures.

What is Juno EHR Surgery Management? Juno EHR Surgery Management (Juno Surgery) module is an intuitive solution for perioperative care. Juno Surgery is designed to reduce documentation time, improve accuracy, patient outcomes, and help facilities improve operational efficiency and revenue capture. All perioperative documentation within Juno Surgery is built on evidence and standards-based practices, and regulatory standards backed by the AORN (Association of Perioperative Registered Nurses) Syntegrity documentation content and data set. Juno Surgery provides the ability to build and customize standardized preference cards for highly used procedures that will help facilities not only standardize the supply utilization but also reduce costs. Patient data is actively updated using Peri-Op documentation features so the care team can track patients across different phases of perioperative care. Juno Surgery provides easy access to generate reports for quality measures, productivity, patient throughput, and cost. Juno Surgery is seamlessly integrated with other modules within the EHR like Registration, Scheduling, Order Entry, Juno Clinical Content Builder, Medication Administration, Laboratory, Radiology reports and imaging systems, etc., to ensure data is shared with the care team. The following features are included in Juno Surgery:

- Surgical scheduling including workflow to incorporate Pre-Anesthesia Testing (PAT) for patients who will undergo procedures requiring anesthesia. PAT testing is generally scheduled within 30 days prior to surgery; the testing and assessment will include a nurse call for medical health history, a review of medications, and a review by an anesthesiologist. It may also include lab tests, EKG, nurse assessment, and review of records from specialists.
- Surgery configuration to create and manage preference card templates and other surgery suite parameters, plus the ability to generate preference card pick lists. Preference cards can be procedure specific or surgeon and procedure specific and can be customized per facility, site, and department.
- Ability to track and document all aspects of point of care treatment during Pre-Op, Intra-Op, and Post-Op. Pre-Op documentation is different than PAT testing; it incl^{ud}es tasks and documentation required on the day of surgery and includes both physical and psychological preparation.
- Use of 3rd party evidence-based perioperative documentation such as AORN Syntegrity Surgical Procedure List and/or Perioperative Nursing Data Set (PNDS) plus the ability to generate forms from clinical templates to be used for manual method for documentation capture.
- The perioperative record reflects data from other areas of the EHR including lab results, radiology reports and PACs image links, medication administration, etc. for a seamless experience.
- Juno Surgery status boards track real time perioperative process flows. Status boards track cases from the point a patient checks in until the patient is released from recovery (either discharged home or admitted to the hospital). It can have different views based on the intended audience and includes information relevant to that group of people. Juno Surgery waiting room status board offers a HIPPA compliant view so the patient's significant other(s) can monitor progress through the surgical process.
- Data collection to support quality measure reporting, also implantable device data capture reported to national database. Juno Surgery provides easy implantable device lookups using APIs from AccessGUDID, the Global Unique Device Identification Database (GUDID) containing key device identification information submitted to the FDA about medical devices.
- Creation of a surgical item master. Item Masters are a single, standardized source of information about items used in the hospital; its content drives not only supply chain processes but a broad range of clinical and financial functions (such as preference cards). Accurate and updated product descriptions and pricing facilitate purchasing accuracy, contract compliance, and revenue capture. A credible and functional Item master also contributes to better patient care, from enabling accurate and timely orders so that clinicians have the right products when they need them, to document detailed product information that can be used to help facilitate recalls.

- Surgical reports using Juno EHR Enterprise Reporting functionality.

Our goal for Juno Surgery is to provide an integrated surgery module that addresses the unique needs of all types of medical specialties involved in the surgery process, to help to improve user workflow and accuracy of data, and as a result, improve patient care. This addition to CPRS will effectively replace the current CPRS Surgery tab with new and enhanced Juno EHR web-based functionality while maintaining all the current functionality of the current CPRS Surgery tab. Both VistA files and Juno EHR tables will be updated as needed by using the previously defined flexible bidirectional interfaces between VistA and Juno EHR.

How will Juno EHR Surgery Management be Introduced into VistA: Juno Surgery is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud. Juno Surgery is dependent on the **Clinical Content/Workflow Builder**. Integration to VistA would require the **Juno EHR Bidirectional Interfaces** enhancement. Juno Surgery can be implemented as a CPRS tab using the **Dynamic CPRS Tab Management** enhancement discussed in this document. This provides a seamless integration with VistA/CPRS. With dynamic, customizable CPRS Tabs functionality, only administrative users will see surgery configuration screens including Item Master and Preference Care setup, while other clinicians (as defined by the site) will have access to other Juno Surgery components such as surgery scheduling and documentation. VA can deprecate the existing CPRS Surgery tab with a simple configuration when the site is ready to fully cut over to Juno Surgery.

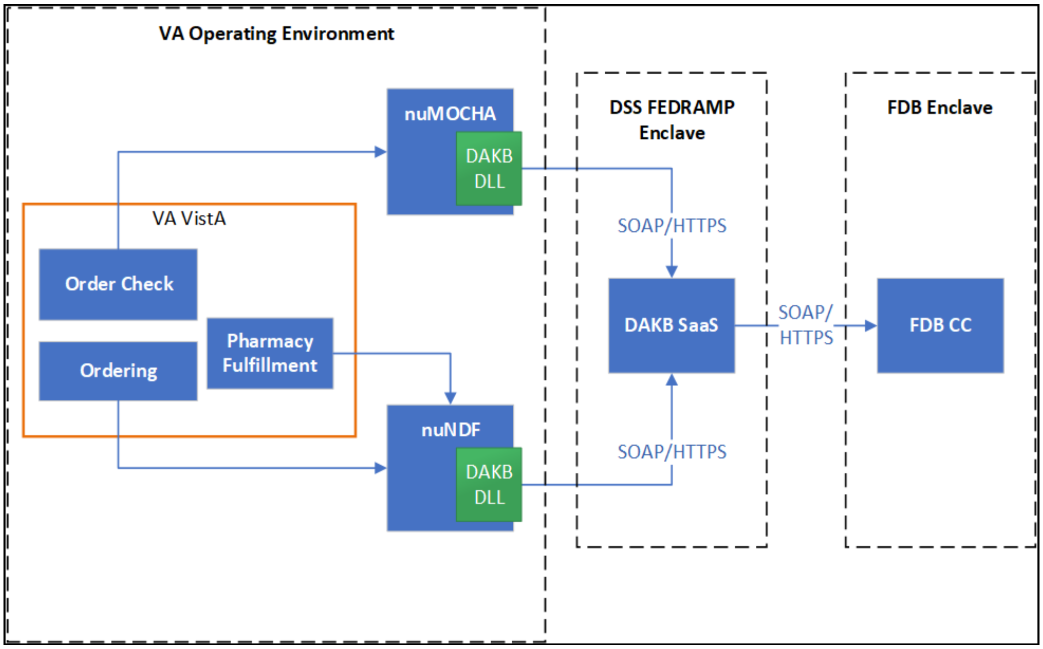
Surgery Dashboards can be made available in Pre-Op rooms and other locations as desired by the sites. Dashboards may require special monitors, mounting, and secure connectivity (typically wireless). AORN content needs to be uploaded and refreshed periodically. Juno Surgery contains an import tool through which IT or other surgery administrators can upload exported CSV files from AORN. The uploaded CSV files are processed, and data is passed to Juno EHR database stored procedures using WebAPI. The stored procedures then insert the AORN data into appropriate tables and return either a success or error response. Error responses include details to help the user fix the issue and try importing the data again.

What is the Juno EHR DAKB Web-Service? The Drug and Allergy Knowledge Base (DAKB) is a middleware service that provides access to the First Databank Cloud Connector (FDB-CC) collection of drug and allergy knowledge base web services. DAKB provides an alternative mechanism to integrate the rich collection of real time, drug, and allergy information sources in FDB CC with customers' Healthcare IT (HIT) applications. Appropriate HIT applications that leverage DAKB could range from a simple allergy check on prescription orders to a complete real time query system that replaces the VistA National Drug File (NDF) package. An added advantage of replacing NDF is no longer requiring local storage of the NDF data, or updates to that data store. The DAKB interface greatly reduces the effort and cost of integrating FDB-CC web services into all forms of HIT applications including web sites, native web applications, and mobile applications. It achieves this by handling all HTTPS, SOAP, REST, or other network integration coding required to access FDB-CC directly, replacing the native FDB-CC API with a simple library of function calls which are programming language independent. It also adds additional services to that integration point including auditing of all transactions, and load balancing concurrent use by multiple users. Providing the DAKB middleware results in a significant reduction in the cost of design, implementation, testing, and administration of software services necessary to access FDB-CC functionality. DAKB is also a true Software as a Service (SaaS) solution, with only the library interface required to be deployed in the customer's HIT environment. All software supporting communication with FDB-CC and recording all system transactions for performance and auditing purposes, etc. is hosted on DSS's cloud based DAKB implementation. This adds the additional cost saving advantages of SaaS to using DAKB, including reducing equipment, operational, and maintenance costs. DAKB is both an underlying service used by DSS HIT solutions, as well as a service available directly to other HIT developers and HIT consumers who wish to integrate their own software solutions directly with DAKB's services.

How will the DAKB Web-Service will be Introduced into the VA Technology Ecosystem: The Drug and Allergy Knowledge Base (DAKB) will be implemented as an abstracted service wrapper around the First Data Bank Common Core (FDB-CC) set of medication knowledge services. The deployment model for DAKB includes a library interface as either a DLL for windows-based systems or as a Shared Object (SO) for LINUX based systems which integrate with DSS-developed backend APIs written as Cache Objects for VistA. The DLL or SO transacts with a set of corresponding DAKB SOAP web services that have a one-to-one relationship with their respective FDB Web Services. The DLL or SO presents this set of Web Services as a single collection of functions to further simplify the integration of FDB services, DAKB services and the applications consuming those services. The DAKB solution is deployed in an elastic cloud environment which can be scaled as needed to meet load demand. This configuration allows for scalability and abstraction through a façade API that any VA technology could bind to for a given VA subscribed product/component.

The FDB-CC publishes its services in multiple protocols, including REST and SOAP, and allows for multiple data payloads in those communication messages via either XML or JSON. DAKB simplifies interfacing to FDB-CC by standardizing the calls using a combination of SOAP and JSON that then are translated into one or more FDB-CC protocol combinations. This provides a unified interface to the FDB-CC content that does not exist in the native FDB-CC calls.

A Notional Integration of VA, DAKB and FDB operating environments



The flexible DAKB architecture allows for significantly higher speed and bandwidth on the same network infrastructure in comparison to a straight RESTful or SOAP integration. The usual issues of deploying and administering a binary protocol add complexity due to security and message routing; this would not be an issue because all communication to VA products would be internal to VA networks and authentication processes. The DAKB system design not only offers FDB-CC pass through calls, but it also offers several ‘value added’ features. These feature sets are capabilities not supplied directly by FDB-CC, but which can be built on the foundation supplied by the FDB-CC exposed technology. Value added features include something as basic as caching query results to reduce duplicate FDB-CC service calls during the week span between FDB data updates. This reduces response time and possibly FDB licensing costs or, perhaps, NDF file updates. Value added features can also be as complex as developing macro-services.

Our analysis has proven, within our own commercial implementation of VistA, that a new service, using the DAKB and FDB services as a foundation can be implemented to replace the labor intensive current monthly NDF process which entails downloading the FDB drug dataset, reformatting and augmenting the information, and creating a KIDS patch file that must be installed in each VistA instance. Though not itself a NDF replacement, using DAKB as a foundation can greatly reduce the effort, cost, and complexity of implementing that new NDF replacement service. The DSS approach uses

DAKB to query FDB Med Knowledge to obtain the updated Drug Knowledge that is then programmatically mapped for storage into the VistA NDF file structures, introducing much needed automation and replacing much of the manual work done today. This master data mapping is maintained within the DAKB backend database for download into each connected VistA system removing the need for a KIDS patch creation, deployment, and installation process so that drug file updates happen seamlessly across the enterprise. Currently, DSS has a success rate of 70% automated coverage for all the VistA NDF fields. While this process is not fully automated, as some of the VA PBM (Pharmacy Benefits Management) processes may still require manual intervention on the DAKB server, overall the DSS solution requires less monthly effort, reduces delay in getting up to date data due to monthly drug file downloads and updates, and reduces cost, risk, and effort by having only one developed and hosted common service which all VA products would be able to use to access current drug data.

Macro-services can be considered the web service equivalent of a transaction process monitor. A potential VA use for a macro-service in DAKB would be supplying a “model” of a given patient’s current drug list and proposed changes to that list. The DAKB client would use the DAKB API to invoke operations which receives a patient’s current prescription list, a list of proposed new drugs to be added, and a list of patient’s allergies and other medical history details. The DAKB macro-service could also return a list of all interaction warnings, recommendations, etc. appropriate to the “model” supplied. These scenarios could be integrated directly into the current underlying VistA clinical event subsystem and leverage TaskMan and the alerting system as a means for review. There are currently 52 APIs available within the DAKB service that allows for different views of the FDB-CC drug/allergy knowledge services, they are as follows: Controlled Substance Service (2 APIs), Unit Service (3 APIs), Dosage Form Service (3 APIs), Order Knowledge Service (1 API), Core Drug Service (6 APIs), Drug Allergy Interaction Service (11 APIs), Drug-Drug Interaction Service (5 APIs), Screening Service (12 APIs), Medication Interoperative Service (9 APIs).

After reviewing the services available through the FDB-CC service and implementing several small software applications to evaluate the performance, functionality, and level of effort to access those service from code, we have concluded that using a common service built by DSS for Juno EHR could also benefit VA as a single source for drug and allergy knowledge. This new offering to VA, instead of accessing data that is 30-60 days past FDB’s latest updates by the time it is downloaded and reformatted, would give access to a real time system that is always no more than seven days past its latest data updates for VA NDF update purposes. Our hands-on analysis of FDB-CC also discovered that the native FDB-CC API is not an optimal interface design and requires far more software engineering/programming effort than it should, particularly to eliminate call chatting and complex web API sequencing. The EHR facade can resolve these inconsistencies by using automation to reduce the amount of code required to integrate with the FDB-CC.

DSS has current projects where DAKB is being integrated as the MOCHA enhancement for its commercial VistA (vxVistA) customers and as a key component for the NDF drug file replacement process. DSS is also exploring using DAKB to provide real time recurring drug order checks based on changes to the patient profile after the initial ordering and dispensing process. Real time monitoring within VistA involves attaching processors to VistA protocols for updates to Allergies, Vitals, Lab Results, Problems, and Diagnosis which would rerun the Drug interaction checks in the background. If the checks return new contraindications, then the ordering provider, pharmacy and/or care team would receive a notification that the drug profile has an issue; providers would be able to review the contraindication to determine if corrective action needs to take place. Essentially this is real time drug order checks post order process based on the latest patient profile offering improvements to drug safety.

What is Juno EHR ProDash? Juno EHR ProDash, accompanied with a native mobile app, is a cloud-based Dashboard to assist physicians and mid-level providers with managing patient's needs real-time. It quickly and efficiently identifies the location of patients by hospital unit and room number. ProDash further identifies patients who have relevant clinical findings to include new patients admitted, those potentially ready for discharge, clinical changes, vital signs, laboratory results, consultation information, other diagnostic findings, and the emergency department note to assist with prioritizing daily tasks. It also provides quick access to all relevant information to assist with timely decision making. ProDash measures and reports Key Performance Indicators specific to the provider, group, service, and hospital. ProDash can effectively replace the current CPRS Consults and Discharge Summary tabs with new and enhanced Juno EHR web-based functionality while maintaining all the current functionality of the current CPRS Consults and CPRS Discharge Summary tabs.

How will Juno EHR ProDash be Introduced into CPRS: Juno EHR ProDash is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud. ProDash is dependent on the **Clinical Content/Workflow Builder**. Integration to VistA would require the **Juno EHR Bidirectional Interfaces** enhancement. Juno EHR ProDash can be implemented as a CPRS tab using the **Dynamic CPRS Tab Management** enhancement discussed in this document. This provides a seamless integration with VistA/CPRS. With dynamic, customizable CPRS Tabs functionality, only administrative users will see ProDash configuration screens while other clinicians (as defined by the site) will have access to other Juno EHR ProDash components. The VA can deprecate the existing CPRS Consults and Discharge Summary tabs with a simple configuration when the site is ready to fully cut over to Juno EHR ProDash. The native mobile app, developed for clinicians, can be made available on the App Store of the VA's choosing such as the VA App Store.

What is Juno EHR Order Management Tool? Juno Order Management (JOMt) encompasses both the CPOE (Computerized Provider Order Entry) feature and CPOE Configuration functionalities within Juno EHR. JOMt consolidates all aspects of the ordering process: configuration of orders, order sets, order viewer and Ordering (order actions) in one web-based system. This advanced tool will give facilities the ability to fully customize the order configuration processes addressing their specific needs, policies, and business rules, as well as provide the customer with a ready to use off the shelf package. JOMt removes handwriting from the ordering process, utilizing only CPOE to avoid the potential for misinterpretation. Workflows designed for Clinicians by Clinicians will ensure less clicks and improved workflows allowing medical professionals to get back to treating patients and spend less time at a computer. Orders will funnel into the Juno EHR Services layer to be used by other applications such as the Clinical Action Center. JOMt is also integrated with the cloud-connected, full Order Check Suite provided by First Databank®, Juno EHR business rules engine, and clinical decision support (CareTRAK™). The following modules are contained within JOMt: Order Builder including all tables/files needed to create an order and identify the ordering authority within the facility, Order Set Builder enables the facility to create clinical, evidence-based order sets, and Customizable Juno Order Entry/CPOE which includes an order viewer and order entry screens allowing all necessary actions on orders. This addition to CPRS will effectively replace the current CPRS Orders tab with new and enhanced Juno EHR web- based functionality while maintaining all the current functionality of the current CPRS Orders tab. Both VistA files and Juno EHR tables will be updated as needed by using the previously defined flexible bidirectional interfaces between VistA and Juno EHR. Additionally, the CPRS Notes tab will be deprecated with the introduction of this functionality to CPRS along with the existing added tabs for Juno EHR Clinical Content/Workflow Builder and Juno EHR ProDash.

> **How will Juno EHR Order Management Tool be Introduced into CPRS:** Juno Order Management is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud. Integration to VistA would require the **Juno EHR Bidirectional Interfaces** enhancement. JOMt can be implemented as a CPRS tab using the **Dynamic CPRS Tab Management** enhancement discussed in this document. This provides a seamless integration with VistA/CPRS. With dynamic, customizable CPRS Tabs functionality, only administrative users will see the Order/OrderSet Builder features, while all ordering clinicians will have access to the CPOE components. VA can deprecate the existing CPRS Orders and Notes tabs with a simple configuration change when the site is ready to fully cut over to JOMt.

What is Juno BCMA/eMAR? Barcode Medication Administration (BCMA) and Electronic Medication Administration Record (eMAR) work together to accurately administer, document, and track patient medication administration events to ensure patients take and receive their prescribed medications with the correct dosage, on time, by the proper route. This allows for an easily accessible, up-to-date record of the current prescribed medication regimen for the patient. BCMA supports 1D and 2D barcode scanning and contains an integrated drug reference. Users can also collect vitals and other clinical information during administration such as 'unable to scan', PRN effectiveness, witness data for high-risk drugs, and other observations. BCMA/eMAR workflow provides capture of the collection of the six rights of medication administration: right patient, right drug, right dose, right route, right time, and right documentation. The combination of BCMA/eMAR supports best practices for clinical medication workflow, provides labor and cost savings, and improves drug inventory management. Future state development includes the ability for notes between nursing, pharmacy, and prescriber.

> **How will Juno EHR BCMA/eMAR be Introduced into CPRS?** Juno BCMA/eMAR is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud. Juno BCMA/eMAR is dependent on the **Clinical Content/Workflow Builder**. Integration to VistA would require the **Juno EHR Bidirectional Interfaces** enhancement. These modules can be implemented as CPRS tabs using the **Dynamic CPRS Tab Management** enhancement discussed in this document. This provides a seamless integration with VistA/CPRS and legacy BCMA data files if these are necessary for downstream reporting. With dynamic, customizable CPRS Tabs functionality, only administrative users will see the BCMA/eMAR configuration screens, while nurses, pharmacists, and other staff will have access to the BCMA/eMAR modules as desired by VA. Much of the current BCMA solution used by VA relies on VistA roll-and-scroll user interfaces such as the Bar Code Medication Administration Manager menu options, Medication Administration Menu Nursing options, and Medication Administration Menu Pharmacy options. Juno BCMA/eMAR provides modern, browser-based configuration utilities and user interfaces to replace the legacy options. An additional benefit of adding the Juno EHR eMAR/BCMA to CPRS is a unification of nursing workflows into a seamless whole.

What is Juno EHR CAF? Juno EHR CAF is the Common Application Framework that is employed by all Juno EHR modules for a common look and feel. As a fully web-based interface it allows for easy integration of both existing Juno EHR developed content as well as nearly seamless integration with other web-based third-party applications. This will effectively allow for CPRS to be officially deprecated in favor of a fully web-based Juno EHR user interface. Depending on the module, CAF will display two different banners: the common patient banner is displayed for all clinical modules when dealing with a specific patient, and the common user banner when either selecting the clinician's current patient list or for system configuration modules. CAF includes the vitally important indicators of Crisis Notes, Warnings, and Directives. The existence of any of these items in the patient record will generate an indicator on the Common patient banner, and similar to CWAD in CPRS today, the underlying documentation can be reached with a single click. Adverse Reactions/Allergies, Pregnancy, and Lactation are always shown on the Common Patient banner. Additionally, unlike CPRS the most critical adverse reactions/allergies are fully spelled out on the patient banner with a visible indicator that more adverse reactions/allergies exist when the space on the Common patient Banner is not sufficient to display all the patient's adverse reactions/allergies.

With the introduction of Juno EHR CAF and the use of Juno EHR Clinical Content/Workflow Builder all remaining CPRS functionality including the Problems tab, Reports tab, Allergy entry and display (replaced as part of the Common Patient Banner), and the Coversheet tab can be deprecated and replaced with new and enhanced Juno EHR web-based functionality while maintaining all the current CPRS functionality.

> **How will Juno EHR CAF be Introduced:** Juno CAF is a web server application that runs on a client's browser. The web server components and enterprise MSSQL Server database are installed in the cloud. Juno CAF is dependent on the **Clinical Content/Workflow Builder**. Integration to VistA would require the **Juno EHR Bidirectional Interfaces** enhancement. At this point in Phase III, much of the legacy CPRS functionality has been replaced by Juno EHR web-based functionality, and CPRS is primarily a thick client GUI that launches Juno EHR modules. The remaining CPRS functionality is either already replicated in the CAF or can be configured with existing Juno EHR CAF tools. Once CPRS is no longer needed, the existing installed footprint can be removed and replaced with a web link to Juno EHR.

About DSS, Inc.:
With over 30 years of experience, DSS knows what works. We are a health information software development and systems integration company, providing services and solutions used daily by thousands of clinicians and administrative staff nationwide, to reduce costs, streamline workflows, and improve patient care.

For More Information
Contact DSS at 561-284-7000 or email sales@dssinc.com to arrange for a custom demonstration at your convenience.

12575 US HWY 1, SUITE 200, JUNO BEACH, FL 33408
dssinc.com