**RFI Technical Evaluation Summary Report**

on

VistA Adaptive Maintenance

*Submitted to:*

**Technology Acquisition Center**

July 27, 2017

**TABLE OF CONTENTS**

[1 Executive Summary 3](#_Toc370733348)

[1.1 Background 3](#_Toc370733349)

[1.2 Evaluation committee 4](#_Toc370733350)

[1.3 evaluation methodology 4](#_Toc370733351)

[2 Evaluation Summary 5](#_Toc370733352)

[3 RFI RELATED QUESTIONS / CLARIFICATIONS / SUBMISSION 6](#_Toc370733353)

[4 Conclusions and recommendations 6](#_Toc370733354)

# Executive Summary

## Background

The Veterans Information System Technology Architecture (VistA) is the Congressionally-approved, authoritative, comprehensive, longitudinal veteran health information system of the U.S. Department of Veterans Affairs (VA). For the past thirty-five years, 131 VistA systems have provided all clinical, financial, and administrative functions to support all clinical and administrative operations of over 1200 VA hospitals and clinics throughout the United States. Each VistA system is comprised of over 180 clinical, administrative, and financial applications integrated within a single database, with many of these applications specific to VA care and benefits, and government compliance and reporting.

The VistA Standardization and Virtualization (VSV) project – the parent project to this Performance Work Statement (PWS) -- is intended to move all possible VistA instances that have been modified at the local facility level to a single, enterprise-deployable, standard software image. The VSV project will benefit Veterans, Clinicians and OI&T staff by achieving commonality across multiple versions of VistA, supporting over 1,600 points of care.

This VistA Adaptive Maintenance project - under the parent VSV project – is intended to progress and further this transformation by modifying select components of VistA to facilitate continued usability in a changed or changing environment. The goal of the VistA Adaptive Maintenance project is to provide backwards and forwards compatibility for selected VistA components and use cases such that VistA’s Graphical User Interface (GUI), the Computerized Patient Retrieval System (CPRS), business functionality is isolated and emulated via a service layer to enable retrieval by CPRS and accessibility to those components via web-friendly interfacing by new clients.

## Evaluation committee

The evaluation committee was comprised of.the Project Manager, Dr. Rafael Richards.

## evaluation methodology

The team established a pass/fail set of criteria (see section 3 for results). Based upon the information provided, an overall determination as to whether a particular company could successfully perform the work was determined based on the initial pass/fail criteria. The respondents evaluated were:

**T4NG**

* Systems Made Simple/Leidos (SMS)
* AbleVets, LLC (AbleVets)
* Liberty IT Solutions, LLC

# Evaluation Summary

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SMS** | **AbleVets** | **Liberty** |
| **SDVOSB/VOSB/Small** | No | Yes | Yes |
| **Overall Capable** |  |  |  |
| **5.1 Project Management** | Yes | Yes | Yes |
| **5.2. Sustainment Services** | TBD | Yes | TBD |
| **5.3 Planning** | Yes | Yes | Yes |
| **5.4 Build and Development** | Yes | Yes | Yes |
| **5.5 IOC Support** | Yes | Yes | Yes |
| **5.6 Release and Deployment Support** | Yes | Yes | Yes |
| **5.7 Transition Support** | Yes | Yes | Yes |

# RFI RELATED QUESTIONS / CLARIFICATIONS / SUBMISSION

Questions provided separately

1. **Conclusions and recommendations**

**T4NG**

* SMS – SMS referenced and described their approach in their JALFHCC project in which the application sends data from CPRS to the DoD EHRs and was developed without using or modifying any legacy MUMPS code. This is accomplished using unique protocols, HL7 Logical links and a generic HL7 message generator. Once these generic HL7 messages are created, our process allows us to modify all of these messages to satisfy the business requirements of the receiving EHR. Our application also has a Java script based unique receiver which is capable of changing not only the HL7 version but also converting the message to XML format which most outside applications can read. They also reference another working part of their application being the ESB. This application is capable of storing, queueing, mapping and directing the incoming or outgoing messages to the appropriate recipient. They state that after successfully creating and sustaining this unique application that transfers CPRS actions to an outside recipient and receives the same from an outside source, they have the necessary resources and knowledge base to effectively support the creation of a similar process that will send and receive data from the newly created VICS. To support this effort, they also have experience maintaining three unique development and SQA environments cloned from a production system that we can use for our development. These existing environments enhance SMS/Leidos’ ability to provide Regression Test Suite to cover all CPRS vital interactions and configuration of a VistA Test System (production clone) to validate interfaces to Vitals VICS. SMS referenced other projects as well in which they used a series of new protocols, HL7 logical links, HL7 messaging and specific namespace routines to accomplish this process and made it complete transparent to the existing functionality of either facility.
  + SMS indicated its Node.js capabilities and expertise, indicating that for the VistA Adaptive Maintenance project, Node.js will be used to develop and deliver Node.js Package Manager (NPM-packaged) REST services that can be accessed by CPRS. CPRS will then need to be configured to disable VistA data calls and enable the REST services created with Node.js, freeing up CPRS dependency on VistA for EHR specific data. They created the initial prototypes for the Medical Care Collection Fund (MCCF) Electronic Data Interchange (EDI) modernization package using Node.js and delivered that code to the development contractors. Their team established the use of the MEAN stack path for MCCF.
  + Leidos’ approach and historical references do not indicate any RPC emulation experience, which is a direct requirement of this PWS.
* AbleVets – AbleVets, in its revised response, elaborated extensively on how its existing work on the VistA (Meta)Data Project (VMDP, also referenced as VDP) directly relates to the VistA Adaptive Maintenance Project and has resulted in addressing many of the pre-requisite objectives to the VistA Adaptive Maintenance Project that would streamline the delivery of the stated capabilities. For VDP, specifically, their project has delivered the following results:
  + CPRS runs unchanged over a secured, model-backed, services-based emulation of its Remote Procedure Call (RPC) interface
  + Services are also exposed as REST for direct use by new web-based clients allowing both CPRS and newer clients to run side-by side
  + Implementation is in industry-standard Javascript, packaged as node.js modules (No MUMPS)
  + A Security module provides for client authentication, auditing and traffic encryption
  + RPCs being emulated are from a variety of clinical domains including allergy, vital, problem and pharmacy as well as patient demographics, PCE, TIU documentation and encounters.
  + A clear distinction is made between RPCs accessing and changing sensitive Patient data from those accessing the meta-data that drives business logic
  + All code, demonstrations and documentation including the VA-mandated project web site (vistadataproject.info) is being developed using Agile methodology and hosted in an open, industry-standard, source code repository.
  + Beyond addressing an RPC’s interface, emulation involves analyzing and accounting for:
    - inter-domain synchronization issues including the maintenance of TIU Document Macros, Alerts and Reminder Dialogs and the use of common functionality by inter-dependent domains (allergy’s use of TIU Document signing)
    - non CPRS user interfaces, specifically the roll and scroll interface built into VistA and its use by Lab Technicians, Pharmacists and Remote Users
    - workload monitoring and other VA business and process management functionality that relies on the clinical data established by CPRS through RPCs
    - third party services used directly or indirectly by RPCs including the MOCHA service used by Pharmacy
  + Two deployment scenarios are being provided:
    - In a pure node.js-based cloud-compatible environment of the kind called for in the PWS
    - Within a node.js-enabled VistA system
  + Testing involves:
    - a Javascript-based, data-driven regression test suite for all emulated RPCs which ensures that emulation matches expected behavior
    - a fully functional test VistA configured with sample patients, users and system configurations required by RPCs
  + VDP has established that model-based emulation can allow an unchanged CPRS to run securely alongside newer web-based clients over centralized, easy to manage services. Its lessons and frameworks provide an ideal basis for the VICS required in the draft PWS.
* Based on AbleVets’ response, they demonstrate sufficient and significant capabilities to meet the requirement.
* Liberty – Liberty’s technical approach in their response to the 2nd round of the RFI and PWS was significantly improved and much more technically focused on PWS 5.2, as requested. They spoke to their high level capabilities to decouple business logic and RPCs and MUMPs from CPRS and indicated work performance addressing the workflow and business logic that is contained in the RPCs being used on CPRS, Vitals, Allergy, Patient Problem, Pharmacy, and FileMan, all of which represent clinical applications and packages referenced in the draft PWS. They indicated that their expertise and historical experience in this decoupling through the HPS Tier3 Sustainment Support: Clinical Task Order. They provided specific examples and numbers of RPCs in these packages that would require modification. They indicated historical experience supporting VistA Services Assembler/VistA.js in both Development, Deployment and Sustainment where Liberty worked to integrate Node.js RESTful services with VistA, which used the cache.node API. Their approach centered on VistA Kernel login code to pass a SAML assertion collected from the VA IAM. This included decoupling business logic, understanding workflows within RPCs as well as eventually identifying issues with cache.node regarding the use of a shared process id that created problems with VistA file locks. Liberty also referenced work on VIA, Vista.js, and VSA projects. Their approaches also involved deep security considerations as indicated by the requirement for enhanced data security for access control, auditing, and RPC encryption.
  + Although each of these projects has required the decoupling of business logic from MUMPS-based RPCs, each of these projects have wrapped MUMPS RPCs, which indicates continuation of a legacy MUMPS code dependency, contrary to the PWS objective stated in the background and in Section 5.2 of the PWS, which states, “Surface specific VISTA/CPRS clinical function and business logic, and emulate with an industry-standard, model-driven, secure service interface with no legacy Massachusetts General Hospital Utility Multi-Programming System (MUMPS) code dependencies while keeping CPRS operational (supporting both backwards and forwards compatibility).” Therefore, it is believed that they may not have experience with RPC emulation.

Based on the revised responses to the second RFI release, we believe that: State the summary.