ElyonStrategies

Cal-Access Replacement System Independent Assessment Advisory Services

High Level Assessment October 14, 2021

Title	CARS High Level Assessment		
Document #:			
Version:	WORKING DRAFT	**DATE:**	3/14/2022
Author(s):	SCOTT OHLUND, KEITH MOREHOUSE, WILLIAM ROETZHEIM		
Customer:	California Secretary of State (SOS)		
Contract:	21C60064		
Project:	Cal-Access Replacement System (CARS) Independent Assessment Advisory Services		
Deliverable ID:	C.1 AND D.1		

VERSION HISTORY

Version	Author	Date	Changes
DED	WHR	9/7/2021	Original DED
Working Draft	Multiple	10/14/2021	Working Draft
Final	Multiple	10/21/2021	Final version, with CRM

Approvals

Role	Name	Signature	Date
Author	Scott Ohlund		10/21/2021
Author	Keith Morehouse		10/21/2021
Author	William Roetzheim		10/21/2021
QM	Karen Morphy		10/21/2021
CEO	Carl Engel		10/21/2021

Table of Contents

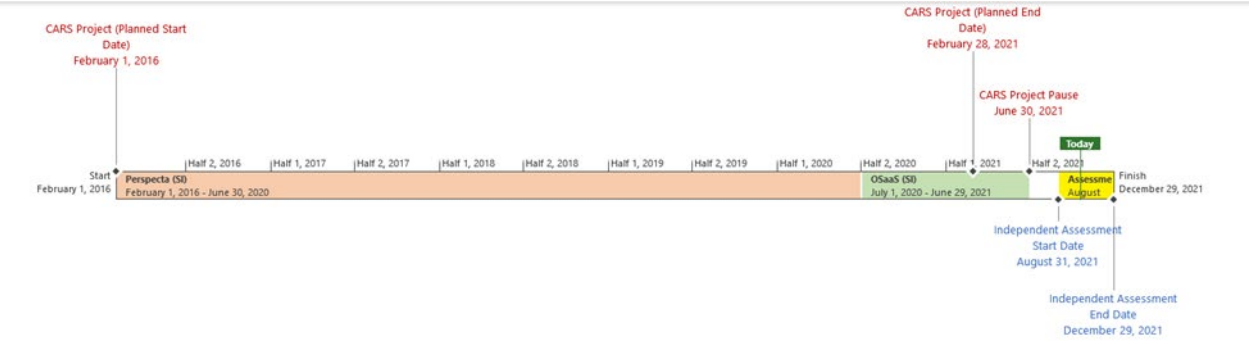
Contents

1 Executive Summary

The California Automated Lobbyist and Campaign Contribution and Expenditure Search System (CAL-ACCESS) is the public's window into California's campaign disclosure and lobbying financial activity, providing financial information supplied by state candidates, donors, lobbyists, lobbyist employers, and others. CAL-ACCESS is the mission-critical legacy system for the Secretary of State (SOS)'s administration of the Campaign Finance and Lobbying disclosure program. The Cal-Access Replacement System (CARS) project began as a roughly five-year project tasked with implementing a new system to replace the legacy CAL-ACCESS solution with a modern technology-based, data-driven system. CARS was operating under a legislative mandate to complete the new system by February of 2021.

CARS business requirements are characterized by complex business rules that are unique to this application and subject to change over time; complex workflow and form-flow requirements that are integrated with the business requirements; significant descriptive, relationship, and financial data that needs to populate to forms and reports; requirements to accept various degrees of "dirty" data for subsequent correction; and extensive version control and redlining at the field level. The application is similar to a taxation system with requirements for form or wizard-based data entry in accordance with complex requirements, plus compliance reviews/audits. A key requirement of the external community portal is the ability to query, tabulate, and compare data across multiple years.

The project encountered difficulties with the selected implementation vendor (Perspecta), and roughly four years into the effort switched to a new primary implementation vendor, Outreach Solutions as a Service (OSaaS). At the time of that OSaaS contract award, the legislatively mandated deadline for completion of CARS was approximately eight months away. In June of 2021, four-months after the target completion date, the project was paused pending an assessment and development of a go-forward strategy. The CARS project timeline is shown in Figure 3. Perspecta was under contract to deliver CARS from February 1st, 2016, through June 30, 2020. Following termination of that contract, OSaaS was under contract to delivery CARS starting on July 1st, 2020, through February 28th, 2021.



The term “death march project” was coined by Edward Yourdon in his book *Death March*[1]. He defined a death march project as one whose project parameters exceed the norm by 50%. These projects are characterized by heroic efforts, long hours, burnout, and in the end, failure. The task assigned to the CARS project team overall, of doing several years-worth of development work in eight months, was quite simply impossible and the result was a classic death march project.

One characteristic of these projects is that time pressure means the vital foundational and architectural work is rushed or skipped altogether. The team moves forward to begin the development phase, without a clear understanding of what needs to be built, and without an optimum and supportable underlying architectural structure. In virtually every case, the project begins to quickly build something, but they are building the wrong thing, and they are building it the wrong way. CARS was no exception to the rule.

Unfortunately, when an Information Technology (IT) project has these foundational problems, much of the software that has been developed has limited use. Even when it can be patched and extended to support the updated architecture, it will suffer from problems in areas including reliability, maintainability, security, and performance. For much of the developed application, the total cost of ownership to rework or repurpose the software correctly is typically prohibitive for many reasons (e.g., cost). We do recommend at least an initial level of examination of the developed application software once the project is recast to determine what is worth keeping versus what needs to be completely replaced.

To analyze the current project, Elyon used quantitative models to assess: the degree to which Salesforce is a match for the given application (the Salesforce fit-gap); the quality of the given Salesforce implementation effort (the Implementation Quality); and the demonstrated capabilities of the given system integration team to perform necessary system integration functions. The result is a score between 1 and 5 where 1 is Very Poor, and 5 is Very Good. The scores produced using Elyon’s enterprise maturity

[1] Yourdon, Edward (2014) [1999]. Death March. Prentice Hall.

readiness index (eMRI) for the CARS project are shown in Table 1 and graphically portrayed in Figure 2. The project has clear challenges in all three areas, and the decision by the Secretary of State to pause the project for an assessment was a wise one. The primary reason that Salesforce is not a good fit for CARS is the complexity of the highly specialized business processes and workflows, and it is the major contributor to the CARS Salesforce fit-gap score of 1.42. In addition, because the PRD business processes, workflows and business rules are specialized and complex, and change fairly often, *and* the current CARS architecture does not address these needs effectively, the resulting implementation quality score is also low.

Table 1: CARS eMRI Scores

	Score (1 to 5)
Salesforce Fit-Gap	1.42
CARS Implementation Quality	2.13
System Integrator Fit-Gap	1.27

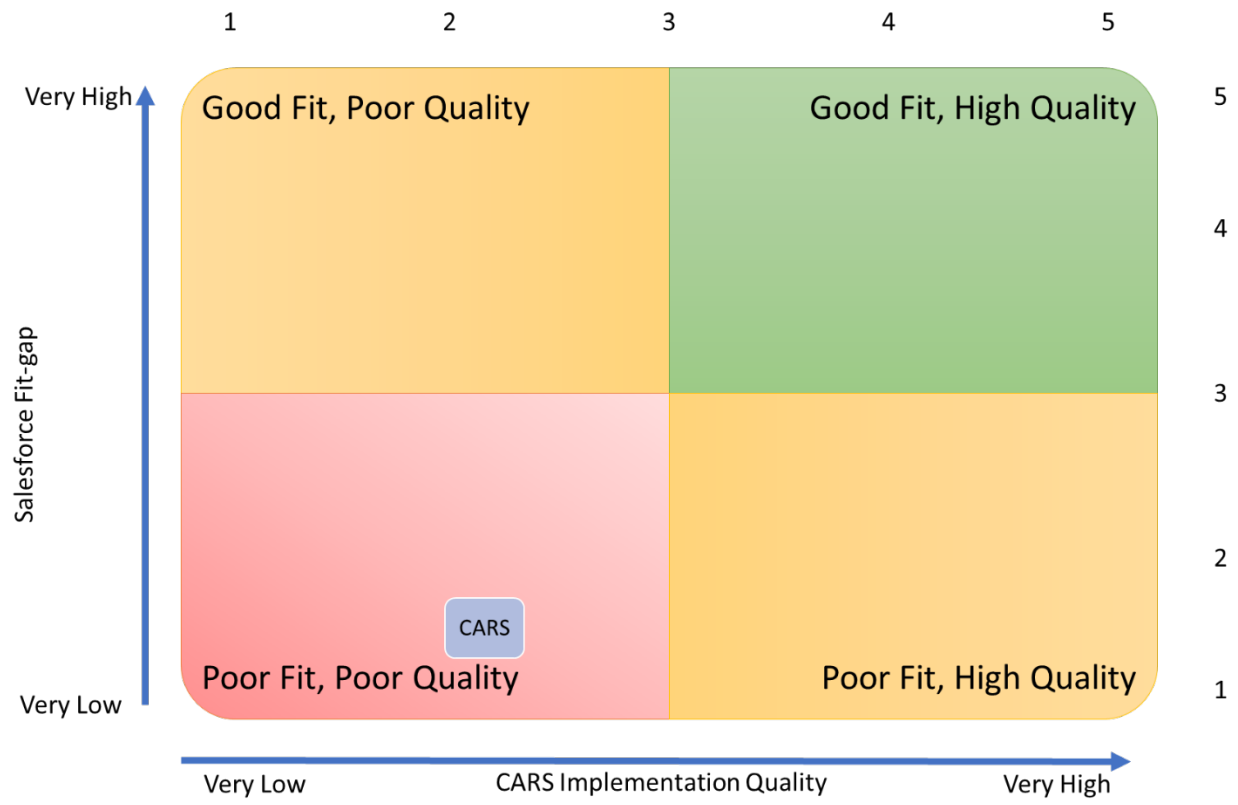


Figure 2: CARS Fit-Gap Quadrant is Poor Fit, Poor Quality

We conclude this executive summary by answering four specific questions:

1. Is the project nearly complete, or if not that, at least in a condition where incremental deployments on top of the existing code base is a wise strategy?
    a. No. The existing system is flawed at the architecture, data structure, middle-tier, and user presentation layers. The architecture, data structures, and middle-tier are not correctly optimized for the necessary business processes, workflows, and data structures. The user presentation layer takes a purely form-centric view of the world, as opposed to an underlying data-centric view. Those flaws are fundamental and recasting the project will be more cost effective than to continue development. Anything less will likely result in an unreliable system with significant functional deficiencies that is expensive and difficult to maintain.

2. Should the restart require Salesforce as the platform?
    a. Probably not. While Salesforce, with an external business rules and data-centric integration architecture could work to deliver a part of the required functionality, the fit-gap analysis makes it clear that Salesforce is not a good/best fit as the foundation for the CARS project, because of the degree of Salesforce customization to meet the CARS requirements.

3. Should the OSaaS contract be modified and extended to support this work?
    a. No. The project should move to a firm-fixed price deliverable-based contract model, and a competitive acquisition would be more appropriate for this new contract. However, we see no reason that OSaaS should be precluded from bidding on the recompete.

4. Was the work performed to date a complete waste of money?
    a. Absolutely not. The CARS restart will greatly benefit from a significant amount of the work that has been completed to date, including work in the areas of requirements, business rules, workflows, data analysis, data conversion, and data clean-up. Our budgets prepared as part of December Roadmap deliverable will include appropriate allowances for the cost savings thus realized.

The CARS ecosystem is complex, and the System Integration (SI) role must be organized and orchestrated effectively to achieve project success and solution quality. This includes:

- The SOS must recast and formalize the new CARS vision, strategy, success metrics, decision matrices, and project charter.

- Within the new strategy, the CARS project must clearly define and assign the SI role to a group with the competency and capacity (e.g., roles, responsibilities, skills, and authority) to support the continued definition and execution of the new CARS project strategy.
- The SOS and selected SI vendor must execute on the recommendations in Section 3.2.2.

In our next report (the 60-day assessment) we will identify specific recommended next steps for the project.

2 Introduction and Approach

2.1 Project Background.

The California Automated Lobbyist and Campaign Contribution and Expenditure Search System (CAL-ACCESS) is the public's window into California's campaign disclosure and lobbying financial activity, providing financial information supplied by state candidates, donors, lobbyists, lobbyist employers, and others. CAL-ACCESS is mission-critical legacy system for the Secretary of State (SOS)'s administration of the Campaign Finance and Lobbying disclosure program. The Cal-Access Replacement System (CARS) project is tasked with implementing a new system that replaces the legacy CAL-ACCESS solution with a modern technology-based, data-driven system. This system should allow campaign and lobbying entities to meet the filing requirements of the Political Reform Act (PRA) more efficiently, improve data quality, expand public access to data, allow for system modifications and improvements to respond to statutory and regulatory changes, allow other system modifications to improve filer efficiency and public access to data, and improve the ability of the SOS, the Fair Political Practices Commission (FPPC) and the Franchise Tax Board (FTB) to fulfill mandated duties.

Elyon Strategies was hired to holistically and objectively assess the current health of the CAL-ACCESS Replacement System (CARS) Project. As part of this assessment, Elyon is working closely with the SOS and the CARS team, including its various vendors, to effectively evaluate the business, technical infrastructure, project management practices, and to provide a corrective action plan and roadmap. The results of the assessment will guide the SOS with a path forward to develop a remediation plan to drive towards achieving successful completion, implementation, and delivery of the CARS system with the goal of meeting or exceeding Political Reform Division (PRD) and external stakeholder business needs, fulfilling legislative and statutory requirements, and functioning consistent with the legislative intent stated in Government Code section 84601 as well as other provisions of the Political Reform Act. The CARS project timeline is shown in Figure 3. Perspecta was under contract to delivery CARS from February 1st, 2016, through June 30, 2020. Following termination of that contract, OSaaS was under contract to delivery CARS starting on July 1st, 2020, through February 28th, 2021.

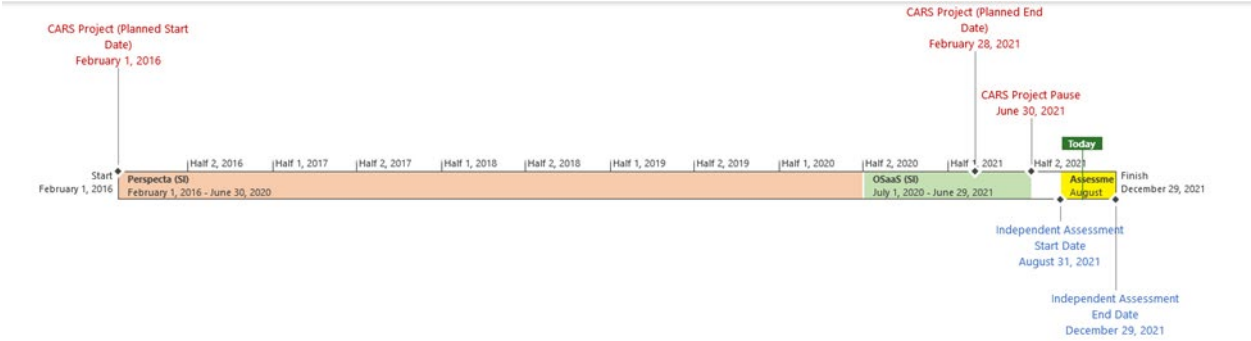


Figure 3: CARS Project Timeline.

2.2 Purpose.

This initial report has two primary purposes. First, it serves as a holistic fit-gap assessment of the current CARS architectural and vendor strategy, answering the following two questions:

- Can the current SOS Salesforce technical architecture, including 3rd party products and integration capability, meet the requirements for this system in a cost effective and supportable manner?

- Can the current OSaaS vendor deliver the required solution?

Second, we share some high-level initial findings and observations as a form of work-in-progress review. Our comprehensive assessment of the project will be completed as part of the second report (due December 1st, 2021), and during that report these initial findings and observations will be expanded, supplemented, and fully supported.

2.3 Approach.

Elyon Strategies is a management consulting and professional services company, providing an integrated service catalog in strategy, architecture, portfolio management, advisory, assessment and improvement services to achieve complex transformation. Because our focus is on providing independent project oversight support to government agencies, we tend to be involved primarily in the largest and most complex projects. Those are the projects where the need for project oversight is recognized and the budget for oversight is available. Here in California, we have provided this service for many of the State's largest and most complex projects, including the following projects (all of which were between $100M and $1B in size; required integration of multiple services/modules across multiple platforms; supported near-real-time/streaming data/event processing; and supported between 12K and 50K concurrent users at

multiple geographically dispersed locations, including data consumers with advanced analytical needs):

- CDSS CWS/CMS project (a child-welfare case management system).
- FTB EDR and EDR2 projects (both are taxation systems).
- CDTFA CROS project (another taxation system).
- CalHEERS (a health insurance portal).
- CDSS SAWS, C-IV, LEADER, and LEADER-Replacement projects (all welfare systems).
- CDSS CWS-CARES project (described below).
- CHHS CCSAS project (a child support case management system).

Elyon is currently under contract as an Independent Advisor to the State of California's Office of Systems Integration for the Child Welfare Services – California Automated Response and Engagement System (CWS-CARES) Project. The Project is delivering a highly complex replacement for the State's legacy child welfare systems comprised of both Salesforce applications and CARES Data Infrastructure services hosted on Amazon Web Services (AWS) infrastructure, and is a highly regulated, safety-critical, and data-intensive system. It is being developed through a user-centered, research-based, iterative, and agile process. Elyon is independently assessing if the Project is on track to deliver a service that meets or exceeds Federal compliance, State program goals and County user needs, and whether there are alternate approaches that could increase speed to value, reduce costs, and increase the usability of the solution.

The CARES Project Salesforce Fit-Gap came back with a score of 3.54 (scale 1 to 5). It is a classic use case of a Case Management solution to be used with serving the families and children in the state of California for Child Welfare needs.

Two other Salesforce implementations that were scored in a fit-gap analysis also used Service Cloud (Case Management) solutions. They respectively scored 4.2 and 4.6 (scale 1 to 5).

We have also provided this independent advisory / assessment service for large government projects in the states of Washington, Oregon, Colorado, Texas, and Florida; as well as for Federal government agencies including the Department of the Interior, the Department of Agriculture, the Department of Veterans Affairs, the Department of Homeland Security, the Department of Energy, the Federal Aviation Agency, and the General Services Administration. We have provided platform assessments for the various state government departments seeking to modernize legacy applications and determining comparative fit for platforms including Salesforce, Microsoft Dynamics, and Pegasystems. Elyon's consultants have provided applications utilizing these platforms as well, giving our team full Solution Development Life Cycle experience from strategy to implementation and positioning us well to understand both

the complexities and benefits of modernization efforts that utilize a platform-based solution.

In performing this work, we model the project using our enterprise Maturity Readiness Index (eMRI) and ExcelerPlan tools. eMRI is an enterprise project process assessment tool, assessing the project process capabilities versus industry best practices, value weighted to the specific processes required for success on this project. eMRI project modeling is in terms of Key Process Areas (KPAs). ExcelerPlan is a benchmark driven system dynamic modelling framework that uses benchmark data to create a model of project success, also tailored to this project. ExcelerPlan project modelling is in terms of High-Level Objects (HLOs) and Function Point Equivalents (FPE), which are industry standard ways to define application scope; plus, Other Direct Charges (ODCs), including infrastructure and licensing; Maintenance and Operations (M&O) support requirements; and project characteristics that impact efficiency. ExcelerPlan's models are based on data from over 40,000 projects.

In configuring and modeling the CARS project we use a combination of analysis of project artifacts (documents and development environments) and stakeholder interviews. To date we have conducted forty-six (46) stakeholder interviews/meetings, reviewed 15,880 documents from a high-level perspective, and identified 1,198 of those documents that are relevant to our analysis. In addition, we reviewed the current Salesforce code in the DevOps system. The purpose of this work was to fully understand the CARS scope and current implementation. The documents that were reviewed were the versions in the SOS SharePoint site, which we believe are the latest version of each document. The list of interviews is included as Appendix B.

2.4 Assumptions and Constraints.

Our analysis is based on the following assumptions:

- The state seeks an optimal go-forward strategy. In accordance with Generally Accepted Accounting Procedures (GAAP), sunk costs are ignored in performing the financial portion of this analysis[2]. This is because those sunk costs will be the same under all potential scenarios going forward.

- The State is not contractually obligated to continue with the current Salesforce based architecture or with the current vendors.

- In terms of the trade-off between quality, scope, schedule and cost we assume that:

[2] For a good discussion of this topic, see: Sunk Cost - Why You Should Ignore Them (the Sunk Cost Fallacy) (corporatefinanceinstitute.com).

- The system must have sufficient quality to be both reliable and maintainable at the time it is deployed.
- The system must have sufficient scope to meet at least the basic needs of all internal and external stakeholders at the time it is deployed, or as an alternative, a phased deployment approach will be approved by the impacted stakeholders. We will explore both alternatives further as part of the next phase of our analysis.
- Schedule and cost should be adjusted to support the above objectives.

It should be noted that our analysis was constrained to a thirty-business-day analysis, which was intended to create extreme focus on product, platform, and approach. This initial assessment is based on a thirty-business-day analysis period, so there are limitations on the number of documents we can study, the number of interviews we can conduct, the amount of independent validation that we can perform, and so on. Our scope for this initial assessment is therefore limited to the areas identified in Section 2.2 above.

2.5 Risks and Issues.

While we requested one-on-one interviews with OSaaS staff, OSaaS elected to have a senior OSaaS executive present in each of the interviews. We cannot assess the degree to which this interfered with the candor of the OSaaS staff being interviewed. Our mitigation strategy for this risk was to ask for State assistance in terms of providing guidance to OSaaS, which was done, but OSaaS continued to have joint interviews.

3 CARS Fit-Gap Analysis

A key question for our analysis is the extent to which the current Salesforce based solution architecture and the current system integrator (OSaaS) are suitable for the CARS application going forward. In other words, is the project almost finished but it needs some guidance along with more time and money? Or is the project in a situation where continuing on the current path would be throwing good money after bad? As discussed in Section 2.2 above, this Chapter presents a holistic fit-gap assessment of the current CARS architectural and vendor strategy, answering the following two questions:

- Can the current SOS Salesforce technical architecture, including 3rd party products and integration capability, meet the requirements for this system in a cost effective and supportable manner?

- Can the current OSaaS vendor deliver the required solution?

3.1 Technical Architecture Assessment.

3.1.1 CARS Architecture.

As shown in Figure 4 and Figure 5, the CARS technical architecture consists of three primary components:

- Salesforce is used for the form entry, business rule processing, and data storage.

- Heroku is used for the public facing portal, with an automated feed from Salesforce. Of note, the automated nature of the data feed means that the public portal and Salesforce are tightly coupled, so that changes in one will require changes in both.

- Mulesoft is used for the Application Programming Interface (API) used to accept data from filing partners (vendors). Again, the automated nature of the Mulesoft connection to Salesforce means that the Application Programming Interface (API) and Salesforce are tightly coupled, so that changes in one will require changes in both.

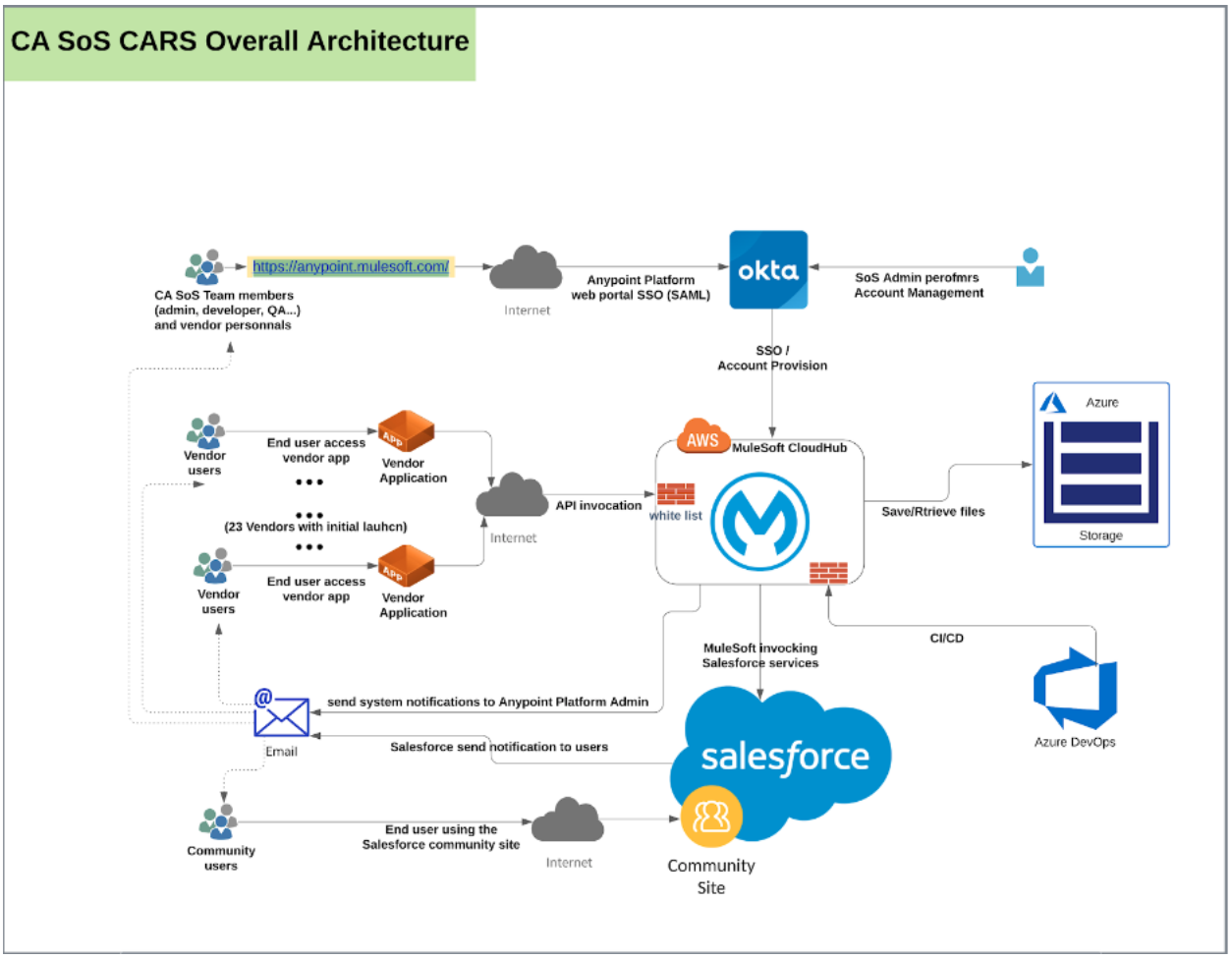


Figure 4: CARS Overall Architecture Overview

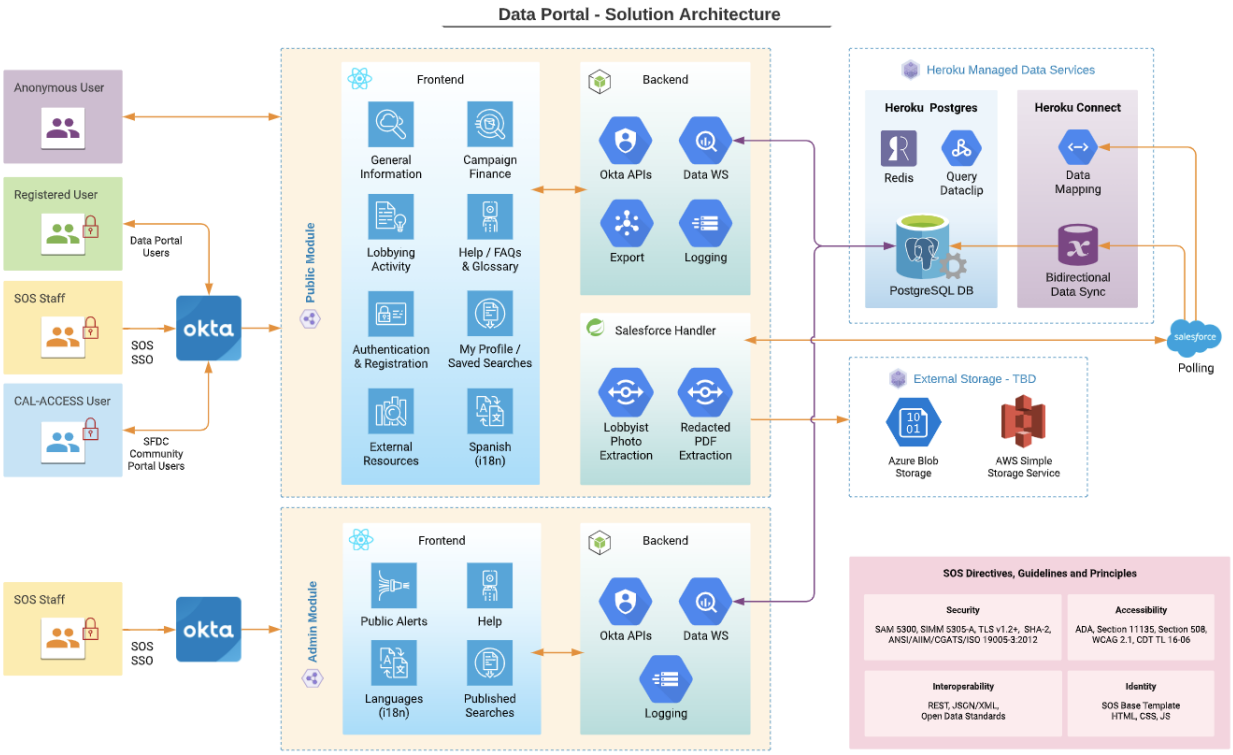


Figure 5: CARS Data Portal Architecture

3.1.2 CARS Requirement Characteristics.

In conducting our analysis, we characterized the CARS requirements as follows:

- Business Functions: CARS business requirements are characterized by complex business rules that are unique to this application and subject to change over time; complex workflow and form-flow requirements that are integrated with the business requirements; significant descriptive, relationship, and financial data that needs to populate to forms and reports; requirements to accept various degrees of “dirty” data for subsequent correction; and extensive version control and redlining at the field level. The application is similar to a taxation system with requirements for form or wizard-based data entry in accordance with complex requirements, plus compliance reviews/audits. A key requirement of the external community portal is the ability to query, tabulate, and compare data across multiple years.

- Performance: Performance loading is significantly predictable over time, with daily peaks at about 6 PM; bi-annual cycles based on the election cycle; and significant loading near known filing deadlines.

- Security: Data integrity considerations are the primary security concern, with significant consequences in the event of unauthorized data modifications. Versioning to the field level is needed to support internal and external auditing.

- 3rd Party Interfaces: Approximately 70% to 80% of the data input to the system comes from external, third-party vendors. In some cases, this data can be large, consisting of up to a half-million records for a single filing.

- Americans with Disabilities Act (ADA) Capabilities: As with all government systems designed for use by the public, CARS must support ADA accessibility requirements.

- Data Conversion: Historic data, currently in Oracle, must be converted and validated. In some cases, the data will not be compliant with current rules, so for example current business rules might require an email address, but there will be historic data with no email address and no reasonable way to obtain an email address. So, you can't convert what is not there.

- Maintainability: Business rules, workflows, and form-flows are subject to change on an on-going basis. It may be necessary to update the central database structures independent of the API so that the system can be modified while coordinating API changes with the external vendors. The complexity of the business rules means that significant self-test, internal diagnostic, and variable debug logging capabilities will be needed to maintain the system.

3.1.3 Fit-Gap.

In conducting our fit-gap analysis we used two independent set of models from our eMRI assessment tools, one to assess the suitability of Salesforce to the CARS application, and the second to assess the overall quality of the current CARS implementation. The models use a weighted multi-variate assessment approach to arrive at an overall assessment of 1 to 5, where 1 is Very Low, 3 is typical or average, and 5 is Very High. Table 2 shows the interpretation of the assessment scores in more detail.

Table 2: eMRI Score Interpretation

eMRI Score	Interpretation
5	Fit-Gap: The application is an ideal fit for the Salesforce platform. Implementation: The implementation is best of breed.

	System Integration: The SI team is best in class, with mature and fully supported processes in place covering all major skill areas.
4	Fit-Gap: The application is a good fit for the Salesforce platform, though it is likely that Salesforce will require some important supporting tools or customization. Architectural analysis is important. Implementation: The implementation is solid and supportable, although there are some areas for improvement. System Integration: The SI team has effective and fully supported processes in place for most areas, but there are some areas of weakness that may decrease efficiency or increase risk, but not to the point of endangering project success.
3	Fit-Gap: Salesforce will work, but other tools/products are an equally good fit. The degree of organizational experience with Salesforce may be a deciding factor in selecting the right approach. Significant architectural analysis is important because the core Salesforce capabilities will likely need to be supplemented. Implementation: The implementation includes areas of strength and areas that need improvement, but the weaknesses can be overcome without major restructuring. System Integration: The SI team has effective and supported processes in place for key areas, but in other areas the team relies on individual skills and actions. Projects will often get into some trouble, but with work by all members of the team success is achievable. These projects tend to require significant oversight and project/portfolio management attention.
2	Fit-Gap: Salesforce is not an ideal choice, but it can potentially be part of the solution. If Salesforce is used, it will likely be in a supporting role with other architectural components adding significant business capability outside the Salesforce environment. A clear allocation of functionality between components, and an understanding of the internal interactions between architectural components, is critical. Implementation: The implementation includes areas of strength and areas that need improvement, and some of the weaknesses will require significant restructuring of the system. System Integrator: The SI team is largely dependent for success on the skills of individuals doing the work, rather than processes. Small

	and simple projects will often still be successful, but large and complex projects will have a high failure rate.
1	Fit-Gap: Salesforce is not a good choice. Other tools/approaches should be considered instead. Implementation: The implementation suffers from major structural issues. These core problems mean that the most effective path forward is to salvage what may be useful and start over. System Integrator: The SI team lacks strong processes and is deficient in several important system integration skills. Project failure is likely, and even smaller projects will often suffer in areas including user satisfaction, maintainability, and cost/schedule control.

The detailed fit and gap in Salesforce functionality (including 3rd party components) is included in Table 9, found in Appendix C, where the degree of fit for each functional area is assessed from Very Low to Very High. This assessment included both Salesforce plus the 3rd party architectural components identified in Figure 4 and Figure 5. The areas assessed as Very Low or Low would be clear gaps. Our assessment of the CARS requirements to ensure that they were correct, complete, and consistent was accomplished using a combination of the interviews and review of documentation in general, and in particular the requirement matrices, business rule matrices, the use-case scenarios, and the user stories.

Implementing CARS with Salesforce would involve relegating Salesforce to the role of a form engine working as the presentation layer for the external filer interface and using an alternate approach for the middle-tier (business rules, workflows), interfaces, and persistence layer (database). While this approach would work, the Salesforce platform would offer limited value within the overall architecture.

As shown in Table 3 and graphically in Figure 6, Salesforce does not appear to be the ideal choice for CARS, and the existing implementation suffers from major structural issues. The primary reason that Salesforce is not a good fit for CARS is the complexity of the highly specialized business processes and workflows, and it is the major contributor to the CARS Salesforce fit-gap score of 1.42. The primary driver of the low implementation quality score is a failure of the system implementation to meet those complex requirements. Some specific and significant issues include:

- CARS was architected without an external business rules engine, and instead the implementation team added the business rules as custom Apex code. This resulted in a large amount of custom code (over 2.6 million characters) that will be very difficult to test and maintain.

- The data portal and external interface are both tightly coupled to Salesforce, meaning that a change in Salesforce will require a simultaneous change in these elements. This will present challenges during on-going system support.

- Salesforce limitations in areas including field versioning, maximum number of records added during load operations, PDF page limitations, and characters of custom code will require additional customization and work-arounds.

- In addition, the custom code that was developed has a large number (over 11,000) of defects and security flaws.

For CARS to work correctly with Salesforce, an external environment (outside of Salesforce) would need to be added to support the business rules, workflows, portals, and interfaces. This external environment would then interact with Salesforce, which would be used for the filer portal. To implement this, the technical team would then need to redesign the database; remove all or most of the Apex custom code; and rebuild the API/External Interface. Portions of the existing community portal and filer portal might then be reusable. We believe that this approach is likely to be less cost effective than a complete redesign, but that belief could be confirmed using an RFI process.

It is our assessment that CARS is in a salvage and start-over scenario and given the poor fit between Salesforce and the CARS application, we recommend that alternate solution approaches be explored as part of the project restart.

Table 3: CARS Overall Assessment

	Score (1 to 5)
Salesforce Fit-Gap	1.42
CARS Implementation Quality	2.13

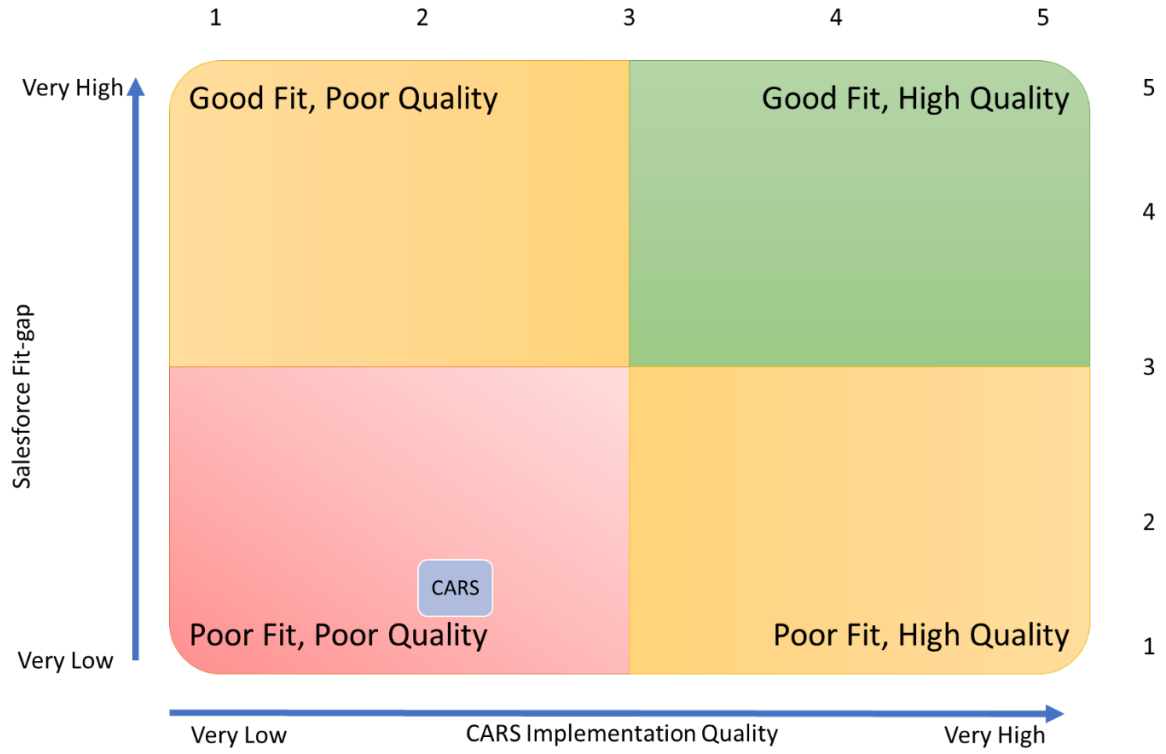


Figure 6: CARS Fit-Gap Assessment Quadrant

Table 4 shows the CARS assessment detailed scores. The detailed Salesforce fit-gap scoring categories are as follows:

- Application: The fit between the business application functional requirements and the Salesforce capabilities. CARS scores as a Very Low fit for Salesforce, primarily because of the complexity of the business rules and workflows.

- Data: The fit between the data structures and storage requirements and the Salesforce approach to data storage. CARS scores as a Low fit for Salesforce, primarily because the vast majority of data structures required for CARS are unique, and the requirements for data versioning and flexible data exception handling are extensive in CARS.

- Lifecycle: Total cost of ownership value including licensing costs, maintainability, and staffing. CARS scores a Very Low fit for Salesforce because the out of the box (OOB) Salesforce solution offers only a very small subset of the required CARS functionality, so the cost to license and support Salesforce is not offset by a corresponding amount of delivered OOB functionality.

And the detailed CARS implementation quality assessment factors are:

- The quality of the design, including the architecture, data, and class design. The architecture and design scores a Low quality score, primarily because of the attempt to implement the business rules and workflows in custom Apex code.

- The quality of the implementation work, including programming and system configuration. The implementation scores a Very Low score, primarily because of the number of defects, the overall code structure, and the poor fit between the code design and the business requirements.

- The maintainability of the system as built. Maintainability scores an Average. The code was mostly developed following standard approaches to error handling, logging, control executive structure, and so on.

Table 4: CARS Assessment Detailed Scores

Detailed Score	
Salesforce Fit-Gap	**Score (1 to 5)**
Application	1.36
Data	1.75
Lifecycle	1.00
CARS Implementation Quality	
Design	2.17
Implementation	1.00
Maintainability	2.55

Our full analysis may be found in Appendix C.

3.1.4 Conclusions and Recommendations.

Information Technology (IT) projects can be thought of as having four layers of functionality. At the lowest level we have the virtual machine, which is the computer hardware (potentially in the cloud), the database management system, and supporting software architecture. So, for example, this is the layer where we would decide how the business rule engine would work, how the workflow engine would work, how security access control and monitoring will work, and so on. Most of this layer involves purchasing and configuring items, rather than building them, although some components might need to be built if the organization has unique requirements.

The second level up is the persistence layer, which simply means the place where data is stored. The persistence layer is where the database design comes into play, which

then also drives the design of the data objects that will be worked with by the higher layers (e.g., interfaces).

The third level up is the middleware layer. This is the layer where the actual business rules, data validation, workflow configuration, and so on resides. To a large degree, the things that make a business unique are captured in here. If the virtual machine layer was architected correctly, then most of this work involves configuring components with the organization specific data, rather than actual programming. When people talk about business process reengineering, or process optimization, they are mostly talking about changes at this level.

The fourth, and final level, is the presentation layer, or User Interface (UI). This is the computer screens, the reports, the dashboards, and so on.

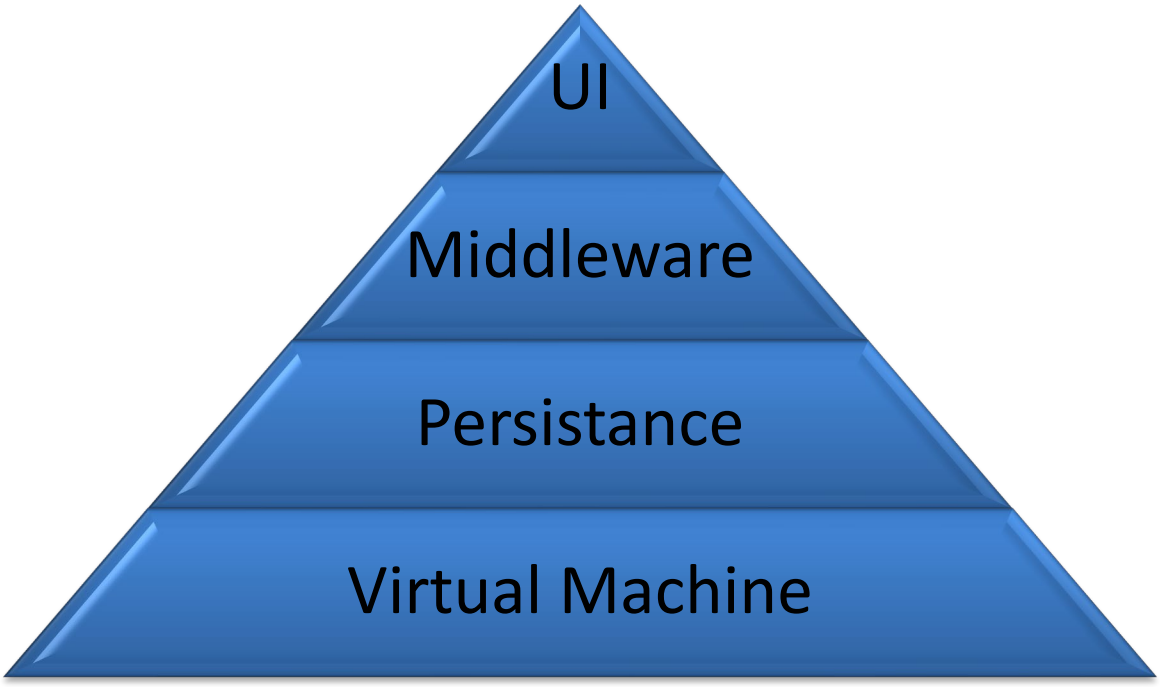


Figure 7: Information Technology Stack

Each layer of the technology stack is dependent on all the layers below it. So, problems at the user presentation layer are easy to fix if the layers below are correct. Problems in the middleware layer will require reengineering that layer, but also require changing the presentation layer. This applies all the way down.

In the case of CARS, what we've found are serious problems with those lower layers, meaning that fixing the problems cannot be accomplished using a "stay the course" approach, but will require a "salvage and start over" approach.

The current CARS implementation is seriously flawed across all dimensions. There is no rules engine or workflow engine. There is no viable approach to data versioning and error management at the level necessary. All system components are tightly coupled, meaning that changes in one area will have a ripple effect on other areas. The underlying architecture does not meet and cannot meet the system requirements and correcting these issues will require major work for virtually all system components. The implementation will be difficult to test, is likely to suffer from on-going reliability issues, and will be a major challenge to maintain going forward. The most cost-effective strategy going forward involves reviewing the project artifacts to identify those that are useful, salvaging those components, and then starting over. In many cases, the most useful artifacts will be from the work done during the early project evolution (in the 2018 timeframe).

With starting over, there is a serious question as to whether or not Salesforce is the right platform for this application. While it would be possible to use Salesforce for some portion of the required functionality, the majority of the application functionality will need to be outside of Salesforce. So, there is the follow-on question of whether the functionality that Salesforce would provide is worth the on-going cost of the licenses, the support costs associated with supporting both Salesforce and another solution, and the cost to implement the interface between Salesforce and the remaining system components. Overall, our assessment is that the most cost effective, and lowest risk, approach would be to build CARS without using Salesforce.

3.2 System Integrator Initial Assessment.

Systems integration (SI) has “two faces” [3]: (1) The internal activities executed to develop and integrate the needs, desires, and requirements into well-defined, effective new products; and (2) the external activities required to integrate components, skills, and knowledge from other organizations into complex products and services. External organizations include suppliers, users, government agencies, regulators, production partners and, sometimes, competitors (preferably working collaborative toward the same outcomes).

As technology continues to advance, platform solutions like Salesforce continue to increase in both functionality and complexity. As states seek to leverage technology platforms to support systemic modernization, the role of system integration is critical for success. The system integration function is tasked with working in conjunction with each component and vendor, connecting disparate information and technology and ultimately responsible to the business and stakeholders to which it is to serve.

[3] https://www.researchgate.net/publication/5212545_Systems_Integration_A_Core_Capability_of_the_Modern_Corporation

Furthermore, benchmark data shows that successful projects of this size and technical complexity rely on a strong system integration capability to achieve vendor accountability and avoid responsibility obscurity. Success is achievable in proportion to the SI's experience and competency in understanding and meeting stakeholder expectations, balancing budget, and resources, and delivering a quality solution.

The sections below provide the identified findings of the assessment (Section 3.2.1), the identified activity to address the findings (Section 3.2.2), the fit-gap scores (Section 3.2.3), and the conclusions and recommendations for system integration (Section 3.2.4).

3.2.1 Organization Summary.

The CARS project has been significantly challenged along its journey in a multitude of ways. Many critical attributes of foundational components from strategy to system development and integration capabilities are either severely constrained or missing altogether. These findings are described within each key area of system integration below. As the project pivoted to OSaaS to develop and implement a technology solution for SOS to replace CAL-ACCESS, it did not have, nor did it develop, the strategy and corresponding system integration supports and controls necessary to succeed. The necessary activities to address these are discussed in Section 3.2.2.

We attempted to find OSaaS reference information from the CARS OSaaS acquisition, but we discovered that no reference checks were conducted, and no description of previous corporate experience was requested as part of the procurement. We asked OSaaS to provide us with references to other projects like CARS, but as of the time of writing this report those have not yet been provided.

The assessment has identified findings in the key areas of the system integration function. The detailed eMRI – Systems Integration model and scores are in Appendix C.

Transformation Governance:

Transformation governance is the exercise of authority; direction; control; management over how the organization executes change. This function defines and governs strategy, principles, architecture, and performance measurement across the complex change portfolio. A transformation in this context is defined as a change that is of a critical size and scope with respect to the fundamental operations of the organization. Many organizations struggle to see the difference between how to support a transactional change versus how to architect and govern a transformational change. When people and organizations focus on the governance of their transformation efforts, quality is proven to rise.

Modernizing CAL-ACCESS is a transformational project. Currently, the concept of transformation governance does not exist within the organization or OSaaS, acting as the SI. The CARS project has been operating without formalized vision, strategy, success metrics, decision matrices, or a project charter. The CARS project must recast its vision and ensure the underlying strategy and success metrics are aligned, measurable, and manageable.

Architecture Governance and Development:

Architecture governance is the exercise of authority; direction; control; management over how the organization architects the enterprise, including transformations. This function defines and governs architectural strategy and principles in alignment with the enterprise strategy. Architecture development generates blueprints and roadmaps for the complex change portfolio. Many organizations struggle to see the difference between architecting the portfolio and executing the portfolio. When people and organizations focus on the quality of their architecture efforts, solution quality is proven to rise, and costs are predictable.

Architecture is not a formalized practice within the organization and has not been formally governed. While some solution architecture has been performed within the CARS project, disciplined architecture development is not evident. Business architecture definitions do not exist currently. Alignment to the solution from a business process perspective has not occurred.

Requirements Development and Management.

Requirements development and management is a key function within systems engineering to ensure that the organization validates and meets the needs of its customers and stakeholders (internal and external). Requirements define the scope in any project in testable detail and maintain traceability through the software development life cycle (SDLC) with other key artifacts (e.g., elaborations, use cases, architectural diagrams, designs, test cases).

CARS project requirements and business rules from 2018 exist but the project has not been disciplined with the traceability and precision of corresponding elaborations and artifacts.

Solution Design.

The purpose of solution design is to provide sufficient detailed data and information about the solution and its elements to enable the development and implementation consistent with the designed architecture and solution requirements. Impact and alternative analysis are critical activities during solution design. Additional key factors must be strategically defined and incorporated into solution design including user experience, support, maintainability, reliability, scalability, time, cost, and efficiency.

The CARS project does not have a clear, holistic solution design. What can be inferred from solution design artifacts is incomplete and lacks design rationale that considers the impact to the Current-State including stakeholders, business processes, and system interfaces.

Solution Development:

Solution Development translates the solution designs, detailed requirements, and business outcomes into a group of interacting, interrelated, and interdependent elements (methods, artifacts, people, technologies). Activities include planning, creating, testing, and deploying the interconnected components of the designed solution. While there are many viable tools, methods, and techniques to support solution development, their success is directly dependent on the other areas within this section (e.g., architecture, requirements, design).

Solution development for the CARS project has been significantly constrained by the lack of architecture, design, governance, and decision-making framework.

Configuration Management:

Configuration management is a system engineering process that tracks, manages, and monitors a solutions configuration capabilities and metadata. Configuration management helps engineering teams define and develop stable solutions using methods and tools that identify, manage, and monitor updates to configuration data. Complex solutions are composed of components that differ in granularity of size and complexity. Configuration components enable the concept of separating the metadata from the code (versus customization).

The core solution platform of the project (Salesforce) is inherently configuration centric. However, the project has not seemed to operate using formalized configuration management protocols, tools, or methods.

Interface Control and Management:

Interface control and management includes defining, designing, and controlling the mechanisms associated with the interaction between different devices, entities, environments, and systems. Interfaces, both external and internal, must be managed and controlled to ensure sustained compatibility and consistency, both among themselves and with the solution.

Interfaces have been primarily developed without adequate consideration for the high-volume of existing data feeds in the Current-State ecosystem.

Enterprise Integrations:

Enterprise integration is the use of multiple integration approaches including technology platform services, API management, application integration, and messaging to leverage enterprise services and assets. This enables organizations to seamlessly integrate, unify and standardize core business capabilities across diverse solution environments.

Strategies for enterprise, business to business (B2B), and legacy integration were not available. These integrations have been addressed ad-hoc.

Program Management:

Program Management is an organizational function that oversees a group of individual projects linked together through a shared organizational strategy and/or common area of impact. This programmatic grouping of multiple projects, or portfolio, provides synergy, consistent management, and greater visibility to stakeholders than individually managed projects.

All key functions related to program management are missing key attributes (e.g., capacity, scope, risk, controls) to implement a multi-stakeholder enterprise system. These key functions include project management, solution testing, quality assurance, and implementation management.

Operations and Performance Management:

Operations Performance Management (OPM) improves the responsiveness, throughput, quality, cost, and efficiency of production solutions. OPM typically includes process optimization, operations intelligence, and forecasting, and often involves technologies such as modeling, process data collection, visualization, and analytics. OPM can interoperate with other performance management capabilities such as Asset Performance Management (APM) systems that focus on improving the reliability and availability of physical assets while minimizing risk and operating costs.

The organization's ability to operate, maintain, and manage performance is missing key attributes (e.g., capacity, scope, risk, controls) to implement a multi-stakeholder enterprise system. At the time of the project pause, the ITD was not equipped in this regard.

3.2.2 CARS System Integrator Requirements.

The CARS project must clearly define and assign the SI role to a group with the competency and capacity (e.g., roles, responsibilities, skills, and authority) to support the definition and execution of the new CARS project strategy. The CARS ecosystem is complex, and the SI role must be organized and orchestrated effectively to achieve project success and solution quality.

The assessment has identified necessary activity in these key areas within the system integration function for the project to be successful.

Transformation Governance: SOS must recast and formalize the CARS vision, strategy, success metrics, decision matrices, and project charter. The concept of transformation governance should be implemented to execute the strategy. This will include the roles and responsibilities within key functions such as system integration.

Program Management: All key functions related to program management including project management, solution testing, quality assurance, and implementation management must be updated based on the new CARS strategy.

Architecture Governance and Development: Architecture must be a formalized practice with formal governance, using modern architecture and engineering discipline that will produce defensible value. This will be foundational in achieving predictable outcomes with respect to success metrics (e.g., stakeholder expectations, time, budget).

Requirements Development and Management. Existing project requirements, business rules, and corresponding elaborations and artifacts (e.g., user stories) should be mined for relevance and traceability to be carried forward with precision.

Solution Practice: The solution practice must be formalized to support the key functions to design, develop and deliver the CARS solution in adherence with the new CARS strategy. The practice will achieve a host of foundational solution assets including a clear, holistic view of the solution design. The solution design, as with other key living artifacts managed by the practice, must be managed in alignment with design principles and decision-making framework. Impacts to the key aspects of the Current-

State (e.g., stakeholders, business processes, system interfaces) must be managed in alignment with the new CARS strategy and practice guidelines.

Operations and Performance Management: All key functions necessary to operate, maintain, and manage performance must be updated based on the new CARS strategy.

3.2.3 System Integrator Fit-Gap.

The existing organizations supporting the CARS project in the SI role currently, may or may not be viable to support the new CARS project strategy. This must be addressed in the context of the new CARS project strategy development as gaps are addressed and targeted improvements are defined. Collectively, the SI function for this project scores 1.27 on the scale of 1-5.

Table 5: CARS System Integrator Assessment

eMRI - System Integrator	Score (1 to 5)
System Integrator Fit-Gap	1.27

As discussed previously, each functional area within the SI assessment is critically low. These foundational components from strategy to system development and integration capabilities are all severely constrained. The table below provides the scores for each area.

Table 6: CARS System Integrator Assessment - detail

eMRI - System Integrator	Detailed Score
System Integrator Fit-Gap Detail	Score (1 to 5)
Transformation Governance	1
Architecture Governance	1
Architecture Development	1
Requirements Development	2
Business Process Integration (BPI)	1
Solution Design	2
Project Management	2
Requirements Management	2
Solution Development	1
Implementation Management	1
Configuration Management	1
Interface Control and Management	1
Legacy Integration	1
Business to Business (B2B) Integration	1

Enterprise Solution Integration	1
Solution Testing and QA	1
Operations (Performance Management)	1
Integrated Program Management	1

3.2.4 Conclusions and Recommendations.

The CARS ecosystem is complex, and the SI role must be organized and orchestrated effectively to achieve project success and solution quality. This includes:

- The SOS must recast and formalize the new CARS vision, strategy, success metrics, decision matrices, and project charter.
- Within the new strategy, the CARS project must clearly define and assign the SI role to a group with the competency and capacity (e.g., roles, responsibilities, skills, and authority) to support the continued definition and execution of the new CARS project strategy.
- The SOS and selected SI vendor must execute on the recommendations in Section 3.2.2.

4 CARS High-Level Findings

As part of our 60-day assessment, due in draft form on 12/1/2021, we will be providing a detailed and supportable assessment of CARS across thirteen technical dimensions. As part of this initial quick-look assessment, we are providing some high-level and preliminary observations in those same dimensions. These observations are based on a combination of analysis of project artifacts (documents and development environments), stakeholder interviews, and an analysis of the current Salesforce code in the DevOps system.

4.1 CARS technical implementation.

As discussed in Section 3.1.3, the current CARS technical implementation is seriously flawed in terms of the strategy, design, implementation, and supportability. Appendix C provides the specific factors that we reviewed in drawing this conclusion. Significantly, the flaws that we identified are foundational in nature, meaning that correcting them will require modifying all, or almost all, of the code that was built on top of that shaky foundation. Trying to correct the problems without re-architecting from the bottom up will result in a system that is neither reliable nor maintainable, and it will require unnecessary compromises in delivering core business functionality.

4.2 Data conversion and migration.

According to Interviews and data conversion related status documentation, data is currently being converted directly from the legacy system into the CARS Salesforce data objects, with clean-up occurring in the form of transformations during the Extract-Transform-Load (ETL) process. Because the current data structures will need to be re-architected, there is very little value in the Salesforce stored data, however business logic and code needed to clean the legacy data will be useful to the project going forward under any approach, because that legacy code will need to be cleaned under all scenarios. There is potential value of the current (OSaaS) data conversion work based on the business logic used for the transport and data cleanup.

Additionally, we did find that there was significant data conversion and migration work performed by the CARS project back in roughly 2018, and in reviewing that work, it appears that there may be useful design work, and potentially useful converted data.

4.3 Legacy data migration challenges.

This area will be addressed in the next report.

4.4 Contract management and vendor negotiations.

A staff augmentation, time-and-material approach to building a complex information technology application like CARS pushes all the responsibility and risk for the project to the State. The vendor's contractual obligation is limited to delivering the requested number of labor hours, with no contractual guarantee of a usable product in the end. The preferred approach is to use a deliverable based approach where the vendor assumes the majority of responsibility and risk for a successful project implementation, at least for the technical implementation work. The Work Order Authorization (WOA) process does not appear to be the optimum approach for this project. We will amplify on this as part of our next deliverable.

4.5 Requirement definition and management.

The current CARS implementation effort defined requirements in the form of user stories, and these stories primarily revolve around the FPPC forms.

The previous CARS implementation effort (Perspecta) defined business requirements, business rules, workflows, and use-case scenarios, all of which primarily revolved around the legislative requirements and the underlying business process. This earlier approach was more appropriate for the CARS application.

The work and knowledge processed over the course of time by both vendors should be mined for value as a suitable starting point for the effort going forward. It should be updated as needed and validated.

4.6 Project schedule management.

As a very rough rule of thumb, for a given technology project schedule using traditional (waterfall) or hybrid development 1/3 of the schedule should be spent on architecture and design, another 1/3 on implementation, and the final 1/3 on testing. Pure Agile will reduce the initial architecture/design stage and the final testing stages somewhat, but they will still each represent roughly 20% of the total schedule. Skipping the initial architecture and design work often results in a system that is poorly architected and difficult or impossible to test and maintain. Skipping the final test phase will result in a system that is unreliable. In the CARS implementation, the team attempted to use a pure Agile approach where virtually the entire schedule was spent on implementation, with the predicted outcomes.

4.7 Communication management.

During the interviews, we consistently found that the implementation vendor and most of the technical team (both State and vendor) felt like there was effective communication.

But we consistently found that the internal and external business users consistently felt that there was very poor communication. Part of this was a mismatch in understanding of the project objectives, resulting in mutual frustration. Part of this was extensive direct communication between technical and business users without a skilled facilitator (e.g., a business analyst) in between, resulting in a situation where technical users did not fully understand the business requirements, and business users did not understand the technology or the implications of technology choices. And part of this was due to undue schedule pressures causing the implementation to shut down “unnecessary” communications to try to meet deadlines.

4.8 Governance and sponsorship.

It’s tempting to say that the project lacked governance and sponsorship. But the fact is that the project did have a strong project sponsor, the sponsor did correctly fulfill her role by insisting that the system meet the business needs of the organization, and the project did have a Project Charter (initially in an approved version, and for the restart in a draft version). However, the project sponsor’s advocating for the required business functionality and quality was overridden by other organizational units that prioritized schedule. This disconnect was resolved at the time in favor of the schedule constraint rather than quality and functionality.

4.9 Organizational change management.

Overall, Organizational Change Management (OCM) was actively and effectively involved in both efforts. We were told that there is an organization-imposed limit on the number of OCM hours that can be awarded to a single contractor (approximately 1,500), so for a multi-year project this key role will turnover approximately once per year, resulting in learning curve related inefficiencies. To eliminate these learning curve related inefficiencies, the SOS could remove this limitation, make this a government role, or incorporate this role into a longer duration contract (e.g., the system integration contract) so that there will be consistency for the project duration. These suggested OCM resourcing best practices are consistent with benchmark successful projects.

4.10 Quality management.

The project suffered from both product and process quality problems, as evidenced by the discussion in Chapter 3. The Independent Verification and Validation (IV&V) contractor identified many of these issues, but their reports seemed to have been ignored.

4.11 Risk management.

Our primary quick-look observation in this area, based on the interviews and some emails provided to us, is that the risk narrative was controlled, resulting in the most critical, strategic risks to the project being hidden. Specifically, business and external stakeholders repeatedly raised risks related to the project goals, objectives, quality, and business functionality. Those concerns, which were central to the ultimate project failure, never made it to the risk management process at all. We were told that they were not included because they were submitted via email rather than on the proper form, or because they were not expressed clearly enough, or because the people raising these risks were just “whining.” Ultimately, the risks that mattered most were ignored.

4.12 Release Management.

This area will be addressed in the next report.

4.13 Testing.

Based on the interviews and the test related documentation reviewed, we found that testing was flawed in almost every way that it could be flawed[4]. There were no consistently agreed to goals and objectives that the system could be tested against. There was no realistic test data set or correct test scripts. Unit testing was inadequate. System Integration Testing (SIT) to verify proper system operation prior to User Acceptance Testing (UAT) was either skipped or so inadequate that the effect was the same. There was little or no regression testing. ADA testing was an afterthought, and there was no attempt to resolve ADA issues. No Test Readiness Review (TRR) milestone was conducted. User acceptance testing (internal and external) did not have sufficient time, clearly defined roles, or objectives. For example, we were told that external testing was scheduled for two-weeks, and that the first scheduled week was the busiest filing week of the year, so no external testers were available. Then during the remaining week, external testers told us that it took them three days to receive the credentials needed to be able to login, and that during the remaining two-day testing window the system kept locking up, requiring them to call the developers to have them manually clear errors. Ultimately, the people best able to test the system simply gave up.

[4] See for reference to testing best practices, ISO/IEC/IEEE 29119, ISO/IEC 9126, IEEE 829, and IEEE 12207.

A. Acronyms

ADA: Americans with Disabilities Act.

API: Application Programming Interface.

B2B: Business to Business.

BIT: Built In Test.

BPI: Business Process Integration.

CAL-ACCESS: California Automated Lobbyist and Campaign Contribution and Expenditure Search System.

CARS: Cal-Access Replacement System.

CM: Configuration Management.

CMDB: Configuration Management Database.

CRM: Customer Relationship Management.

eMRI: enterprise Maturity Readiness Index.

ETL: Extract-Transform-Load.

FPE: Function Point Equivalents.

FPPC: Fair Political Practices Commission.

FTB: Franchise Tax Board.

GAAP: Generally Accepted Accounting Procedures.

HLO: High-Level Object.

IT: Information Technology.

IV&V: Independent Verification and Validation.

KPA: Key Process Areas.

LCAP: Low-Code Application Platform.

M&O: Maintenance and Operations.

MVP: Minimum Viable Product.

OCM: Organizational Change Management.

ODC: Other Direct Charge.

OOD: Object Oriented Design.

OOTB: Out of the Box.

OSaaS: Outreach Solutions as a Service.

PM: Project Management.

PMD: Programming Mistake Detector.

PRA: Political Reform Act.

PRD: Political Reform Division.

QA: Quality Assurance.

SD: Solution Design.

SDLC: Software Development Lifecycle.

SI: System Integration

SIT: System Integration Testing.

SOS: Secretary of State.

TRR: Test Readiness Review.

UAT: User Acceptance Testing.

UX: User Experience.

WOA: Work Order Authorization.

XML: Extensible Markup Language.

B. Interviews

Table 7 contains a list of stakeholders interviewed during our assessment thus far, along with the date and time of the interview. The interviews represent internal and external stakeholders who participated in the CARS project since 2016 and leading up to the pause in June of 2021. The purpose of the interviews was to understand multiple perspectives of the CARS scope of work, current status, and areas for potential improvement going forward.

Table 7: Stakeholder Interview List

Stakeholder	Organization	Topic/Description/Role	Date	Time
Pam Parra and Dawn Hadid	External-FTB	Use of data for audits	9/23/2021	10:30 AM
David Montgomery	External-Netfile	Largest software vendor	9/17/2021	11:30 AM
Jen Broadbent and Ben Katz	External-SVS	Software Vendor Subgroup	9/23/2021	12:30 PM
Taylor Kayatta	Legal	Legal Perspective	9/9/2021	4:00 PM
Janet Fong	SOS-CDT[5]	General Observations	9/23/2021	11:30 AM
Lisa Martin and Reggie Fair	SOS-Exec	General Observations	9/24/2021	9:30 AM
Madame Secretary	SOS-Exec	Exec Perspective	9/30/2021	10:30 AM
Michael Carter	SOS-Exec	Executive Perspective	9/15/2021	11:00 AM
Tristian Cormier	SOS-Exec	CTO	9/23/2021	10:00 AM
Gurnam Basra	SOS-ITD	ITD Lead	9/20/2021	9:00 AM
Joe White	SOS-ITD	CIO and ITD Chief	9/23/2021	2:30 PM
Krishna Dhulipala	SOS-ITD	ITD supervisor	9/21/2021	11:30 AM
Dana Furby	SOS-Legal	Legal-Acquisition/Contract	9/20/2021	3:00 PM
Cruz Nieto	SOS-PMO	PMO Director	9/17/2021	10:30 AM
Han Ha	SOS-PMO	Project Manager-Budget/Cost	9/17/2021	3:00 PM
John Bryce	SOS-PMO	Contract Manager	9/23/2021	1:30 PM
Preeti Narang	SOS-PMO	Risks and Issues	9/20/2021	11:00 AM
Kathryn Whelan, Kira Rasmussen, and Sean Jensen	SOS-PRD	Staff Services Mgr.	9/16/2021	10:30 AM
Lacey Keyes	SOS-PRD	Software Input Group (ind. filer)	9/20/2021	12:00
Lorna Semana	SOS-PRD	IT Lead	9/9/2021	3:00 PM
Samantha Brown	SOS-PRD	Training/Outreach	9/16/2021	4:00 PM

[5] Upon SOS' request during the CARS Project pause, on 6/28/21 CDT was asked to help procure an independent advisor to assess the health of the CARS Project and to oversee the vendor engagement. Throughout the engagement, CDT remains a neutral independent entity and has no bearing or influence on the outcome of the assessment findings.

Julie Waddell	SOS-Sponsor	Prev. Project Sponsor	9/22/2021	9:00 AM
Margie Hieter	SOS-Sponsor	PRD Perspective	9/21/2021	9:00 AM
Margie Hieter	SOS-Sponsor	PRD Perspective	9/22/2021	10:00 AM
Project Approach	SOS-Sponsor	Assessment Status	9/15/2021	8:30 AM
Hyla Wagner and Lynda Cassady	Vendor-Bus Rules	Bus. Rules and Reqs	9/20/2021	3:30 PM
George Conley	Vendor-IV&V	IV&V Observations	9/9/2021	10:00 AM
George Conley	Vendor-IV&V	IV&V Follow-up	9/21/2021	10:30 AM
Joan Rene	Vendor-OCM	OCM	9/16/2021	1:30 PM
Anthony Montero	Vendor-OSaaS	Integration Lead	9/15/2021	3:00 PM
Carlo Grifone	Vendor-OSaaS	Demo + Client Success Partner	9/23/2021	4:00 PM
Courtney Montero	Vendor-OSaaS	PMO Lead	9/22/2021	3:00 PM
Curt Cadwallader	Vendor-OSaaS	Conversion/Migration Lead	9/13/2021	10:30 AM
Geo Shannon	Vendor-OSaaS	Data Portal Lead	9/13/2021	9:00 AM
Matthew Grifone	Vendor-OSaaS	Test/Trace Lead	9/14/2021	3:30 PM
Rahal Rathore	Vendor-OSaaS	Salesforce Lead	9/16/2021	9:00 AM
Venkata Kasturi	Vendor-OSaaS	API Lead	9/21/2021	3:00 PM
Keven Star	Vendor-PMO	Solution Implementation Manager	9/17/2021	9:00 AM
Carlos Armenta	Vendor-Testing	Testing	9/14/2021	2:00 PM

C. Detailed Fit-Gap and Implementation Assessment

This Appendix includes the detailed fit-gap and implementation assessment models, input parameters, and justifications. We begin with the Salesforce Fit-gap, then address the implementation quality assessment, and then the system integration assessment. For each of these three eMRI models, we begin with the model itself, after which we provide the input parameters along with the supporting justification for each input setting.

Table 8 contains the eMRI model used to evaluate the Salesforce fit-gap. The model consists of categories, which are simply broad areas of fit analysis, and sub-categories within each category. For each sub-category, the application will be scored somewhere between Very Low (poor fit) and Very High (exceptional fit). The specific criteria used when assigning this score are also shown in the table. Finally, different sub-categories may have more or less importance to the final (total) fit-gap score. This subjective setting, shown in the Importance column, ranges from Very Low (not important) to Very High (exceptionally important).

Table 8: Salesforce Fit-Gap eMRI Model

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
Application	Customization	2.5M or more characters of custom Apex code required.	1 M characters of custom Apex code required.	500K characters of custom Apex code required.	250K of custom Apex code required.	100K or less of custom Apex code required.	Very High
Application	Data Warehouse	Extensive near-real-time data warehouse and reporting required.	Extensive data warehouse with nightly updates.	Some data warehouse requirements with nightly updates.	Limited external reporting required.	No data warehouse.	Low
Application	Functional Requirements	Many business unique functional requirements.	Many externalized or custom functions with some case management or CRM functionality.	Primarily case management or CRM but requires largely custom configuration.	Case management and case support with appropriate tailored configuration available.	Sales, Marketing, post-sale support (CRM).	Average

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
Application	Process Improvement	Salesforce does not provide processes with equivalent business value without significant customization.	Salesforce OOTB processes need to be modified to provide equivalent business value to existing business processes.	Salesforce OOTB processes are equivalent in business value to existing business processes.	Salesforce OOTB processes somewhat improve existing business processes and offer business value.	Salesforce OOTB processes significantly improve existing business processes and offer business value.	Average
Application	System Interaction	Mostly electronic interfaces, batches.	Much of the system functionality is through interfaces and batches.	Several electronic interfaces or batches.	Few electronic interfaces or batches.	All user screens and reports.	Low
Data	Data structures	Existing Salesforce data structures meet <= 60% of the requirements.	Existing salesforce data structures meet 70% of the requirements.	Existing Salesforce data structures meet 80% of the requirements.	Existing Salesforce data structures meet 90% of the requirements.	Existing Salesforce data structures meet 95% of the requirements.	Average
Data	Data Versioning	Versioning for >60 fields per object.	Versioning for >30 fields per object.	Versioning for 20 fields per object.	Versioning for 10 fields per object.	Versioning for 5 or fewer fields per object.	Average
Data	Online Data Storage	5 million records.	10 MBytes per user.	5 MBytes per user.	3 MBytes per user.	1 million records.	Average
Data	Online File Storage	1.5 GBytes per user.	1 GByte per user.	500 MBytes per user.	250 MBytes per user.	100 MBytes per user.	Average
Lifecycle	Licensing	Salesforce license costs are significantly higher than the offsetting reduction in M&O Costs.	Salesforce license costs are somewhat higher than the offsetting reduction in M&O Costs.	Salesforce license costs are roughly equal to the offsetting reduction in M&O Costs.	Salesforce license costs are lower than the offsetting reduction in M&O Costs.	Salesforce licensing and support covers most M&O activities.	Average
Lifecycle	Maintainability	Design requires many tools and significant external functionality.	System requires some to several tools and	System requires few tools but requires some functionality	System requires some to several tools, but all	System requires few tools, and all functionality is met by Salesforce.	Very Low

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
			significant external functionality.	external to Salesforce.	functionality is met by Salesforce.		
Lifecycle	Staffing	System requires staff with Salesforce configuration, Apex programming, and skills with 2 or more external development environments.	System requires staff with Salesforce configuration, Apex programming, and 1 or more external development environments.	System requires staff with Salesforce configuration and either Apex programming or staff with 1 or more external development environments.	System requires staff with Salesforce configuration skills.	System can be supported by staff without any Salesforce configuration or programming technical skills.	Average

Table 9 shows the assigned CARS fit-gap score for each of the eMRI Saleforce fit-gap sub-categories, along with the justification for each setting. Justifications were based on a combination of manual analysis of the actual implementation of Salesforce and supporting tools within Azure DevOps, automated analysis of those components, review of documentation from SharePoint, and interviews. We have repeated the importance column from Table 8 for convenience. In reading the CARS Score column:

- A score of Very Low or Low would be considered a gap.
- A score of Very High or High would be considered a fit.
- A score of Average would be neutral, neither a strong fit nor a significant gap.

Table 9: Salesforce Fit-Gap Model Input Settings

Category	Sub-Category	CARS Score	Justification	Importance
Application	Customization	Very Low	The choice to go with internal business rules vs utilizing an externally based rules engine has helped lead to custom Apex code sprawl with over 2.6 million allowable characters in Apex code is unsupportable in our estimation. If a true BRMS solution was procured and implemented, the amount of custom Apex code would go down significantly.	Very High

Category	Sub-Category	CARS Score	Justification	Importance
Application	Data Warehouse	Average	The Public Reporting portal is a work in progress with a Heroku based Postgres database backend and JavaScript based front-end mixed in with Java for querying the database. It is a less than desirable solution. Instead of reinventing the wheel with a custom from the ground up reporting solution, we recommend that a commercial or open-source reporting solution be used instead (e.g., Tableau, Tableau CRM, Splunk or ElasticSearch).	Low
Application	Functional Requirements	Low	Case Management is an OOTB classic use case for a CRM solution. Clarification – It has since been learned that the CARS project does not and will not use Case Management functionality within Salesforce to fulfill the user requirements for the project.	Average
Application	Process Improvement	Low	Salesforce CRM and platform does provide high configurable without custom code in relation to process improvement and process automation. This is part of the LCAP (Low-Code Application Platform) family of solutions that Gartner provides analysis for. Salesforce is recognized as one of the leaders in this category of configurable process improvement and automation.	Average
Application	System Interaction	Low	There is one main Interface/service that is used for 70% to 80% of the filings in the CARS solution. It uses the Bulk API that is offered by Salesforce OOTB for ingesting large XML files.	Low
Data	Data structures	Very Low	Since this is a custom filing solution for Campaigns and Lobbyists with a Case Management element that is exposed via the Salesforce Communities solution (now called Experience Cloud), there are only a handle of OOTB objects for this use case. However, the LCAP element of Salesforce does allow for quick configuration of additional custom objects (think database tables) with clicks and not code.	Average
Data	Data Versioning	Very Low	This is a very normal use case to version and use configurable OOTB Field Audit History to track data history in multiple fields. The only caveat is that there is a limit to how many fields that can be "versioned" or tracked via Field Audit History (e.g., 20 per object).	Average
Data	Online Data Storage	Average	More online data storage is not considered a poor feature in Salesforce, but a good and OOTB standard feature that the CRM solution provides. The only drawback is that the online data storage can be expensive for larger requirements. I would recommend that we revisit this section.	Average

Category	Sub-Category	CARS Score	Justification	Importance
Data	Online File Storage	Average	More online file storage is not considered a poor feature in Salesforce, but a good and OOTB standard feature that the CRM solution provides. The only drawback is that the online file storage can be expensive for larger requirements. I would recommend that we revisit this section.	Average
Lifecycle	Licensing	Very Low	OOTB, Salesforce Licensing costs do offset most M&O activities. In the case of the CARS implementation, due to the amount of customization in addition to the licensing costs, this DOES NOT offset the M&O activities and costs. Please let me know if you need additional clarification.	Average
Lifecycle	Maintainability	Very Low	Business Rules, Business Rules, Business Rules. OOTB, Salesforce does not offer an enterprise and easily maintainable solution for custom business rules. Therefore, an external solution for implementing, updating, and maintaining the business rules is highly recommended here.	Very Low
Lifecycle	Staffing	Very Low	This is where the CARS solution becomes unwieldly and untenable in our estimation due to the internalizing of the business rules and 2.6 million allowable custom Apex characters (or 36% of the allowable custom code). This is going to require a high degree of expertise, tribal knowledge, and a larger staff to support this application going forward if implemented.	Average

Table 10 contains the eMRI model used to evaluate the CARS implementation quality. As with the Salesforce fit-gap model, this model consists of categories, which are simply broad areas of fit analysis, and sub-categories within each category. For each sub-category, the implementation quality will be scored somewhere between Very Low (poor quality) and Very High (exceptional quality). The specific criteria used when assigning this score are also shown in the table. Finally, different sub-categories may have more or less importance to the final (total) implementation quality score. This subjective setting, shown in the Importance column, ranges from Very Low (not important) to Very High (exceptionally important).

Table 10: eMRI Implementation Quality Assessment Model

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
Design	Apex class architecture	The class design does not follow	Many Apex classes are	Apex classes generally use	Apex classes mostly use	Apex classes correctly use	Average

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
		basic OOD principles.	poorly designed from the perspective of encapsulation, inheritance, coupling, or alignment with the business domain.	encapsulation and inheritance, are loosely coupled, and align with the business domain.	encapsulation and inheritance, are loosely coupled, and align with the business domain.	encapsulation and inheritance, are loosely coupled, and align with the business domain.	
Design	Architecture	The architecture fails in 2 or more dimensions (functionality, allocation, interfaces, maintainability). It does not address all functional requirements, does not properly allocate functionality to architectural components, does not have clear internal interfaces, or is not designed to be maintainable.	The architecture fails in 1 dimension (functionality, allocation, internal interfaces, maintainability.	Required functionality is achieved through well designed and maintainable external systems to house processing areas not well supported by salesforce. The allocation between Salesforce and external systems is appropriate, and the interfaces are clear.	Required functionality is achieved with some added third-party tools and customization.	Required functionality can be achieved using configuration of Salesforce tools.	Very High
Design	Database/Object design	The database design does not follow good	The database design does not follow good	The database design is 3rd normal form and	The database design is 3rd normal form and	The database design is 3rd normal form,	Very High

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
		database design principles and the necessary business data is not correctly mapped to the database (e.g., missing data, incorrect data).	database design principles, but the necessary data is included in the design.	mostly aligned to the business processes.	aligned with the business processes but does not make adequate or correct use of encapsulation.	aligned to the business processes, and uses encapsulation to group related data together.	
Design	Functionality-Business	The system fails to meet critical functional needs of the internal business stakeholders, and the needed improvements will require changes in the architecture or core data structures.	The system fails to meet critical functional needs of the internal business stakeholders, but the design will support continuous improvement.	The system meets the most important functional needs of the internal business stakeholders, and the design will support continuous improvement.	The system mostly meets the functional needs of the internal business stakeholders, and the design will support continuous improvement.	The system fully meets the functional needs of the internal business stakeholders.	High
Design	Functionality-Partner	The system fails to meet critical functional needs of partner (e.g., supplier) stakeholders, and the needed improvements will require changes in the architecture or	The system fails to meet critical functional needs of partner (e.g., supplier) stakeholders, but the design will support continuous improvement.	The system meets the most important functional needs of partner (e.g., supplier) stakeholders, and the design will support continuous improvement.	The system mostly meets the functional needs of partner (e.g., supplier) stakeholders, and the design will support continuous improvement.	The system fully meets the functional needs of partner (e.g., supplier) stakeholders.	Average

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
		core data structures.					
Design	Functionality-Public	The system fails to meet critical functional needs of the public/customer stakeholders, and the needed improvements will require changes in the architecture or core data structures.	The system fails to meet critical functional needs of the public/customer stakeholders, but the design will support continuous improvement.	The system meets the most important functional needs of the public/customer stakeholders, and the design will support continuous improvement.	The system mostly meets the functional needs of the public/customer stakeholders, and the design will support continuous improvement.	The system fully meets the functional needs of the public/customer stakeholders.	Average
Design	Functionality-Technical	The system fails to meet critical functional needs of the internal technical (IT) stakeholders (e.g., support personnel), and the needed improvements will require changes in the architecture or core data structures.	The system fails to meet critical functional needs of the internal technical (IT) stakeholders (e.g., support personnel), but the design will support continuous improvement.	The system meets the most important functional needs of the internal technical (IT) stakeholders (e.g., support personnel), and the design will support continuous improvement.	The system mostly meets the functional needs of the internal technical (IT) stakeholders (e.g., support personnel), and the design will support continuous improvement.	The system fully meets the functional needs of the internal technical (IT) stakeholders (e.g., support personnel).	Average
Design	Interfaces	External interfaces are flawed in two or	External interfaces are flawed in one or	External interfaces are clear for both	External interfaces are clear for both	External interfaces are clear for both	Average

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
		more dimensions. They are not well defined in terms of data and control information; they do not meet business functional requirements. They are not efficient from a performance perspective. They will break when the database is changed.	more dimensions. They are not well defined in terms of data and control information; they do not meet business functional requirements. They are not efficient from a performance perspective. They will break when the database is changed.	data and control information, meet business functional requirements, include clear diagnostic information to the sender for common errors, are efficient, are insulated from changes in the main system and data structures.	data and control information, meet business functional requirements, include clear diagnostic information to the sender for all errors, are efficient, are insulated from changes in the main system and data structures.	data and control information, meet business functional requirements, include clear diagnostic information to the sender for all errors, are efficient, are insulated from changes in the main system and data structures, and support backward compatibility.	
Design	Performance Engineering	The system has some areas of poor performance and those have moderate impact of internal and external stakeholders. Correcting those areas will require changes in the underlying architecture or	The system has some areas of poor performance and those have moderate impact of internal and external stakeholders. Correcting those areas will not require changes in the underlying architecture or	The system has some areas of poor performance but those have acceptable impact of internal and external stakeholders.	Performance bottlenecks are identified and understood. Performance considerations are incorporated in both data structures and algorithms.	Performance bottlenecks are identified and understood. Performance considerations are incorporated in both data structures and algorithms. Performance testing using realistic test data	Average

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
		core data structures.	core data structures.			sets has been conducted.	
Design	Security - Custom Programming	Software was not tested for security flaws, or security defects remain.	Software was automatically tested for security flaws and discovered problems were corrected.	Software was automatically (tool) and manually tested for security flaws and none were found.	Software was designed, developed, and tested with security in mind.	Software was designed, developed, tested, and externally verified with security in mind.	High
Implementation	Apex code implementation	>1K automatically identified defects.	>500 automatically identified defects.	>100 automatically identified defects.	>10 automatically identified defects.	0 automatically identified defects.	High
Implementation	Apex code structure	The code is complex, unclear, and difficult to maintain (spaghetti code). There are instances of repeated code where fixing or changing one will result in different behavior for the same desired function. The code contains "magic numbers."	The code is generally clear, but not structured and specifically written to be maintainable so there may be "magic numbers," hard coded logic, and so on.	The code is well commented and clear, variable naming is standardized, but in places there are repeating code segments (so that if one is updated, the other must also be updated).	The code is not well commented but it is clear, variable naming is standardized, repeating code is minimized.	The code is well commented and clear, variable naming is standardized, repeating code is minimized.	High

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
Implementation	Security - Configuration	Security roles are implemented incorrectly.	Security roles are partially defined and implemented.	Appropriate security roles are defined and implemented, but not fully tested and not assigned to individuals along with the related processes.	Appropriate security roles are defined, implemented, and tested, but assignment to individuals and related processes are not in place.	Security roles are defined, appropriate permissions are included in the design and implementation, are tested, and individuals are assigned to the correct roles. Automated procedures are in place to remove access when appropriate (e.g., an employee departs.)	Low
Maintainability	Built in Diagnostics	Errors are not visible unless the underlying data is manually validated.	The system implements a built-in-test (BIT) capability and proactively alerts users to identified serious errors.	The system implements a built-in-test (BIT) capability and proactively alerts users to identified errors, but this capability is not pervasive in the application.	The system implements a built-in-test (BIT) capability and proactively alerts users to identified error and warning conditions.	The system implements a built-in-test (BIT) capability and proactively alerts users to identified error and warning conditions with information needed to correct the problem.	Average
Maintainability	Control Executive	Both logic and data are	Parameterization is used and clear,	Business specific logic and	Control executive code is	Control executive code for areas	Very High

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
		scattered throughout the system, and there is repeated logic and data (e.g., the same logic/data in multiple places, so fixing one leaves the other unfixed).	but logic is scattered.	parameters are mostly separated out and grouped in a manner to simplify finding and modifying them.	used for most parameters and logic, but some business logic and data are grouped or otherwise clearly called out. The domains of each are clear.	likely to change (business rules, workflows, interfaces, etc.) are separate from business specific parameters and logic (e.g., business rules, specific variable values).	
Maintainability	Encapsulation	Functionality is not encapsulated, or the encapsulation is not aligned with the business domain.	Functionality is somewhat encapsulated and aligned with the business domain (modularity).	Functionality is mostly encapsulated and aligned with the business domain (modularity).	Functionality is strongly encapsulated and aligned with the business domain (modularity) but there are some specific, clear gaps or weaknesses.	Functionality is strongly encapsulated and aligned with the business domain (modularity).	Average
Maintainability	Error Diagnostics	>1K automatically identified defects.	>500 automatically identified defects.	>100 automatically identified defects.	>10 automatically identified defects.	0 automatically identified defects.	Average
Maintainability	Error Handling	Error handling is haphazard, error codes are not meaningful, and errors do not clearly point to	Errors are handled but the error messages may not be clear, or they may not clearly point to	Error handling varies, with some of it strong and some of it weak, and the weak	Error handling is mostly strong but there are some specific, clear gaps or weaknesses.	Errors are handled, and the diagnostic codes clearly identify the area of	Average

Category	Sub-Category	Very Low	Low	Average	High	Very High	Importance
		the underlying problem.	the underlying problem.	areas not clearly defined.		failure and specific problem.	
Maintainability	Interface Testing	Interface testing is only possible using actual interface partners and data.	Some controlled interface testing is possible, but other testing requires coordinating with actual interface partners.	Test data from actual interface operation is available for testing, using the actual interface in a test environment.	Controlled and actual test data and test scripts are available for testing, using the actual interface in a test environment.	Test harnesses, controlled test data, and test scripts are available for controlled interface testing.	Average
Maintainability	Logging Capabilities	The system has minimal or no logging capability.	Logging is designed to support the basic functions of recording errors and security related events, but not designed to support test and debug.	Logging is incorporated throughout the design, but the degree of logging is predefined and not configurable.	Logging is incorporated throughout the design, and the degree of logging can be set globally to assist with testing and debugging.	Logging is incorporated throughout the design, and the degree of logging can be set both globally and at the module level to assist with testing and debugging.	Low

Table 11 shows the assigned CARS Salesforce implementation quality score for each of the eMRI Salesforce implementation quality sub-categories, along with the justification for each setting. Justifications were based on a combination of manual analysis of the actual implementation of Salesforce and supporting tools within Azure DevOps, automated analysis of those components, review of documentation from SharePoint, and interviews. We have repeated the importance column from Table 8 for convenience. In reading the CARS Score column:

- A score of Very Low or Low would be considered poor implementation quality, which will show up as some combination of poor user satisfaction, system caused inefficient business operations, poor reliability, or poor maintainability.

- A score of Very High or High would be considered good implementation quality, which will show up as some combination of satisfied users, system enabled efficient business operations, high reliability, or high maintainability.
- A score of Average would be neutral, with some good characteristics and some areas for improvement.

Table 11: CARS Implementation Quality Model Input Settings

Category	Sub-Category	CARS Score	Justification	Importance
Design	Apex class architecture	Low	Encapsulation is pretty good here, but the application as currently architected and implemented, is tightly coupled. In interviews with OSaaS and with code review, it has been determined that any code change in the system would require many other areas to also be changed and updated, thus tightly coupled and difficult to update. This creates challenges for the business to not be able to be nimble and respond to changes in legislation and user requirements in a timely and cost-effective manner.	Average
Design	Architecture	Very Low	This architecture first fails in attempting to implement business rules internally within Salesforce. This has led to a high amount of unnecessary customization and configuration within the CARS solution. An Apex PMD (Program Mistake Detector) report came back with 11,000+ hits. Secondly, the user interface in the Filer's portal is fraught with UX errors. Third, the Data Portal does not need to be a custom Heroku based solution with a Postgres database with JavaScript and Java interacting and querying the database. It should not have to reinvent the wheel with a complete solution. The solution could use a more optimal enterprise and/or open-source solution like Splunk, Tableau, Tableau CRM or ElasticSearch.	Very High
Design	Database/Object design	Very High	After reviewing the CARS Data Model more closely, this does adequately serve the needs of the solution needs and does not have missing data and/or incorrect data. It uses approximately 50 delivered and custom Salesforce objects to fulfill the application needs with the proper relationships such as parent/child lookups.	Very High
Design	Functionality-Business	Very Low	It is our estimation that due to the very large volume of custom Apex code that has been developed in the CARS solution, the solution will need to be redesigned and re-implemented in this area. Also, business rules were not externalized for maintainability. It is very questionable that the system as	High

Category	Sub-Category	CARS Score	Justification	Importance
			currently designed will meet the critical functional needs of the business stakeholders and will be very difficult to iterate and improve as time goes on due to the design and complexity of the solution as it currently stands.	
Design	Functionality-Partner	Average	This applies to the Supplier in this case (Net File). The Filer's MuleSoft hosted API/Service is the saving grace here as 70 to 80% of the filings through submitting an XML file to CARS for candidates and lobbyists. The outlier is the direct submission through the Salesforce hosted Community for the Filer's Portal. The only caveat here is to potentially take the opportunity and go through the provided custom API which is probably very old and revisit the design and implementation of it. It really should not take 30 to 60 minutes to submit an application and have Salesforce ingest it and process it.	Average
Design	Functionality-Public	Average	this applies to the Public as well. The Filer's MuleSoft hosted API/Service is the saving grace here as 70 to 80% of the filings through submitting an XML file to CARS for candidates and lobbyists. The outlier is the direct submission through the Salesforce hosted Community for the Filer's Portal. The only caveat here is to potentially take the opportunity and go through the provided custom API which is probably very old and revisit the design and implementation of it. It really should not take 30 to 60 minutes to submit an application and have Salesforce ingest it and process it.	Average
Design	Functionality-Technical	Very Low	This is where the CARS Solution really fails to provide a proper solution with a very high cost of Maintenance & Operations due to the many customizations and resulting complexity of it. This will be very difficult to support going forward without a continued high degree of customization and work. There will need to be high expertise and familiarity of the application.	Average
Design	Interfaces	High	The Filer's MuleSoft hosted API/Service is the saving grace here as 70 to 80% of the filings through submitting an XML file to CARS for candidates and lobbyists. The outlier is the direct submission through the Salesforce hosted Community for the Filer's Portal. The only caveat here is to potentially take the opportunity and go through the provided custom API which is probably very old and revisit the design and implementation of it.	Average

Category	Sub-Category	CARS Score	Justification	Importance
			It really should not take 30 to 60 minutes to submit an application and have Salesforce ingest it and process it.	
Design	Performance Engineering	Average	The main area of concern is that it can take up to 30 to 60 minutes to file an application via the MuleSoft API/Service to CARS. An opportunity should be taken to revisit this API to streamline and improve and update its design to be more performant. *Note* - Performance testing has not been done on the Filer's or Data Portal.	Average
Design	Security - Custom Programming	Very Low	The built-in System Health Checks have provided feedback regarding security flaws that need to be rectified if the CARS solution were to go live. There are 3 findings in the Optimizer Report that require “Immediate Action Required”. The System Health Check revealed a 64% Poor rating with 5 critical areas.	High
Implementation	Apex code implementation	Very Low	There are currently 11,446 defects that came back from the Apex PMD plugin utility. These are for the custom Apex code that has been developed for the CARS Solution. These categories include: Best Practices (2,772), Code Style (2,949), Design (1,699), Documentation (1,729), Error Prone (110), Performance (1,252) and Security (935).	High
Implementation	Apex code structure	Very Low	There are currently 11,446 defects that came back from the Apex PMD plugin utility. These are for the custom Apex code that has been developed for the CARS Solution. These categories include: Best Practices (2,772), Code Style (2,949), Design (1,699), Documentation (1,729), Error Prone (110), Performance (1,252) and Security (935).	High
Implementation	Security - Configuration	Very Low	This CARS application is not ready for go-live due to security flaws and security roles being partially implemented. There are zero security roles currently implemented in the UAT environment, zero permissions sets and very few user profiles. We would expect more security configuration at this point to go live. Maybe this is planned for later in the MVP project.	Low
Maintainability	Built in Diagnostics	Average	The CARS solution is built on the Salesforce platform; therefore, it has delivered error and warning conditions with the information that is needed to correct the appropriate problem within the UX from the Apex Code. The MuleSoft API/Service also has built-in error handling. Also, if there is	Average

Category	Sub-Category	CARS Score	Justification	Importance
			additional action that needs to take place in relation to a submitted Filing, a Salesforce Case is submitted where support staff needs to follow up.	
Maintainability	Control Executive	Average	Data is organized well via the custom CARS data model. Business logic is mostly separated throughout the solution in configuration and custom Apex code. The latter is more difficult to track down and troubleshoot.	Very High
Maintainability	Encapsulation	Average	Encapsulation is pretty good here, but the application as currently architected and implemented, is tightly coupled. This means that with any future change, there are many dependencies that must be considered going.	Average
Maintainability	Error Diagnostics	Very Low	There are currently 11,446 defects that came back from the Apex PMD plugin utility. These are for the custom Apex code that has been developed for the CARS Solution. These categories include: Best Practices (2,772), Code Style (2,949), Design (1,699), Documentation (1,729), Error Prone (110), Performance (1,252) and Security (935).	Average
Maintainability	Error Handling	Average	There is error handling built-in throughout the CARS Application. However, the error messages that are produced should be updated so that they are more useable and readable, and thus the problem or issue can be resolved more quickly.	Average
Maintainability	Interface Testing	Average	This has been done just with a manual testing process using Postman and a sample test file. There has not been much in the way of performance testing except for uploading a large XML file that was double the size that took 60 minutes for the load and ingestion into Salesforce.	Average
Maintainability	Logging Capabilities	Very High	There is a custom Apex Code log object that is used throughout the CARS application by the Apex Classes. this captures such things as the class name, LogDate, Loglevel and stacktrace errors. There is also Salesforce delivered functionality that allows for this.	Low

Table 12 contains the eMRI model used to evaluate the CARS system integrator capabilities. This model is used to evaluate the ability of a project team to effectively manage the implementation of a large and complex information technology project. While the primary organization evaluated is the assigned system integrator (OSaaS, in this case), the evaluation also includes other stakeholder groups within the organization that play key roles with respect to system

integration activities (e.g., governance by the State stakeholders). As with the previous eMRI models, this model is broken down into categories (key process areas, or KPAs), which are simply broad areas of capability analysis. For each category, the demonstrated system integration capabilities will be scored somewhere between Very Low (area of weakness, no repeatable process strength) and Very High (area of strength, repeatable and optimum processes). The specific criteria used when assigning this score are also shown in the table. Finally, different categories may have more or less importance to the final (total) system integrator score. This subjective setting, shown in the Importance column, ranges from Very Low (not important) to Very High (exceptionally important).

Table 12: eMRI System Integration Assessment Model

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
Transformation Governance	The organization has limited or no decision-making framework.	Governance members are identified and Decision-making structures, process are under development. Governance meetings are more informational and less about decisive and intentional strategic direction.	A set of structures, processes, and qualified members comparable to the transformation proposed for the organization.	Established structures, processes, criteria, and qualified members required for decisioning directly aligned with the Organization vision, business outcome, and applicable regulations and policies.	Structures, processes, criteria, and members are rewarded for decisions that directly influence achievement of the desired transformation directly aligned with the Organization vision, business outcome, and applicable regulations and policies.	Average
Architecture Governance	The organization has limited or no architecture governance framework aligned	Governance members are identified and decision-making structures, process	A set of structures, processes, and qualified members comparable to (or appropriate for) the	The Governance Model has an active role in periodic decisioning associated with	Architecture Governance is stable and refined based on the quantifiable	High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
	with Transformation or Business Outcomes.	are under development. Governance meetings are more informational and less about decisive and intentional strategic direction.	transformation proposed for the organization. Meeting focus on Architecture improvement or refinement based on defined business outcomes, strategic direction, and principles.	enterprise debt, technical debt or adjustments in desired business outcomes, strategic direction and/or principles.	effectiveness of how well transformation efforts manage cost, risk, flexibility, and quality of solutions that achieve desired business outcomes.	
Architecture Development	The organization has identified the architecture discipline as important in a large transformation. Role(s) and Responsibilities have been defined for the function.	The organization has an architect or architecture group that reviews current or proposed architecture and provides input to the construct and content included in architecture artifacts.	The organization has an architecture development group who develops architecture artifacts that guide transformation efforts (solution design, project execution, etc.)	The organization has a group that develops, maintains, and keeps architecture artifacts current. Architecture is addressed pre-project, supports all levels of governance, and is measure for effectiveness.	Architecture Development directly reflects the effectiveness, efficiency, agility, and durability of the organization by supporting the management of cost, risk, flexibility and quality of solutions, systems, and services, through Architecture modeling, and supporting enterprise governance.	Very High
Requirements Development	The organization's approach to	The organization calls in Subject	The organization has a defined process	The organization has the budget, roles	In addition to Level 3-4 capabilities, the	Very High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
	requirements development is point in time, adopted from other sources, and minimally effective for guiding transformation efforts.	matter experts to develop requirements per project effort based on point in time understanding of desired business outcomes.	for deriving requirements from motivational factors (strategies, outcomes, regulation, etc.) and architecture specifications.	and processes required to manage a requirements knowledge base to guide and constrain transformation efforts.	organization measures the effectiveness and efficiency of requirements sets against the outcomes of project efforts.	
Business Process Integration (BPI)	The organization reacts to new regulation, systems, and process change as or after they are implemented to meet performance metrics.	The organization has a role and emerging process for BPI that participates in projects or transformation efforts proactively to synchronize internal operations and proposed system changes or new solutions.	The organization has a defined discipline (roles, responsibilities, and processes) to plan business improvement and new system integration.	The organizations BPI discipline has the budget and resources to analyze motivational factors, architecture, and the current environment to recommend architecture changes, drive OCM, and measure business improvements.	Business process integration (BPI) synchronizes internal operations across department, program functions and external partners using forecasted and real-time data to measure effectiveness of BPI and the associated transformation efforts.	Average
Solution Design	The organization does not have a design system or framework, has limited or no solution specification, principles, or	The organization outsources solution design and has little or no solution design knowledge transfer during the process. Solution Design system or	The organization adopts a design system or framework and trains staff to participate in or own the solution design effort.	The organization's SD unit or team drives solution design based on a framework, standards, and past designs, according to an improvement	Solution Design is a standardized, quantified function, managed by a team of solution designers according to a SD Framework	Very High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
	constraints to govern the solution design process.	frameworks vary from solution to solution.	Training or hiring increases the solution design skills and matures the framework.	blueprint or roadmap. The SD team participates in governance of organization improvement.	that is continuously improved.	
Project Management	The organization operates in a relatively random manner, with limited project control, and low predictability of project success, particularly when faced with a crisis. Success on projects is possible (on time on budget) quality and repeatability is unlikely.	The organization operates PM according to a structured or framework-based approach, with basic project management practices at an individual project level. Overall project success depends on key individuals or specific management support rather than on adoption of standards.	The organization has adopted and uses well-defined project management procedures are documented and used as a standard of operations. Defined at an organizational level, personnel are trained and informed, and typically are proactively executing the function.	The organization manages PM as a discipline, measuring project performance using well-defined metrics. Standards are reviewed, improved, and agreed to across the organization, and common metrics are used to manage business decisions and processes.	The organization focuses on deliberate and continual process improvement of the PM function to optimize performance through unique and innovative techniques that enhance the framework and are tailored specifically to the organization's vision and strategies.	High
Requirements Management	The organization does not manage requirements in a formal manner and: * has little or no documentation for project, user & system	The organization relies on static, point in time requirement documents, created and maintained by subject matter experts per project	The organization manages requirements centrally according to a standard and shared process, for the purpose of compliance and risk	The organization manage all improvement cycles (strategic and tactical) based on the current `requirements knowledge base	In addition to Level 3-4 capabilities, the organization continuously improves the requirements universe, and influences overall	Very High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
	requirements * Relies primarily on a concept level or procurement document. * Leverages user stories with no clear distinction between the functionality of the system and the expected outcome or user experience.	and not as an enterprise asset.	management, typically in a standard or shared tool (spreadsheet, library, or requirements tool), making the requirements accessible and visible to solution design, project management, development, testing and implementation management concurrently.	which includes structured and standard data, supports validation and verification, and provides clear traceability to enterprise motivational factors (strategy, business outcomes, regulation, etc.)	quality management and process control across the portfolio of improvement initiatives and projects.	
Solution Development	The organization does not provide a stable culture or environment for developing new solutions. Development is often undermined by ineffective planning, reaction-driven change process short-cuts and risks, limited architecture and engineering	The organization establishes policies for managing or outsourcing solution development efforts. The organization names a SDLC pattern or framework but has emerging procedures to improve implementation both policies and framework. Solution	The organization has a defined SDLC with processes, policies and parameters for successful solution development shared across the improvement initiatives and projects. Solution Development works from a planned improvement cycle according to architecture,	The organization manages solution development according to Vision, Business outcomes and project objectives and is measured quality, time to develop, and value of solutions.	In addition to Level 3-4 capabilities, the organization measures the effectiveness and efficiency of solution development through a continuous improvement cycle as a part of overall enterprise effectiveness and efficiency.	Very High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
	specifics. Performance depends on the capabilities of individuals or teams and varies with skills, knowledge, and motivations.	Development is structured for each solution development effort or team. Development is supported by external teams like: Architecture, Engineering, project management and or oversight groups (UX, QA, Testing, IV&V, etc.)	engineering and roadmaps, at an established velocity.			
Implementation Management	The organization has limited or no implementation experience or approach. Implementations are ad-hoc with success based on architecture, engineering, and development quality.	The organizations approach implementation adopts or develops a methodology for the steps, controls, and principles of implementation of solutions across multiple stakeholders, uses and business partners. The Implementation methodology is used for the initial deployments, releases, and	The organizations Implementation framework defined, trained, repeatable and sufficient for most improvement cycles. Implementations are successful across multiple customer scenarios and solution environments. Implementation Framework provides: Processes, procedures, accelerators,	The organization's Implementation Framework is managed and measured for performance as a key part of improvement projects and initiatives. implementation management support governance and reports on implementation efficiency, customer satisfaction, and solution quality.	In addition to level 3&4 capabilities, Implementation Management is managed through a continuous improvement cycle based on each Improvement initiative.	Very High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
		upgrades retaining solution implementation knowledge including traceability to integration points and test results from each implementation.	checklists, standard documentation Standard approaches and roadmaps.			
Configuration Management	The organization's approach to CM has limited or no structure, experience, or defined process. due to absence of prior need or governance. Practice is not seen as vital or necessary by senior management based on current systems, solution ownership or need for improvement.	The organization adopts or develops an approach as assigns a team or group to CM, that incorporates or develops CM standards and principles. CM experience and/or training is pursued. CM data is not recognized as an enterprise asset across the improvement lifecycle.	The organization has a defined Framework for CM including codified methodology, standardized artifacts, CM training and governance, as well as a CM data Environment (e.g., CMDB) The proactive approach is a strategic part of planning and executing improvement initiatives and projects, especially new system or solution development efforts.	The organization has a measured CM program operating from an organization specific CM framework and is considered as a vital part to the effectiveness and efficiency of Improvement initiatives and projects.	In addition to level 3&4 capabilities, Configuration Management is managed through a continuous improvement cycle based on the effectiveness of each Improvement initiative.	Average

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
Interface Control and Management	The organization has little or no defined approach to interface control and management. The function is considered outside the scope of organization responsibilities.	The organization adopts an initial process to establish interface standards, policies, and controls. Quality of interface management is based on subject matter expertise and domain knowledge of personnel.	The organization has a defined process for interface control and management including processes, governance, standards, and policy or principles.	The organization has an established practice for Interface control and management that is measured for quality, consistency and effectiveness of solution interfaces and user experience.	In addition to Level 3-4 capabilities, the organization measures the satisfaction, effectiveness, and efficiency of interfaces to drive a continuous improvement cycle.	High
Legacy Integration	The organization has little or no defined approach to Legacy Integration. The function is considered outside the scope of organization responsibilities.	The organization adopts an initial process to establish integration standards, policies and controls based on currently defined legacy systems (internal and external). Quality of legacy integration is based on subject matter expertise and domain knowledge of personnel.	The organization has a defined approach for Legacy Integration including processes, governance, standards, and policy or principles. The process describes and delineates enhancements and new integrations.	The organization has an established practice for Legacy Integration that is measured for quality, consistency and effectiveness of solution interfaces and user experience. The function is staffed with Solution and Legacy experts.	In addition to Level 3-4 capabilities, the organization measures the effectiveness and efficiency of integrations to drive a continuous improvement cycle.	Average
Business to Business (B2B) Integration	The organization has little or no defined approach to B2B	The organization adopts an initial process to establish	The organization has a defined approach for B2B Integration	The organization has an established practice for B2B	In addition to Level 3-4 capabilities, the organization	High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
	Integration. The function is considered outside the scope of organization responsibilities.	integration standards, policies and controls based on currently defined B2B systems. Quality of B2B integration is based on subject matter expertise and domain knowledge of personnel.	including processes, governance, standards, and policy or principles. The process describes and delineates enhancements and new integrations.	Integration that is measured for quality, consistency and effectiveness of solution interfaces and user experience. The function is staffed with Solution and B2B experts.	measures the effectiveness and efficiency of integrations to drive a continuous improvement cycle.	
Enterprise Solution Integration	The organization has little or no defined approach to Enterprise solution Integration. The function is considered outside the scope of organization responsibilities.	The organization adopts an initial process to establish integration standards, policies and controls based on new solutions and systems being integrated into the current environment. Quality of legacy integration is based on subject matter expertise and domain knowledge of personnel.	The organization has a defined process for enterprise solution integration including processes, governance, standards, and policy or principles. The process describes and delineates enhancements and new integrations.	The organization has an established practice for enterprise solution integration that is measured for quality, consistency and effectiveness of solution interfaces and user experience. The function is staffed with Solution and solution experts with domain knowledge.	In addition to Level 3-4 capabilities, the organization measures the effectiveness and efficiency of integrations to drive a continuous improvement cycle.	Very Low
Solution Testing and QA	The organization has little or no defined approach to solution Testing and	The organization adopts an initial process to establish testing and QA	The organization has a defined process for solution testing and QA including	The organization has an established Testing and QA practice that is	In addition to Level 3-4 capabilities, the organization measures the	High

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
	QA. The function is considered outside the scope of organization responsibilities.	standards, policies and controls based on new solutions and systems being developed. Or the organization outsources the function to industry experts.	processes, governance, standards, and policy or principles. The process describes and delineates enhancements and new integrations.	measured for quality, consistency and effectiveness of solutions and solutions with traceability to architecture, engineering, and solution requirements. The function is staffed with testing and QA professionals with domain knowledge.	effectiveness and efficiency of testing and QA to drive a continuous improvement cycle.	
Operations (Performance Management)	The organization has little or no defined approach to Performance management for Operations.	The organization adopts a standard approach to Operations Performance management to establish integration standards, policies, and performance levels. Quality of operations is directly proportionate to the expertise and domain knowledge of the operations team.	The organization has a defined Operations Framework for all program functions and services, guided by governance, standards, and policy or principles.	The organization has an established Operations practice and environment that is measured for quality, compliance and effectiveness all defined services. The function is staffed with Domain Service Professionals with extensive domain knowledge.	In addition to Level 3-4 capabilities, the organization measures the effectiveness and efficiency of integrations to drive a continuous improvement cycle.	Average

Category	Initial Level - Ad-hoc	Level 2 – Repeatable	Level 3 – Defined	Level 4 – Managed	Level 5 – Optimized	Importance
Integrated Program Management	The organization has little or no defined approach to Program Management.	The organization adopts an initial approach to program management to increase standards, policies and capabilities required for enterprise improvement cycles.	The organization has a defined framework for Integrated Program management including processes, governance, standards, and policy or principles. The process describes how Program management operates the entire change portfolio and improvement lifecycle.	The organization has an established practice for Integrated Program Management that is measured for quality, consistency and effectiveness of Transformation budget, time, and resources throughout the improvement lifecycle. The practice is directly tied to executive leadership, and enterprise or transformation governance.	In addition to Level 3-4 capabilities, the organization measures the effectiveness and efficiency of integrations to drive a continuous improvement cycle.	Low

Table 13 shows the assigned CARS system integration score for each of the eMRI system integration capability categories, along with the justification for each setting. Justifications were based on a combination of review of documentation from SharePoint and interviews. We have repeated the importance column from Table 8 for convenience. In reading the CARS Score column:

- A score of Very Low or Low would be considered weak system integration capabilities, which will show up as some combination of risks to project success in terms of scope, budget, schedule, quality, technical objectives, and total cost of ownership. These projects tend to be unpredictable, and they have a high failure rate.

- A score of Very High or High would be considered a strong system integration score, which will show up as successful projects in terms of scope, budget, schedule, quality, achieving technical objectives, and total cost of ownership. In this context, success involves setting realistic objectives and then achieving those objectives. Strong system integration skills result in predictability, not necessarily "cheap" projects in terms of cost.
- A score of Average would be neutral, with some good characteristics and some areas for improvement.

Table 13: CARS System Integration Model Settings

Category	CARS Score	Scoring Rationale	Importance
Transformation Governance	Very Low	The concept of transformation governance does not exist within the organization or the SI. The project has been operating without formalized vision, strategy, success metrics, decision matrices, or a project charter. The organization is missing many key attributes in this area.	Average
Architecture Governance	Very Low	Architecture is not a formalized practice and has not been formally governed.	High
Architecture Development	Very Low	Some solution architecture has been performed. However, disciplined architecture development is not evident.	Very High
Requirements Development	Low	Project requirements and business rules from 2018 exist but corresponding artifacts and elaborations lack traceability and precision.	Very High
Business Process Integration (BPI)	Very Low	Business process definitions do not exist currently. Therefore, alignment to the solution from a business process perspective has not occurred.	Average
Solution Design	Low	There is not a clear, holistic view of the solution design. Solution design is lacking design rationale that considers impact to the current state including stakeholders, business processes, and system interfaces.	Very High
Project Management	Low	The project management organization is missing key attributes (e.g., capacity, scope, risk, controls) to manage a multi-vendor enterprise system development and implementation project.	High
Requirements Management	Low	Original project requirements exist from 2017. However, the organization's understanding and documentation of requirements elaboration activities, related artifacts and traceability are lacking.	Very High
Solution Development	Very Low	Solution Development has been significantly constrained by the lack of architecture, design, governance, and decision-making framework.	Very High

Category	CARS Score	Scoring Rationale	Importance
Implementation Management	Very Low	The Implementation Management function is missing key attributes (e.g., capacity, scope, risk, controls) to implement a multi-stakeholder enterprise system.	Very High
Configuration Management	Very Low	The core solution platform of the project (Salesforce) is inherently configuration centric. However, the project has not seemed to operate under a formal configuration management protocol.	Average
Interface Control and Management	Very Low	Interfaces were primarily developed without adequate consideration for the high-volume of existing data feeds in the current-state ecosystem.	High
Legacy Integration	Very Low	A strategy for legacy integration was not available. Legacy integration has been addressed ad-hoc with respect to key legacy systems (internal and external).	Average
Business to Business (B2B) Integration	Very Low	A strategy for B2B integration was not available. B2B integration has been addressed ad-hoc.	High
Enterprise Solution Integration	Very Low	A strategy for enterprise solution integration was not available. Enterprise solution integration has been addressed ad-hoc.	Very Low
Solution Testing and QA	Very Low	The Solution Testing and QA function is missing key attributes (e.g., capacity, scope, risk, controls) to implement a multi-stakeholder enterprise system.	High
Operations (Performance Management)	Very Low	The Performance Management function to support Operations is missing key attributes (e.g., capacity, scope, risk, controls) to implement a multi-stakeholder enterprise system.	Average
Integrated Program Management	Very Low	The organization has not managed the CARS project from the context of Integrated Program Management. The organization is missing key attributes (e.g., capacity, scope, risk, controls) to support effective Integrated Program Management for multi-stakeholder enterprise systems.	Low