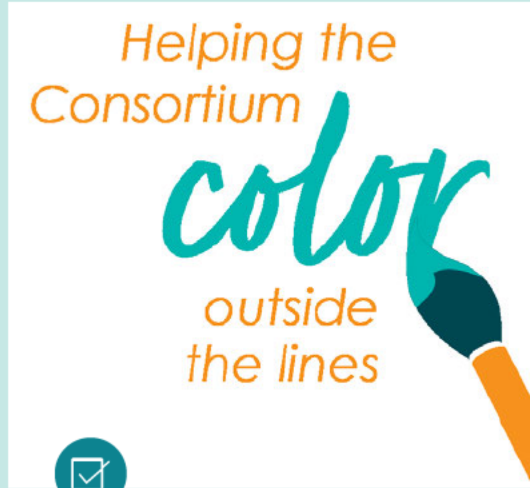
Understanding and Approach to M&E Services

Section 4



RFP Reference: 6.3.8.6 Section 4 – Understanding and Approach to M&E Services

The Bidder shall provide a detailed narrative response to the Understanding and Approach topics outlined in Section 5.3. Bidders will respond to the following areas to satisfy or exceed the RFP requirements as described in Section 5 - Requirements, addressing the following topics:

- Sub-Section 5.3.3.1 - Integrated Multi-Contractor Environment
- Sub-Section 5.3.3.2 - Application/Architecture Evolution
- Sub-Section 5.3.3.3 - System Change Requests
- Sub-Section 5.3.3.4 - Innovation
- Sub-Section 5.3.3.5 - Transition-In

By "coloring outside the lines" with Deloitte, the CalSAWS Consortium can deliver County-requested changes faster, provide better integration across vendors, and deliver a next-generation CalSAWS' architecture. To help the Consortium rethink the status quo, we bring a fresh perspective informed by delivering 31 Eligibility and Enrollment (E&E) systems, including California. This experience enables us to implement improved M&E processes via user-centered design and enabling technologies (as we have demonstrated on both the BenefitsCal project and the operation of CalHEERS for Covered California).

SECTION HIGHLIGHTS

- **An M&E approach informed by current E&E delivery projects in 26 states.**
- **An approach that puts humans in the center rather than technology.**
- **An evolved CalSAWS architecture that delivers higher levels of responsiveness while retaining stability.**
- **An approach that helps the Consortium color outside the lines to better serve California Counties.**
- **Processes and tools that accelerate delivery of System Change Requests (SCRs).**

The End Result: The Consortium obtains a responsive, reliable, and innovative vendor that leverages national experience and technical knowledge that supports the Counties with their mission to provide timely health and human services.

4.2.1 Breaking Down the CalSAWS Application (ME-UA4)

RFP Reference: 5.3.3.2 - M&E Understanding and Approach to Application Evolution	
ME-UA4	Describe your strategy and approach to breaking down the large CalSAWS application into feature modules, prioritizing, and refactoring the application to evolve the application architecture. Describe how this strategy will be integrated with the DB Migration effort including decoupling the CalSAWS application from the database structures into feature modules. Describe how this strategy will address security considerations, reduce costs, and improve optimization, scalability, and flexibility.

4.2.1.1 Strategy and Approach

CalSAWS modernization requires careful consideration of policy, CalSAWS operational impact, County operations (current and future), technology, and financials. All the while, Counties must maintain continuity of services to vulnerable customers, respond to policy and program changes, and enable workers to address increasing caseloads with care and compassion. Deloitte walks this modernization tightrope with the Consortium and its partners to deliver a resilient CalSAWS^{CN} that enables the Counties to continue delivering mission critical services with the aim of enhancing overall operations.

LOOKING FORWARD

- A CalSAWS^{CN} world where individual application components automatically scale to meet user volumes across the mix of transactions.
- The Consortium can make complicated changes to meet complex policy needs with no impact to the business operations.

As the leading public sector AWS partner, as noted in Figure 4.2.1-1, we deliver cloud native modernization at scale. For example, California now reaps **better scalability** and is capable of operating at **120% of peak without additional infrastructure** and **30% reduced resource usage for non-peak workloads from our CalHEERS cloud-native modernization** approach and AWS alliance. BenefitsCal also uses AWS cloud native services that added new features. For example, BenefitsCal uses Amazon Lex to implement a public facing virtual assistant. It also uses Amazon Textract that reads SAR7 status report images from Customers to identify if changes are reported. Deloitte's BenefitsCal serverless architecture increased has driven down the overall cloud costs and limited non-production environments to less than the production environment. This comes with a month over month reduction in the per user cost as the adoption and user population continues to grow.

Deloitte is the Premier Partner to take Advantage of AWS Capabilities

aws partner network	aws partner network	aws partner network	aws partner network	aws partner network
Premier Consulting Partner	Premier Consulting Partner	Premier Consulting Partner	Premier Consulting Partner	Premier Consulting Partner
	DevOps Competency	Security Competency	Machine Learning Competency	Public Sector Partner

CA_CON_CalSAWS MandO_MO-379038_2022-0161_02

Figure 4.2.1-1. An Industry-Leading Partnership in Your Focus Areas.

Technology Architecture Evolution Guidelines

We gather our architectural evolution guidelines by consistently learning from modularization projects of similar complexity within the HHS field. In Figure 4.2.1-2, we indicate the "must have" considerations in a cloud native, serverless context with our cloud alliance partner, AWS, retaining responsibility for container platforms, the operating system, virtualization, and hardware. Through our partnership we regularly complete AWS Well Architected Reviews to continually evolve and improve. We are also certified to conduct these reviews, for clients to further evolve their capabilities and systems. This helps create a shared understanding of leading practices, providing sound, equitable technology decisions.

Architecture Modernization Principles – Cloud Native Thinking

- **Modularize Business Services** for flexibility and reuse
- Build **Cloud Native Serverless Microservices** to enable configurability and agility
- Apply the **AWS Well Architected Framework** to enable best solution and process optimization
- Leverage **Infrastructure as Code (IaC)** for initial and ongoing builds and deployments; to enable automation and reliability
- Optimize compute and operations cost with on-demand elastic scaling with a simplified, standardized technology stack
- **Zero-trust** and Design in Depth policy for **Layered Security Architecture**
- Attain separation of concerns across application and platform
- **Operational Efficiency through Distributed Data** using cloud native databases for better scalability and cost rationalization

CA_CON_CalSAWS MandO_MO-379038_2022-0029a_6

Figure 4.2.1-2. Guiding Principles for CalSAWS^CN Architecture Evolution Roadmap.

The Design Guidelines in Figure 4.2.1-3 represent how we implement the Guiding Principles during the architecture and design to deliver CalSAWS^CN.

Modular E&E Business Services

- Modularize business capabilities into smaller **cloud native microservices** to optimize cost and maximize on-demand scalability such as:
  - Deploy Appeals and Hearings as a Serverless Function
  - Host Case Management on serverless containers
- Enable reuse of atomic business services (e.g., same case summary service works for both the worker and contact center worker)

Infrastructure as Code (IaC) and DevSecOps

- Leverage Cloud-formation and Terraform scripting with externalized configuration for environment definition
- Create configurable and modular build scripts to enable faster continuous, on-demand deployment cycles
- Integrate code quality, code coverage, and static vulnerability scans as part of build pipeline

Security

- Implement DevSecOps principles with security phase gates defined throughout the application development lifecycle
- Apply principle of least privilege across the layers via policy configurations
- Design and implement security controls to demonstrate compliance with security standards applicable to CalSAWS operations
- Improving security, without diminishing the customer experience through human-centered design

Transform Operational Efficiency

- Refactor and break down large processes into smaller operations to **leverage real-time and event streaming** to increase user and batch throughput
- Implement configuration driven modular workflows (e.g., STEP Functions) to orchestrate operations like task management and tailor these to Counties
- Align data to the processes owning the data to reduce errors and improve data quality
- Increase observability through improved logging, monitoring, and the application of AIOps for early detection and prevention of failure

CA_CON_CalSAWS MandO_MO-379038_2022-0029b_6

Figure 4.2.1-3. Design Guidelines for CalSAWS^CN Architecture Evolution Roadmap.

Case Study – Evolving CalHEERS

We recently refined these guiding principles in California, where CalHEERS successfully moved away from an operationally rigid and expensive legacy technology to a nimble, cloud native architecture. CalHEERS completed its transformation in a 2-year timeframe applying the activities in Table 4.2.1-1.

Our CalHEERS Evolution Activities	How We Delivered Value
Migrated Oracle database to Aurora PostgreSQL.	• Improved database scalability, resilience, and cost-effective resource usage.
Rearchitected the existing monolith to lightweight services and replaced legacy COTS (e.g., Oracle) stack with cloud native and open-source software.	• Reduced implementation and deployment cycle time for functional and policy changes using faster, flexible, minimum to no-downtime deployment end state architecture.
Implemented a self-healing and self-recovering OpenShift Container Platform architecture with increased redundancy and fault tolerance.	• Reduced deployment downtime. • Introduced rapid system response to user demand. • Simplified adoption of innovative, cloud native technologies.
Migrated legacy data warehouse to Snowflake data platform.	• Increased business intelligence and visibility. • Introduced rapid change / enhancement cycles.
Replaced a legacy security solution with a modern security architecture built on the principles of Zero Trust.	• Improved the maturity of CalHEERS security program with enhanced security capabilities. • Leveraged tools with mature security capabilities (e.g., Palo Alto Prisma Cloud, ForgeRock, Okta, 3-Scale, TrendMicro).
Implemented IaC using automation pipelines and tech stack modernization with Snowflake, ForgeRock, Red Hat Fuse, Apache Drools.	• Provided enhanced security and serverless deployment. • Reduced maintenance, operations, and infrastructure costs over time.
Implemented an elastically scalable infrastructure to respond to user traffic.	• Improved system speed and user demand response time, providing a better user experience (UX) and reduced wait times.
Replaced legacy Java Server Pages (JSP) screens for service center with modern React architecture.	• Provided responsive UX and equitable access to modern technology across users and stakeholders.
Replaced legacy integrations with modernized, standardized, highly secure interfaces between the different partners.	• Enabled collaboration and cohesive response across partners.

Table 4.2.1-1. Application/Architecture Evolution at CalHEERS.

CalSAWS technological complexity is similar, though programs and functions vary. We plan a similar journey with the Consortium in modernizing CalSAWS considering the uniqueness of E&E programs.

Organizing CalSAWS into Feature Modules

Distilled from our 31 E&E implementations, [redacted] This is a set of clearly defined feature modules that logically organize the business features needed to serve the E&E domain. Shown in Figure 4.2.1-4 below, it serves as our reference architecture for analyzing CalSAWS and making recommendations for aligning system features, business functions and capabilities into independent microservices. This removes the guess work of about which E&E functions go into which microservices.

[redacted]

Building the Approach to Application / Architecture Evolution Deliverable

We work with the Consortium to define the CalSAWS[CN] end state vision through the **Approach to Application / Architecture Evolution deliverable.** Table 4.2.1-2 lays out the key activities in our approach. As the transition begins, we begin work on the key activities with the expectation that we will complete this work during the first six months of transition (see Figure 4.2.1-5). The first two activities increase the understanding for the transition team as well as the architecture team. We plan to complete this work in the first six months of transition so we can start the database migration efforts by month 7 of the contract.

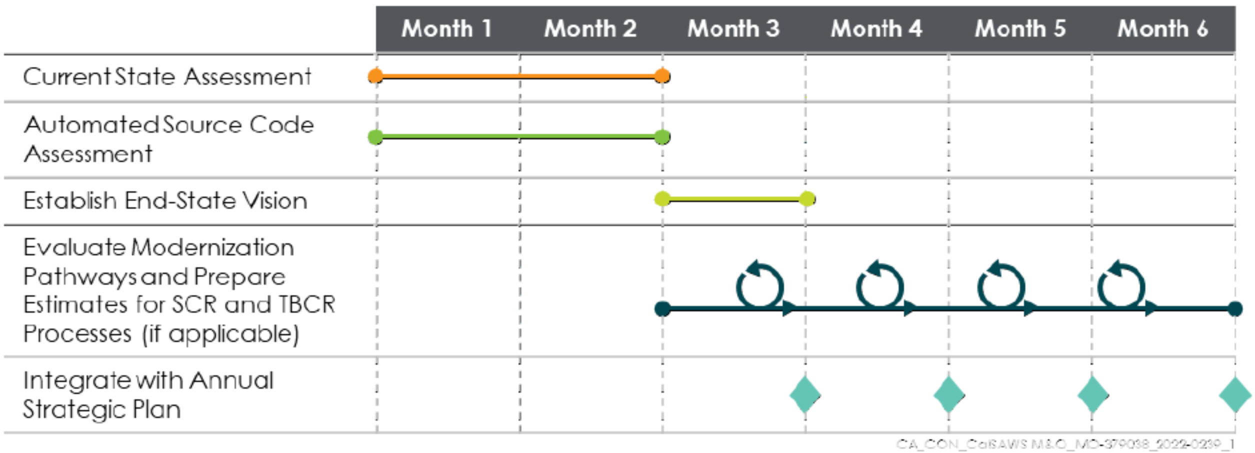


Figure 4.2.1-5. Key Activities Timeline.

Activity	Timeframe	Outcomes
Current State Assessment	Months 1-2	• Review current state with stakeholders. • Determine architecture modernization priorities and concurrent business technology changes.
Automated Source Code Assessment	Months 1-2	• Establish cloud native standards with the Consortium. • Create a current state baseline from the database and source code [redacted] • Identify conformance to cloud native standards and cloud anti-patterns using Deloitte's StormFury tool to remove technical debt.
Establish End-State Vision	Month 3	• Establish an end-state vision to guide the evolution journey, using assessment outputs, stakeholder input, production observations, and innovation opportunities. • Confirm technologies to support end-state vision (e.g., COTS product replacements for Oracle.)

Activity	Timeframe	Outcomes
Evaluate Modernization Pathways	Months 3-6 Iteratively	• Determine the optimal path to modernization for each feature module (e.g., rearchitect, refactor, re-platform, rehost.) • Prioritize feature module projects considering County needs, program / policy needs, funding, and return on effort. • Calculate the Evolution Impact Score, a rubric to assess progress on select dimensions, and choose optimal modernization sequencing.
Prepare Estimates for SCR and TBCR Processes (if applicable)	Months 3-6 Iteratively	• Consider effort estimates holistically, include detailed design, stakeholder input and review, technology and platform proofs of concept, feature modifications/ rewrites/ evaluations of replacements, and underlying technology platform features.
Integrate with Annual Strategic Plan	Months 3-6 Iteratively	• Establish cloud native baseline modules to determine journey progress to end-state vision. • Set up projects that enable the target state vision for the Consortium after the cloud native baseline is established.

Table 4.2.1-2. Application Evolution Roadmap from Architecture Through Operations.

As highlighted in Figure 4.2.1-6, and detailed below, [redacted] to support the design of new CalSAWS feature modules:

- [redacted]
- [redacted]
- [redacted]

[redacted]

The insights from the automated source code analysis and the feature modules from the Reference Architecture (Figure 4.2.1-4) provide microservice candidates and application programming interfaces (APIs) used to finalize the CalSAWS^{CN} architecture.

After completing the analysis and mining with the source code analysis, we will apply Generative AI to further refine and generate the base modules and automated test base code for developers finalize. [redacted]

These items inform the complexity and sequence recommendation based on the size of the modules generated and the complexity rating of the generated code.

Evolution Workstream and Sequence Recommendation

The Approach to Application / Architecture Evolution deliverable outlines the sequencing and activities for each phase of the database and application changes to achieve the target state architecture. Our current, high-level recommendations for the sequence of application and database changes are expressed in Figure 4.2.1-7 on the next page. Our recommendations begin with the database evolution. One of the biggest challenges to application evolutions is the data entanglement with the legacy application. Another is the limitation a legacy database places on the modules as they are decoupled from the monolith. Evolving the database and the data access layer first will address these challenges first and identify any areas of additional complexity that aren't obvious outside of the code.

Deloitte reviews and update these recommendations based on Consortium input when developing the **Approach to Application / Architecture Evolution** deliverable.

Each phase of the application / architecture evolution includes the evaluation of which **existing CalSAWS software products should be replaced** in coordination with the refactoring of the monolithic Java application. This coordination minimizes risk and accelerates adoption of an overall cloud native architecture that is independent of the technology stack. Additionally, **Batch Modernization** takes place incrementally throughout each phase to minimize risk while providing immediate benefit and improvement. The user interface changes flow along with the service phases after the workflow is completed. This allows CalSAWS users to have an improved UX module by module.

Estimation, change request creation, and implementation considerations are resolved via the Consortium's SCR and annual planning processes.

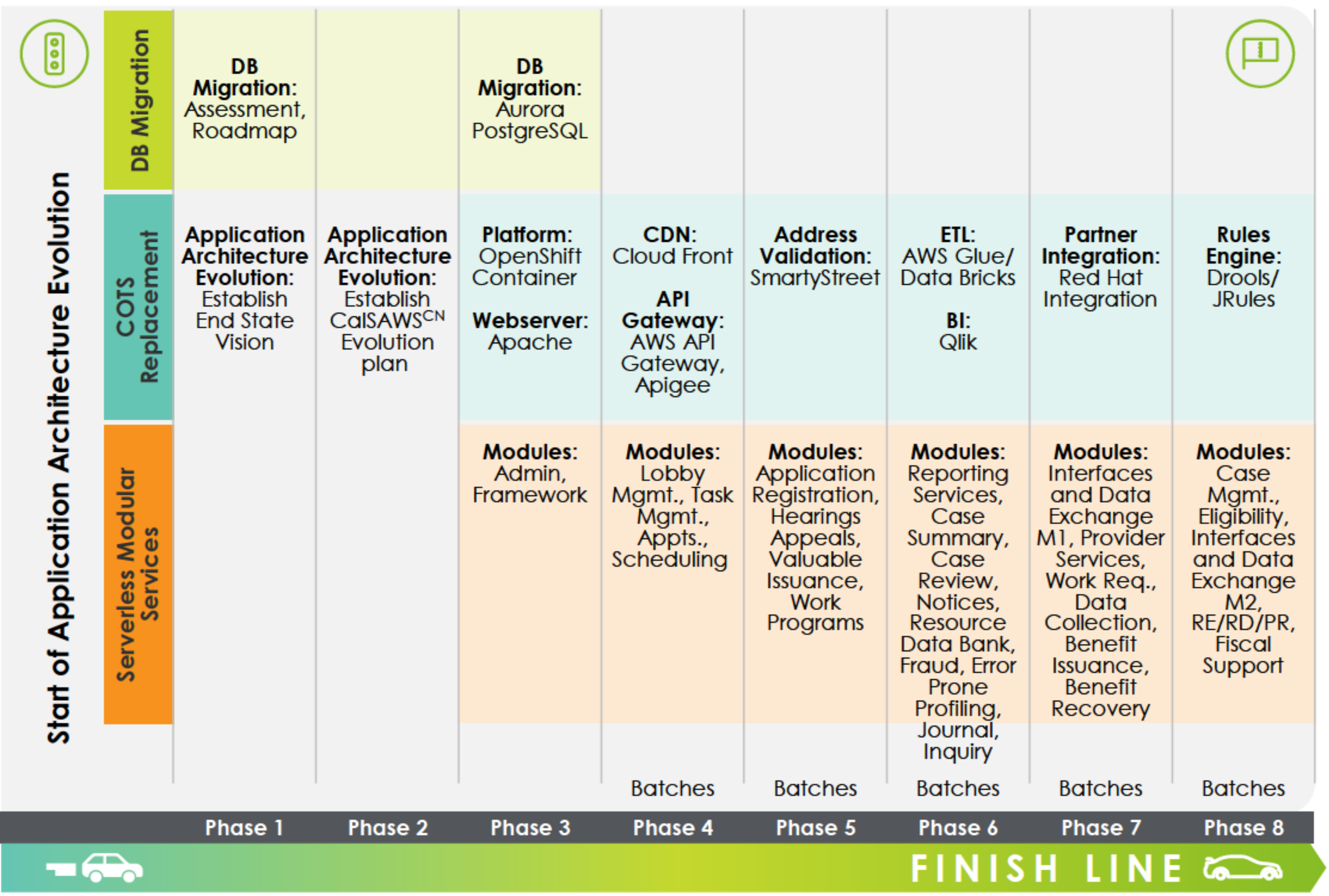


Figure 4.2.1-7. Journey Towards Serverless Cloud Native Adoption.

Table 4.2.1-3 describes the value delivered to the Consortium based on our recommended sequence and phasing. The database migration begins during month 7 of the transition, delivering a cloud native database in 12 months, by month 18 of the contract for increased reliability, increased scalability to support more users, on-demand database scalability for peak usage periods and reduced licensing costs by reducing Oracle database licenses early in the application evolution.

Workstreams	Phase	How We Deliver Value to the Consortium
Application Architecture Evolution – Plan	1	• Faster refactoring which can begin as soon as 6 months after transition starts.
Database Migration – Plan and Releases	1	• Provides a complete and transparent plan for DB migration across impacted stakeholders.
	2, 3	• Completion of the DB migration in 12 months. • Reduction of Oracle database licenses. • Provides elasticity, scalability, and dynamic environment capabilities.
		• Database Migration applies automation for the conversion of database structures and code. Conversion exposes detail of where and how the application uses the database and provides correlation for the planned feature modules. • Introduction of data access layers wherever possible to centralize data responsibilities and begin feature module decoupling
Application Architecture Evolution – COTS Replacement and Modular Services	3	• Establishes serverless platform to replace Oracle Web Server and WebLogic. • Minimizes risk by reviewing monolithic code and deep automated analysis to confirm functional services.
	4	• Provides Task Management changes early and leverages new database. • Establishes a new API Gateway thar is expanded on in later phases.
	5	• Brings in new Address Validation software along with Application Registration.
	6	• Improves the flexibility and consistency of Case Views. • Replaces Oracle BI Publisher with Qlik and ELK leveraging AWS Glue and Lambdas.
	7	• Replaces first half of Oracle SOA. • Migrates first interfaces, mitigating risk.
	8	• Completes Interface refactoring. • Completes Oracle SOA replacement. • Replaces IBM ODM to reduce cost.
Application Architecture Evolution – Batch Modernization	4 to 8	• Addresses batches with performance issues first and most quickly. • Improves processing time and exception management by replacing legacy batch processing with real time processing using event streaming and microbatching increasing the observability and recoverability.
Application Architecture Evolution – UX	4 to 8	• Provide opportunities for County workers to receive improved UX with React interfaces created module by module.

Table 4.2.1-3. Value Derived from the Application / Architecture Evolution Phases.

Prioritizing and Decoupling the Databases

Deloitte prioritizes migrating the database from Oracle to Aurora Postgres as the first step. We propose to migrate CalSAWS databases incrementally with four waves **over a one-year period** (i.e., beginning in month 7 and completing six months after the transition completes) to minimize risk to completion of the migrations and production downtime. Each wave is defined during assessment and planning. We determine which database components can be migrated independently, along with any dependent application changes. Each wave is its own workstream and independently delivered. The waves begin with the introduction of data access layers aligned to domain-driven architecture boundaries to decouple the underlying database physically and logically from the CalSAWS application.

DID YOU KNOW?

- Deloitte's data migration factory leads the nation in successful migration of Oracle to AWS Aurora Postgres for large complex databases.
- Costs are optimized with on-demand provisioning.
- On-demand replicas can offload data read operations for reporting and batch processing.

During this process we expect to create 3 outcomes:

1. Address one of the most common challenges and risks to any application evolution: data entanglement with the business logic by decoupling the application from the database.
2. Improve data quality by analyzing and identifying common data issues like duplicate person records that have not been addressed previously. We will move any data issues to a quarantine that is accessible and recoverable to prevent issues that would impact the users of the system.
3. Align the data access layer to the bounded contexts. We finalize the logical domain data boundaries where the data supports aligned to the bounded contexts. This provides API access and the AWS API gateway to be used by the monolith and then the domain microservices into which CalSAWS will evolve.

We use the bounded contexts for the APIs to organize the four waves of database migration and take care of any reporting integration to not interfere with the priorities for functional changes and enhancements to the application.

Each wave's SDLC **overlaps** (i.e., testing begins on Wave 1 as development begins on Wave 2). Concurrent waves support accelerated development timeframes, with complete validation and no interference between waves.

Applying Tools and Accelerators

Given CalSAWS' large-scale, high-volume databases, the heightened risk due to undocumented code, and the need to avoid negative County impacts or cost extensions from delays, Deloitte uses our production-proven, industrial-strength **Database Migration Factory (DMF) tools.**

DISTINGUISHING FACTORS

- Automated conversions achieve 90+% accuracy and reduced the level of effort by 1/3 over manual approaches
- Launch databases as needed with IaC.
- Hydrate Dev and Test DBs and data using source- controlled IaC and test data.
- Architect databases to be cloneable and sanitized from production.
- Validate batch, API, and partner expectations of migrated data.

As Deloitte has migrated other clients' systems to Platform-as-a-Service (PaaS) databases, including Aurora PostgreSQL, we've learned the limits of the out-of-the-box (OOTB) AWS tools. Accordingly, we have built our own suite of tools to augment and accelerate the database migration process to Aurora PostgreSQL. We use these tools, in addition to **AWS OOTB Database Migration Services** tools, to reduce cycle time and increase the quality of the migration. Our DMF tools are aligned to delivering the six phases of the migration approach described in Figure 4.2.1-8. We have found that the OOTB tools achieve a 65-75% automated conversion leaving the remaining 25-35% of the work to do manually, including building tests and data validations. With our tools, we achieve a 90-95% automated conversion rate, reducing the manual efforts reliably and generating automated tests and data validations to further increase the quality of the migration and decrease the time necessary. For the consortium and the Counties, this decreases the risk of the migration and accelerates the reduction of costly database licenses from Oracle. It also means that we can provide features, enhancements, and performance improvements sooner with less risk due to the elastic nature of the cloud native database and its ability to scale on demand and not get slower when more people need it to be available.

[redacted]

Deloitte Cutover Playbooks

A set of playbooks that leverages our experience and define best practices for database cutover (including production dry runs) and mitigation for known challenges

CA_CON_CalSAWS MandO_MO-379038_2022-0090_02

Figure 4.2.1-8. Deloitte's DMF Toolset.

As shown in Figure 4.2.1-9, work begins with a High-Level Migration Assessment and Setup phase to finalize the overall database migration roadmap and migration development and test environments. We group CalSAWS databases logically into the migration waves. We evaluate application integration, data size, archival requirements, and estimate the time to migrate each database. Each database and each migration will have the same due diligence that we give to production. This will enable us to migrate training and sandbox databases with the same reliability and importance as production. We will plan with the Counties to work with training schedules and team needs. The same way we work to minimize production outages, we will minimize outages in any environment. This level of discipline will provide a smooth production migration with many practices occurring beforehand.

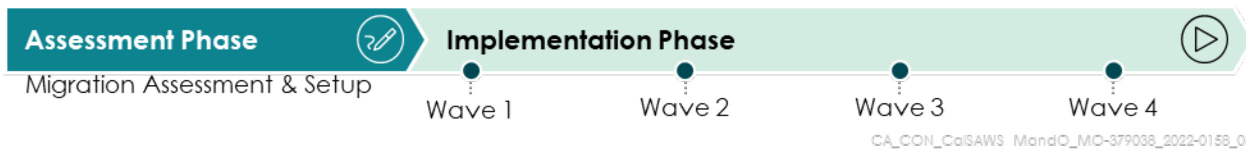


Figure 4.2.1-9. Incremental Migration Plan.

After the high-level assessment, a series of phases and activities are executed as outlined in Figure 4.2.1-10. This is done for each database in each Wave. It begins with a deep-dive assessment to confirm the Wave plan and activities and continues until the cutover phase.

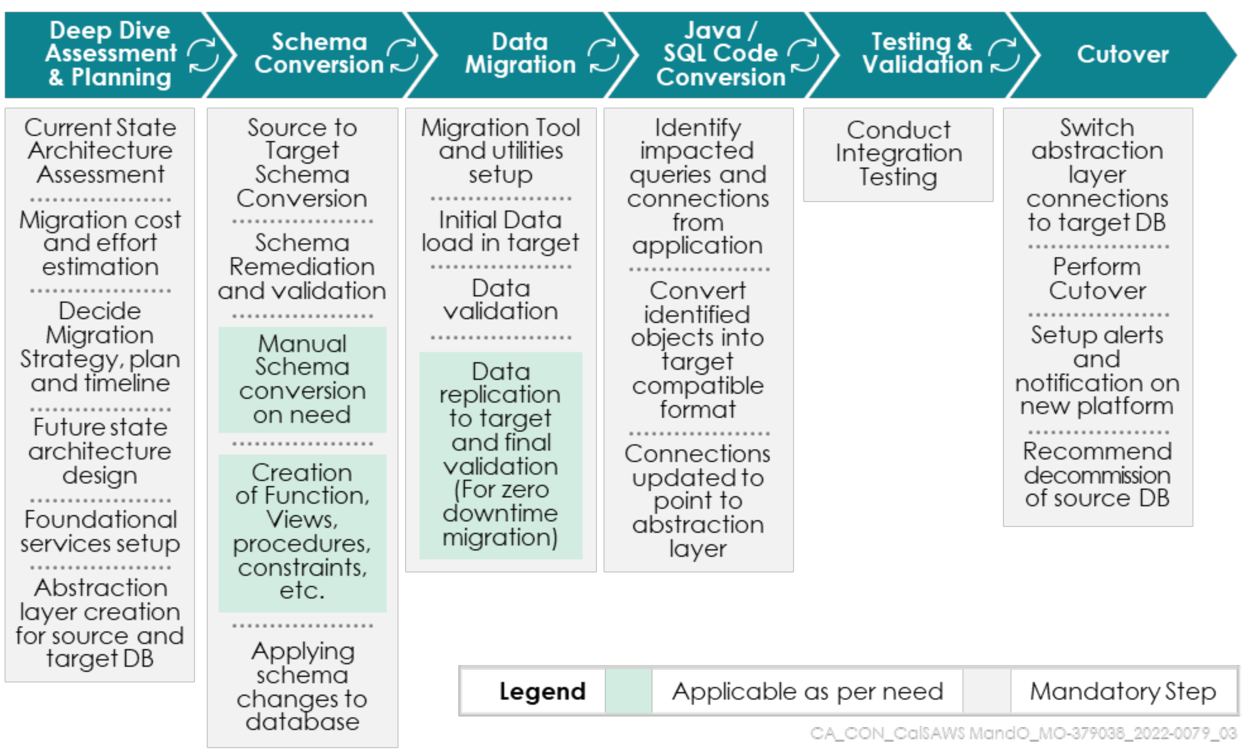


Figure 4.2.1-10. Migration Approach Built on Deloitte's DMF Framework.

Deloitte's database migration approach results in a stable and elastically scalable CalSAWS[CN] databases. Once the databases have been migrated to an available serverless database engine, the Consortium has a baseline database to support rapid modularization and decoupling activities that result in a refactored, cloud native, stable, and scalable CalSAWS[CN]. Additionally, Aurora Postgres will be able to replicate data in minutes. This means we can setup a replica data for testing for modularization activities and then remove it when no longer needed. As we work through the feature module evolution, we can take advantage of this capability to create cleansed test data and test at production volumes. This approach will reduce costs for test databases by only having them when they are needed for testing and allow us to automate test data creation for each of the phases when we break up the monolith reducing overall regression testing time and improving quality.

Refactoring CalSAWS to CalSAWS[CN]

After we migrate the databases and have established a flexible cloud native foundation, we turn to refactoring the application and evolving the architecture along three dimensions:

- Modernizing the CalSAWS Technology Stack
- Modernizing the CalSAWS Monolithic architecture to Microservices
- Modernizing the Functional Features of CalSAWS

For each of these dimensions we are highly opportunistic in identifying SCRs from the CalSAWS backlog that can be addressed. This will provide the Counties and the consortium the ability to prioritize and complete functional changes and enhancements, which may have been previously put on hold, throughout the evolution of the application architecture, modules, and operations.

Modernizing the Technology Stack

Our team works with the Consortium to weigh the tradeoffs across technology, cost, and features as the technology stack is modernized. This includes whether full replacement of a technology makes sense or whether or new feature modules should be built using an existing technology. We evaluate and apply these approaches as we iterate through the CalSAWS technology stack.

In Table 4.2.1-4, we outline the technology stack components that we plan to target for the CalSAWS modernization. This aligns to requirement ME-5.1-03 to move from the existing Oracle stack to a new architecture. Replacing these components enable the Consortium to have a lightweight set of microservices which consume fewer resources.

Software Capability	Current Products	Proposed Product	Replacement Sequence
Database	Oracle RDBMS	• Aurora PostgresSQL	First
Application Server	Oracle WebLogic	• Java CLI • RedHat Runtimes for SpringBoot, Tomcat • Lambda	Iteratively as modules are developed
HTTP Server	Oracle HTTP	• Apache HTTPD	Iteratively with UI updates in React for each module
API Gateway	Oracle API Gateway	• AWS API Gateway & Apigee	With the first phase of modules
Service Bus	Oracle SOA/OSB	• Red Hat Integration or AWS Event Bridge	Iteratively as modules migrate to the new architecture
Rules Engine	IBM ODM	• Drools or alternate cloud native rules engine	Iteratively as modules migrate to the new architecture
BI Tools/ Reports	Oracle BI Suite	• Qlik and ELK	Aligned with reports for each module developed and the purpose of the reporting

Table 4.2.1-4. Technology Stack Modernization Targets.

Modernizing the CalSAWS Monolith to Microservices

Organizing the CalSAWS monolith into smaller, microservice-based feature modules enables us to align business boundaries. This improves the flexibility and resource usage of the system. As we define microservices boundaries for feature modules, we work with the Consortium's technology team to confirm the serverless approach. We will evaluate each module to best apply our two core options: **serverless functions** or **serverless containers**. Table 4.2.1-5 below, outlines criteria used in making the decision. The BenefitsCal Team went through a similar exercise with the Consortium as part of the BenefitsCal architecture design phase.

Criteria	Serverless Function (AWS Lambda or Knative)	Serverless Container (AWS Fargate or ROSA)
Functional/ Use-Case Fit	• Event-driven applications or jobs; Short asynchronous processes • Actions that can be expressed as single functions • Activities with spiky traffic patterns and need dynamic scaling	• Long running compute jobs taking more than 15 minutes • Compute jobs that need more than 3GB of memory • Jobs with predictable scaling or where longer start times are acceptable
Performance	• Lower startup time (up to 5 secs.) • Cold start if no concurrency provisioned • Max runtime of 15 minutes	• Higher startup time – 15-30 seconds • Dedicated resources – warm state • No max runtime

Criteria	Serverless Function (AWS Lambda or Knative)	Serverless Container (AWS Fargate or ROSA)
Scalability	• Auto scale by design • Rapid scaling up possible – ideal for use cases with large spikes / unpredictable traffic and low complexity • Can scale down to 0 compute instances	• Configured for auto scaling • Scale out and scale in happens pod by pod • Cannot scale in / down to 0 compute instances, ideal for use cases with constant demand and medium to high complexity
Pricing/AWS Cost	• Cost calculated per millisecond usage based on number of requests, memory, compute, and network traffic • Price per GB-second	• Cost calculated per second based on allocated memory, vCPUs, OSs, CPU infrastructure, and resources used • Price per vCPU-hour + per GB-hour

Table 4.2.1-5. Serverless Evaluation Criteria for Modularization.

Figure 4.2.1-11 illustrates the application of the evaluation criteria above for select CalSAWS modules.

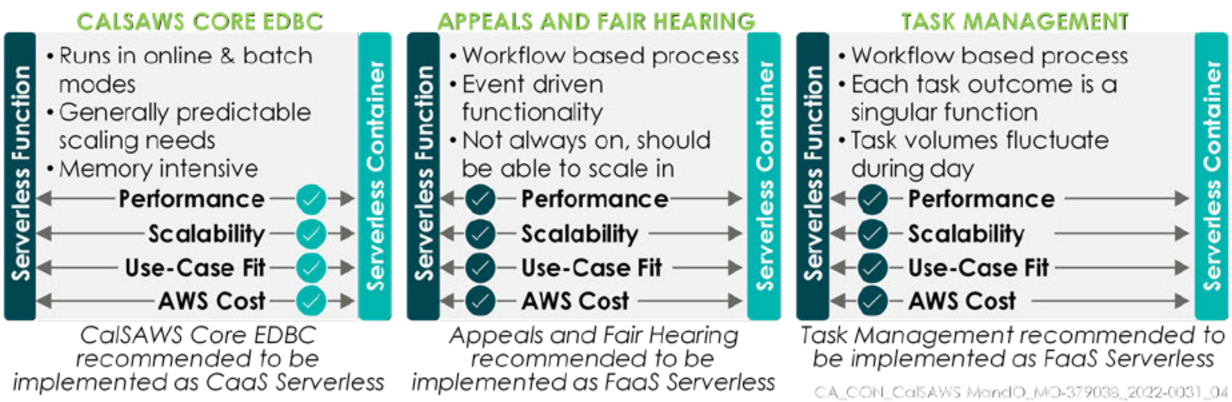


Figure 4.2.1-11. Possible Serverless Destinations for CalSAWS Modules.

Based on our experience with CalSAWS and other E&E systems, and guided by Deloitte's DSDP Reference Architecture, we recommend modularizing the monolith into microservices in the sequence and phases outlined in Figure 4.2.1-12. This sequence is based on our experience with the complexity of E&E systems and on the expected complexity of each of these modules. This sequence helps to spread the work out more evenly as well as order the dependencies that will need to be addressed for each module. During the transition period we review and finalize the schedule and approach with the Consortium as part of the **Approach to Application and Architecture Evolution** deliverable. During planning we confirm the timelines and initial estimates of work to define which work is completed in what timeframe based on the fixed capacity defined for the evolution activities.

[redacted]

Scaling Batch Operations

One of the first priorities in evolving CalSAWS is modernizing its batch processes. This is particularly important given CalSAWS ongoing challenges with completing batch processing on time, and the increased caseloads it will contain once the CalWIN migration is complete. We replace the batch processing architecture with an event-driven architecture using modern cloud native solutions and serverless containers performing real-time processing integrated through batch adapters where necessary to interface with partners.

DID YOU KNOW?

Deloitte implemented an event-driven architecture to replace batch processing at the IRS in 2020 and realized these benefits:

- Processed 8,000+ transactions per second
- Introduced near real-time processing
- Increased flexibility
- Reduced demand on RDBMS resources

The core differences between the CalSAWS' current batch architecture and our proposed event driven architecture are:

- The new architecture gathers data incrementally throughout the day in preparation for creating the batches.
- The database resources usage is reduced by replacing large complex queries.
- The batch processing window is used only for processing activities that cannot be executed prior to the batch window.

This approach provides flexibility, scalability, and higher throughput capabilities for batch processing resulting in reduced timeframes for processing the same volume of activity. The real-time modules will execute on demand and will smooth the database demand throughout the day rather than creating abnormally high demand for batch queries and writes. Additionally, the status of the batch processes that integrate with partners becomes more visible and can be more granular, and processes are more reliable (e.g., less errors and exceptions) and more recoverable (e.g., less data loss or repetition).

Modernizing the Functional Features of CalSAWS

As we modernize the application architecture for the system, we will also separate the presentation layer from the business services, creating both micro-frontends and microservices. We minimize the changes to the user interface to deliver working modules in the shortest time possible.

Once the micro-frontends exist, this approach provides opportunities for the Counties and the Consortium to modernize the business capabilities and use interfaces independently where they are most visible to the users. For example, a key area for business modernization is **Task Management** where significant standardization and improvement in workflow for Counties is needed. It is one of the first areas we transform to increase flexibility and meet the operational needs of the Counties. How we modernize task management is a pattern for how we will be able to modernize user activities across the applications levering HCD. Some tasks as defined today do not require any worker action and are more of an alert. This creates a lot of "noise" and time spent by County workers to determine what they really need to address versus what is just informational.

Our CalWIN ISS Team has been supporting the CalWIN Counties with defining their task configurations. They have had a front row seat to the challenges the current task management functionality has created. Supervisors are challenged with being able to plan for staffing without the right tools. Furthermore, in looking at the task configurations that are available less than half are used by the majority of counties.

Flexible and Optimized Task Management

As per the RFP, Deloitte prepares the requirements and design to reimagine Task Management. Table 4.2.1-6 below highlights key opportunity areas to explore when evolving CalSAWS' Task Management module.

Functional	Operational	Analytics	Architecture
• Ease of configuration (reduce effort, expertise, and errors) • Intelligent, prioritization with case attributes and County operations models	• Work de-duplication • Accounting of work • Automation/closure of tasks • Intelligent work distribution • Lobby management • County configuration flexibility (e.g., large v. small office) • Move to true tasks vs alerts and tasks mixed together	• Worker, office, County, and State-level metrics • Quantitative and qualitative analysis	• Serverless task creation and updates • Serverless next task to improve system resource allocation to consumption model • Optimization, scalability, and security

Table 4.2.1-6. Opportunities to Evaluate and Modernize Task Management.

Our approach for evolving task management aligns with our innovation approach documented in Section 4.4 Innovation. Rooted in HCD principles, it balances equity, stakeholder engagement, and repetitive workload reduction. We bring production-proven task management practices from across states and industries to address the Consortium's needs. Figure 4.2.1-13 provides timelines, high-level activities, and artifacts in our task management redesign approach. The artifacts make up the **Task Management Design and Recommendations Report.**

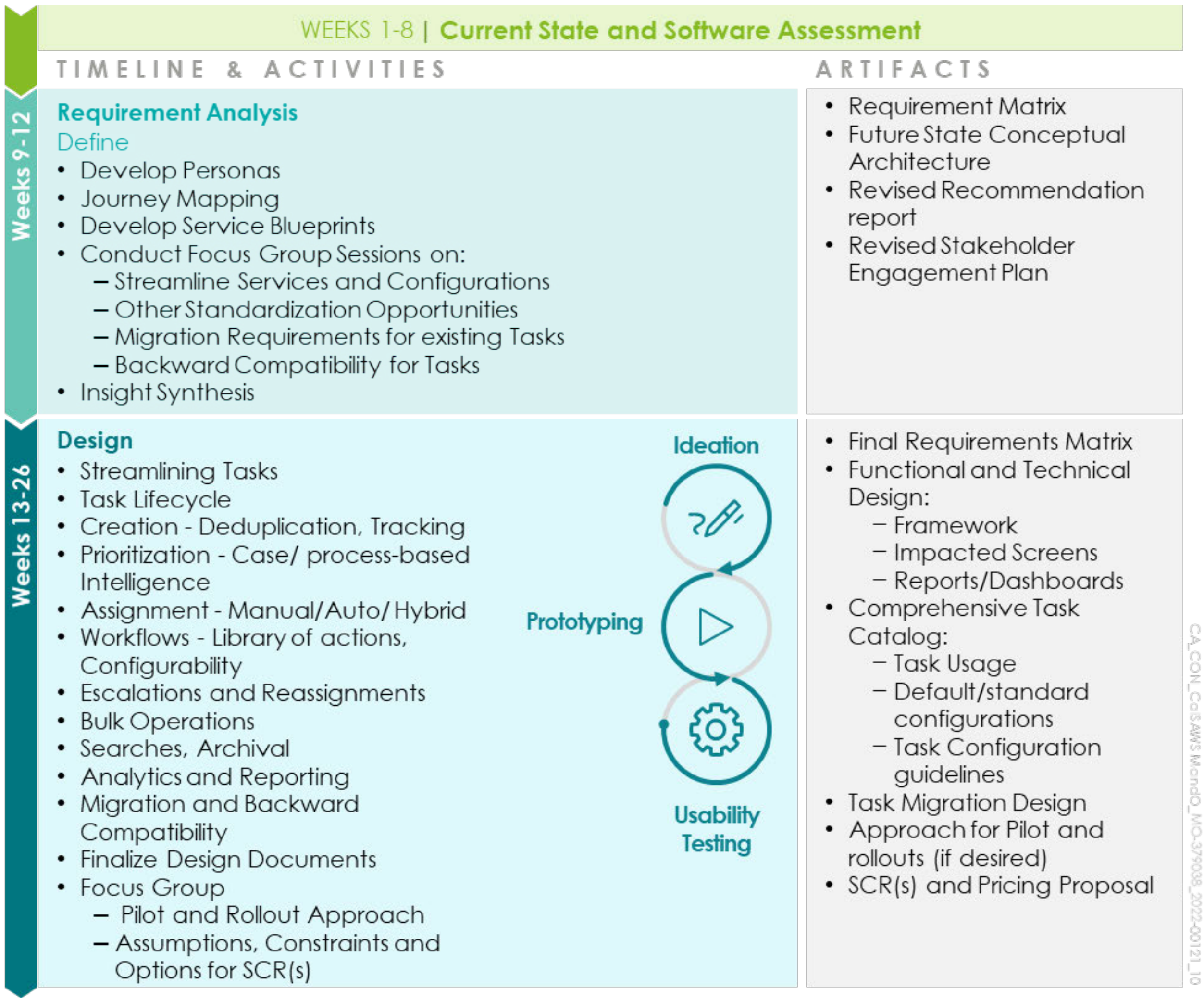


Figure 4.2.1-13. Proposed Plan for Task Management Redesign.

Following the task management redesign, we integrate task management changes with the CalSAWS[CN] modularization development plan and leverage integrated AI components.

Optimizing with Artificial Intelligence (AI) and Automation

Keeping workers engaged and focused on high-value customer facing activities is the top priority of HHS agencies nationwide. To effectively prioritize and engage workers on customer-facing activities, we utilize AI and machine learning (ML) assisted technology, which enhances efficiencies, increases worker capacity, and diminishes errors that hinder advancements toward business outcomes. Figure 4.2.1-14 provides a visual representation of the comprehensive support offered by AI/ML.

AI/ML Enabled Operations – From Assisted Intelligence to Augmented Intelligence

Robotic Process Automation (RPA)	Intelligent OCR (iOCR) Engines	Conversational AI (Chatbots)	Machine Learning (ML) Model Engines
Acts as a "digital worker" in the User Interface (UI) layer to support workers with mundane tasks and mimic worker actions across applications.	Enables automatic reading and extracting of undigitized data (e.g., identifying and reading scanned SSN, applications, forms) and learning for continuous improvement.	Uses natural language processing to conduct a conversation through auditory or textual methods and includes continuous learning and sentiment analysis.	ML models to predict unique conditions and events from available system data and trigger customized alerts to case workers or residents directly for various use cases.

CA_CON_CalSAWS MandO_MO-379038_2022-0032_a

Figure 4.2.1-14. AI/ML Operations Continuum.

To prepare the **Approach to Automation, Artificial Intelligence and Machine Learning deliverable**, we find automation opportunities through HCD techniques. This helps us discover user stories that we discuss with County workers to find scenarios for assisted or augmented intelligence. To speed up this process, [redacted] designed to enhance efficiency, capacity, and quality by reducing errors. These engines support the features in Figure 4.2.1-15.

[redacted]o Evolve Operations

Worker Efficiency	Operations Capacity	Increased Processing Time	Increased Quality Outcomes
Workers can spend more time on tasks that have more impact on the constituents.	Workers can spend more time on tasks that have more impact on the constituents.	RPA, iOCR, and chatbots perform tasks 24/7. The AI Engines enter data, automatically read and extract information, and communicate with constituents when workers are unavailable.	Machine learning to identify error prone processors, and enable remediation at supervisor /manager discretion.

CA_CON_CalSAWS MandO_MO-379038_2022-0032_b

Figure 4.2.1-15. AI Engines That Can Be Considered.

With the Counties and the Consortium, we identify the Deloitte AI Engines to integrate with the task management changes and other parts of the CalSAWS^{CN} workflow to lighten the burden on case workers them to focus on other things by implementing an innovative, AI-driven system with Deloitte. A more intuitive task management capability is easier for new staff to learn.

Table 4.2.1-7 highlights some of the additional benefits our clients have experienced when redesigned task management and AI have been combined, significantly improving the Case Worker administrative workload.

[redacted]

4.2.1.2 Addressing Security Considerations, Reducing Costs and Improving Optimization and Flexibility

Security Considerations

Our strategic vision for security consists of **three cornerstones** connected by our commitment to secure data for Californians and the Consortium on the road to supporting a **Zero-trust architecture**. As part of the application/architecture evolution, we work with the Consortium to prioritize the security considerations strategically.

- **Modernize –** Address security operations through advanced monitoring, cloud native security solutions, and continuous scanning. The increased usage of the cloud platform features reduces the risk of misconfiguration and improves the Consortium's threat monitoring position. Our accelerators include a set of deployment patterns to rapidly implement DevSecOps automation and secure cloud services configuration.

We work with the Consortium to prioritize key capabilities such as Privileged Access Management to reduce the risk of service/non-human accounts and improve the operational security of the solution.

- **Evolve –** Focus on proactively understanding the future needs of the CalSAWS security program. With a **customer-centric focus,** we improve the security posture, leveraging the enhanced authentication and authorization identity platform with password-less authentication, **integrated threat modeling**, security-by-design, architecture reviews, and **continuous application security testing.**
- **Fortify –** This addresses CalSAWS' core resiliency. Fortify activities shield and protect CalSAWS against evolving threats and attacks. Together, we prioritize goals for cyber reconnaissance to identify potential security vulnerabilities from an attacker's perspective**.** We implement Deloitte's Detect and Respond services to focus on advanced threat intelligence, threat hunting, and **application security monitoring**.

Reducing Costs

The application / architecture evolution to serverless delivers cost savings by fully using AWS Cloud pay for usage and value models. The Consortium gains full transparency into the cost and usage of individual components providing optimized consumption manageability. Design considerations must factor in correct patterns, choice of native services, accurate configurations, and expected transaction volumes. Deloitte uses our Financial Operations (FinOps) practice, highlighted Figure 4.2.1-16, to review cloud financial management capabilities, address peaks and valleys in spend, foster mid- and long-range planning.

REMEMBER **WHEN**

- Deloitte optimized cloud spend on BenefitsCal by reviewing service transaction volumes by month and across environments.
- Deloitte helped the Consortium reduce AWS costs for BenefitsCal by 65% per month.

FinOps Foundation

Our FinOps evangelists passionately enable education and continuous improvement. Their most recent contribution is to the FinOps's Foundation's US Public Sector Playbook, a trusted guide for Government Cloud Engineering teams.

CA_CON_CalSAWS MandO_MO-379038_2022-0162

Figure 4.2.1-16. Driving FinOps Discipline Nationwide.

Improve Optimization, Flexibility, Scalability

The new CalSAWS^{CN} system architecture optimizes resource usage based on actual user need rather than large, static allocations for peak resource demand. The new CalSAWS^{CN} also takes advantage of the elastic nature of the cloud, enabling scalability for each module to grow and shrink, releasing resources when not needed. This reduces manual processes and the associated delays in legacy architectures like the current CalSAWS. CalSAWS^{CN} enables resources to be rapidly provisioned in real-time, even for unforeseen peak loads and disproportionate growth such as during natural disasters, seasonal influx, and open enrollment. Lastly, flexibility increases with the ability to deploy individual serverless feature modules maintained, updated, or changed independently. This reduces the time needed to test changes as they are more contained. Features can be added independently and deployed frequently without creating downtime or impacting system performance. This enables the Consortium to deliver more SCRs and deliver them more quickly.