PIs and affiliations: Aaron Weiskittel, Professor of Forest Biometrics and Director, Center for Research on Sustainable Forests, University of Maine; Director, NSF Center for Advanced Forestry Systems

Kasey Legaard, Research Associate of Geospatial Analytics and Machine Learning, Center for Research on Sustainable Forests, University of Maine

Kenneth Bundy, Software and Machine Learning Engineer, Center for Research on Sustainable Forests, University of Maine

Michael Premer, Assistant Professor of Forest Management, University of Maine

Collaborators: Jimm Domingo, Software Engineer, Center for Research on Sustainable Forests, University of Maine

Jereme Frank, Forest Biometrician and Analyst, Maine Forest Service

Phil Radtke, Associate Professor of Forest Biometrics, Virginia Tech University

Jim Westfall, Research Forester, USFS Forest Inventory and Analysis, Northern Research Station

Project Title: Cloud-based small area estimation based on fast, on-demand processing of large-area data sets and mid- to high-resolution geospatial auxiliary remote sensing

Period for report: July 1, 2024 to Dec. 31, 2024

Progress: The goal of this project is to support collaborative development and application of SAE methods by prototyping a cloud-based system that connects users to data, algorithms, and computing resources. System design considerations simultaneously address the needs of research scientists, data analysts, and data end-users. A high level of interdependence among design objectives has necessitated parallel development of various software subsystems. We have adopted a hierarchical structure to guide architectural design, formalizing the requirements of individual software components and interactions between components. Progress during this period has touched on all major software subsystems (Figure 1), but with particular focus on the job management subsystem, isolated runtime environments, and client-facing and administrative interfaces.

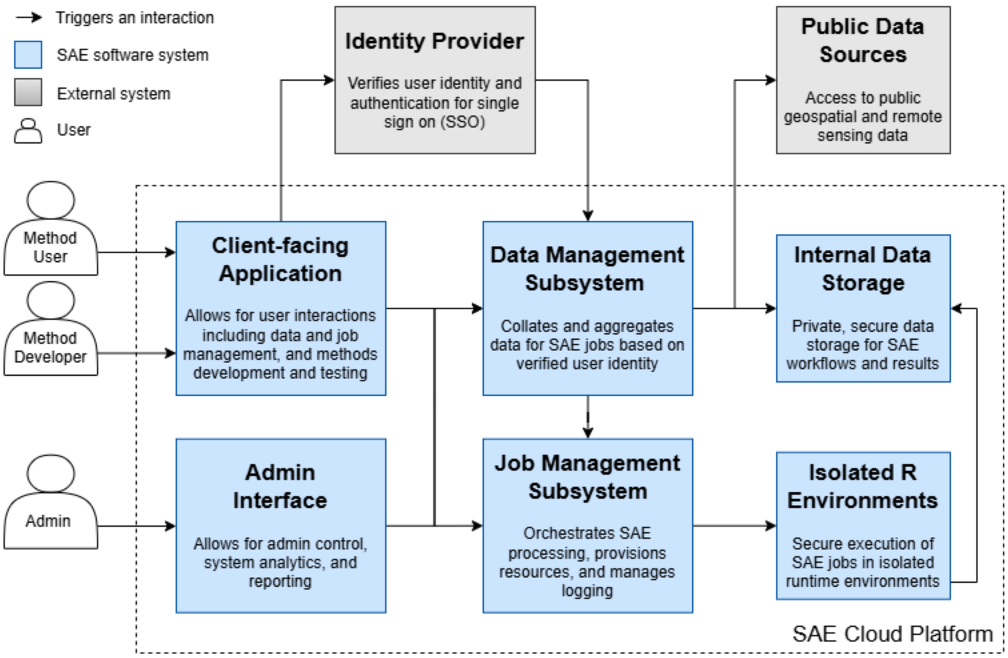
Triggers an interaction
SAE software system
External system
User
Identity Provider
Verifies user identity and authentication for single sign on (SSO)
Public Data Sources
Access to public geospatial and remote sensing data
Method User
Method Developer
Admin
Client-facing Application
Allows for user interactions including data and job management, and methods development and testing
Data Management Subsystem
Collates and aggregates data for SAE jobs based on verified user identity
Internal Data Storage
Private, secure data storage for SAE workflows and results
Admin Interface
Allows for admin control, system analytics, and reporting
Job Management Subsystem
Orchestrates SAE processing, provisions resources, and manages logging
Isolated R Environments
Secure execution of SAE jobs in isolated runtime environments
SAE Cloud Platform

Figure 2: Primary SAE platform subsystems (blue) and key external systems (grey).

The job management subsystem orchestrates and provisions resources for SAE workflows. Workflows are implemented as processing pipelines that initiate input data extraction and collation, and link processing functions as separate blocks of code. We have defined pipelines using simple, human readable and editable text files in the YAML format. This provides transparency and close control over pipeline functionality, and a simple basis for saving, editing, and sharing pipeline definitions amongst users. We have made substantial progress prototyping pipeline orchestration software to validate and execute pipelines on the cloud.

Pipelines themselves are composed of separate blocks of code that implement data handling or computational functions. We plan to provide a set of pre-built blocks constructed around existing R functions required to deploy a broad set of established SAE methods, including those commonly utilized by FIA (e.g., within the FIESTA package and web application). Additionally, our system will support user-contributed blocks written in either R or Python. We intend to make construction of compatible code blocks as simple as possible, and progress made during this period has included prototyping of utility software to support the development of user-contributed blocks. Prototype code has been written in Python but will be ported to R to enable user development in either language.

Progress during this period included extensive investigation of alternative approaches to managing blocks and their software dependencies. We broadly considered the relative strengths and weaknesses of two different architectural frameworks in which 1) individual blocks are containerized with their specific dependencies, and 2) the pipeline is containerized but blocks are managed within that container using process isolation. Both approaches have distinct advantages and disadvantages, warranting careful consideration and a certain level of prototyping. We have since committed to the framework based on a single pipeline container. There are several compelling advantages to this approach, including simplification of code development by users, particularly for those who are less comfortable working with software containers. SAE pipelines will be executed within secure, fully isolated containers (i.e., no network access at runtime) that provide temporary instances of a runtime environment that includes all software resources and dependencies needed for pipeline execution. We have prototyped Docker containers that include full R and SQL integration, plus relevant geospatial packages and software utilities. Progress has been made with the integration of relevant Python dependencies. Our system's containerized runtime environment will be shared for user development. Any code developed and tested within that environment will be compatible with our cloud platform.

User-facing and administrative interfaces have been an additional focus of recent progress, with particular attention placed on the exploration of ideas for a graphical interface to facilitate no-code construction of new pipelines or customization of existing pipelines. We are prototyping a simple web-based interface that provides functionality similar to the construction of a flowchart. Once we have more fully evaluated the feasibility of this interface design, we will expand its feature set and refine its functionality with iterative feedback from project collaborators.

Lastly, we have made additional progress on the design of backend services to support a multi-tenant data management subsystem, which will collate and aggregate data for SAE jobs based on verified user identity. We have evaluated different technologies and third-party services, and have prototyped a backend based on Supabase, an open-source project which offers storage, real-time data management and database services, and user management services that integrate third-party authentication through providers like Google or Apple.

Next period plans: Parallel development of all major subsystems will proceed through the next reporting period, but with particular focus on the user interface and connectivity to the data management subsystem. Our goal is to provide meaningful demonstrations of multiple aspects of system functionality by the end of the next period.