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

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

Overview: The goal of this effort is to support collaborative development and application of SAE methods through an accessible, secure, and efficient cloud-based system that seamlessly connects users to data, algorithms, and computing resources (Figure 1). Design considerations will address the needs of scientists, analysts, and data end-users. Supporting objectives include: (1) Developing and hosting a cloud-based SAE system for collaborative R&D and data analysis by parties invested in the Partnership for Small Area Estimation (PSAE); (2) Closed prototyping, testing, and verification of all software subsystems to ensure the security of proprietary or restricted resources; (3) Online system deployment using non-sensitive data sets for further evaluation and testing.

System design and implementation will include: (a) Analysis tools and workflows to model inventory attributes from auxiliary data with support for very large data sets (i.e., “big data”); (b) Integration of 3D LiDAR and photogrammetric point cloud processing supported by fast, scalable cloud computing; (c) Support for secure hosting of proprietary or restricted inventory and auxiliary data; (d) Integration of user-supplied SAE algorithm implementations and tooling to support local algorithm development; (e) Opt-in and selective sharing of data and algorithms between individuals and organizations; (f) Authentication, runtime isolation, and other measures to ensure secure execution of SAE workflows on the cloud; (g) User and administrative interfaces including no-code customization of SAE workflows and tools to control access to data and services.

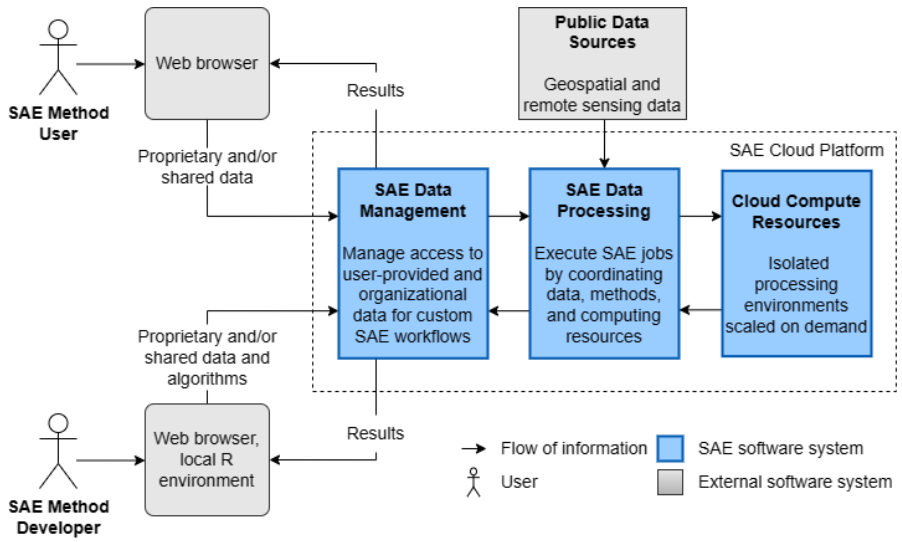

Figure 1: Overview of the SAE cloud platform including primary user types.

Progress for reporting period (February 1 - July 31, 2024): A major development focus of this reporting period has been the data processing subsystem, which orchestrates and provisions resources for SAE workflows. SAE jobs will be executed within secure, fully isolated, and temporary instances of a runtime environment (i.e., no network access at runtime) that includes all resources and dependencies required to deploy a broad set of SAE methods, including those implemented by FIA (e.g., within the FIESTA package). We have prototyped R and Python runtime environments, including relevant geospatial packages, common software utilities, and full SQL integration. We conceptualize SAE workflows as processing pipelines that initiate input data extraction and collation, and link processing functions as separate blocks of code. We have developed a prototype pipeline daemon (a process to launch blocks in succession) in Python. This may eventually be supplanted by a daemon written in another language like Rust, which offers faster processing, lower memory consumption and, potentially, improved security.

Further progress related to the processing subsystem has primarily focused on complexities pertaining to the management of blocks and their dependencies. We have broadly considered the relative strengths and weaknesses of subsystem architectural frameworks in which 1) individual blocks are containerized with their specific dependencies, or 2) the pipeline is containerized but blocks are managed within that container using process isolation, or sandboxing. Both approaches have distinct advantages. A container-based approach offers excellent security and will simplify system development and maintenance through language-independence, simple block registration and management, and ample opportunity to leverage existing tools. However, the execution of workflows may be slower with greater disk usage, and additional development of tooling will be required to support local development by users who are not comfortable building their own containers. On the other hand, an architectural framework based on process isolation within a single pipeline container could simplify tooling for local methods development, but at the possible cost of increased overall complexity of pipeline process management and system development. Overall, the tradeoffs associated with these different approaches are substantial enough to warrant further assessment and prototyping. It may also be the case that further development will reveal good justification to mix these approaches.

Our development team has further investigated options for the implementation of various components of the data provider subsystem, which collates and aggregates data for SAE jobs based on verified user identity. We have evaluated a number of different technologies and third-party services, with particular emphasis placed on identity verification and security. Related to this, we have evaluated different options for back-end services associated with user-facing applications and interfaces. We have found significant advantages to the use of a backend-as-service provider, specifically 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. We have made significant progress prototyping a Supabase backend for a generic project that shares the critical design objectives of this PSAE project.

Next period plans: During the coming months, we will continue development of the data processing subsystem, specifically focusing on further evaluation and probable full prototyping of a container-based framework for pipeline and block management, including tooling to support local methods development. We will more fully evaluate backend-as-service providers as a foundation for the data provider subsystem, and prototype custom tooling needed to connect data provider and data processing subsystems. Contingent upon progress on these matters, we will shift attention to front-end development of user interfaces. We will continue to hold regular meetings with collaborators and look for alignment with related projects (e.g., FIESTA) through continued dialog with their leadership. We have requested participation in PSAE development group meetings, and plan to participate in the 2024 FIA Science Symposium.