SN Computer Science (2026) 7:281
https://doi.org/10.1007/s42979-026-04861-5

ORIGINAL RESEARCH

Building Collaborative Intelligence: The Translational Journey of the Smart Amplified Group Environment Across Research, Education, and Practice

Jason Leigh[1] · Luc Renambot[2] · Ryan Theriot[1] · Lance Long[2] · Christopher Lee[1] · Jesse Harden[4] · Nurit Kirshenbaum[1] · Mahdi Belcaid[1] · Chris North[3] · Andrew Johnson[2] · Maxine Brown[2]

Received: 23 September 2025 / Accepted: 25 February 2026

Abstract

The SAGE Suite–SAGE1, SAGE2, and SAGE3–translates advances in visualization, cyberinfrastructure, and human-computer interaction into an open, scalable platform that aligns with embodied cognition to support collaborative, spatial reasoning on large displays and personal devices. Over two decades and hundreds of deployed walls worldwide, SAGE has enabled scientists, educators, and students to juxtapose heterogeneous media, sustain shared context, and accelerate sensemaking across the research lifecycle. This paper contributes: (1) a synthesis of the Suite's translational impact across domains–from biology and atmospheric science to disaster management, health care, public outreach and workforce development; (2) a comparative framing of SAGE3 (the Smart Amplified Group Environment) among Computer Supported Cooperative Work and infinite-canvas tools; (3) the design rationale and user experience foundations of SAGE3's "spatial thinking operating system," including boards, rooms, wall viewports, and multi-user attention/flow mechanisms; (4) a modular architecture that delivers low-latency synchronization, extensibility via plugins, and privacy-aware deployment; and (5) a paradigm for human–Artificial Intelligence (AI) collaboration that spatializes notebooks and conversational workflows, enabling multi-user, multi-AI interaction grounded in shared visual context. We also surface systemic challenges in recognizing software-as-instrument within academic incentives and document emergent usage patterns spanning synchronous/asynchronous, co-located/distributed work. SAGE3 demonstrates how open, research-driven cyberinfrastructure can couple spatial cognition with collective intelligence to advance scientific collaboration and decision-making.

Keywords Cyberinfrastructure · Collaboration · Computer supported cooperative work · Artificial intelligence · Human computer interaction · Embodied cognition

Introduction

The SAGE Suite–comprising three generations of software (SAGE: Scalable Adaptive Graphics Environment, SAGE2: Scalable *Amplified Group* Environment, and SAGE3: *Smart* Amplified Group Environment)–serves as a powerful lens, bringing the outputs of modern, computationally enhanced science into sharp focus. Grounded in principles of embodied cognition, the suite leverages spatial organization and large-scale displays to align with how humans naturally perceive relationships, retain context, and reason about information in space. Designed to enable scientists, researchers, and students to collaborate with both colleagues and data using scalable tiled display walls, the SAGE Suite supports the sharing of information and digital media–particularly large-scale visualizations and animations–to accelerate discovery,

Jason Leigh, Luc Renambot, Ryan Theriot, Lance Long, Christopher Lee, Jesse Harden, Nurit Kirshenbaum, Mahdi Belcaid, Chris North, Andrew Johnson, and Maxine Brown have contributed equally to this work.

Extended author information available on the last page of the article

Published online: 17 March 2026

build consensus, and make decisions with greater speed, accuracy, comprehensiveness and confidence [1].

SAGE3 (Fig. 1) is a complete redesign and re-implementation of it predecessors, providing its users with significantly improved versions of prior capabilities as well as new capabilities in response to evolving needs and computing advances. SAGE3 enables multiple users to simultaneously drag and drop information - e.g., numeric data files, images, movies, PDFs, and web pages - from their laptops onto an infinitely sized, shared canvas where it can be interactively juxtaposed with other related information. SAGE3's collaboration features enable multiple users to simultaneously point to and interact with this digital media on the canvas, and compare and contrast particular features. Multiple users can simultaneously share their laptop screens to add additional information from active applications. SAGE3's asynchronous and synchronous collaboration capabilities enable researchers to seamlessly transition from individual work to group work. SAGE3's AI capabilities enable users to bring in raw data and leverage modern Data Science and AI libraries to perform analysis and create new visualizations in real time.

Over more than two decades, the three SAGEs have primarily been translational science projects. Translational science can be broadly described as the process of taking discoveries and methods from basic research and transforming them into practical tools, technologies, and systems that deliver immediate benefits to user communities. In the context of SAGE, this means bridging fundamental research in visualization, distributed systems, and human-computer interaction with the engineering necessary to create robust, scalable platforms for scientific collaboration.

The first SAGE project launched in 2004 with funding from the National Science Foundation's (NSF) Information Technology Research (ITR) program and has continued to receive support for more than 20 years from the Office of Advanced Cyberinfrastructure (OAC). As a translational effort, its success has been measured primarily by adoption and use by end-users.

Translational science faces unique challenges in traditional academic settings, particularly when it comes to recognition [2]. Scientific software like the SAGE Suite plays a crucial role in *enabling* modern science, yet its impact is often underrepresented in conventional publication venues. Such software embodies not only significant research contributions but also the substantial engineering needed to transform foundational ideas into reliable, widely used tools. Despite this, domain scientists who benefit from SAGE often view it as an instrument – much like a beaker or a telescope – critical to discovery but not seen as advancing disciplinary knowledge in its own right. As a result, its contributions, like those of many scientific software systems, are seldom formally acknowledged.

This lack of recognition creates two key problems. First, it discourages the publication of detailed accounts of software development, limiting the broader community's understanding of the complexities involved in architecting, building, and sustaining these systems. Second, it can dissuade talented developers from entering or staying in the field, as the work is too often dismissed as "just engineering" rather than as a cornerstone of scientific progress.

This paper highlights the multifaceted contributions of the SAGE Suite, revealing the expertise and innovation required to develop scientific software and demonstrating the impact of its deployment across diverse research

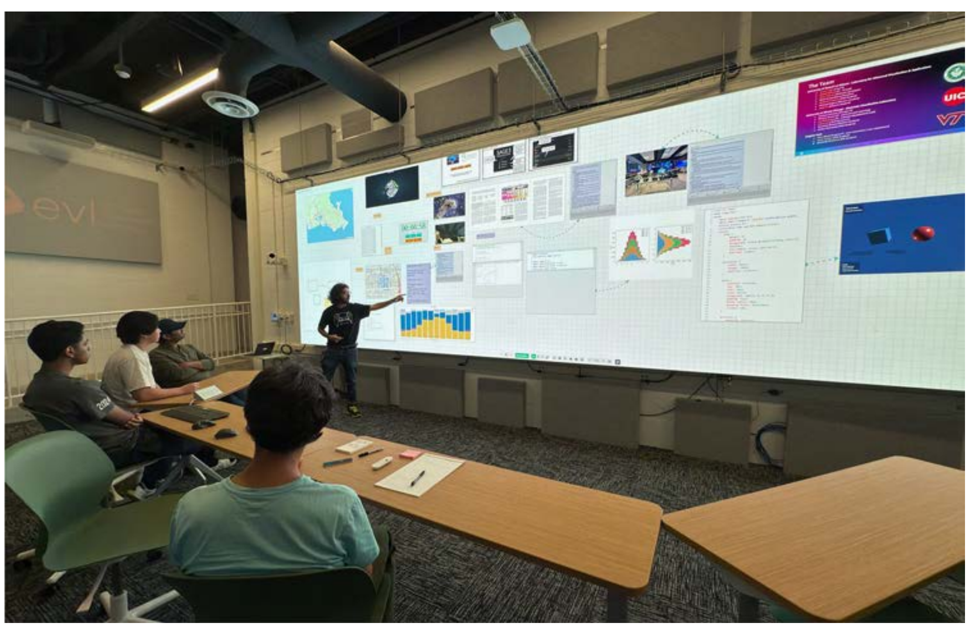

Fig. 1 Public demonstration of the AI features of SAGE3, in the Arcade visualization space at the University of Illinois Chicago - Electronic Visualization Laboratory. (NSF-funded grant #2320261, MRI: Track 2 Acquisition of data observation and computation collaboratory)

communities. The paper is organized into two main parts. The first examines the impact of the three generations of SAGE systems, illustrating the range of problems they have helped address across disciplines. The second focuses on SAGE3, beginning with a comparison to other computer-supported collaborative work (CSCW) technologies, then detailing the design decisions behind its user interface, the integration of AI to support embodied cognition, and its system architecture, and concluding with an analysis of common usage patterns within the SAGE community.

In the remainder of this paper, we refer to the SAGE Suite collectively as SAGE, and to each generation individually as SAGE1, SAGE2, and SAGE3.

SAGE's Impact

Since 2004, we estimate that more than 800 SAGE-driven display walls have been deployed in over 18 countries, representing a hardware infrastructure investment of over $120 million. These systems have supported users in a wide variety of fields – not limited to Computational Science disciplines – including agriculture, aquaculture, archaeology, architecture, art, atmospheric science, biology, bioinformatics, chemistry, cybersecurity, civil engineering, communication, computer science, creative media, education, geoscience, health science, library science, medical sciences, meteorology, network engineering, neuroscience, physics, psychology, workforce development, and statistics. During the five years of SAGE2 (the Scalable *Amplified Group* Environment), we recorded approximately 1,600 users, and in the two years since the release of SAGE3 in late 2022, that number has risen to 6,400 users due to users who now use SAGE3 on their laptops as well as on display walls. Figure 2 show the organizations that attended the 2025 SAGE3 Summit.

Figure 3 illustrates the range of tasks historically associated with SAGE use (Conceptualization (C), Data Collection (W), Data Analysis and Visualization (V), Knowledge Crystallization (X), and Knowledge Presentation (P)) within the Computational Science research, development, and training enterprise. Note that the icons represent: group work, individual work, cyberinfrastructure use, visual analysis, interactivity, intensive writing, and intensive coding. *Group Work* represents work that can benefit from collaborations among group members. E.g., when ideating, scientists may sit together and display possible directions for investigation. Or, in the interdisciplinary analysis phase, group members may meet to present individual progress in the project and form an overarching narrative of the cumulative progress, which then fuels further rounds of analysis. *Individual Work* represents work performed by a scientist, independent of their group, either on their personal computer or a display wall. E.g., during product development, an individual may require concentration to write large amounts of text for a publication, which can later be refined in a group setting. *Cyberinfrastructure* represents stages that may require considerable cyberinfrastructure facilities to complete, such as High Performance Computing clusters. Depending on the size of the data collected and the complexity of analysis, scientists need to have advanced infrastructure locally. This icon is included in data gathering and cleaning, analysis and refinement, as well as for presentation modes that are interactive, since all these need to access data and potentially manipulate it.

The following sections present examples of how a number of disciplines have applied SAGE over time across these categories of use.

Biology (VP)— One of the earliest adopters of SAGE1 was the National Center for Microscopy and Imaging Research (NCMIR) at the University of California San Diego, which in 2004 built the BioWall tiled display to analyze extremely high-resolution microscopy images captured by NCMIR's laser scanning multiphoton and confocal microscopes. By allowing scientists to zoom, pan, and navigate interactively through details far beyond the limits of desktop systems, SAGE1 effectively served as their microscope lens for high-resolution image analysis and biological discovery (Fig. 4).

Atmospheric Science (VXP)— In 2005, the University of Michigan's Atmospheric Sciences Department used SAGE1 to replace its traditional paper-based simulation pin-up board with animated computer graphics that displayed multiple synchronized weather simulations side-by-side for easy comparison. The display wall was installed in a commons area, allowing users to view the latest results each day – even over coffee or lunch – with the goal of stimulating discussion and collaboration among the team (Fig. 5).

Polar Research (WVXP) — In 2011, SAGE2 was used to support the planning, validation, and analysis of large-scale scientific expeditions. The NASA ENDURANCE (Environmentally Non-disturbing Under-ice Robotic Antarctic Explorer) project [3, 4] deployed a robotic submersible under the ice of Lake Bonney in Antarctica's Taylor Valley over two seasons to collect biochemical, photographic, and sonar data about the lake and the adjacent Taylor Glacier. Researchers used CAVE2[5] running SAGE2 for multiple stages of the mission: planning the expedition with satellite imagery, validating data through integrated tables, graphs, and maps, and analyzing environmental patterns by combining satellite photos with underwater images (Fig. 6). SAGE2 also supported comparison of complex sonar datasets, enabling scientists to test parameter settings and refine models of the lake's structure to create a new bathymetric map

Fig. 2 Organizations that attended the 2025 SAGE Summit

SAGE3 Summit 2025 Attendees

UNITED STATES
American Museum of Natural History
Argonne National Laboratory
Assets High School
Baylor University
Boise State University
California State University, Northridge
California State University, San Bernardino
Case Western Reserve University
Central Wyoming College
Chaminade University of Honolulu
Clemson University
FishEye Software, Inc.
Florida A&M University
Hawaii Community College
Hawaii Office of Recovery and Resiliency
Hawaii State Judiciary
Indiana University
James Madison University
Kauai Community College
Lewis University
Maui Institute of Art and Technology
Montclair State University
MultiVis LLC
NASA
National Institute of Standards and Technology
National Science Foundation
National Weather Service
Network Startup Resource Center
Northwestern University
NYSERNet
Oceanit
Oregon State University
Pacific Business News
Pacific Interface
Purdue University
Qlik
San Diego Community College District (SDCCD)
Texas Advanced Computing Center
Universities Space Research Association
University of Arkansas at Little Rock
University of California, Merced
University of California, Riverside
University of California, San Diego
University of Chicago
University of Cincinnati
University of Hawaii (4)
University of Illinois Chicago (9)
University of Illinois Urbana-Champaign, NCSA
University of Utah
University of Wyoming
Virginia Tech – Applied Research Corporation (VT-ARC)
Woodwell Climate Research Center

AUSTRALIA
RMIT University
Western Sydney University

BRAZIL
RNP
Universidade Federal de São Paulo

CAMBODIA
Institute of Technology of Cambodia

CANADA
Université de Sherbrooke
Universite Laval

CZECH REPUBLIC
CESNET
Masaryk University

ENGLAND
Rolls-Royce PLC

GERMANY
Rhein Main University of Applied Sciences

GUATEMALA
Universidad Francisco Marroquín

INDONESIA
Universitas YARSI

JAPAN
Professional Univ. of Information and Management for Innovation

NEW ZEALAND
Victoria University of Wellington

SAUDI ARABIA
King Abdullah University of Science and Technology

SOUTH KOREA
Gwangju Institute of Science and Technology

SPAIN
Instituto de Astrofisica de Canarias

SWEDEN
Stiftelsen Folkets Hubb (Sweden)

TAIWAN
National Center for High-Performance Computing (NCHC)
National Yang Ming Chiao Tung University
Thammasat University

THAILAND
Chulalongkorn University
Mahidol University
Walailak University

Educational Institutions, State, National or International Government Research Organizations, Not-for-Profits, Businesses

of the bottom of the ice covered lake. By allowing simultaneous visualization of diverse data types at both overview and detail scales, the system helped externalize mental models of the lake's biochemical processes and fostered shared understanding across the interdisciplinary research team of geoscientists, computer scientists, roboticists, and biologists. Based on the success of the ENDURANCE project, SAGE2 was also used to help plan the successor SIMPLE (Sub-ice Investigation of Marine and PLanetary-analog Ecosystems) mission to the Antarctic, where the team sent a different robotic submersible out under the Ross Ice Shelf to compare data taken from below the ice to data captured from an airplane flying above the ice, to better understand how to calibrate data captured from the air.

Disaster Management (CVXP) - In 2020, SAGE2 was used to support disaster management, where rapid access to integrated data is critical for decision-making. The LandSAGE project [6] used SAGE2 to combine real-time sensor data and high-resolution visualizations for monitoring and mitigating landslides, mudflows, and floods in Southeast Asia. The project was a multi-national collaboration involving the National Institute of Advanced Industrial Science and Technology (AIST), Japan; the National University of Laos; Walailak University, Mahidol University, and Thammasat University in Thailand; the FIMO Center at the University of Engineering and Technology, Vietnam National University; the Institute of Technology of Cambodia; and the University of Hawaii at Manoa. With funding from the Trans-Eurasia Information Network program (www.tein.asia), developers of LandSAGE built display walls at each of their institutions and used SAGE2 both in the conceptualization and the eventual development of the LandSAGE tool (Fig. 7).

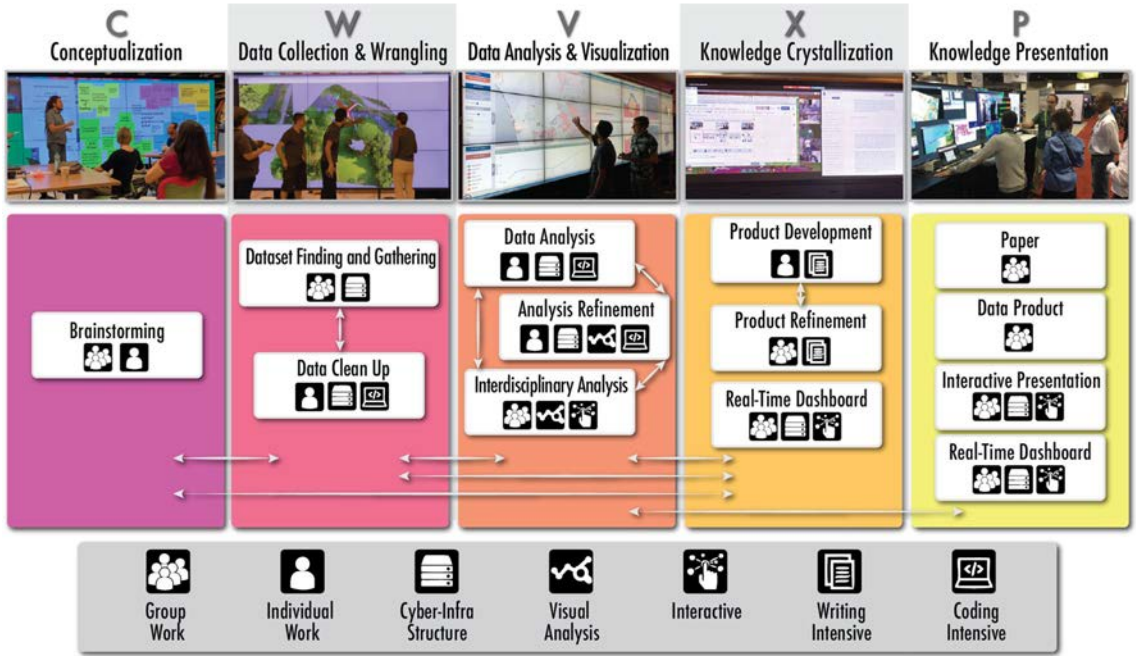

Fig. 3 Illustration of how SAGE supports the computational science research, development and training enterprise

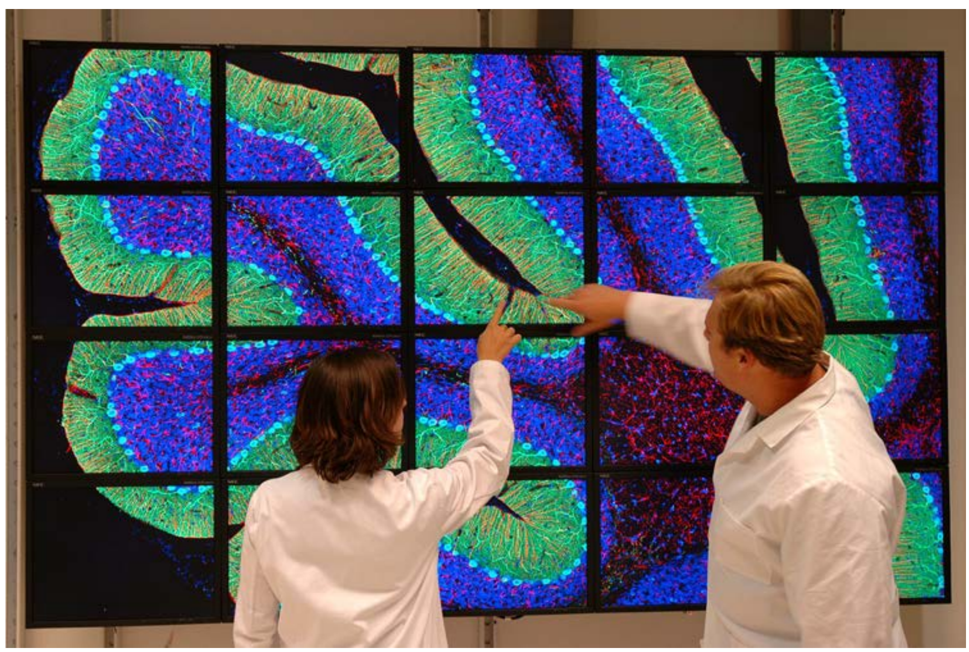

Fig. 4 SAGE1 showing high resolution microscopy images at the National Center for Microscopy and Imaging Research (2004)

Health Science (VXP) — In 2017, SAGE2 was used to improve clinical workflows and patient safety [7]. At the University of Illinois Hospital, researchers and clinicians used SAGE2 to develop a novel interactive visualization system for Intensive Care Unit (ICU) nurses to manage treatment plans and visualize patient test results during shift handoffs (Fig. 8). It is critically important for patient care continuity that nurses going off-shift communicate current patient status and the effectiveness of the patient's current care plan over the past several shifts to the nurses coming on-shift so they will be able to make appropriate decisions. SAGE2 improved the development process of this new tool by making it easy for the interdisciplinary development team of computer scientists and nurse practitioners to compare, discuss, and iterate through various design prototypes, showing alternative ways to display and interact with care plans, and visualize patient test results to better understand how effective those care plans have been. SAGE2 also allowed the team to easily compare results from the user

Fig. 5 Atmospheric science simulations at the University of Michigan, Global Change Laboratory. Left shows the paper-based board that was eventually replaced by the display wall running SAGE1 (2005)

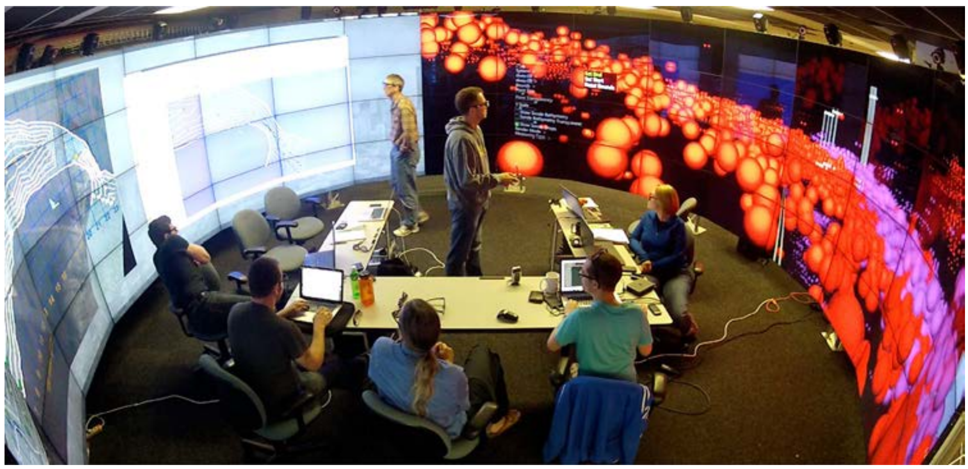

Fig. 6 SAGE2 supporting a NASA ENDURANCE project meeting in the CAVE2 Hybrid Reality Environment (2011)

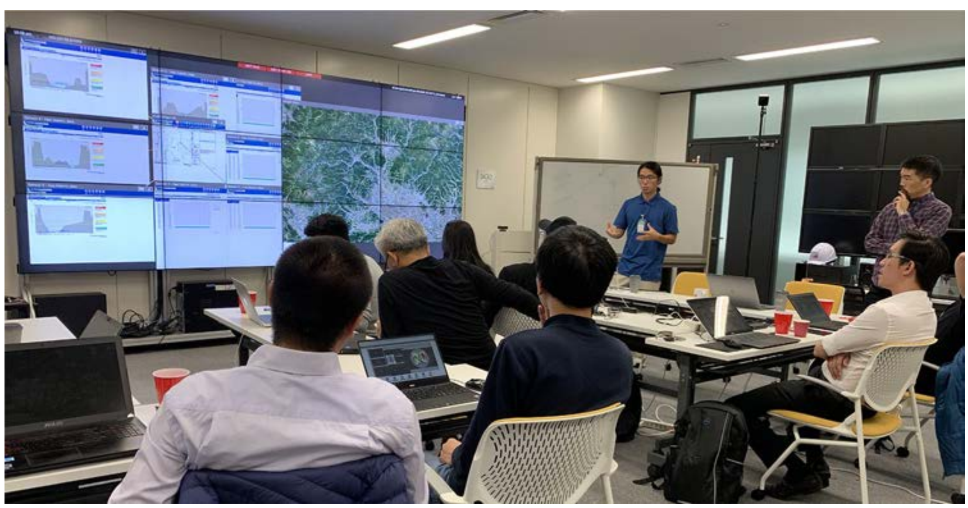

Fig. 7 LandSAGE Project Meeting showing SAGE2 in use at Mahidol University, Thailand (2020)

studies conducted on these prototypes and feed those results back into the design process.

Public Outreach (P) — In museum and public education settings, SAGE2 has been used to create immersive, interactive experiences that bring scientific research to life. At the Imiloa Astronomy Center in Hilo, Hawaii, a three-screen display wall powered by SAGE2 became a centerpiece for daily exhibits and outreach, enabling visitors to engage with complex data in an accessible way. For Imiloa's 13th anniversary in 2019, the Carnegie Airborne Observatory

Fig. 8 Interactive visualization system developed in SAGE2 for managing intensive care unit data during patient handoffs (2017)

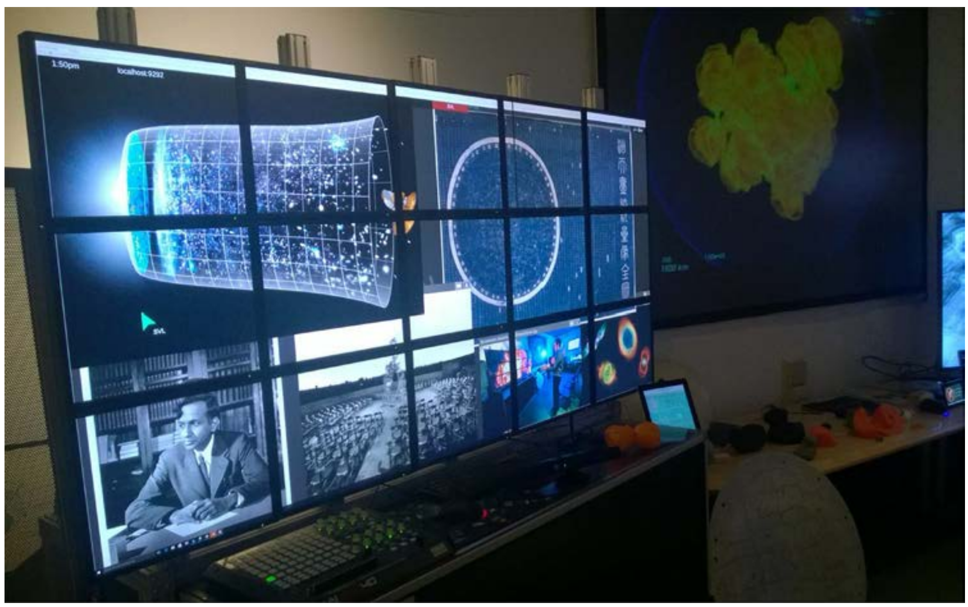

Fig. 9 SAGE2 at Adler Planetarium's Space Visualization Laboratory (2007)

presented high-resolution aerial surveys of Hawaii Island on the system, displaying environmental data across the tiled wall. Similarly, at the Adler Planetarium in Chicago (Fig. 9), the Space Visualization Lab (SVL), established in 2007, used SAGE2 to showcase high-resolution telescope imagery on a tiled wall, connecting visitors with the latest research conducted by the museum's affiliated scientists. By integrating diverse media–from airborne environmental surveys to astronomical observations–SAGE2 enabled scientists and educators to deliver richer learning experiences and foster deeper public engagement with science.

Biotechnology (CVXP) — Around 2012, Monsanto installed display walls at its four offices in Cambridge, Massachusetts; Research Triangle Park, North Carolina; St. Louis, Missouri; and Bangalore, India (Fig. 10). Using SAGE2, these walls enabled teams to share research results in precision agriculture and comparative genomics across locations. This led to the development of BactoGeNIE [8], a bespoke tool designed to compare hundreds to thousands of bacterial genome neighborhoods–something tools at the time struggled to do.

Creative Media (CVP) — In collaboration with Nippon Telephone and Telegraph and CineGrid, the Electronic Visualization Laboratory demonstrated SAGE2 to the Walt Disney Company in 2010, as Disney explored how advanced cyberinfrastructure could transform film production workflows and enable artists to collaborate more effectively across geographic distances (Fig. 11). This work garnered the 2010 Corporation for Education Network Initiatives in California (CENIC) Innovations in Networking

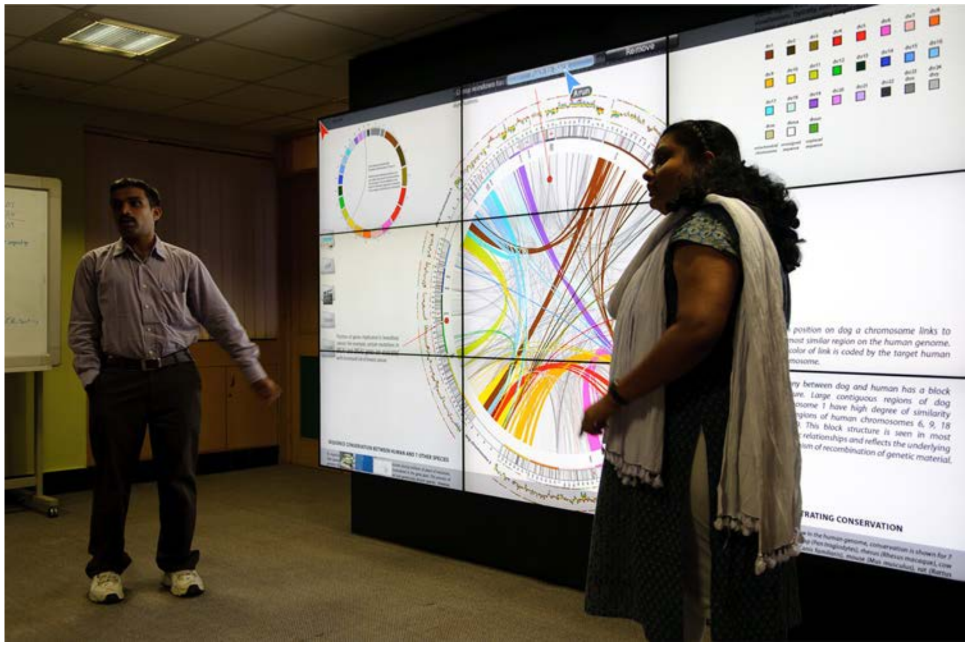

Fig. 10 SAGE2 at Monsanto in Bangalore, India (2012)

Fig. 11 SAGE2 at Walt Disney Studios (2010)

Award [9]. More recently, in 2024, SAGE3 was featured in the Amazon film Red One, where it served as the command center of a fictional weather-tracking agency. For this scene, SAGE3 powered an array of simultaneously playing video files, arranged as dynamic data dashboards to create a realistic on-screen environment. Unlike traditional approaches, where such visuals are fixed on the day of shooting, SAGE3 allowed the director to make last-minute adjustments to better match the needs of the scene.

Large Scale Scientific Visualization (CVXP) — In 2016, the Army Research Lab, in Baltimore, Maryland used SAGE2 in their visualization laboratory to support the visualization needs of their scientists. They developed ParaSAGE[10] to enable SAGE2 to integrate with ParaView - a widely used scientific visualization tool. With ParaSAGE, ARL scientists were able to visualize high fidelity physics based models enabling investigations of complex aero-engine processes relevant to Army and Department of Defense Aviation and Hypersonics science and technology efforts (Fig. 12).

Commercial Licensing — As SAGE2 matured, it was licensed to two companies: MultiVis (in 2022) and Metaform (in 2024). MultiVis (multivis.com) is headquartered in Brazil, with subsidiaries in the U.S. and Europe, and specializes in building command-and-control center systems. Metaform (metaform.com) partnered with Little Diversified

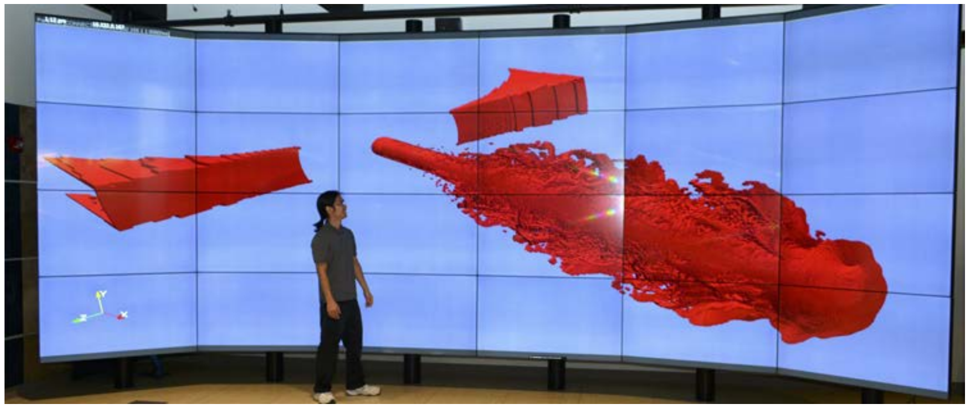

Fig. 12 SAGE2 at Army Research Laboratory showing ParaSAGE visualizing a physics based model of engine combustion, enabling users to inspect the shape of the plumes to help guide the design of the containment object (2016)

Fig. 13 SAGE3 used for Cyberinfrastructure Training (2021)

Consulting to develop Project Fusion, an ambient computing laboratory designed for hybrid work environments at Little's Uptown Charlotte, North Carolina office. At the core of Project Fusion is a highly customized and extended version of SAGE2. Metaform's mission is to help customers address the challenge of analyzing growing and diverse datasets using their technologies, and SAGE2 provided a powerful substrate on top of which to scaffold their product offerings.

Workforce Development/Education (CVP) — Since SAGE is a university research project, arguably its most widespread use is in classrooms, where it has been adopted as a tool for enhancing interactive learning and collaboration. For example, in 2021, the Cyberinfrastructure Training to Advance Climate Science (CI-TRACS) project was an NSF-funded initiative run by the University of Hawaii's Data Science Institute. Its mission was to build vital cyberinfrastructure skills among climate scientists and students across the Hawaii–Pacific region. CI-TRACS provided hands-on training for undergraduates and graduate students, hosted publicly available workshops, ran a summer research immersion program, and fostered a community of practice through the CyberClub, which served as a hub for training materials, datasets, and stakeholder consulting services in climate data management and visualization. As part of its cybertraining approach, CI-TRACS incorporated SAGE3 into its curriculum–enabling students to create dynamic, interactive research posters using the platform (Fig. 13). This hands-on use of SAGE3 gave learners immersive, collaborative visualization experiences, reinforcing the program's emphasis on cutting-edge cyberinfrastructure and real-world data practices.

As another example, in 2025 the University of Wyoming School of Computing uses SAGE3 in its Data-X Studio - a computation-focused space for data science-oriented collaborative learning and data analytics and visualization. The studio is equipped with four 98-inch 4 K displays/computer combos; an 86-inch interactive touch table; cameras and audio; adjustable rollable desks and seating. SAGE3 enabled its users to collaborate between Data-X's stations using hyper screen sharing that allowed them to share the screens of multiple laptops between stations simultaneously, alleviating their need to pass video cables between laptops when sharing results with each other.

One final example of the impact of SAGE3 in the classroom is through its use in various classes at Virginia Tech, University of Illinois Chicago, and the University of Hawaii; classes such as these helped the SAGE team explore how users organize content when using an online whiteboard with a large in-person display, what the potential benefits of such a classroom setup might be, and whether such a classroom setup had advantages over other, more common setups [11]. SAGE3 served as the online whiteboard software in this study, and was found to be beneficial for student engagement with materials, likely due to the permanence of the SAGE3 boards and the collaborative ways in which the classroom setup was used. As of this writing, classes that have used SAGE3 with a large display range from data visualization and analytics to biochemistry and video game development class.

Comparison of SAGE to Other Tools

Scientists need a variety of software tools to support their work process. For example, to share data and its products in many different file types, from text and data tables, to three-dimensional imaging and videos, they may use tools like Google Drive, Microsoft One Drive, Box, and Dropbox. To communicate ideas, inspire discussion and organize thoughts among one or more people they may use Mindmap, Google Jamboard, and infinite canvas boards such as Miro, Freeform, Figma, Mezzanine and Thinkhub. To share ongoing or finished data products, either as a slide deck, screen sharing, or a working demo, they may use Google Slides, Prezi and the many infinite canvas tools mentioned above, as well as communication tools like Zoom. To conduct data transformation and analysis they may use Jupyter, Google Colab, Tableau or Spreadsheets. Additionally, recent workflows (e.g., Einblick and CoCalc) opt to overcome the limitations of a linear computational notebook [12] by developing computational environments that use spatial features.

In Fig. 14, we compare and map the evolution in capabilities of the major implementations of the SAGE platform. The letters C, W, V, X, and P map to the tasks associated with the computational science research, development, and training enterprise defined in Fig. 3. With SAGE3, we set out to create an all-in-one toolset that integrates rich media content and computational capabilities within a desktop-like environment, designed around the principles of embodied cognition and the meaningful spatial placement of information [13–19]. Any web application can run within SAGE3, bringing users together in a unified workspace that amplifies the capabilities of existing scientific gateways and portals[20]. SAGE3 surpasses competing systems through its

Fig. 14 Comparison of the capabilities of the three SAGE platforms. The CWVXP columns denote the various phases each feature of SAGE supports (*C* Conceptualizing ideas, *W* Data collection and wrangling, *V* Data analysis and visualization, *X* Knowledge crystallization, *P* Knowledge presentation)

	C	W	V	X	P	SAGE1	SAGE2	SAGE3
Compute Cluster Support						Required	N	N
Content mirroring					O	Only via pixel broadcasting	N- laptop only show thumb nails	Y- laptop shows same content as wall
Screen Sharing	O	O	O	O	O	Multi- but only on local wall	Multi - performance limited	Multi & unlimited
Embedded Video Conferencing	O				O	N	With web based app	N
Stickie Notes	O			O	O	Y	Y	Y
Web Browsing	O		O	O	O	N	Y but out of sync	Y- 3 modes - web link, streaming, embedded browser
Annotation	O			O	O	N	N	Y
Computational Notebooks	O	O	O		O	N	N	Y
Map Viewer	O		O		O	N	Y	Y
Video Player	O		O		O	Y	Y	Y
PDF Viewer	O			O	O	N	Y	Y
Polling tool	O			O	O	N	N	Y
AI tools	O	O	O	O	O	N	N	Post it, pdf, image, web
Drawing Tool	O			O		N	N	Web based
Provenance	O	O	O	O		N	Y- save and play back sessions	Y-arrows
Wall Organization	NA	NA	NA	NA	NA	N	Dividers	Rooms and boards
Authentication	NA	NA	NA	NA	NA	N	N until later	Y
Custom Applications	NA	NA	NA	NA	NA	N	Y	Y- apps and plugins
Change Notification	NA	NA	NA	NA	NA	N	N	N
Collaboration Model	NA	NA	NA	NA	NA	Use laptops to control wall- all users must gather in front of wall. Server needed per wall.	Same as SAGE1	Server manages multiple boards and rooms. Wall is just another client
Content Rendering	NA	NA	NA	NA	NA	In back end and pixels are streamed	On wall client	Client side

flexibility, extensibility, and deliberate design for collaboration across distributed groups using varied display devices and sizes – building on SAGE1 and SAGE2's historical support for large, wall-sized displays. Its AI core further extends these capabilities, enabling countless applications for real-time analysis and interaction. Finally, as with all prior versions of SAGE, SAGE3 is open source [21].

Like many modern collaboration tools, SAGE3 supports users placing content on an infinite canvas called a *board*. Boards are organized within *rooms*, and rooms and boards can be password protected by the user that creates them. Each board is assigned a unique URL to make sharing easy. At the room level, SAGE3 manages assets (files uploaded or created on any board in the room) and kernels (both public and private), giving every board in a given room access to related content.

Users can use any number of SAGE3 tools within boards, most commonly: image viewers, web viewers (load fully functional browser windows within SAGE3), PDF viewers, video viewers, map, screen share (as many as supported by one's network bandwidth), and sticky notes (see Fig. 15). Many of these start by simply dragging content onto the board or copying and pasting files from one's laptop. Every launched tool can be positioned and resized anywhere on the board. Other users on the board can also interact and manipulate windows; for example, they may resize each other's screen share, edit each other's notes, flip through pages in a PDF document regardless of who added it, and also download a copy of any document from the board.

A board's user interface enables direct manipulation of tools and a set of control widgets, similar to toolbars, that can be moved about and called/dismissed as needed. These widgets let users: launch tools, track or follow other users on the board, access the assets manager, launch plugins (a form of additional web applications), add annotations, and navigate across the board.

SAGE3 also introduces a paradigm for Human–AI collaboration that moves beyond the limitations of linear interfaces like chatboxes and notebooks by embedding AI within a spatially organized, collaborative environment. Spatiality serves as both context and framework, enabling AI to reason in ways aligned with human embodied cognition. To support this, SAGE3 provides multiple entry points for AI access–from global and per-tool buttons to contextual operations on selected objects–and extends notebook-style workflows with spatially linked SAGECells (Figs. 19 and 20) and provenance links for branching conversations (Fig. 21). Interaction is designed to scale from single objects to large heterogeneous collections, with future integration of workflow systems for very large datasets. Crucially, SAGE3 supports parallel, multi-user, multi-AI collaboration across the shared canvas, fostering a form of collective intelligence in which space itself becomes a medium of thought.

The next section provides details about how the user-interface design interacts with, and motivates, SAGE3's software architecture.

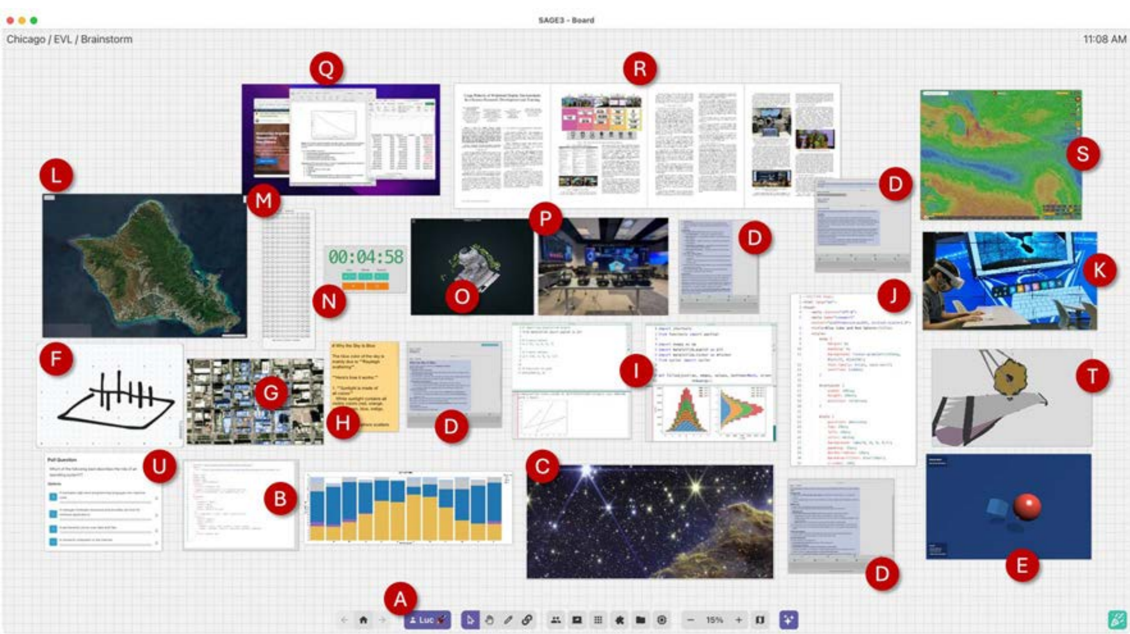

Fig. 15 SAGE3 board showing its tools. **A** main toolbar, **B** VegaLite visualization, **C** Deep image viewer, **D** Spatialized AI chat, **E** 3D Web rendering, **F** Drawing app, **G** GeoJSON viewer, **H** Sticky note, **I** SAGECell, **J** Code editor, **K** Movie player, **L** Map viewer, **M** CSV Table viewer, **N** Timer, **O** Potree point cloud viewer, **P** Image viewer, **Q** Screen share, **R** PDF viewer, **S** Web viewer, **T** GLTF model viewer, **U** Poll tool

Designing SAGE3's User Experience

The goal of the SAGE project, as it evolved over the years, has been to support science work in academia, government laboratories, industry and museums, to encourage high levels of collaboration among users, and to showcase high-resolution data on large displays. The emphasis on large displays was based on many factors; large displays have been shown to assist with sensemaking tasks, support visual-analytic comparisons that would be difficult on a more limited canvas, use the human propensity for spatial memory to organize large amounts of information, and facilitate large group collaboration by their sheer physical size [1, 15].

Today, display and projection technologies are both higher in quality and resolution and significantly lower in cost, making large displays a more affordable option in many settings. Also, SAGE3's capabilities provide all the benefits mentioned above in scientific workflows (see Sect. 2 earlier and citation [1] for scenarios from different phases of e-Science work on large displays using SAGE2). And, during the past few years, as people worldwide were first mandated to work from home due to COVID and now enjoy the benefits of hybrid work environments, it became clear that remote collaboration [22] using personal displays must be addressed.

SAGE3 design goals are an amalgamation of historical SAGE goals along with the critical need to support collaborative work on displays of varying sizes, as well as our desire to purposefully match the needs of scientists with modern-day computational capabilities and Data Science and AI tools. The following section is organized in four categories: (1) Flexible display sizes, quantity and locations; (2) Collaboration, (3) Computational Science support, and (4) AI Collaboration support.

SAGE3 is being developed using a continual near-live prototype process [23]. At the time this paper was written, a SAGE3 prototype has been used for approximately three years by several SAGE3 research and development groups, multiple times per week. It is used to teach classes, hold meetings, give presentations, brainstorm and foster sensemaking of information.

Display Flexibility

Figure 16 shows various SAGE3 collaboration modes supporting local and remote users working in front of multiple devices, from large display walls to small laptops. In location 1, users are in front of a large display wall using their laptops to contribute and manipulate content on the wall. In location 2, users at a remote site use a large display wall that is part of the same collaboration. Location 3 shows a user viewing the same board while sitting in front of a mid-scale display; this user can control the board's content either from the computer driving the mid-scale display or from their own laptop, mimicking the workers in the large display settings. Location 4 shows a mid-scale display where a user is a passive spectator in the collaboration. In location 5, a user might be at home or in a hotel room, collaborating from their laptop.

We made several design decisions that improved our support for these configurations. A key SAGE3 design component is the *wall viewport*. SAGE2 was designed such that users with laptops sat around a large display - a SAGE wall - where they could see and control the layout of information on the wall, but could only see thumbnails of the content on their laptops. While in SAGE2 a wall was a physically large display, in SAGE3 it is easy to designate any client to the role of "wall" which is now more conceptually thought of as the space for focused attention for all collaborators regardless to their physical location. Walls make their current view within the infinite canvas (viewport including position and zoom level) visible to all users on the board by showing a bounding box. This bounding box does not actively limit the

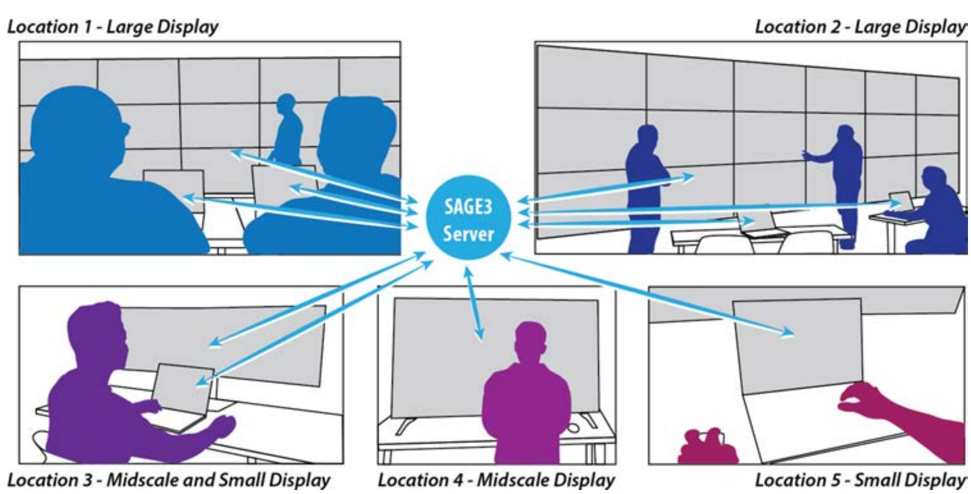

Fig. 16 Various collaboration modalities supported by SAGE3

users' access to the rest of the board, but highlights the area of focus. It supports easy manipulation as this bounding box can be dragged along the board and resized from any client - this is one of several interaction schemes in SAGE3 that support followship and attention re-direction.

There are challenges that arise from designing user interfaces (UI) for both extremely large and extremely small displays. In SAGE3, the UI is a separate layer on top of the content that does not change as the board is zoomed and panned. Large displays can have very-high resolutions and it is important that UI text labels remain legible, so users can choose the size of UI elements per client. In addition, the UI is designed with the thought of touch-based large wall displays in mind. We do not place any interactive elements at the top of the UI since that area can be inaccessible on large displays that are often taller than a person, and a widget replicating the capabilities of the main toolbar is accessible anywhere on the board with a right click (or long press using touch interactions).

Collaboration

Several design decisions we made improved upon our goal to support collaboration. Using an infinite canvas provides users with space to place content without overlapping it with other contributors. It also causes problems; an infinite canvas lacks ways to anchor users to specific locations on the board and, when used over multiple sessions in time, can result in "Window Inflation" where newer content is positioned to look bigger and more prominent than older content.

One solution we developed based on our experience as large-display advocates is the concept of wall viewport we described above. While this is a simple UI feature, we found that having a wall boundary box helps collaborators focus on a specific area of the board and tells them where to place content if they want it to be in view (on the wall) or out of view for the moment. We also learned that having a bounding box significantly helps with window inflation.

So, while there are benefits of using an infinite canvas for collaboration, we found it necessary to complement it with viewports (such as our wall viewports) and supplying many ways to rapidly navigate across the board; for example, in SAGE3, a user can jump to any location using a mini-map of the board, or they can use a keyboard shortcut ('z') over a window to zoom there. We introduced a feature to follow the viewport of a selected user to help get all collaborators on the same page in the absence of other concrete shared viewports (like that of a physical wall). We also added a "Party" feature which allows a host to bring collaborators along across different rooms and boards.

Computational Science

Several design decisions improved upon our efforts to support computational science. SAGE3 promotes the idea of a new paradigm for computational notebooks, one that uses space to organize the "messy" process of scientific work. On SAGE3, researchers can mix code cells with other forms of media (notes, images, tables, and documents) into a story board or an interactive poster. We discuss in depth the SAGECells system below.

A major enhancement of SAGE3 over SAGE2 is its integration with data science and AI capabilities. A short-term goal is to provide our end-users with familiar tools (such as computational notebooks), but improve upon them by leveraging the advantages of spatialized information in SAGE3 using SAGECells (Figs. 19 and 20). A long-term goal is to make AI a first-class collaborator in a meeting, where it can examine information deposited by users, analyze it, and produce useful insight, with feedback from users.

Jupyter Notebook is a staple for scientists and their students to perform data analysis. Our user study, "Exploring Organization of Computational Notebook Cells in 2D Space," discovered that although Jupyter displays its code cells linearly, users prefer and are more productive arranging cells spatially [24]. This was the motivation behind SAGECell, a code cell that users can freely position on SAGE boards, grouping cells in a way that aligns with their objectives or enhances their workflow.

To use SAGECell, the user first selects a kernel from the toolbar. Kernels can be created for Python, R or Julia and can be private or public. For convenience, the user creating a kernel assigns an alias to it, which is then immediately available to all SAGECells. Switching between multiple kernels enables task "segregation" for enhanced prototyping. For instance, users can independently experiment with or prototype analyses in separate kernels. The Code Input Box is used to enter and edit source code and the Output Box below it displays text and standard rich media mimetypes, such as PNG, JPG, PDF, SVG, HTML, which may result from the code execution. To organize the order of the cell execution, users can create manual links that explicitly define which cell precedes which cell. Though we experimented with various other heuristics for establishing run order, the explicit, visually marked with arrows, method seems to be most robust. The user can choose to run the code of the current cell, all the cells prior to and including the current cell, or all the cells starting from the current cell onwards.

SAGECells are currently used to discuss data during meetings, teach code to students, perform pair programming, create visualizations, carry out data wrangling, and handle some basic automation tasks. We plan to leverage existing machine learning models to include natural language interactions, allow users to write their own models using data in each room, and offer more ways to think about writing code. We are actively building the mechanisms to see which models are most effective. In addition to their functionality for data analysis, SAGECell enables users to programmatically “drive” SAGE3 through a dedicated Python interface so users can perform any action available through the SAGE3 GUI interface via the Python programming.

We accommodate scientists in other ways as well. First, we note the importance of data organization and data privacy for users from scientific fields. In SAGE3, we support privacy with password-protected rooms and boards, and with private SAGECell kernels that can only be used by the user that created them. Second, many users involved with computationally intensive tasks prefer keyboard shortcuts to work more efficiently, so SAGE3 supports shortcuts within specific tools; e.g., navigation (using arrow keys to pan the board or ‘z’ to zoom to a window) and launching tools (shift + ‘s’ will create a new sticky note). Third, support for computational science is yet another motivator to leverage high-resolution large displays and the design decisions made specifically for that purpose (in the previous two sections); large display walls can simultaneously show multiple, high-resolution visualizations in a way that cannot be replicated on smaller displays. Last, scientists often need to extend and customize their tools. SAGE3 code is open source so users can write native SAGE3 apps, and we offer ways to add plug-ins to SAGE3 with the ease of developing a simple JavaScript web application.

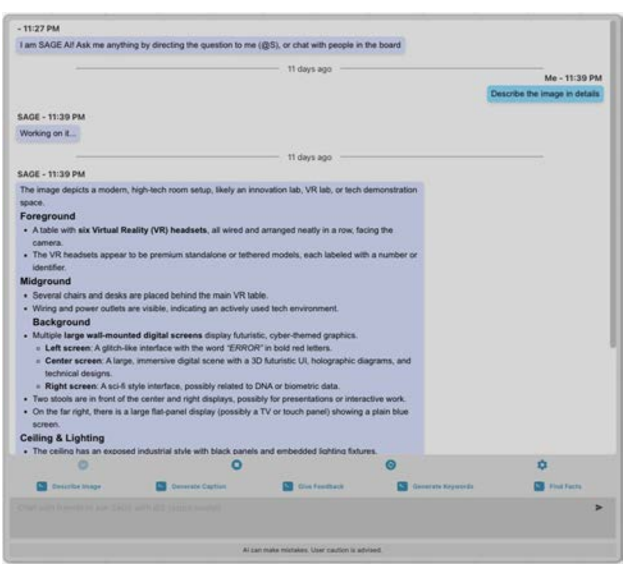

Fig. 17 SAGE3 Chat user interface

Human-AI Collaboration

SAGE3’s paradigm for human-AI collaboration is still evolving. Nevertheless, the guiding design question is: How can we foster collective intelligence in an environment where multiple users collaborate across a rich, spatially arranged information landscape–where the spatial arrangement itself is an essential part of the context for AI?

In this vision, spatiality is not merely a visual convenience but a semantic framework. It enables AI to reason about the workspace in a manner analogous to human embodied cognition, allowing the shared spatial environment to act as a common ground for communication between humans and AI. In essence, we seek to endow AI with a form of spatially grounded cognition.

The Role of Spatiality In SAGE3, content is arranged spatially on infinite canvases, as such layouts provide distinct advantages for sensemaking. Spatial organization enables users to perceive relationships, group related concepts, and retain contextual cues in ways that linear interfaces cannot.

To support this, we needed to incorporate multiple entry points for accessing AI:

1. A global AI button on the main toolbar that enables quick questions to be asked of AI.
2. The AI chat tool, accessible from the tool menu to enable longer engagements with AI.
3. Per-tool AI access, available through each tool’s toolbar, uses the information managed by that tool as the context for AI queries.
4. Contextual AI access triggered by selecting a group of objects in the environment.

Figure 17 shows the user interface of the Chat tool, where multiple users can collaborate on a specific topic. In addition to conversing with each other, users can engage the AI by either posing direct questions to the Large Language Model (LLM) or selecting from a set of predefined prompts designed to accelerate exploration. These prompts are tailored to the type of data being explored (e.g., text, code, PDF, image, or web). Figure 18 illustrates the panel where users can select the LLM they wish to employ.

From Linearity to Spatial Reasoning The prevailing user interface paradigm for AI–the chatbox–is inherently linear. While its familiarity makes it accessible, it constrains the flow of information to a single ordered sequence. A similar limitation exists in Jupyter Notebooks, where cells are presented and executed in a fixed order.

Our research, however, found that users prefer to *spatialize* Jupyter Notebook cells when given the opportunity [24]. To support this, SAGE3 introduced SAGECells–an adaptation of notebook-like cells that can be placed anywhere

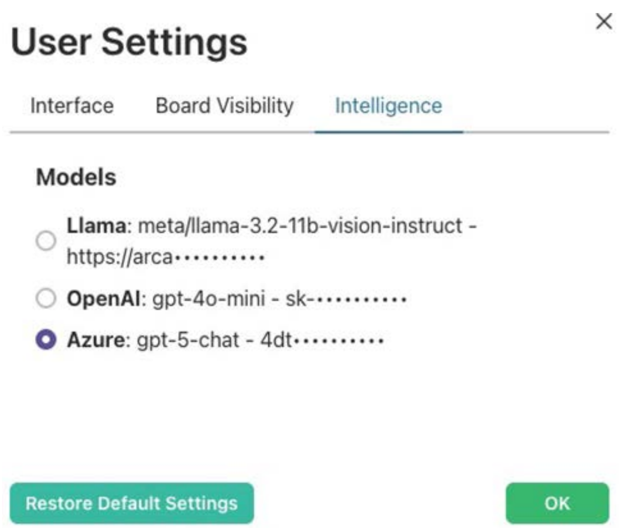

Fig. 18 Large Language Model (LLM) selection (pre-configured by the SAGE3 site administrator)

on the canvas and linked in arbitrary sequences (Figs. 19 and 20). In this paradigm, execution order is not limited to its vertical position as in Jupyter Notebooks but by explicit links created by the user to connect spatially organized code cells.

SAGE3 further advances this concept by spatializing AI outputs, supporting scenarios in which users explore multiple branching conversational paths in parallel. To aid this process, SAGE3 introduces provenance links (dotted lines), which can be used as visual traces that reveal the evolution and divergence of conversations (Fig. 21).

Interacting with AI at Different Scales Working within a spatial environment introduces distinct interaction patterns depending on the number of objects involved:

1. Single Object – The user selects an individual object and accesses AI tools via the object's toolbar.
2. Small Collection (small n) – The user lasso-selects a small group of objects, triggering a collection-specific toolbar with an AI button to initiate context-aware operations.
3. Large Collection (large N) – For large sets of objects, SAGE3 employs container references (e.g., links to Google Drive or Dropbox folders) to enable AI operations without requiring direct manipulation of each individual item. At the time of writing, support for very large numbers of objects has not been implemented, as additional backend infrastructure is needed to handle such a scale. To address this limitation, we are collaborating with the teams are developing scientific workflow systems such as Pegasus [25] and Tapis [26].

Importantly, SAGE3's AI also needs to be able to operate over heterogeneous collections–for example, a combination of sticky notes, PDF documents, images, and code snippets–providing integrated responses to queries that draw upon diverse media types. This feature is also under development.

Parallel, Multi-User, Multi-AI Collaboration SAGE3 is designed for true collaborative concurrency. In the chat windows in Fig. 21, multiple users can engage in each of these distributed chat sessions simultaneously. SAGE3's AI can participate in all these interactions in parallel, maintaining context for each conversation thread. This enables scalable,

Fig. 19 SAGECell tool containing the code section, the output section, the kernel and font size selection, and the execution control on the side

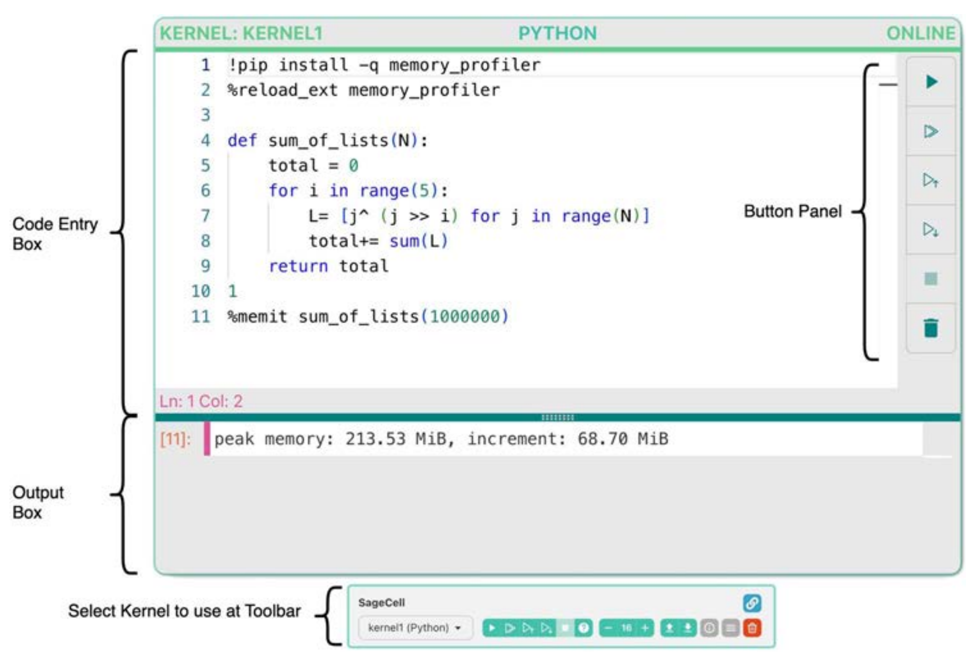

Fig. 20 SAGECell execution order can be defined by the user by creating arrows linking cells

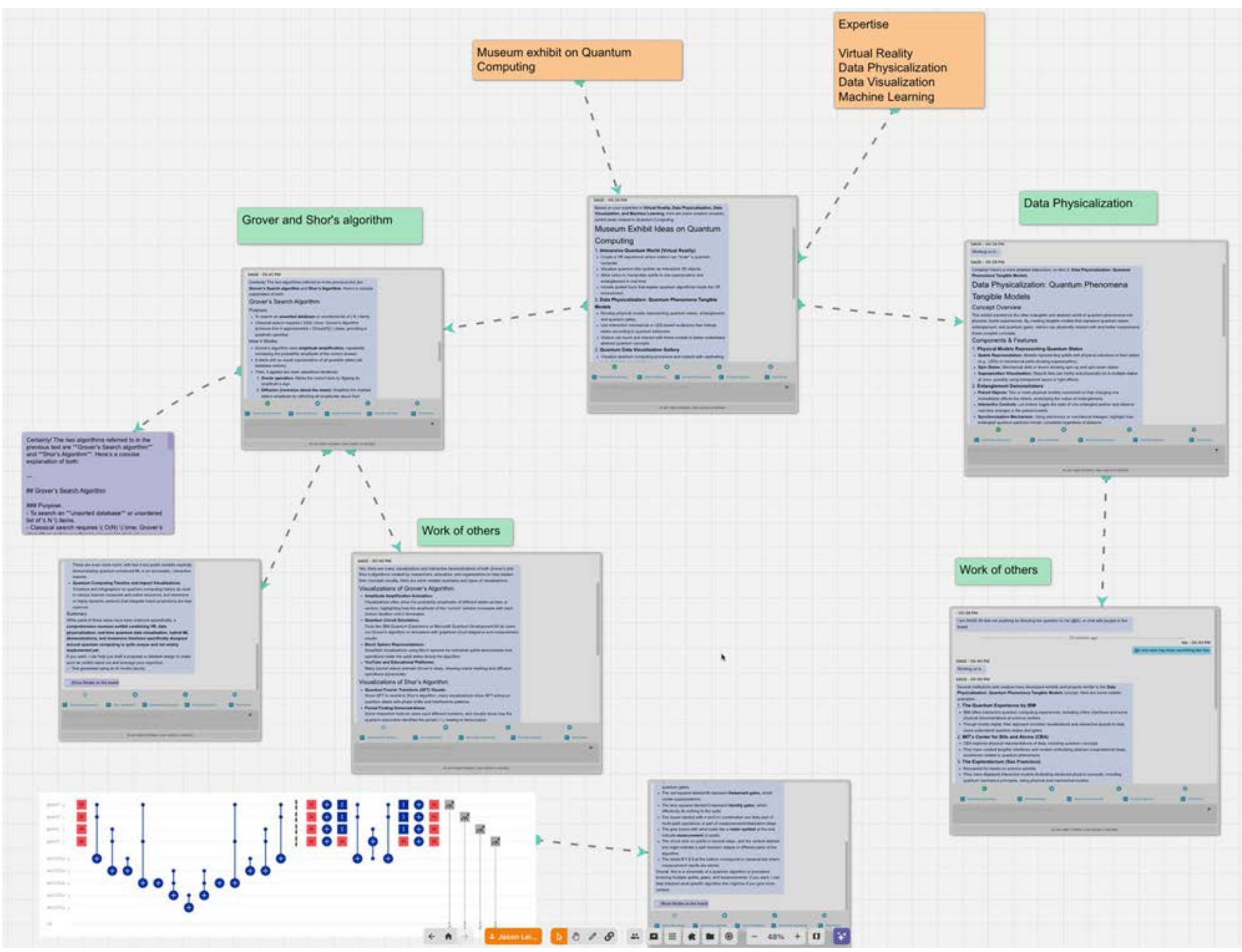

Fig. 21 Ideating with AI in a branching conversation structure (depicted by dotted lines connecting parent conversations with child conversations)

multi-threaded human–AI collaboration in a unified shared environment.

To summarize, the SAGE3 AI interface is guided by a philosophy that merges the familiarity of traditional, linear AI interaction paradigms with the cognitive advantages of spatial reasoning. By enabling AI to interpret and operate within a spatially organized, multimodal, and collaborative environment, SAGE3 fosters a form of shared, embodied intelligence where space itself becomes a medium of thought.

In the next section we will describe the technological challenges that had to be overcome to support SAGE3's complex user experience requirements.

The Architecture

A Brief Technological History of the SAGE Suite

In 2004, the OptIPuter project envisioned a future where computational science would no longer be constrained by network bandwidth [27]. It demonstrated that high-speed computer networks could serve as the backplane for planetary-scale computers, linking distributed resources–compute clusters, data storage systems, scientific instruments, and visualization clusters–into a seamless whole. This infrastructure enabled scientists to interact with ultra-resolution visualizations on large, high-resolution display walls, themselves driven by compute clusters.

Using OptIPuter's distributed architecture, SAGE1 streamed visualizations from remote servers over 10 Gbps networks to these display walls [28] (hence its name - Scalable Adaptive Graphics Environment). When users moved windows from one location to another, pixel-routing automatically directed the data to the appropriate compute nodes, maintaining a coherent, uninterrupted image–a capability that foreshadowed NVIDIA Grid nearly a decade later [29].

By 2011, advances in web graphics made it possible to perform GPU-accelerated graphics rendering within web browsers. Furthermore, it was possible to drive large display walls with single PCs rather than compute clusters, making it easier for scientists to acquire, use, and maintain display walls without requiring an IT staff - provided the necessary software was available. In 2014, SAGE2 (*Scalable Amplified Group* Environment) [30, 31] emerged to leverage this emerging technology trend, by re-architecting SAGE1 from C++ to a fully JavaScript code base, where all graphics were rendered in the browser.

By 2020, JavaScript frameworks such as React and Angular were widely used, scientists were increasingly interested in leveraging data science and AI capabilities, and COVID made it impossible for them to go into their laboratories to use their tiled display walls. The confluence of these events motivated us to re-architect SAGE2 from the ground up. The alpha version of SAGE3 (the *Smart* Amplified Group Environment) was released in November 2021 [21]. SAGE3 leverages robust JavaScript frameworks and cloud services to re-implement and improve upon SAGE2's capabilities, and adds provisions to run and manage data science and AI services. Furthermore, SAGE3 enables users to both work at home on their desktop computers/laptops and in their research laboratories on display walls, while having the content seamlessly shared across infinitely sizable workspaces and infinite numbers of workspaces.

SAGE3's Architecture

SAGE3 aims to create a seamless collaborative environment that supports real-time interaction across distributed teams, from individual laptop users to groups working on large-tiled display walls. The system must handle several critical challenges: maintaining consistent shared state across multiple concurrent users, providing low-latency synchronization for real-time collaboration, and delivering a unified user experience across diverse hardware configurations–from laptops to room-scale tiled displays.

To address these challenges, SAGE3's architecture (Fig. 22) is organized into three complementary parts, each solving specific aspects of the collaborative problem. The backend (SAGE3 Server) provides the foundational infrastructure for state synchronization, data persistence, and computational services, ensuring that all users maintain a consistent view of the shared workspace. The frontend (Client) delivers a responsive, cross-platform user interface that adapts seamlessly from individual desktop environments to large collaborative displays, while maintaining identical functionality across all deployment scenarios. External services extend the platform's core capabilities by integrating specialized cloud-based functionality for AI, authentication, and media streaming–capabilities that would be prohibitively complex and resource-intensive to develop and maintain in-house.

The Backend

The backend architecture directly addresses the core challenge of maintaining a consistent, synchronized state across multiple concurrent users in real-time collaborative sessions. When dozens of users interact simultaneously with shared content by moving windows, editing documents, or uploading files, the system must ensure that every participant sees identical updates without conflicts or data loss. In addition, the back-end must provide computational resources for data

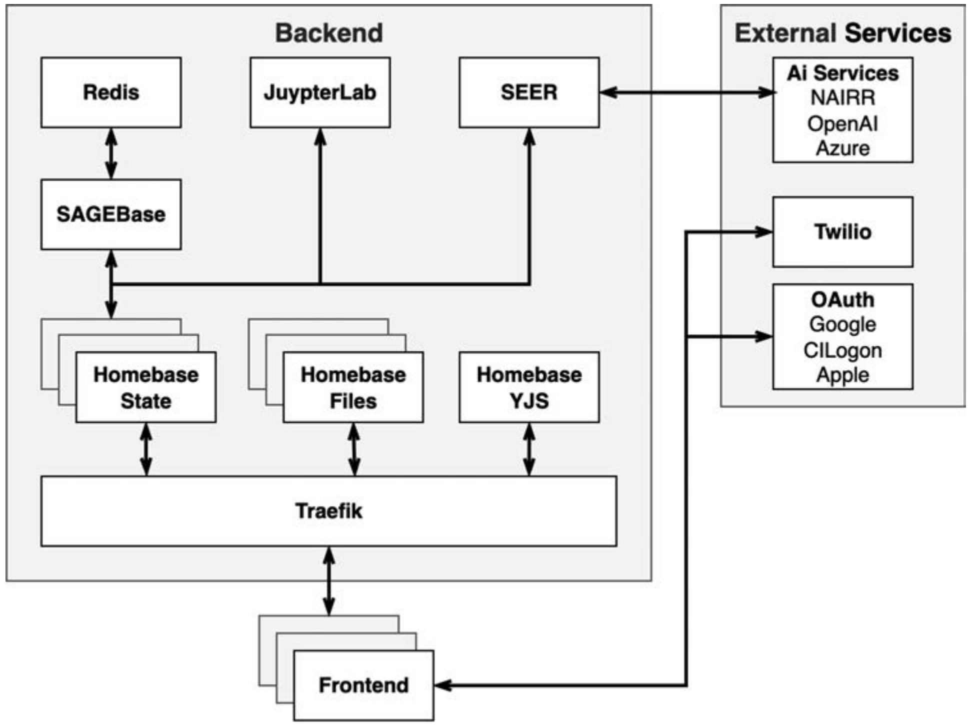

Fig. 22 SAGE3 Architecture with backend (server), frontend (clients), and external services

analysis, coding, and AI processing while maintaining security and isolation between users and sessions.

To solve these challenges, the SAGE3 backend implements a distributed service architecture composed of eight specialized components: a reverse proxy (*Traefik*) for load distribution and SSL termination, three Node.js servers collectively called *Homebase* (State, Files, and Yjs) for coordination and synchronization, a high-performance *Redis* in-memory database for persistence, a database abstraction layer (*SAGEBase*) for structured data access, a Python AI runtime (*Seer*) for LLM services, and a *JupyterLab* instance for code execution.

Reverse Proxy

When multiple users connect simultaneously to a distributed system, incoming requests must be efficiently routed to the appropriate backend services while maintaining security and performance. A reverse proxy solves several critical challenges: it provides a single entry point for all client connections, enables SSL termination to secure communications, and distributes load across multiple service instances to prevent bottlenecks.

SAGE3 implements this layer using Traefik (traefik.io), a modern reverse proxy that simplifies SSL certificate management by centralizing encryption at a single point and that enables load balancing across multiple service instances. By handling SSL termination at the proxy level, backend services can focus on core logic without managing certificates. Additionally, Traefik allows SAGE3 deployments to run multiple copies of *Homebase State* and *Homebase Files* at deployment time, distributing client requests across these instances to handle higher concurrent user loads and improve system resilience through redundancy.

Coordination Layer

The coordination layer manages the complex challenge of synchronizing collaborative activities across multiple users and maintaining a consistent state. In a multi-user environment, this layer must handle simultaneous user interactions, manage shared resources like files and documents, and ensure real-time synchronization, all while providing secure access control and reliable data persistence.

SAGE3 implements this coordination layer through Homebase, a set of Node.js and Express web servers divided into three specialized services: *State*, *Files*, and *Yjs*. Homebase exposes both REST APIs and Websocket endpoints to update and synchronize state, and serves the frontend React web application.

State Synchronization Maintaining a consistent shared state across all participants is critical in multi-user environments–when one user moves a window, creates a sticky note, or joins a board, all other users must see these changes immediately without conflicts or data loss. This requires a centralized authority that can process updates, resolve conflicts, and broadcast changes to all connected clients in real-time.

SAGE3 addresses this through the Homebase State service, which maintains the persistent state of collaborative sessions, including boards, tools, and user interactions. The service uses SAGEBase as the authoritative data store and ensures synchronization through both HTTP requests for CRUD operations (create, read, update, and delete) and Websocket subscriptions for real-time updates. Multiple instances of the State service can be deployed to improve scalability and resilience under heavy load.

Asset Management Collaborative environments demand efficient handling of diverse digital assets–images, videos, PDFs, datasets, and other files that users need to share and view together. These assets must be securely stored, quickly accessible to all participants, and optimized for different viewing contexts, from individual laptops to large display walls with varying resolutions and capabilities.

SAGE3 handles this through the Homebase Files service, which provides secure upload and download capabilities for all asset types. The service includes metadata extraction and pre-processing functionality that converts PDFs to images and generates multi-resolution versions of images to optimize performance across devices with varying capabilities. Like the State service, multiple instances of the Files service can be deployed to support horizontal scaling.

Real-time Text Collaboration When multiple users simultaneously edit the same text document, sticky note, or code cell, the system must handle conflicting edits gracefully while ensuring all participants see a consistent final result. Traditional approaches like locking prevent concurrent editing, while naive merging can lead to data corruption or lost changes. This challenge requires sophisticated conflict resolution algorithms that can automatically merge concurrent edits without user intervention.

SAGE3 addresses this through the Homebase Yjs service, which enables real-time synchronization of text-based tools such as SAGECell, Sticky notes, and Notepad. The service integrates the Yjs library (yjs.dev), which implements conflict-free replicated data types (CRDTs) to guarantee consistency across clients, even under conditions of high latency or concurrent edits [32]. Unlike the State and Files services, the Yjs service requires a single central server since the CRDT algorithm depends on a consistent authoritative instance to merge concurrent updates, and therefore cannot be horizontally scaled.

Database

Multi-user systems require databases that can handle high-frequency updates, real-time queries, and rapid data synchronization across multiple concurrent users. Traditional relational databases can become bottlenecks in collaborative environments due to complex schema constraints, slower read/write operations, and limited support for real-time subscriptions. The database must support document-oriented storage for flexible data structures, provide fast in-memory access for low-latency operations, and enable real-time notifications when data changes.

SAGE3 addresses these requirements using Redis (redis.io), an in-memory data store, combined with SAGEBase, a custom abstraction layer that provides document-oriented access patterns and manages authentication.

Redis Redis serves as SAGE3's primary database, chosen for its exceptional performance in high-frequency read/write scenarios and built-in support for real-time data structures [33]. In-memory databases like Redis offer significantly faster response times compared to traditional web-oriented databases such as MongoDB or PostgreSQL, which is critical for highly interactive collaborative tools. The deployment uses Redis Stack with RedisJSON and RedisSearch modules, enabling efficient storage and querying of JSON documents while maintaining Redis's sub-millisecond response times.

Within SAGE3, Redis operates as a document-oriented store where keys are structured hierarchically to represent collections (users, rooms, boards) and their corresponding JSON documents. This organization provides a schema similar to NoSQL databases like MongoDB, but with Redis's superior performance characteristics. RedisSearch extends this model by enabling indexed queries, full-text search, and filtering capabilities, providing the flexibility needed for collaborative tools.

SAGEBase SAGEBase serves as the middleware layer that abstracts Redis's key-value interface into a structured document-oriented API (Fig. 23). It organizes data into collections of JSON documents that can be queried, updated, and subscribed to by clients, ensuring that Homebase services have a consistent access model for all persistence operations. Multi-user collaboration extends requirements beyond standard REST operations by necessitating real-time subscriptions and bi-directional communication, which SAGEBase addresses through Websocket connections that enable clients to subscribe to data changes and receive immediate updates.

The system defines all data objects through JSON schemas that can be exported and mapped across multiple programming languages, supporting diverse client types including TypeScript web clients, Python AI backends, and C# AR/VR applications. SAGEBase automatically generates REST and Websocket routers for accessing and modifying collections (users, rooms, boards, assets, tools).

Beyond data access, SAGEBase centralizes authentication and session management, integrating with PassportJS (www.passportjs.org) to support and delegate multiple login strategies through several identity management platforms

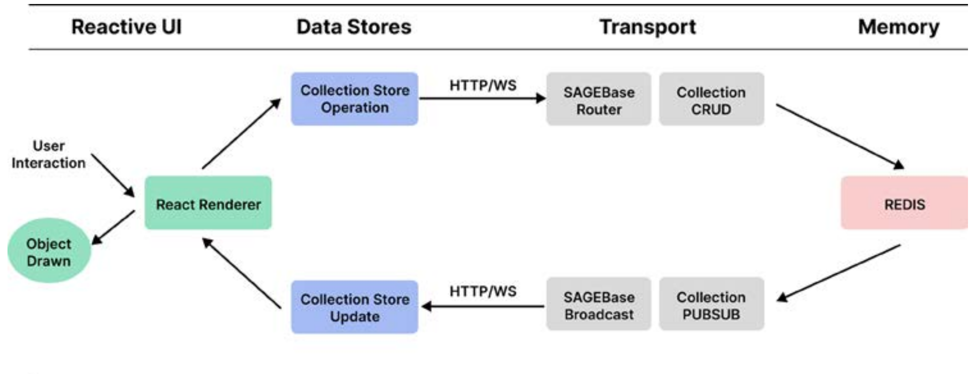

Fig. 23 SAGEBase with React: User operates on a data collection through a data store, transports the operations to the backend, updates the Redis database and broadcasts updates to all clients

such as Google OAuth, CILogon, and Apple OAuth. By consolidating both data management and authentication, SAGEBase allows higher-level services to focus on core logic rather than low-level database operations and security concerns.

SAGE3 Python Library

In addition to its primary JavaScript backend, SAGE3 also incorporates a Python library *pysage3*. It plays an important role in enabling AI functionality in SAGE3 by providing access to the tool states. A Python class, called "SmartBit," for each native tool, which is responsible for holding the tool's data. The Python engine, like any other client, receives and processes updates to maintain an up-to-date version of the data. With this capability, tools can perform code execution out-of-the-box, utilize the layout engine, and more. For example, a SAGE3 sticky note can access a LLM to translate its content. Similarly, a PDF document can access functions to identify its main topics. Every tool can use these functions out of the box without the programmer having to do anything explicitly.

Code Execution Environment and SAGECell

Data science workflows require the ability to execute code, generate visualizations, and perform computational analysis directly within shared workspaces. Users need access to rich ecosystems of scientific libraries–Python's NumPy, Pandas, Matplotlib, Scikit-learn, and R's extensive statistical packages–without leaving the collaborative environment or managing local installations. The challenge is to provide secure, isolated code execution that can handle multiple concurrent users across different programming languages while seamlessly sharing results throughout the collaborative session.

SAGE3 addresses this by embedding JupyterLab as a backend service that provides managed execution kernels for multiple programming languages, including Python and R. Rather than directly exposing JupyterLab's notebook interface, SAGE3 integrates these kernels through the *SAGECell* tool, enabling users to execute code for data analysis, visualization, and machine learning directly within the collaborative workspace. This approach delivers the computational power of Jupyter while maintaining SAGE3's unified collaborative interface, allowing code results and visualizations to be immediately shared and discussed by all users.

SAGECell leverages the messaging specification for how Jupyter frontends and kernels communicate. Specifically, we uses the ZeroMQ library[34] to manage new kernels (start, delete, interrupt), and holds code execution and output. Currently, SAGECell supports Python and R, with additional kernels that can be installed seamlessly using the same tried and tested approach that is used to install new kernels in Jupyter. This flexibility allows users to work with a wide range of tools and libraries in their preferred language, making SAGECell a versatile platform for data analysis and scientific computing. By leveraging the Jupyter infrastructure, we can provide users with the ability to scale both the number and specifications of the kernels they can launch in a vertical manner.

Seer

Seer is SAGE3's AI agentic system, written in Python with various agent libraries (LangChain, LangGraph, HuggingFace, and OpenAI). Seer enables users to leverage AI functionalities and tools within SAGE3, where users can ask questions and engage in discussion with a LLM on the specific data shown on the SAGE3 board. For example, a user can ask a question about a specific PDF publication, about a piece of code, or a specific visualization or dataset. Those discussions can be one-on-one with the AI or as a group of users. Multiple documents or data sets can also be selected to ask more general questions.

The access to LLMs is managed by the backend service *Seer*, which uses the *pysage3* library to interact with the internals of SAGE3. *Seer* currently operates a set of basic agents (e.g. summarization, generic object detection, PDF processing, image analysis). Each agent, written in Python, processes one query at a time for a specific data type (text, image, PDF, web, etc.). Each query contains the name of the LLM that the user prefers (according to various criteria, such as speed, cost, or privacy) within a list configured by the administrator of a SAGE3 site.

To date, SAGE3 supports commercial OpenAI models and hosted LLMs that support an OpenAI-style API. AI models currently supported are the Meta Llama family of models (*Llama 3.3* for general text chat, *Llama 3.2 Vision* for image and visualization analysis) and various OpenAI models (GPT-5, GPT-4). In addition to the LLMs that SAGE3 supports, site administrators who host local SAGE3 servers can select specific LLM models by editing its configuration file. It is up to administrators to choose the most effective models according to cost, performance, and privacy. Seer enabled us to quickly develop a system for SAGE3, while studying more complex systems such as LangGraph and now Model Context Protocol (MCP), which is a new open standard for connecting AI assistants to data sources. The AI software landscape is a fast-moving target and we try to use the most promising and lasting software packages that are supported by large developer communities.

Seer consists of a FastAPI server with specific routes for each query (/image,/chat,/pdf,...). The frontend sends queries to the backend, forwarded to Seer with the parameters necessary to fulfill the request (LLM selected, parameters, prompt, data sources such as PDF files, etc.). The queries and answers are typed using the Pydantic library, which makes sure the data sent and received from the Typescript frontend is well formatted. This software architecture protects the security and integrity of the system where only valid requests are sent to Seer. For complex documents (mainly PDF files at this time), we use ChromaDB (www.trychroma.com), an open-source search and retrieval database for AI applications. A PDF document is converted to text (Markdown) and then segmented into smaller units of text, such as paragraphs or sections. Each unit is then transformed into a high-dimensional vector representation using an embedding model (for instance OpenAI's *text-embedding-3-large*, which captures its semantic meaning. These vectors are stored in the ChromaDB vector database, enabling efficient similarity search. When a user submits a query, the query is likewise converted into a vector. The system retrieves the most semantically similar vectors from the database, corresponding to the most relevant portions of the original document, and uses them to generate an informed response.

We received an NSF National AI Research Resource (NAIRR, Pilot award #240471) for SAGE3 resources through Microsoft Azure credits. For the SAGE3 community, we deployed various OpenAI models on Azure cloud (GPT-5, text embedding model, document intelligence model, etc.) at no-cost to the users and integrated OpenAI APIs into Seer.

Deployment

The architecture is designed so that organizations can control and configure their own infrastructure, addressing requirements for privacy, security, and regulatory compliance while providing more predictable performance than commercial cloud solutions. Each service runs in its own container, with the complete system deployed as a collection of containers managed through Docker Compose within a secure private network. This containerized approach enables consistent deployment across multiple architectures (Windows, macOS, Linux) and simplifies both development and production deployment scenarios.

The virtual network created with the Docker deployment allows the following services to communicate freely without being exposed to the users and the Internet at large. Only the reverse proxy configured with SSL certificates is exposed to the external network. Consequently, the security layer of *SAGEBase* guarantees that all requests are properly authenticated. The services are as follows:

- Reverse proxy: *Traefik* instance exposed over HTTPS,
- Redis instance: in-memory real-time database,
- Fluentd instance: logging service to track errors and messages,
- Kernel management: small *FastAPI* python service we created using the Jupyter REST API, to manage the Jupyter kernels on behalf of SAGE3 users using SAGECells,
- State service: main Node.js collaboration hub (multiple instances are run for performance and reliability),
- Files service: file upload service and asset manager (also replicated for scalability),
- Yjs service: peer-to-peer service for CRDT functionality,
- Seer service: AI backend interfacing with various LLMs,
- Vector database: AI embedding database used by *Seer*,
- Jupyter: standard Jupyter instance (we use the *datascience-notebook* Docker image).

This architecture makes it easy to deploy a SAGE3 instance: one needs a machine running Docker, with a fully qualified domain name (FQDN) and a set of SSL certificates from a valid certificate authority. Over the years, we deployed SAGE3 servers on Windows, macOS, and mostly Linux. The main SAGE3 hubs at the University of Hawaii and the University of Illinois Chicago are deployed on powerful Linux servers, starting at 16 cores (to deploy all the services and several replicas, but more cores would be beneficial if SAGECells and Jupyter kernels are expected to be used extensively), 64GB of RAM (the more RAM the more users the Redis database can keep in memory), at least 1Gbps of outside networking (mainly used to upload large assets, delivering videos, and updating all the clients with application updates), lastly an available GPU could be used by the Jupyter instance but is not a requirement. We host our foundational LLMs on other servers with large dedicated AI Nvidia GPUs (Nvidia H100 and newer). Moreover, each team maintains a development server, using continuous integration from Github, on much lower end hardware (i.e. Mac Mini with 16GB of RAM and 256GB of storage). We also leveraged NSF-funded Jetstream2 - an NSF funded on-demand cloud computing environment - for quick developments (such as short-term user studies requiring a blank-slate server). Virginia Tech and University of Wyoming deployed SAGE3 servers on their own cloud infrastructure to follow security and privacy guidelines required to support classes and their collaborative work.

The Frontend

The frontend must deliver a responsive, intuitive interface that works seamlessly across vastly different environments–from individual laptops with limited screen space to room-scale display walls spanning multiple monitors. Users need consistent functionality whether working alone on a 13-inch laptop or collaborating with a team on a 20-foot display wall. The interface must handle real-time updates from multiple users with minimal performance degradation, provide immediate visual feedback for user actions, and maintain state consistency as users connect and disconnect from collaborative sessions.

SAGE3 addresses these requirements through a two-component frontend architecture: (1) a React-based web application that provides the core user interface, delivered through (2) an Electron desktop client that enables cross-platform deployment and system-level integrations. This approach combines the flexibility of web technologies with the capabilities and consistency of native desktop applications.

User Interface

The user interface must handle the complex task of presenting collaborative content that updates in real-time while maintaining responsive performance across different screen sizes and resolutions. The interface needs to be modular enough to support diverse content types–from simple sticky notes to complex data visualizations–while ensuring that all users see identical content regardless of their device capabilities.

SAGE3 implements this through a React-based web application built in TypeScript, chosen for its component-driven architecture that enables modular, reusable interface elements. React is mainly designed to maintain a local application state which, when updated, triggers automatically a re-render of the page. In our case, we receive asynchronous updates from Redis in the backend, when other users change the state of a tool (i.e. window size and position, state values internal to a tool, etc.). To maintain consistency between the backend state and the local user state, the web application uses local data stores built with the library Zustand (zustand.docs.pmnd.rs). Those local data stores are updated with data from Websocket messages received from Redis updates, and will also trigger a re-render of the page. Reversely, when a user moves a window or changes a tool (i.e. press a button or change a slider), state values are changed in the data store and updates are sent automatically to the backend to update Redis through Homebase/State service and the SAGEBase data management layer. This bi-directional communication pattern is encapsulated easily with the Zustand library which ensures that React components automatically re-render only when collaborative content changes (Re-render events are very costly in React and the main causes of performance issues). This architecture offers good performance and separates user interface logic from networking concerns, allowing the interface to respond immediately to user interactions while background processes handle synchronization with other participants.

Cross-Platform Client

Collaborative software must run consistently across different operating systems and hardware configurations without requiring users to install complex dependencies or navigate browser-specific limitations. The client needs access to system-level features like file system access, and hardware interfaces while maintaining security.

SAGE3 addresses these requirements through an Electron (www.electronjs.org) desktop application that combines Chromium (www.chromium.org) and Node.js (nodejs.org) into a unified runtime. This approach enables a single codebase to run on Windows, macOS, and Linux across different architectures (x64, arm64) while providing access to system APIs unavailable in standard browsers.

Experience with SAGE2 demonstrated that browser-only deployments created usability challenges including security restrictions and additional setup overhead. By consolidating functionality into a dedicated desktop client, SAGE3 eliminates these barriers while delivering consistent performance and a safe user experience. For instance, Electron provides better screen sharing capabilities, enables the rendering of embedded web pages (webviews), automatic updates, and code signing and notarization with certificates issued by trusted authorities. The client supports connections to multiple SAGE3 servers, allowing users to choose from public installations or connect to private deployments seamlessly.

External Services

While SAGE3's core architecture handles collaboration fundamentals, modern platforms require specialized capabilities that would be prohibitively expensive and complex to develop in-house: advanced AI for content analysis and generation, authentication systems that integrate with institutional identity providers, and scalable media streaming infrastructure for real-time screen sharing. Building and maintaining these capabilities internally would require significant development resources and ongoing operational expertise that diverts focus from core collaboration features.

The SAGE3 team selected to strategically integrate proven cloud-based services for AI, authentication, and media streaming. This approach allows SAGE3 to deliver advanced capabilities while maintaining a focused development effort on collaborative innovations.

Artificial Intelligence Integration

Collaborative environments generate diverse content–text documents, code, images, PDF files –that users need to analyze, summarize, translate, or enhance. Modern AI capabilities can dramatically accelerate these tasks, but implementing AI systems requires specialized expertise, significant computational resources, and ongoing model updates. Users expect AI assistance to be contextual and immediate, integrated seamlessly into their collaborative workflows rather than requiring separate tools.

SAGE3 integrates AI services from OpenAI and Azure directly into collaborative tools. OpenAI's API provides access to LLMs like GPT for text generation, code explanation, and natural language processing tasks, while Azure's Cognitive Services offer additional capabilities including image analysis, document processing, and multi-language translation. These services enable users to translate sticky notes into different languages, generate text summaries in Notepad, explain or annotate code within SAGECell, analyze PDF documents, and describe images. By embedding AI capabilities directly into user workflows, SAGE3 transforms collaborative sessions into AI-augmented environments where computational intelligence enhances human collaboration.

Identity Management

Multi-user platforms must provide secure user authentication that integrates with existing institutional systems while minimizing user friction. Managing credentials directly introduces security risks, administrative overhead, and compatibility challenges with diverse organizational identity systems. Authentication systems for collaborative projects are complex, involving federated login, multi-factor authentication, biometric data, and tokens, all of which can create security vulnerabilities that are challenging to address properly.

SAGE3 leverages federated identity management systems to avoid handling user credentials directly, reducing security risks while supporting institutional requirements for privacy, security, and regulatory compliance. Through PassportJS, the system implements OAuth 2.0 authentication with multiple providers: Google OAuth (which can itself federate with users' home organizations), Apple OAuth, and CILogon–an NSF-funded federated identity system that provides researchers access to cyberinfrastructure systems across universities and national laboratories. For service-to-service authentication, SAGE3 uses JSON Web Tokens (JWT) that site administrators can easily generate, enabling secure communication between system components such as Python execution kernels authenticating with the backend.

Media Streaming

Screen sharing across multiple participants necessitates real-time video streaming to all users simultaneously. Implementing this capability demands substantial server resources for video processing and sophisticated routing algorithms to efficiently distribute streams to multiple viewers. Without proper infrastructure, screen sharing can quickly overwhelm server bandwidth and computational capacity, particularly when multiple users share screens simultaneously or when supporting large display walls with high-resolution requirements.

SAGE3 integrates Twilio's Video API (www.twilio.com) to provide scalable screen sharing capabilities that support multiple simultaneous streams across diverse deployment scenarios. Twilio's Selective Forwarding Unit (SFU) technology efficiently manages video stream distribution, decrypting, processing, and re-routing streams to connected users without requiring substantial computational resources from SAGE3 servers. This approach ensures reliable, high-quality screen sharing while allowing SAGE3 to focus development resources on collaborative innovations rather than media infrastructure.

The End-User Tools

Collaborative environments need immediate access to diverse content types and interactive capabilities without forcing users to switch between external tools. Users need to view documents, edit text collaboratively, perform calculations, communicate through chat, and share their screens–all within the same shared workspace. The challenge is providing comprehensive tooling that works seamlessly across different devices and supports real-time multi-user interaction while maintaining consistent behavior and synchronization.

SAGE3 addresses this through an integrated suite of built-in tools (Fig. 15) organized into several categories. Content viewers automatically handle diverse file formats–PDFs, images, videos, text, CSV files, and GIS data–each rendered through dedicated viewers that optimize display for collaborative viewing. We provide a few base components for each tool to use (a window, a toolbar, a data store, etc.), but each tool can leverage any React libraries to build its visual components. In SAGE3 tools, we use widely deployed and tested components to load specific data formats (CSV files, GIS formats, PDF files, etc.) and make extensive use of the Chakra-UI React library of user interface elements (chakra-ui.com) to maintain a consistent visual design across the SAGE environment. The data store system (described in Sect. 5.4.1) handles automatically the bidirectional communications with the backend.

Each tool is built from a set of React components. Text-based collaboration tools including Notepad, CodeEditor, Sticky notes, and SAGECell enable real-time multi-user editing powered by Conflict-Free Replicated Data Types (CRDTs) for seamless synchronization. Communication and coordination utilities such as Calculator, Chat, Drawing, Poll, Timer, and Screensharing support the social aspects of collaboration. Additionally, the Webview tool allows users to integrate external web content and web applications directly into collaborative sessions, enabling custom dashboards and shared web browsing experiences.

Multi-User Tool Design

Creating tools that support simultaneous multi-user interaction presents fundamental challenges not encountered in traditional single-user tools. Each tool must maintain a consistent state across all participants while handling conflicting user actions, managing network latency, and scaling to support varying numbers of concurrent users. Different types of tools require different synchronization strategies, beyond the standard state management: streaming real-time media has different requirements than collaborative text editing or shared web browsing.

Screen Sharing Screen sharing presents a particularly challenging scaling problem: direct peer-to-peer streaming works well for small groups, but becomes prohibitive as participation increases. In a session with 10 participants, a single user sharing their screen must maintain nine simultaneous streams, creating significant performance bottlenecks and system-dependent issues. Server-based approaches can reduce client load but transfer computational and bandwidth demands to the server infrastructure.

SAGE3's screenshare implementation evolved through multiple approaches to address these scaling challenges. Initial WebRTC (webrtc.org) peer-to-peer streaming was replaced with a server-based model where clients stream to a central server for rebroadcasting. Ultimately, we adopted Twilio's Video Cloud Services with Selective Forwarding Unit (SFU) technology, which decrypts, processes, and re-routes streams to connected users while providing scalable performance with minimal client overhead.

Collaborative Text Editing Real-time text editing presents the fundamental challenge of maintaining consistent shared state when multiple users simultaneously edit the same document. Concurrent edits can create conflicts that require sophisticated resolution algorithms–users typing at the same location, deleting text being modified by others, or making overlapping changes. Traditional approaches like operational transformation are complex to implement correctly, while simpler locking mechanisms prevent true concurrent editing.

Text-based tools including Sticky notes, Notepad, and CodeEditor address this challenge using Conflict-Free Replicated Data Types (CRDTs) implemented through the Yjs library [35]. CRDTs provide mathematical guarantees that concurrent edits can be merged consistently across all clients without requiring complex conflict resolution logic, enabling seamless real-time collaborative text editing. These SAGE3 tools use the Homebase/Yjs service that we created to setup peer-to-peer connections between all the clients. Regularly, the current state of the CRDT values are pushed to the Homebase/State service for persistence.

Web Content Synchronization Synchronizing web content across multiple users presents unique challenges because websites often render different content based on user-specific data. Cookies, geolocation, login states, and session data cause identical URLs to display different content for different users. Single-page web applications compound this problem since navigation changes driven by JavaScript may not be reflected in URLs, making it impossible to reliably synchronize navigation events.

The Webview tool addresses these challenges through two approaches. Basic synchronization shares only URLs across clients, ensuring users view the same pages while accepting that content may differ based on individual user contexts. For true content synchronization, SAGE3 employs a VNC-based [36] approach where the server runs a Docker container with a web browser, streaming the rendered output to all clients while routing user interactions back to the server. This guarantees identical web content across all participants, though at the cost of increased server computational load.

SAGECells

Jupyter Notebook is a staple for scientists and their students to perform data analysis. Our user study, "Exploring Organization of Computational Notebook Cells in 2D Space," discovered that although Jupyter displays its code cells linearly, users prefer and are more productive arranging cells spatially [24]. This was the motivation behind SAGECell, a code cell that users can freely position on SAGE boards, grouping cells in a way that aligns with their objectives or enhances their workflow.

To use SAGECell, the user first selects a kernel from the toolbar. The Code Input Box is then be used to enter and edit source code. The Output Box below the Input Box displays text and standard rich media mime-types, such as PNG, JPG, SVG, HTML, with interaction for Plotly and Vega charts. Figure 24 shows the use of SAGECells to process and visualize global surface temperature data.

SAGECell also enables users to switch between multiple Python and R kernels, which enables task "segregation" for enhanced prototyping. For instance, users can independently experiment with or prototype analyses in separate kernels. To uphold greater granularity in terms of privacy among collaborators, access to kernels can be set to "private", thus allowing only owners or authorized collaborators to access the data they hold or use them for executing code. This mechanism also helps safeguard data from accidental overwriting, thereby maintaining data integrity and security.

SAGECells are currently used to discuss data during meetings, teach code to students, perform pair programming, create visualizations, carry out data wrangling, and handle some basic automation tasks. We plan to leverage existing machine learning models to include natural language interactions, allow users to write their own models using data in each room, and offer more ways to think about writing code. We are actively building the mechanisms to see which models are most effective.

In addition to their functionality for data analysis, SAGECell enables users to programmatically "drive" SAGE3 through a dedicated Python interface. With this interface, users can generate and execute macros, construct simple data flows (where the output of one program serves as the input to another), or develop reactive tools (where a variable

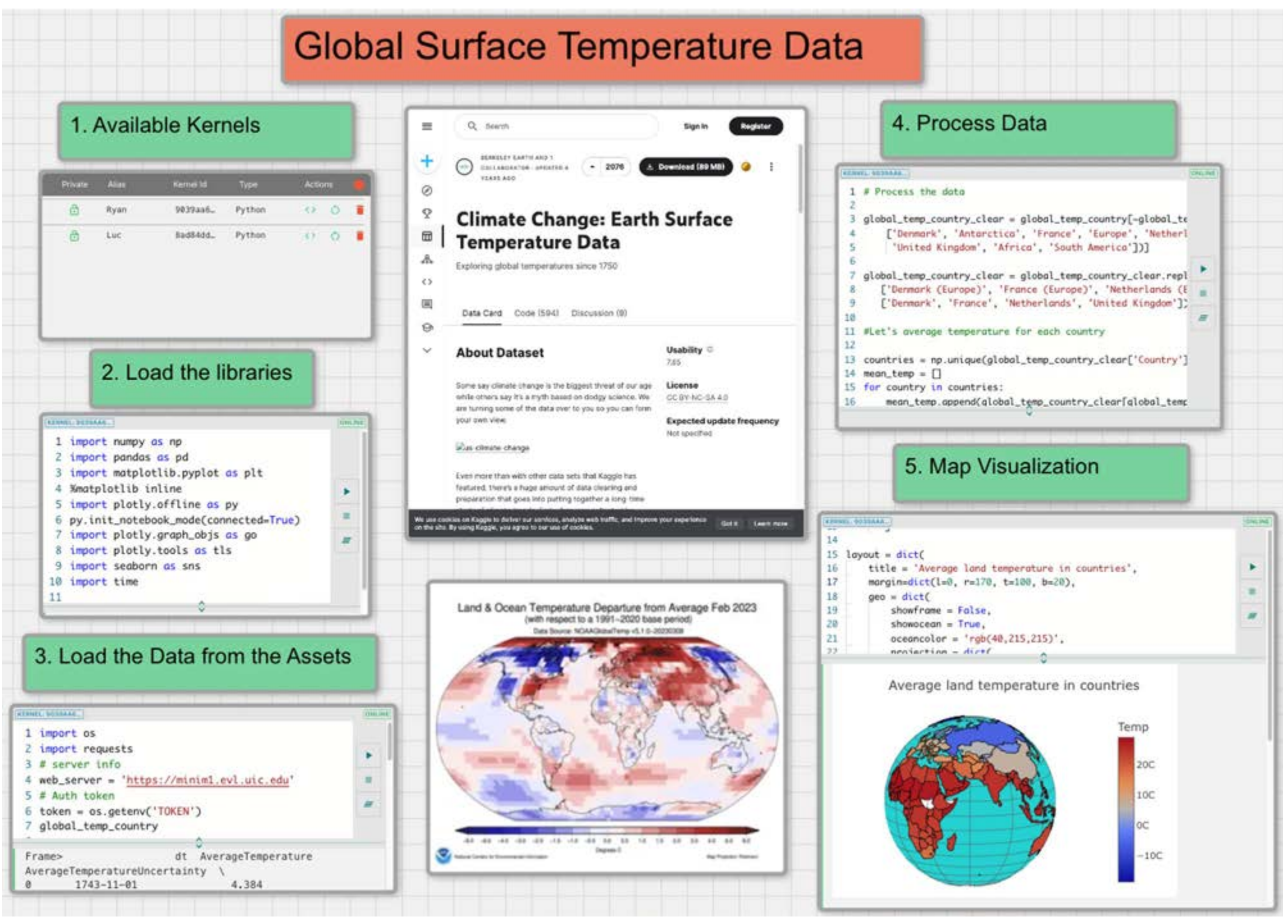

Fig. 24 SAGECell workflow for Global Surface Temperature dataset

is updated when another variable changes). As a result, users can perform any action available through the SAGE3 graphical user interface via the Python interface.

Extensible Plugin Architecture

Collaborative platforms must balance providing comprehensive built-in functionality with the ability to support specialized tools and workflows that cannot be anticipated by platform developers. Users need access to domain-specific tools, custom visualizations, and experimental tools without requiring platform modifications or complex development environments. The challenge is enabling third-party extensions while maintaining security, stability, and collaborative synchronization.

SAGE3 addresses this through a plugin system that allows any web application to be integrated as a collaborative tool. Plugins are web applications written in HTML, JavaScript, CSS, or WebAssembly that run within a sandboxed *iframe* environment with content security policies for safe execution. The SAGE3 `@sage3/sageplugin` JavaScript library provides a communication channel between plugins and the SAGE3 runtime, enabling state synchronization and data exchange while maintaining isolation.

Developers can package web applications into compressed archives and upload them directly to SAGE3 servers, where they become available as collaborative tools alongside native tools. This approach has successfully integrated diverse third-party web applications including D3.js visualizations, Unity3D (unity.com) interactive applications, and Potree (github.com/potree) point cloud renderer (Fig. 15). The plugin architecture lowers development barriers, promotes experimentation, and extends SAGE3's capabilities without compromising core platform stability.

Security and Vulnerability Management

For SAGE2, we partnered with Trusted CI (trustedci.org)–the NSF Cybersecurity Center of Excellence at Indiana University–to design and implement an Identity and Access Management (IAM) approach based on SAGE community feedback. We also benefited from the experience of KLM Royal Dutch Airlines (KLM), an early adopter using SAGE2 for research meetings, who shared security findings that helped us identify and remediate vulnerabilities.

With SAGE3, the security landscape has grown markedly more complex. The platform depends on a layered software stack with extensive transitive dependencies, and vulnerabilities can arise anywhere along this supply chain. Understanding these weaknesses–and how they relate within the dependency graph–is essential to maintaining secure collaborative environments, especially for organizations deploying SAGE on their own infrastructure. Traditional vulnerability-management practices that rely on prose reports or simple network diagrams quickly become unwieldy when confronted with dependency structures spanning hundreds or thousands of components.

For SAGE3 we developed a Software Bill of Materials (SBOM) vulnerability visualization tool that provides intuitive understanding of security risks within the platform's dependencies. The tool analyzes Docker images and other components to identify vulnerable, transitively vulnerable, and safe libraries, presenting this information through advanced visualization techniques.

Three-Dimensional Vulnerability Visualization

Complex dependency networks with vulnerability relationships cannot be effectively represented using traditional two-dimensional network diagrams, which become visually overwhelming and difficult to interpret. Users need to understand not only which components are vulnerable, but also how vulnerabilities propagate through the dependency chain and how different vulnerability types relate to each other. Three-dimensional visualization approaches have been shown to help users better disambiguate complex graphs, particularly when combined with interactive perspective controls [37].

Drawing from our five decade long expertise in data visualization and virtual reality [38], we developed a three-dimensional approach that contextualizes SBOM and Vulnerability Exploitability eXchange (VEX) data through spatial organization and interactive exploration. The visualization approach separates software libraries into vulnerability categories– directly vulnerable components positioned in the top layer (in red), transitively vulnerable components (vulnerable by association) in the middle layer (in yellow), and safe libraries at the bottom layer (in green). Dependencies between libraries are displayed on-demand through animated connections, dramatically reducing visual clutter while maintaining contextual understanding.

Furthermore, we allowed users to select collections of libraries to visualize their individual VEX vulnerability vectors akin to DNA strands that can be compared side by side - a technique borrowed from our prior work in comparative genomics research (BactoGeNIE) described earlier [8]. Lastly we also applied k-means clustering to the VEX CVSS (Common Vulnerability Scoring System) [39] vectors and Principal Component Analysis to provide an overview of the similarity between libraries with respect to their vulnerabilities.

The final visualization tool was deployed on Jetstream2 for review by the team during the development process (Fig. 25).

Emergent Usage Patterns

Previously, when observing how our user community worked with SAGE2, we discovered a number of signature usage patterns. These include: conceptualization (brainstorming), data collection and wrangling, data analysis and visualization, knowledge crystallization, and presentation [1]. SAGE3 enhanced these existing patterns, but has also resulted in new usage patterns.

Enhancing Existing Usage Patterns

Across Space

The classical SAGE1 and SAGE2 collaboration mode was a group of users co-located in front of a display wall and using their laptops to manipulate documents on the wall. They could only see the content on the display wall, not on their laptops. With SAGE3, however, users can see the full extent of the content on every device type, and as a result, have exhibited new modes of collaboration: (1) remote users working on laptops/desktops can collaborate with users in front of display walls (co-located/distributed hybrid); (2) remote users on laptops/desktops can collaborate with other remote users also using laptops/desktops (fully distributed), and (3) locally situated users on laptops/desktops can collaborate with other locally situated users on laptops/desktops (fully co-located).

Across Time

In addition to accommodating use scenarios across space, SAGE3 can accommodate Computer Supported Cooperative Work (CSCW) across time, completing the designed space defined by CSCW quadrants [40]. While most work we have seen falls under synchronous collaboration, the permanence of each SAGE3 board, and the ease with which a board can be dedicated to specific use without compromising the work of other SAGE3 users, means that we also see asynchronous work patterns. Group members can load or update content in between work sessions in order to prepare, communicate, and add referents on the board.

Across Boards

While SAGE2's servers could only support a single group meeting at a time, SAGE3 can support multiple meetings concurrently, as long as there is sufficient computational capacity to support those meetings. More details about usage patterns across space and across time are available in our discussion of approaches for teaching with SAGE3 [11].

Creating New Usage Patterns

2D Programming

SAGE3 enables data collection, wrangling and analysis tasks to be performed using two-dimensionally-arranged code cells (SAGECells) that can be collaboratively viewed

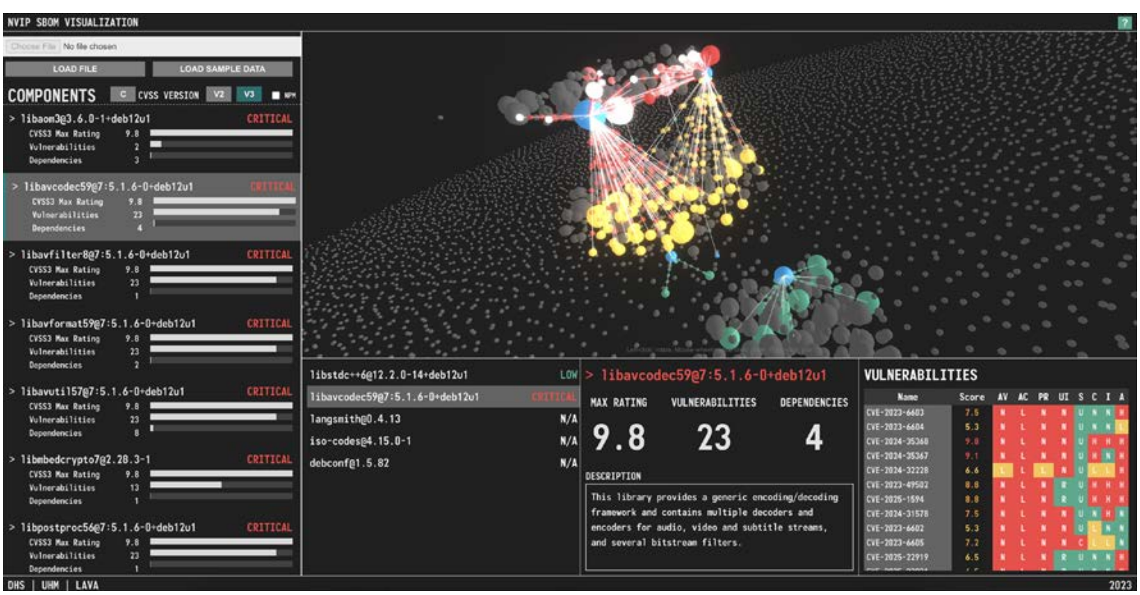

Fig. 25 Software Bill of Materials visualization of the SAGE3 Homebase Docker image. The left panel lists software libraries; the lower center panel shows user-selected items; the middle panel details the current selection; and the right panel displays vulnerabilities with CVSS metrics in a colored matrix. The 3D view groups components into layers: red = vulnerable, yellow = dependent on vulnerable components, green = components with dependencies, and gray = self-contained with no vulnerabilities

and edited. This enables users to simultaneously contribute code to different parts of the data analysis pipeline at the same time.

Viewport Guide

SAGE2's workspace was limited to the resolution of the physical display wall. SAGE3 provides infinitely sized screen spaces (rooms and boards) as well as an infinite number of rooms/boards where users can spread out their content. Often, when a display wall is filled with content, users will search the entire screen to find more open space in which to work. Alternatively, they can create a whole new room/board to work in.

Multi-Dashboard

With SAGE3's ability to support multiple simultaneous collaboration sessions, it became popular for users to build dashboards comprised of visualizations of streaming data such as from sensors. The data sources are often websites that were originally built to provide live streaming data. SAGE3 makes it easy to create custom placements of these visualizations. For example, pictured below (Fig. 26) is a custom dashboard for the Hawaii Climate Mesonet [41], constantly showing data from its network of weather sensors. Mesonet stakeholders can use SAGE3 to create their own custom dashboard layouts by simply launching various charts and then placing them in configurations that best match their needs. This concept won a grand prize in the Michael H. Freilich Visualization Competition at the 2024 American Geophysical Union meeting.

Virtual Reality

The SAGE3 team is often asked whether SAGE can support Virtual Reality (VR) or Extended Reality (XR). To explore this, we conducted research and prototyping [42], and demonstrated that SAGE3 boards can indeed be experienced through VR headsets such as the Meta Quest (Fig. 27). This capability allows users to view boards as though they were standing in front of a large physical display wall, even when working in small, confined spaces such as a hotel room or airplane cabin. In addition, VR makes it possible to surround oneself with multiple boards of varying sizes simultaneously–an arrangement that is difficult, if not impossible, to replicate in the physical world.

Our study revealed a notable finding: participants used roughly twice as much virtual space as physical space, creating and arranging layouts that could not be achieved on a laptop or even a large physical display wall–for example, positioning displays to float freely in midair. Interestingly, this observation echoes one of our earliest insights. In our 2003 study that was a precursor to the original SAGE concept, we found that once users experienced large display walls during meetings, they resisted returning to the traditional single-screen, slide-by-slide presentation format. Instead, they wanted the ability to view more information side by side [43].

Together, these findings highlight a consistent theme: whether in physical or virtual environments, users naturally seek out more *space to think* [15] when given the opportunity. In this sense, our research has come full circle. Looking ahead, we are optimistic about the future. As head-mounted displays continue to become lighter (as in the XReal or Meta Ray-Ban glasses) and achieve ever-higher resolutions, we

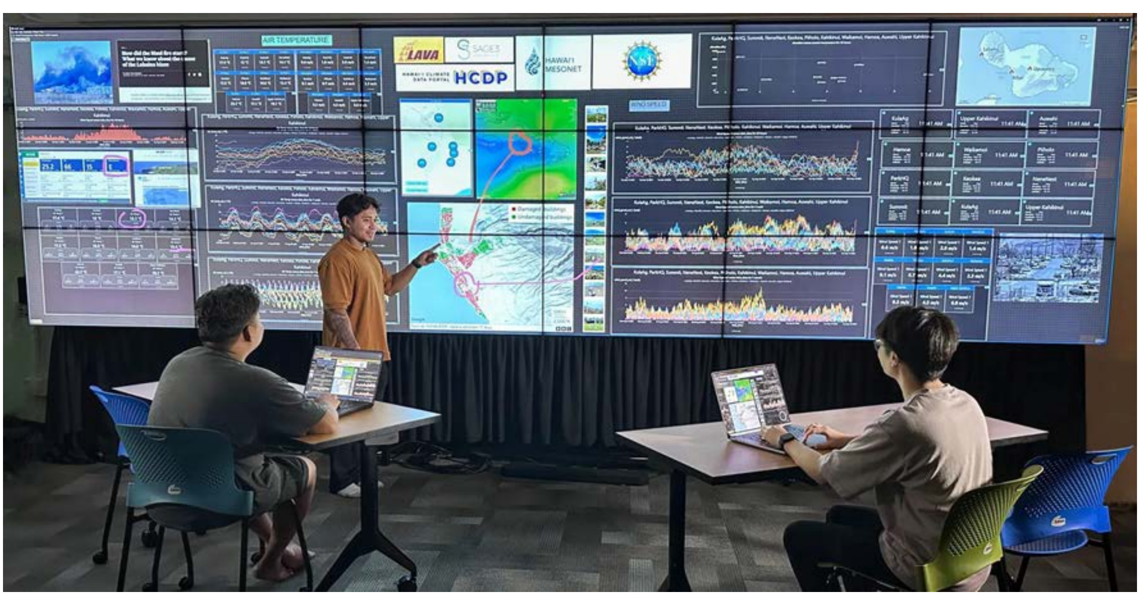

Fig. 26 Dashboard-Students demonstrating a Hawaii fire risk dashboard consisting of data feeds from the Hawaii Climate Mesonet

Fig. 27 Multiple SAGE3 boards viewed on multiple tabs of the Meta Quest Browser in Meta Quest 2

envision SAGE operating seamlessly in augmented reality glasses–moving beyond today's display walls and laptops.

However, this is not the only or even the ultimate application of XR. Our long-term goal is to design interactions that fully leverage the affordances of 3D environments–for instance, enabling users to view and interact with inherently three-dimensional datasets, in ways similar to recent explorations of 3D computational notebooks by In et al. [44].

Research Conducted Across the Evolution of SAGE

As noted in the introduction, all SAGE projects were funded through the National Science Foundation's advanced cyberinfrastructure programs. These programs explicitly prioritized the translation of emerging computer science techniques into robust, production-quality tools, rather than basic research, to provide domain scientists with useful and usable cyberinfrastructure capable of advancing the scientific sensemaking and knowledge-production practices – work that is often significantly more complex and demanding than is typically assumed in research-focused contexts. The nature of these translational challenges, and their implications for the long-term sustainability of cyberinfrastructure, are discussed in greater detail by our prior paper, Belcaid et al. [2].

Each instantiation of SAGE (SAGE1, SAGE2, SAGE3) incorporated state-of-the-art and emerging hardware and software technologies, Computer Supported Collaborative Work (CSCW), Human Computer Interaction (HCI), and evolving trends in how people collaborate in groups, both locally and remotely. In parallel with this translational effort, targeted empirical and technical studies were conducted alongside successive system iterations to examine the feasibility, performance, and collaborative implications of large-scale, spatially organized visual computing environments.

SAGE1: Understanding the Performance Characteristics of Pixel Routing

In the mid-2000s, work on SAGE1 focused on establishing whether software-based pixel streaming could serve as a viable foundation for interactive, ultra-high-resolution, collaborative visualization. The primary research goal was to characterize the performance limits, scalability, and reconfiguration costs associated with dynamically routing pixel streams from distributed rendering resources to compute-cluster nodes that drove the tiled display walls [28].

Through detailed architectural analysis and extensive benchmarking, this work demonstrated that dynamic pixel stream routing could be achieved in software without specialized hardware support while maintaining interactive frame rates at extreme resolutions. SAGE1 sustained multi-gigabit throughput with low latency and achieved over 90% network utilization on both local- and wide-area advanced research networks, even as application windows were moved or resized. These findings validated pixel-based streaming as a practical and scalable approach, directly informing subsequent SAGE architectures that emphasized flexibility, parallel rendering, and dynamic spatial layout.

SAGE2: Enabling New Collaborative Workflows on Shared Displays

Building on the technical foundation established in SAGE1, research conducted in the early 2010s when developing SAGE2 shifted toward understanding how large, shared displays could enable new forms of data-intensive collaboration. A central goal of this phase was to explore how web-based technologies could lower barriers to entry while supporting collaborative workflows that were not achievable with existing tools of the time.

Deployments and observational studies of SAGE2 revealed that its browser-based architecture enabled forms of collaboration, such as lightning collaboration, parallel investigation, and simultaneous multi-application interaction [31]. By enabling multiple users to contribute content concurrently and spatially organize information at scale, SAGE2 supported collaborative practices that were fundamentally different from those afforded by desktop-centric or screen-sharing systems. These findings reinforced the idea that the value of shared displays lay not only in resolution, but in their ability to support parallel, spatial workflows, shaping SAGE2's emphasis on openness, ease of access, and multi-user concurrency.

SAGE2: Empirical Comparison of Spatial Collaboration Practices

Further research during the SAGE2 era explicitly examined how collaborative practices supported by spatially persistent shared displays differed from those enabled by contemporary technologies, including projector-based meetings and Zoom [19].

By analyzing research meetings conducted using SAGE2 on large-scale displays versus traditional projection technologies and Zoom, the study identified distinct modes of content contribution and showed that SAGE2 uniquely supported fully parallel interaction. Participants were able to add, manipulate, and reference content without interrupting ongoing discussion, leading to richer feedback, sustained engagement, and stronger collective memory through persistent spatial traces. In contrast, Zoom and projector-based meetings largely enforced sequential interaction. These findings provided empirical evidence that spatial persistence and parallel contributions fundamentally alter collaborative scientific workflows, reinforcing SAGE2's design direction toward large, continuously evolving shared canvases.

SAGE3: Spatializing Computational Notebooks

Research conducted alongside the development of SAGE3, starting in 2020, extended these ideas into the domain of computational notebooks [24, 45]. The goal of this work was to investigate whether spatialized layouts of notebook cells could better support data science workflows, and to use these findings to inform SAGE3's version of computational notebooks (SAGECells).

Controlled user studies demonstrated that 2D, multi-column notebook layouts significantly improved efficiency and usability for common data science tasks, including comparison, parameter exploration, and iterative analysis, without compromising correctness. Participants consistently preferred spatial layouts, citing reduced scrolling, improved cognitive offloading, and more natural side-by-side reasoning. These results validated the hypothesis that computational notebooks benefit from spatial organization and directly motivated SAGE3's support for spatialized, collaborative notebooks embedded within a free-form canvas.

SAGE3: Interaction Survey Study

Section 3 compared SAGE3 with a range of ideation tools. In designing SAGE3's user interaction paradigm, we surveyed and analyzed interaction techniques employed by these systems, with the goal of adopting an interaction scheme that would be familiar to most users while remaining capable of supporting the broader range of display environments addressed by SAGE3, including tablets, laptops, and large display walls. The bottom row of Fig. 28 shows SAGE3's control scheme.

A central challenge was the need to support interaction across devices with markedly different scales and input characteristics, particularly ultra-high-resolution display walls, which are not typically supported by comparison systems. Touch interaction at this scale is especially difficult due to reduced spatial accuracy of wall-sized touch technologies, and the increased likelihood of unintended touches resulting from incidental contact by the user's hands or fingers. To address these issues, we introduced a modified interaction scheme for touch-enabled display walls that required users to explicitly switch between navigation (moving across the wall) and selection (interacting with objects). Although this additional mode-switching step introduced some overhead, it represented a practical compromise given the competing constraints imposed by accuracy, scalability, and usability.

To mitigate the cost of mode switching and reduce physical effort, we also implemented a quick-access menu that appears at the location of a two-finger tap (Fig. 29). This menu provides rapid access to interaction modes and frequently used commands without requiring users to reach the fixed menu bar at the bottom of the display wall.

The above are but five examples of research we conducted to inform the design of the three SAGEs. Others include, a study of a how users would use information spaces that are fully situated in virtual reality [42]; an evaluation of voice interaction interfaces in SAGE2 and SAGE3 [46, 47]; and a study of user behavior when sitting at the extreme of large display walls [48].

	Touch Screen Interaction			Mouse Interaction		
System	Moving Around Canvas	Zooming	Lasso	Moving Around Canvas	Zooming	Lasso
Excalidraw	Two finger scroll	CMD/Ctrl + Scroll up/down or Pinch to Zoom	Left Click + Drag	Middle Click + Drag Space + Left Click + Drag	CMD/Ctrl + Scroll up/down	Left Click + Drag
Draw.io	Two finger scroll	CMD/Ctrl + Scroll up/down or Pinch to Zoom	Left Click + Drag	Middle Click + Drag Space + Left Click + Drag	CMD/Ctrl + Scroll up/down	Left Click + Drag
Figma	Two finger scroll	CMD/Ctrl + Scroll up/down or Pinch to Zoom	Left Click + Drag	Middle Click + Drag Space + Left Click + Drag	CMD/Ctrl + Scroll up/down	Left Click + Drag
Lucidcharts	Two finger scroll	CMD/Ctrl + Scroll up/down or Pinch to Zoom	Left Click + Drag	Middle Click + Drag Space + Left Click + Drag Right Click + Drag	Scrolling CMD/Ctrl + Scroll up/down	Left Click + Drag
Miro	Two finger scroll	CMD/Ctrl + Scroll up/down or Pinch to Zoom	Left Click + Drag	Middle Click + Drag Space + Left Click + Drag Right Click + Drag	CMD/Ctrl + Scroll up/down	Left Click + Drag
Creately	Two finger scroll	CMD/Ctrl + Scroll up/down or Pinch to Zoom	Left Click + Drag	Space + Left Click + Drag Right Click + Drag	CMD/Ctrl + Scroll up/down	Left Click + Drag
Flowmapp	Two finger scroll	Pinch to Zoom	N/A	Left Click + Drag	Ctrl + Scroll up/down	N/A
SAGE3	Two finger scroll (for tablets) Move mode & then 1 finger scroll (for touch walls)	CMD/Ctrl + Scroll up/down or Pinch to Zoom	Left Click + Drag	Middle Click + Drag Space + Left Click + Drag Right Click + Drag	CMD/Ctrl + Scroll up/down	Left Click + Drag

Fig. 28 Comparison of interaction schemes of SAGE3 (bottom) with other related systems. Each control scheme assignment is color coded for easy comparison across systems and functions

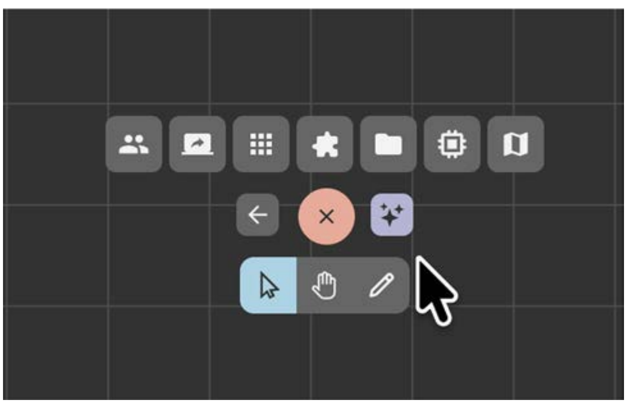

Fig. 29 SAGE3's Quick Access Menu to provide touch wall users with rapid access to SAGE3's menu bar without having to constantly reach for it at the bottom of the display wall (as shown in Fig. 15A)

Conclusion

Over the course of two decades, the SAGE Suite has evolved from an experiment in tiled display walls into a robust, open, and extensible platform that redefines how distributed teams collaborate with data, media, and one another. Through SAGE1, SAGE2, and now SAGE3, the project has consistently demonstrated the value of translational science – transforming advances in visualization, networking, and human–computer interaction into usable, scalable systems that have reached hundreds of organizations worldwide and supported discovery across disciplines ranging from biology and atmospheric science to health care, disaster management, education, and creative media.

SAGE3 marks a culmination and a new beginning. Its "spatial thinking operating system" extends beyond prior generations by unifying synchronous and asynchronous collaboration across laptops and large displays, embedding data science and AI capabilities directly into shared canvases, and enabling multi-user, multi-AI workflows grounded in embodied cognition. These advances illustrate how spatial reasoning, collective intelligence, and computational augmentation can be purposefully integrated into everyday scientific and educational practice.

At the same time, the trajectory of the SAGE Suite surfaces enduring systemic challenges: sustaining open scientific software, recognizing it as an instrument of discovery in academic incentives, and ensuring that collaborative cyberinfrastructure keeps pace with rapidly changing user expectations and technological ecosystems. Addressing these challenges requires not only technical innovation but also cultural and institutional commitment.

Looking forward, we see SAGE3 as both an infrastructure and an invitation – a living platform for advancing research, pedagogy, and public engagement, while also serving as a testbed for the next generation of human–AI collaboration. By coupling embodied spatial cognition with extensible architectures and open community-driven development, SAGE demonstrates how scientific gateways can expand beyond portals into shared cognitive environments that help society think together at scale.

While commercial systems are beginning to offer overlapping capabilities, they remain proprietary and opaque. SAGE3, in contrast, provides an open, research-driven model of collaborative cyberinfrastructure. In doing so, it advances the state of practice and offers a blueprint for how thoughtfully designed systems can sustain and amplify the future of scientific collaboration.

Author Contributions All authors contributed equally to this paper.

Funding The SAGE3 project is funded in part by the National Science Foundation awards ACI 2004014, 2003800, 2003387, OIA 2149133, and NAIRR Pilot award 240471.

Data Availability No new data were collected or generated for this paper. SAGE3 can be downloaded from sage3.sagecommons.org. The SAGE3 source code is available at sage-3.github.io. Source codes for all three SAGEs are available at sagecommons.org.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no Conflict of interest.

Informed Consent All photographs included in this manuscript depict public or educational settings and are used for illustrative and historical purposes. The individuals shown are not research subjects, are not identified or analyzed, and no sensitive personal information is revealed. In accordance with Springer Nature's policies, informed consent was therefore not required.

Research Involving Human/Animals No new data were collected or generated for this paper.

References

1. Leigh J, Kobayashi D, Kirshenbaum N, Wooton T, Gonzalez A, Renambot L, Johnson A, Brown M, Burks A, Bharadwaj K. Usage patterns of wideband display environments in e-science research, development and training. In: 2019 15th International Conference on eScience (eScience), pp. 301–310. IEEE, San Diego, California, USA (2019)
2. Belcaid M, Leigh J, Theriot R, Kirshenbaum N, Tabalba R, Rogers M, et al. Reflecting on the Scalable Adaptive Graphics Environment Team's 20-year translational research endeavor in digital collaboration tools. Comput Sci Eng. 2023;25:50–6. https://doi.org/10.1109/MCSE.2023.3297753.
3. Richmond K, Febretti A, Gulati S, Flesher C, Hogan B, Murarka Kuhlman, Sridharan M, Johnson A, Stone Priscu, Doran: Sub-ice exploration of an antarctic lake: Results from the endurance project. In: 17th International Symposium on Unmanned Untethered Submersible Technology (2011)
4. Febretti A, Richmond K, Gulati S, Flesher C, Hogan B, Johnson A, Stone W, Priscu J, Doran P. Poisson reconstruction of extreme submersed environments: the endurance exploration of an under-ice antarctic lake. In: Proceedings of the 8th International Symposium on Visual Computing (ISVC), pp. 425–434. Springer, Rethymnon, Crete, Greece (2012). https://doi.org/10.1007/978-3-642-33179-4_38
5. Febretti A, Nishimoto A, Thigpen T, Talandis J, Long L, Pirtle JD, Peterka T, Verlo A, Brown M, Plepys D, Sandin D, Renambot L, Johnson A, Leigh J. CAVE2: a hybrid reality environment for immersive simulation and information analysis. In: Dolinsky, M., McDowall, I.E. (eds.) The Engineering Reality of Virtual Reality 2013. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 8649, p. 864903 (2013).https://doi.org/10.1117/12.2005484
6. Guillermo C, Wooton BT, Kobayashi D. Landsage: A decision support tool for monitoring and mitigation of natural disasters. In: Practice and Experience in Advanced Research Computing 2020: Catch the Wave. PEARC '20, pp. 549–550. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3311790.3404543
7. Keenan G, Dunn Lopez K, Yao Y, Sousa V, Stifter J, Febretti A, et al. Toward meaningful care plan clinical decision support: feasibility and effects of a simulated pilot study. Nurs Res. 2017;66(5):388–98. https://doi.org/10.1097/NNR.0000000000000234.
8. Aurisano J, Reda K, Johnson A. Bactogenie: a large-scale comparative genome visualization for big displays. BMC Bioinformatics. 2015;16(Suppl 11):6. https://doi.org/10.1186/1471-2105-16-S11-S6.
9. CineGrid@Disney (2010). https://www2.evl.uic.edu/entry.php?id=999
10. Kobayashi D, Su S, Bravo L, Leigh J, Shires D. Parasage: Scalable web-based scientific visualization for ultra resolution display environment. In: IEEE VIS, Baltimore, Maryland (2016). Poster
11. Harden J, Kirshenbaum N, Tabalba R, Theriot R, Rogers M, Belcaid M, North C, Renambot L, Long L, Johnson A, Space to teach: Content-rich canvases for visually-intensive education. In: 2024 IEEE VIS Workshop on Visualization Education, Literacy, and Activities (EduVIS), pp. 51–59 (2024). IEEE
12. Chattopadhyay S, Prasad I, Henley AZ, Sarma A, Barik T. What's wrong with computational notebooks? pain points, needs, and design opportunities. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–12. ACM, Honolulu, HI, USA (2020)
13. Tan DS, Gergle D, Scupelli P, Pausch R. With similar visual angles, larger displays improve spatial performance. In: Cockton, G., Korhonen, P. (eds.) Proceedings of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, April 5-10, 2003, pp. 217–224. ACM, Ft. Lauderdale, Florida, USA (2003). https://doi.org/10.1145/642611.642650
14. Ball R, North C. Effects of tiled high-resolution display on basic visualization and navigation tasks. In: Veer, G.C., Gale, C. (eds.) Extended Abstracts Proceedings of the 2005 Conference on Human Factors in Computing Systems, CHI 2005, April 2-7, 2005, pp. 1196–1199. ACM, Portland, Oregon, USA (2005). https://doi.org/10.1145/1056808.1056875
15. Andrews C, Endert A, North C. Space to think: large high-resolution displays for sensemaking. In: Mynatt, E.D., Schoner, D., Fitzpatrick, G., Hudson, S.E., Edwards, W.K., Rodden, T. (eds.) Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, April 10-15, 2010, pp.

55–64. ACM, Atlanta, Georgia, USA (2010). https://doi.org/10.1145/1753326.1753336
16. Bi X, Balakrishnan R. Comparing usage of a large high-resolution display to single or dual desktop displays for daily work. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '09, pp. 1005–1014. ACM, New York, NY, USA (2009). https://doi.org/10.1145/1518701.1518855
17. Andrews C, Endert A, Yost B, North C. Information visualization on large, high-resolution displays: issues, challenges, and opportunities. Inf Vis. 2011;10(4):341–55. https://doi.org/10.1177/1473871611415997.
18. Knudsen S, Jakobsen MR, Hornbæk K. An exploratory study of how abundant display space may support data analysis. In: Malmborg, L., Pederson, T. (eds.) Nordic Conference on Human-Computer Interaction, NordiCHI '12, October 14-17, 2012, pp. 558–567. ACM, Copenhagen, Denmark (2012). https://doi.org/10.1145/2399016.2399102
19. Kirshenbaum N, Davidson K, Harden J, North C, Kobayashi D, Theriot R, Tabalba Jr RS, Rogers ML, Belcaid M, Burks AT. Traces of time through space: Advantages of creating complex canvases in collaborative meetings. Proceedings of the ACM on Human-Computer Interaction 5(ISS), 1–20 (2021)
20. SGCI: Science Gateway Community Institute. SGX3. https://sciencegateways.org
21. SAGE3 (2023). https://github.com/SAGE-3/next
22. Olson GM, Olson JS. Distance matters human-computer interaction. Hum Comput Interact. 2000;15(2–3):139–78.
23. Horst W, Matthews B. Live prototyping. In: Proceedings of the 2016 ACM Conference on Designing Interactive Systems, pp. 632–642. ACM, Brisbane QLD Australia (2016)
24. Harden J, Christman E, Kirshenbaum N, Wenskovitch J, Leigh J, North C. Exploring organization of computational notebook cells in 2d space. In: 2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 1–6. IEEE, Rome, Italy (2022)
25. Deelman E, Vahi K, Rynge M, Mayani R, Silva RF, Papadimitriou G, et al. The evolution of the pegasus workflow management software. Comput Sci Eng. 2019;21(4):22–36. https://doi.org/10.1109/MCSE.2019.2919690.
26. Stubbs J, Cardone R, Packard M, Jamthe A, Padhy S, Terry S, Looney J, Meiring J, Black S, Dahan M, Cleveland S, Jacobs G. Tapis: An api platform for reproducible, distributed computational research. In: Advances in Information and Communication. Advances in Intelligent Systems and Computing, vol. 1363, pp. 878–900. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-73100-7_61
27. Smarr LL, Chien AA, DeFanti T, Leigh J, Papadopoulos PM. The optiputer. Commun ACM. 2003;46(11):58–67. https://doi.org/10.1145/948383.948410.
28. Jeong B, Renambot L, Jagodic R, Singh R, Aguilera J, Johnson A, Leigh J. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing. SC '06, p. 108. Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1188455.1188568
29. Herrera A, NVIDIA GRID vGPU: Delivering Scalable Graphics-Rich Virtual Desktops (2015). NVIDIA Corporation. https://images.nvidia.com/content/pdf/grid/whitepaper/NVIDIA-GRID-WHITEPAPER- vGPU-Delivering-Scalable- Graphics-Rich-Virtual-Desktops.pdf
30. Renambot L, Marrinan T, Aurisano J, Nishimoto A, Mateevitsi VA, Bharadwaj K, et al. SAGE2: a collaboration portal for scalable resolution displays. Future Gener Comput Syst. 2016;54:296–305. https://doi.org/10.1016/j.future.2015.05.014.
31. Marrinan T, Aurisano J, Nishimoto A, Bharadwaj K, Mateevitsi VA, Renambot L, Long L, Johnson AE, Leigh J. SAGE2: A new approach for data intensive collaboration using scalable resolution shared displays. In: Bertino, E., Chen, S., Aberer, K., Krishnamurthy, P., Kantarcioglu, M. (eds.) 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2014, October 22-25, 2014, pp. 177–186. ICST / IEEE, Miami, Florida, USA (2014). https://doi.org/10.4108/icst.collaboratecom.2014.257337 .
32. Shapiro M, Preguiça N, Baquero C, Zawirski M. Conflict-free replicated data types. In: Défago X, Petit F, Villain V, editors. Stabilization, Safety, and Security of Distributed Systems. Berlin, Heidelberg: Springer; 2011. p. 386–400.
33. Redis Benchmark. https://redis.io/docs/latest/operate/oss_and_stack/management/optimization/benchmarks
34. Hintjens P. ZeroMQ: Messaging for Many Applications. Sebastopol, CA: O'Reilly Media, Inc.; 2013.
35. Nicolaescu P, Jahns K, Derntl M, Klamma R. Yjs: A framework for near real-time p2p shared editing on arbitrary data types. In: Engineering the Web in the Big Data Era: 15th International Conference, ICWE 2015. Springer, Rotterdam, The Netherlands (2015). https://doi.org/10.1007/978-3-319-19890-3_55
36. Richardson T, Stafford-Fraser Q, Wood KR, Hopper A. Virtual network computing. IEEE Internet Comput. 1998;2(1):33–8. https://doi.org/10.1109/4236.656066.
37. Ware C, Mitchell P. Visualizing graphs in three dimensions. ACM Trans Appl Percept. 2008. https://doi.org/10.1145/1279640.1279642.
38. Johnson AE, Renambot L, Marai GE, Tsoupikova D, Papka ME, Long L, et al. Electronic visualization laboratory's 50th anniversary retrospective: look to the future, build on the past. PRESENCE: Virtual and Augmented Reality. 2024;33:77–127. https://doi.org/10.1162/pres_a_00421.
39. Feutrill A, Ranathunga D, Yarom Y, Roughan M. The effect of common vulnerability scoring system metrics on vulnerability exploit delay. In: 2018 Sixth International Symposium on Computing and Networking (CANDAR), pp. 1–10 (2018)https://doi.org/10.1109/CANDAR.2018.00009
40. Borghoff UM, Schlichter JH. Computer-Supported Cooperative Work. Berlin, Heidelberg: Springer; 2000. p. 87–141. https://doi.org/10.1007/978-3-662-04232-8_2.
41. Giambelluca T. Hawaii Mesonet. Hawai'i Climate Data Portal. https://www.hawaii.edu/climate-data-portal/hawaii-mesonet
42. Kobayashi D, Kirshenbaum N, Tabalba RS, Theriot R, Leigh J. Translating the benefits of wide-band display environments into an xr space. In: Symposium on Spatial User Interaction, pp. 1–11. ACM, New York, NY, USA (2021)
43. Park K, Renambot L, Leigh J, Johnson A. The impact of display-rich environments for enhancing task parallelism and group awareness in advanced collaboration environments (2003)
44. In S, Krokos E, Whitley K, North C, Yang Y. Evaluating navigation and comparison performance of computational notebooks on desktop and in virtual reality. In: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, pp. 1–15 (2024)
45. Harden J, Christman E, Kirshenbaum N, Belcaid M, Leigh J, North C. "there is no reason anybody should be using 1d anymore": Design and evaluation of 2d jupyter notebooks. https://openreview.net/forum? id= Gkogn48LeI, ??? (2023)
46. Kobayashi D, Ready M, Gonzalez Martinez A, Kirshenbaum N, Seto-Mook T, Leigh J, Haga J. Sage river disaster information (sagerdi): Demonstrating application data sharing in sage2. In: Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces. ISS '18, pp. 33–42. ACM, New York, NY, USA (2018).https://doi.org/10.1145/3279778.3279798 .
47. Tabalba J, Roderick S Lee, CJ, Tran G, Kirshenbaum N, Leigh J. A pragmatics-based approach to proactive digital assistants for data exploration. In: Proceedings of the 7th ACM Conference on

Conversational User Interfaces. CUI '25. Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3719160.3736632

48. Kirshenbaum N, Kobayashi D, Gonzalez A, Leigh J. On a side note: Observations on using digital notes on a large display with users sitting at extreme sides. In: Proceedings of the 8th ACM International Symposium on Pervasive Displays (PerDis '19). ACM, New York, NY, USA (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affil tions

Jason Leigh[1] · Luc Renambot[2] · Ryan Theriot[1] · Lance Long[2] · Christopher Lee[1] · Jesse Harden[4] · Nurit Kirshenbaum[1] · Mahdi Belcaid[1] · Chris North[3] · Andrew Johnson[2] · Maxine Brown[2]

✉ Jason Leigh
leighj@hawaii.edu

Luc Renambot
renambot@uic.edu

Ryan Theriot
rtheriot@hawaii.edu

Lance Long
llong4@uic.edu

Christopher Lee
clee48@hawaii.edu

Jesse Harden
jharden@radford.edu

Nurit Kirshenbaum
nuritk@hawaii.edu

Mahdi Belcaid
mahdi@hawaii.edu

Chris North
north@cs.vt.edu

Andrew Johnson
ajohnson@uic.edu

Maxine Brown
maxine@uic.edu

[1] Department of Information and Computer Sciences, Laboratory for Advanced Visualization and Applications, University of Hawaii at Manoa, 1680 East West Rd., Honolulu 96822, Hawaii, USA

[2] Department of Computer Science, Electronic Visualization Laboratory, University of Illinois, Chicago, 850 W. Taylor St., Chicago 60607, Illinois, USA

[3] Department of Computer Science, Center for Human-Computer Interaction, Virginia Tech, 3625 Potomac Ave., Blacksburg 24061, Virginia, USA

[4] School of Computing and Information Sciences, Radford University, 905 East Main St., Radford 24142, Virginia, USA