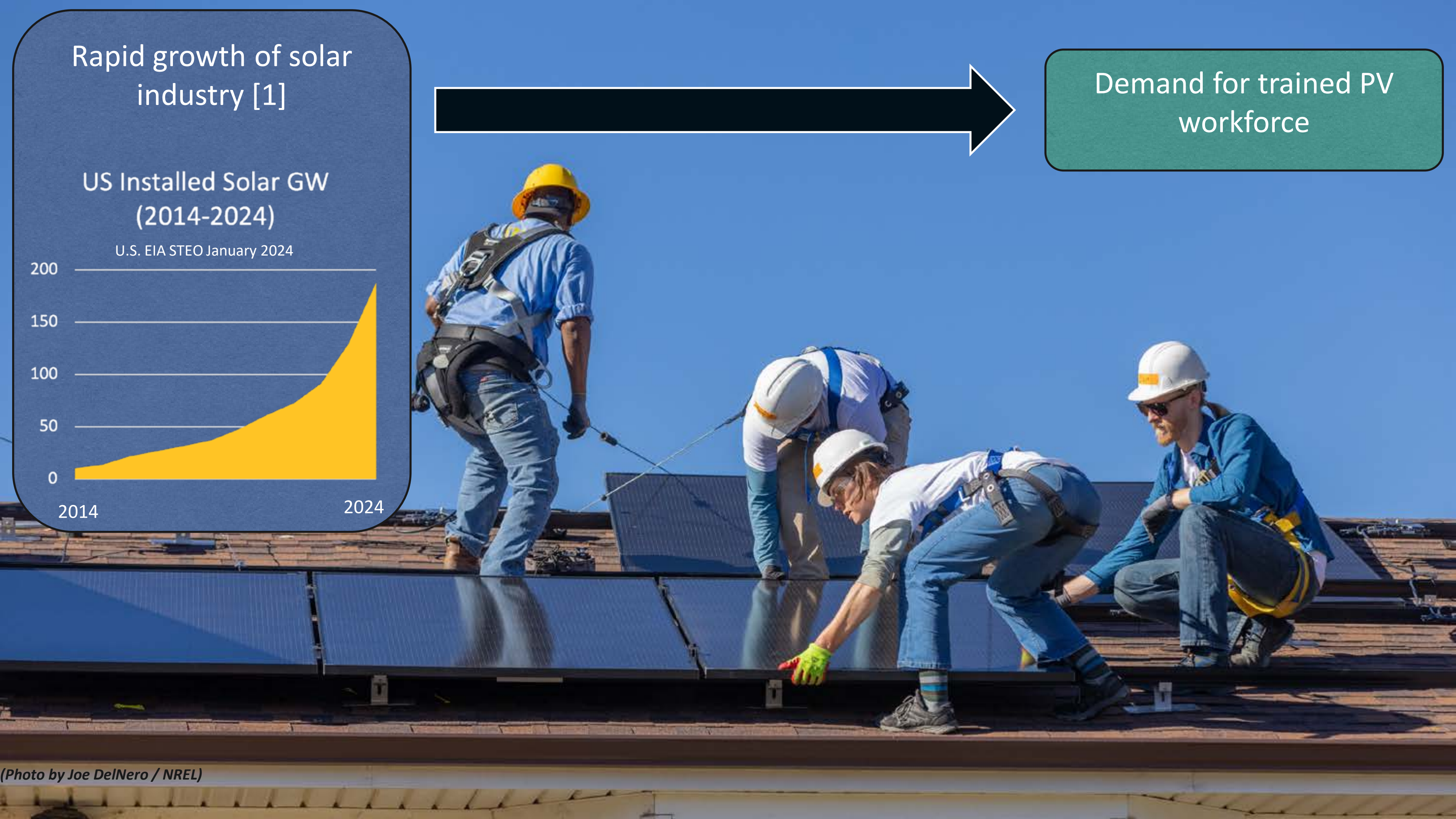
Situated Visualization of Photovoltaic Module Performance for Workforce Development

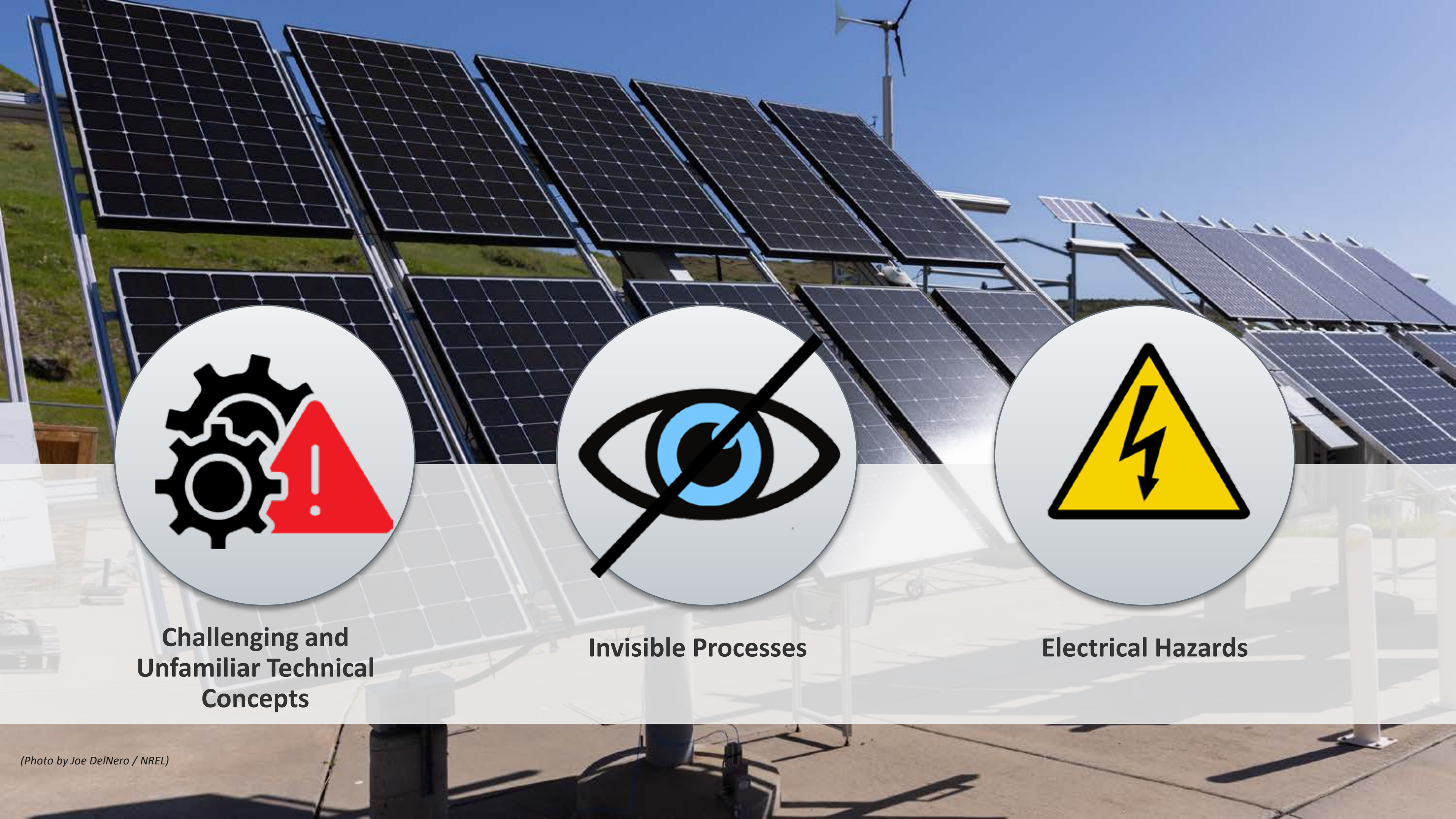
Nicholas Brunhart-Lupo, Kenny Gruchalla, Laurie Williams, Steven Elias

Kenny Gruchalla
Energy Visualization 2024
October 14, 2024

Photo by Dennis Schroeder, NREL 55200



(Photo by Joe DelNero / NREL)



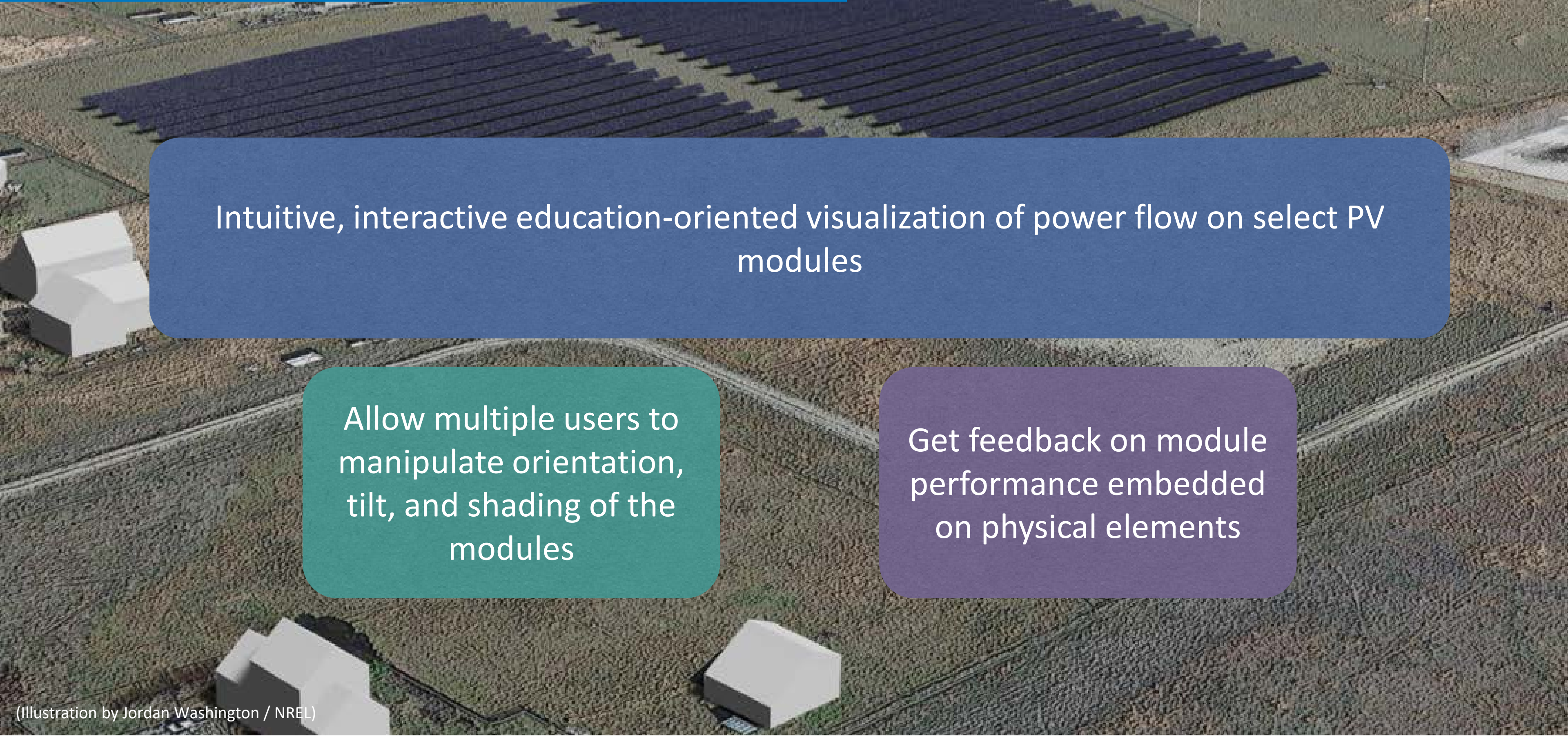
(Photo by Joe DelNero / NREL)

AR, VR, and Situated Visualization can help

- Shown to improve learning gains and motivation [2]
- Improve students' attitudes to technical environments [3]
- Contextualizing visualizations and presentations on physical equipment [4].
- Provide additional degrees of freedom to represent and communicate complex technical concepts [5].

Photo by Dennis Schroeder / NREL)

Objective



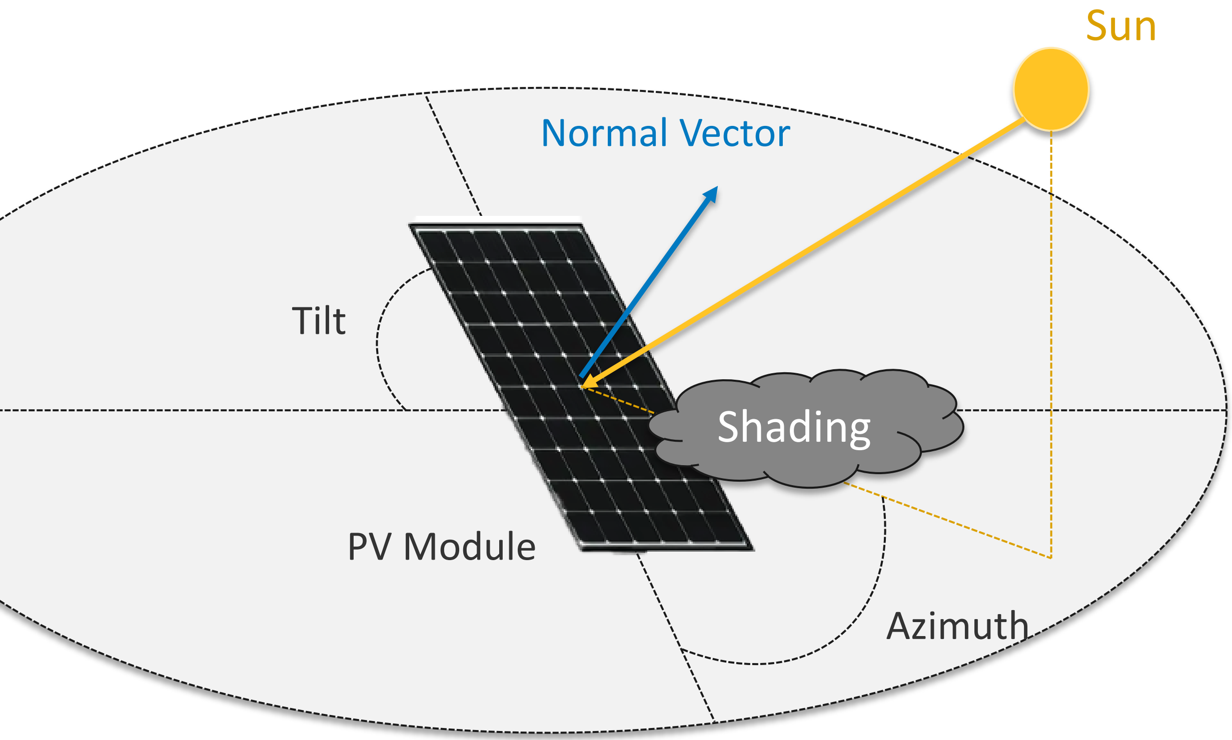
Intuitive, interactive education-oriented visualization of power flow on select PV modules

Allow multiple users to manipulate orientation, tilt, and shading of the modules

Get feedback on module performance embedded on physical elements

(Illustration by Jordan Washington / NREL)

PV Module



Position a Proxy Sun:

- Time of year
- Installation Geolocation

Physically Manipulate:

- Module angles
  - Tilt
  - Azimuth
- Module Shading

Compare & Contrast:

- Different module technologies

Design - Principles

D1

Simplicity

- Simplicity in system design
- Remove extraneous interactions
- Emphasize easily-configurable and disruption-resilient components

D2

Adaptability

- Commodization of AR and VR
- New devices/updates are frequent
- Solution must be hardware agnostic

D3

User Heterogeneity

- Group participation in education
- Multiple users on multiple devices
- Must support multi-user collaboration

D4

Content Maintainability

- Evolving content
- New modules, new hardware
- Configuration changes

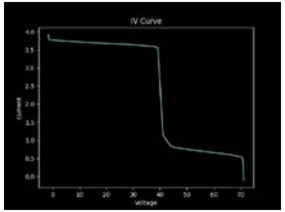
Visualization

- Power flow across cells with tubes and arrows.

- No flow => spherical glyphs

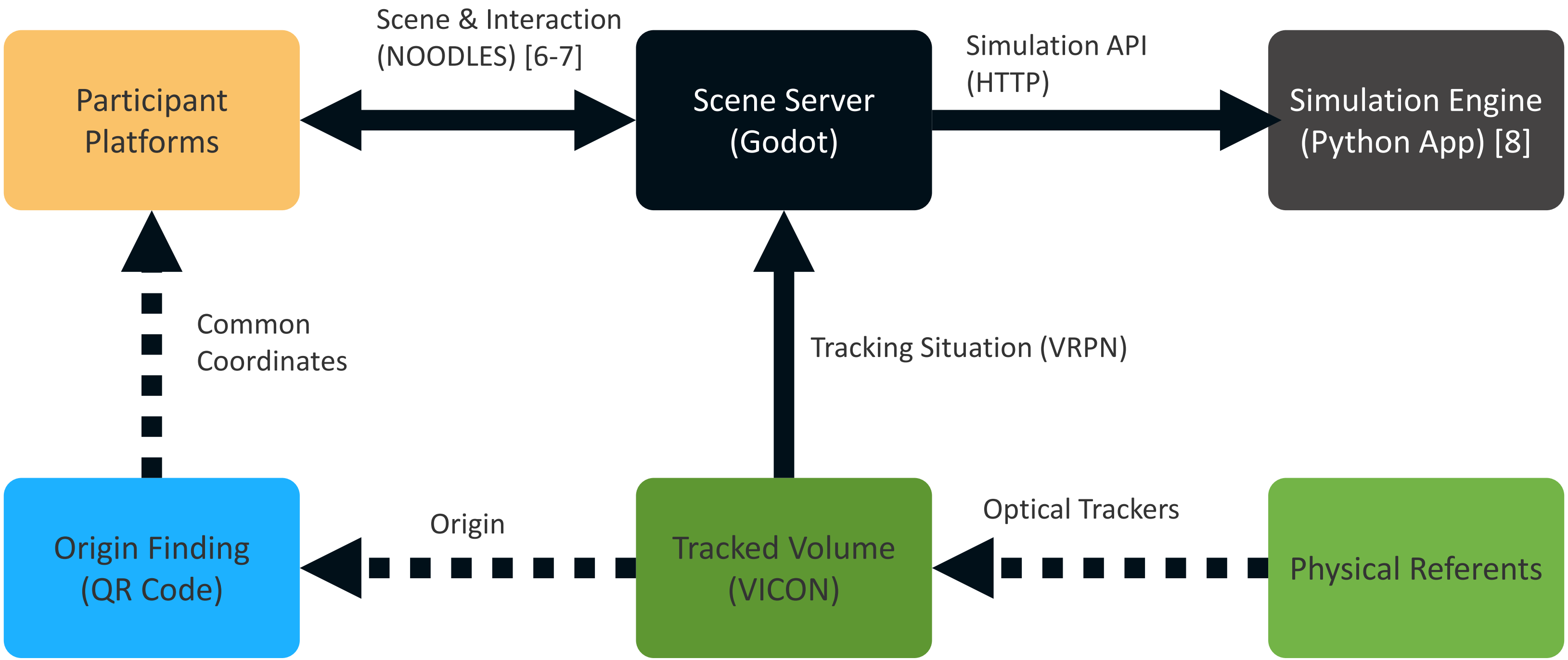
- Yellow overlay for unshaded cells

- Real-time IV plot situated above module



(Photo by Kenny Gruchalla/ NREL)

Approach



Participant Platforms
Scene & Interaction (NOODLES) [6-7]
Scene Server (Godot)
Simulation API (HTTP)
Simulation Engine (Python App) [8]
Common Coordinates
Tracking Situation (VRPN)
Origin Finding (QR Code)
Origin
Tracked Volume (VICON)
Optical Trackers
Physical Referents

Simulation Engine

- Serves scenario simulation requests
- Implemented in Python using Pvlib [8] and Flask
  - Consumes module type, module and sun positioning, PV module occlusion, and temperatures.
  - Produces flow, current and voltages
  - Also produces proper sun-sky position for graphical representation of sun
- Minimal-state HTTP API promotes simplicity
- Python chosen for maintainability

D1

D4

```
class PVSystem:
    def simulate(self, irradiance, shaded_cells=[], temp_unshaded=25, temp_shaded=35):
            # Convert a single irradiance dict to a list of irradiance dicts for each string
        if isinstance(irradiance, pd.DataFrame):
            irradiance = [irradiance] * self.num_strings

        self.shaded_cells = shaded_cells

        # Initialize containers for Ee (effective irradiance) and cell_temps for each string a
        Ee_multi = [[] for _ in range(self.num_strings)]
        cell_temps_multi = [[] for _ in range(self.num_strings)]

        for string_idx, irr in enumerate(irradiance):
            # Get the number of modules for the current string
            if not isinstance(self.num_mods, list):
                num_mod = self.num_mods
            else:
                num_mod = self.num_mods[string_idx]
            # In case of bifacial arrays, if a single module, we will do the naive way
            # In theory--rear irradiance is sum of diffuse, reflected (albedo), and direct rad
            Ee_front = irr["poa_global"][0]
            Ee_rear = 0
            if self.module_params["Bifacial"]:
                Ee_rear = irr["poa_diffuse"][0]

            # Combine front and rear irradiance for bifacial modules
            Ee_total = Ee_front + Ee_rear

            # Convert irradiance W/m2 to "Suns" for pvmismatch
            Ee = np.full(self.cell_layout.shape, Ee_total / IRRADIANCE_CONVERSION_FACTOR)

            # Create array of cell temperatures
            cell_temps = np.full(self.cell_layout.shape, temp_unshaded)

            # Handling shaded cells
            # Ensure shaded_cells is a nested list if multiple modules
            if shaded_cells and any(isinstance(i, list) for i in shaded_cells):
                for mod_idx in range(num_mod):
                    # Copy current Ee and cell_temps for the current module
                    Ee_mod = np.copy(Ee)
                    cell_temps_mod = np.copy(cell_temps)

                     # Check if there are shaded cells for the current module of the current s
                    if string_idx < len(shaded_cells) and mod_idx < len(shaded_cells[string_id
                        tmp_shaded = shaded_cells[string_idx][mod_idx]
                        indices = get_cell_indices(self.cell_layout, tmp_shaded)
                        for idx in indices:
```

Physical Referents

- Incorporated a full-cell, a half-cell, and a bifacial module.
- Mounted on mobile, motorized tilt tables
- Tilts (0° - 90°) and rotates
- Incorporated a shade referent proxy to simulate building or tree
- Use of referent proxies and real-world control allow for simple modification of complex scenarios

D1

Tracked Volume

- Tracks the position and orientation of all physical referents
- Use Virtual Reality Peripheral Network (VRPN) to communicate position
  - Device and network-agnostic open-source framework
- Served through a separate server application
- Ensures robust and reliable tracking, adaptable to future devices

D2

Origin Finding

- Mobile devices operate with independent coordinate system
- Difficult to coordinate different devices and tracking systems
- Counter with QR code placed at the origin of tracker
- Mobile devices establish the root of the graphics scene by simply observing code.
- Does not rely on brand-specific solutions
- QR codes are simple to deploy and maintain

D2

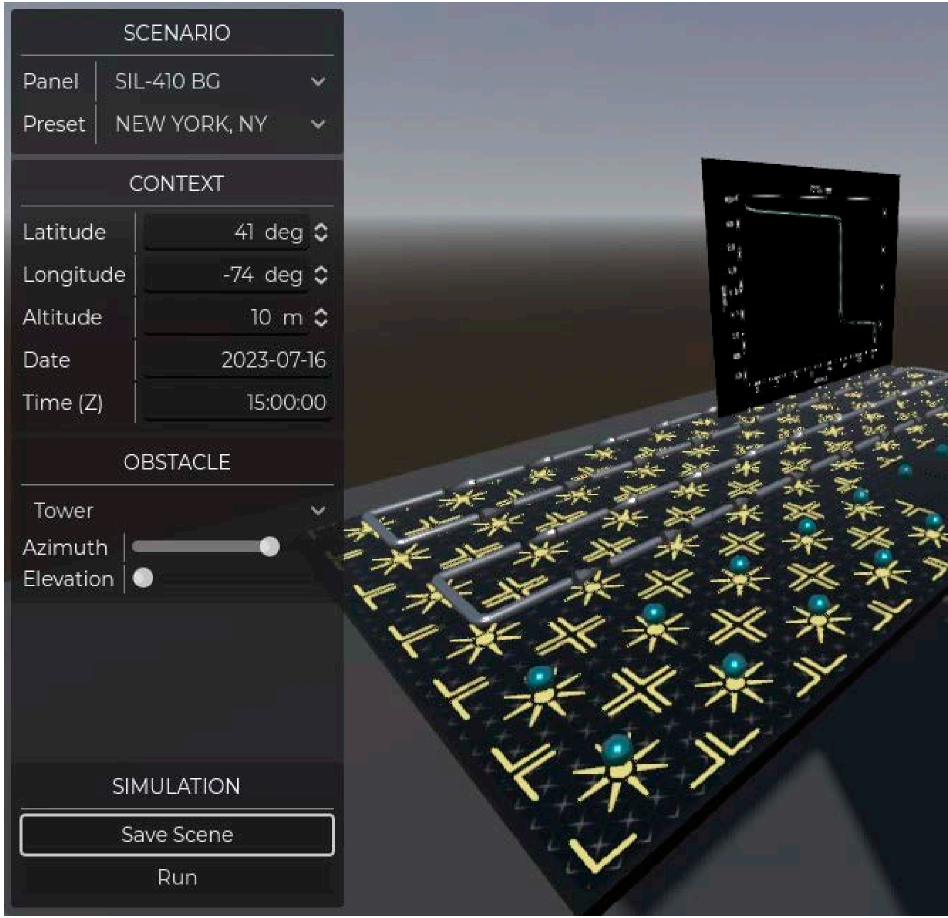
D1

Scene Server

- Enables creation, distribution, and inter-action with 3D scene
- Populated with information from tracked volume and simulation provider
- Written in Godot, with GDScript
- Provides GUI, streams over NOODLES[6,7]

D4

D1-3



Participant Platforms

- Support desktop, mobile, and VR/AR headsets
- Any NOODLES client can participate D3
- Clients are general, not restricted to this project
- Modifications or updates are server-only, not client-side D1
- Greater flexibility and ease of maintenance D4

(Photos by Kenny Gruchalla/ NREL)

Clients Used

Device	Implementation
Magic Leap 2	Unity
Apple Vision Pro	Swift+RealityKit
Browser	Three.js
Immersive	Qt/C++

NREL/PR-2C00-91700

Thanks to:
J.D. Laurence-Chasen
Jordan Washington
Graham Johnson

Acknowledgements

www.nrel.gov

This work was authored in part by the National Renewable Energy Laboratory, managed and operated by Alliance for Sustainable Energy, LLC for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by Colorado Office of Economic Development & International Trade (OEDIT). The research was performed using computational resources sponsored by the Department of Energy's (DOE) Office of Energy Efficiency and Renewable Energy (EERE) located at the National Renewable Energy Laboratory, and used resources at the Energy Systems Integration Facility, which is a DOE EERE User Facility. The views expressed do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Photo from iStock-627281636

Transforming ENERGY

References

1. Solar energy technologies office workforce request for information and convenings (summary), SETO, 7 2021, DOI: 10.2172/1859676

2. J. Garzón, J. Pavón, and S. Baldiris, Systematic review and meta-analysis of augmented reality in educational settings, Virtual Reality, 23(4):447–459, December 2019, DOI: 10.1007/s10055-019-00379-9

3. M. Akçayır, G. Akçayır, H. M. Pektaş, M. A. Ocak,.Augmented reality in science laboratories: The effects of augmented reality on university students' laboratory skills and attitudes toward science laboratories. *Computers in Human Behavior*, 57:334–342, 2016. doi: 10. 1016/j.chb.2015.12.054

4. M. Whitlock, D. A. Szafir, and K. Gruchalla. HydrogenAR: Interactive data-driven presentation of dispenser reliability. In 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 704–712, 2020. doi: 10.1109/ISMAR50242.2020.00101

5. K. Gruchalla, and N. Brunhart-Lupo. The Utility of Virtual Reality for Science and Engineering. In W. R. Sherman, ed., *VR Developer Gems*, chap. 21, pp. 383–402. Taylor Francis, 2019. doi: 10.1201/b21598-21

6. N. Brunhart-Lupo, B. Bush, K. Gruchalla, K. Potter, and S. Smith. Collaborative exploration of scientific datasets using immersive and statistical visualization. *Proceedings of the 2020 Improving Scientific Software Conference (No. NCAR/TN-567 +PROC)*, 2020. doi: 10. 26024/p6mv-en77

7. N. Brunhart-Lupo, NOODLES: cooking up collaborative visualization, 2022, NREL/PR-2C00-83800

8. K. Anderson, C. Hansen, W. Holmgren, A. Jensen, M. Mikofski, and A. Driesse, Pvlib python: 2023 project update, Journal of Open Source Software, 8:5994, 12 2023, DOI: 10.21105/joss.0599

Future Work

Remove QR code?

- Noise in tracking
- Requires constant tweaks

Remove tracker?

- Otherwise requires constant headset observation
- Additional expense

Improve lighting

- Headsets use realistic shading
- Occlude or obscure content of interest

(Photos by Kenny Gruchalla/ NREL)