Alfalfa Virtual Building Service: Software Engineering Best Practices Applied to Runtime Interaction with Building Energy Models

Kyle Benne, Tobias Shapinsky, Anya Petersen
National Renewable Energy Laboratory

NREL
Transforming ENERGY

When buildings consume energy is just as important as how much energy they consume

Demand management can help:

Flatten the duck curve by consuming energy when it's available and reducing the evening ramp

Reduce peaks to avoid expensive capacity upgrades when demand for electricity grows

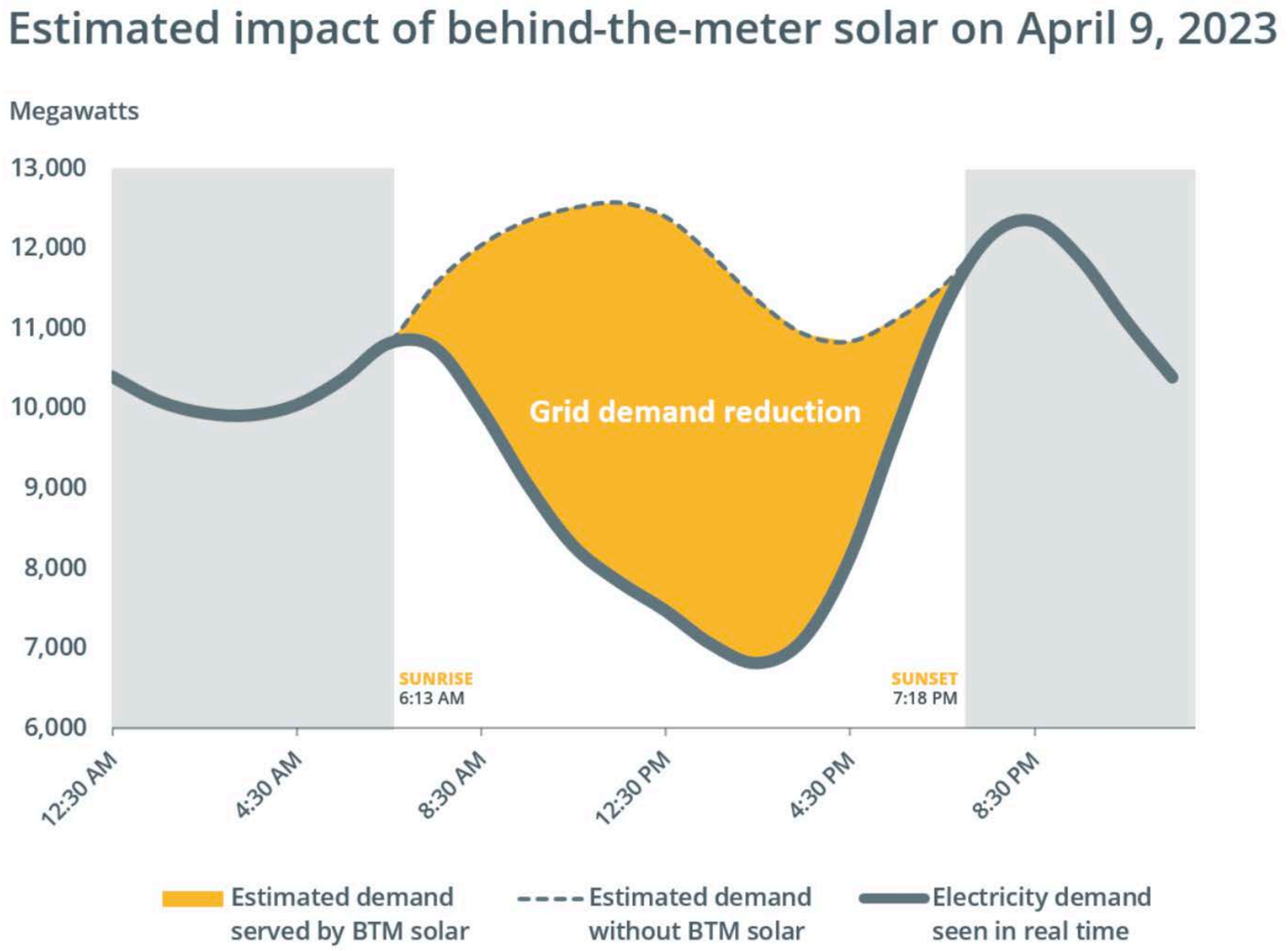


Figure from https://isonewswire.com/2024/02/28/duck-curve-days-becoming-more-frequent-as-solar-power-spreads/

Industry Is Pursuing New Opportunities for Buildings

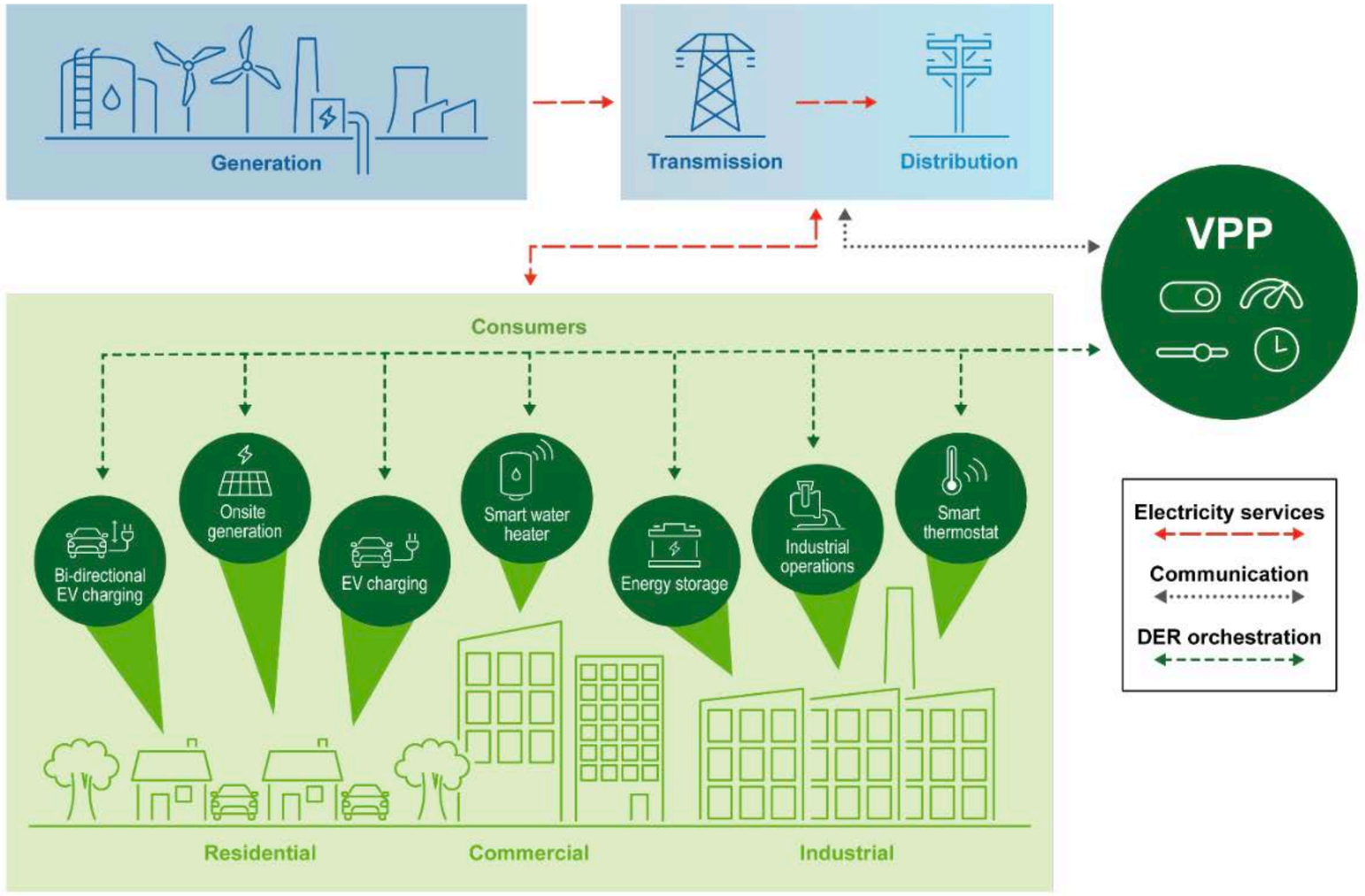


Figure from https://liftoff.energy.gov/wp-content/uploads/2023/09/20230911-Pathways-to-Commercial-Liftoff-Virtual-Power-Plants_update.pdf

For example, virtual power plants (VPPs) coordinate buildings and equipment to provide grid services

New equipment and multiple layers of control are introduced...

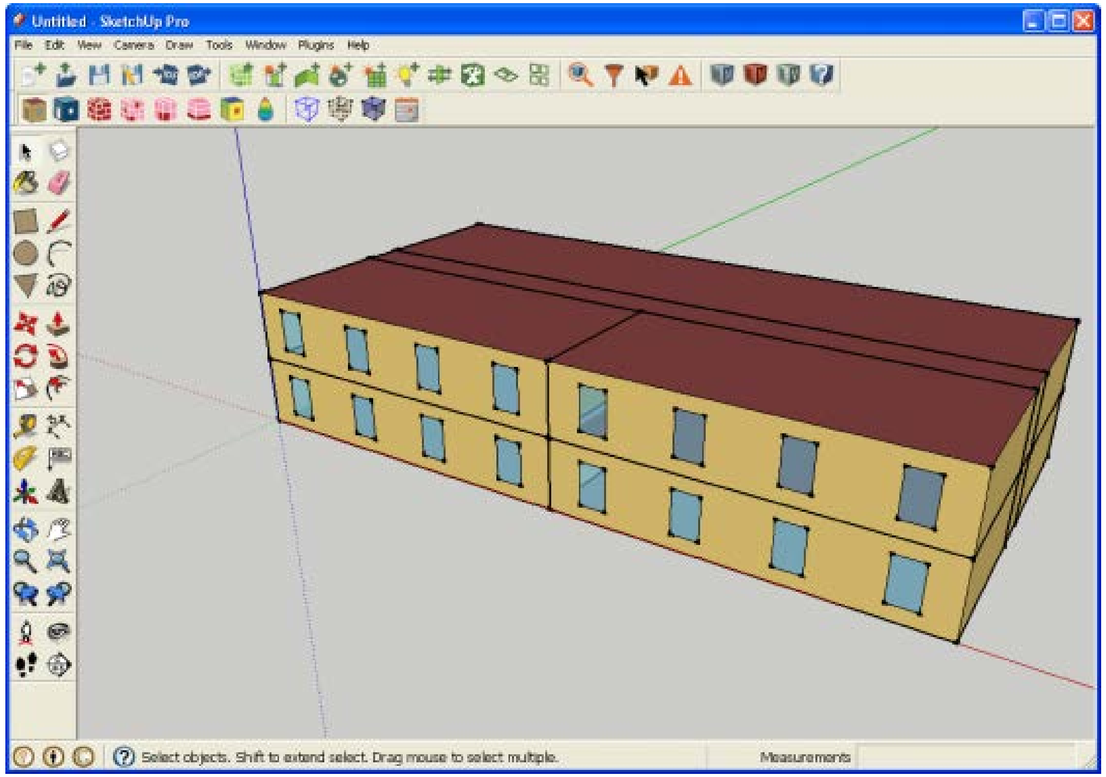
How do we study these interconnected systems?

- Buildings are difficult test subjects
- Field deployments are expensive
- Virtual buildings can represent a range of conditions and system-of-systems interactions for the development and de-risking of hardware and software technologies

Photo from iStock 478821794

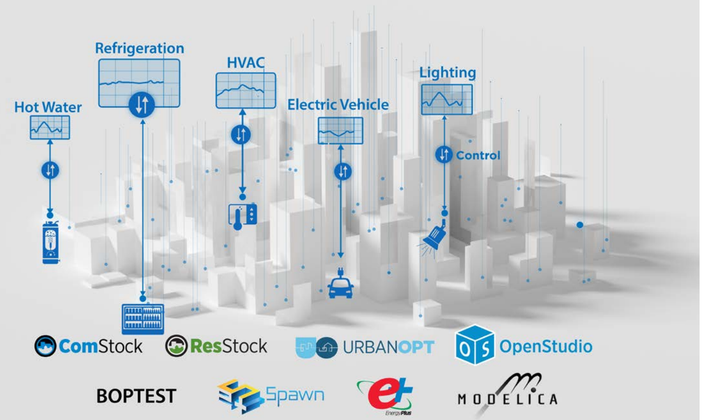
Building energy modeling (BEM) can stand in for real buildings during development and de-risking

The Problem Is Our Analysis Tools Were Born in a Different Era



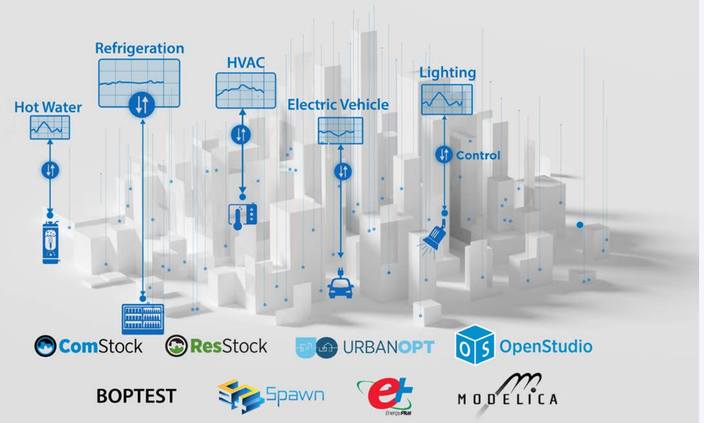
- **Annual end use intensity** used to be the **prime key performance indicator**
- Typically, **one building per simulation**
- **Monolithic executable programs** (static model input file(s) -> **executable program** -> output files -> analysis)
- **Focus on building construction industry** (mechanical engineers, architects, building code officials, and building-focused policy makers)

Alfalfa Virtual Building Service is a web service designed to meet the BEM requirements we face today.



- Run and **interact** with **building energy models** using a simple **web interface**
- **Start** simulation, **stop** simulation, **read** outputs, **write** inputs, **advance** time
- Models run interactively, **in real time or faster**
- One programmer interface for many simulation engines
- Highly **scalable—**one building or **thousands**
- Supports **local** deployments and **“big cloud”**
- Supports a wide range of models. **ComStock**, **ResStock**, and **user supplied models**

Alfalfa is simple and intuitive




```python
client = AlfalfaClient("https://alfalfa.nrel.gov")
```

1 Initialize Alfalfa Client

```python
run_id = client.submit("model_directory")
```

2 Submit model to Alfalfa

```python
client.start(run_id,
             start_datetime=datetime(2020,1,1),
             end_datetime=datetime(2020,1,2),
             external_clock=False,
             timescale=1
             )
```

3 Start run

```python
inputs = client.get_inputs(run_id)
outputs = client.get_outputs(run_id)
```

4 Get control points

```python
client.set_inputs(run_id,
                  {
                    "Heating Setpoint": 18,
                    "Cooling Setpoint": 24
                  })
```

5 Override setpoints

```python
client.stop(run_id)
```

6 Stop run

- Alfalfa and BOPTEST both offer web APIs for live interaction with building simulations.

- The development histories and target use cases are different. **(circa 2017)**

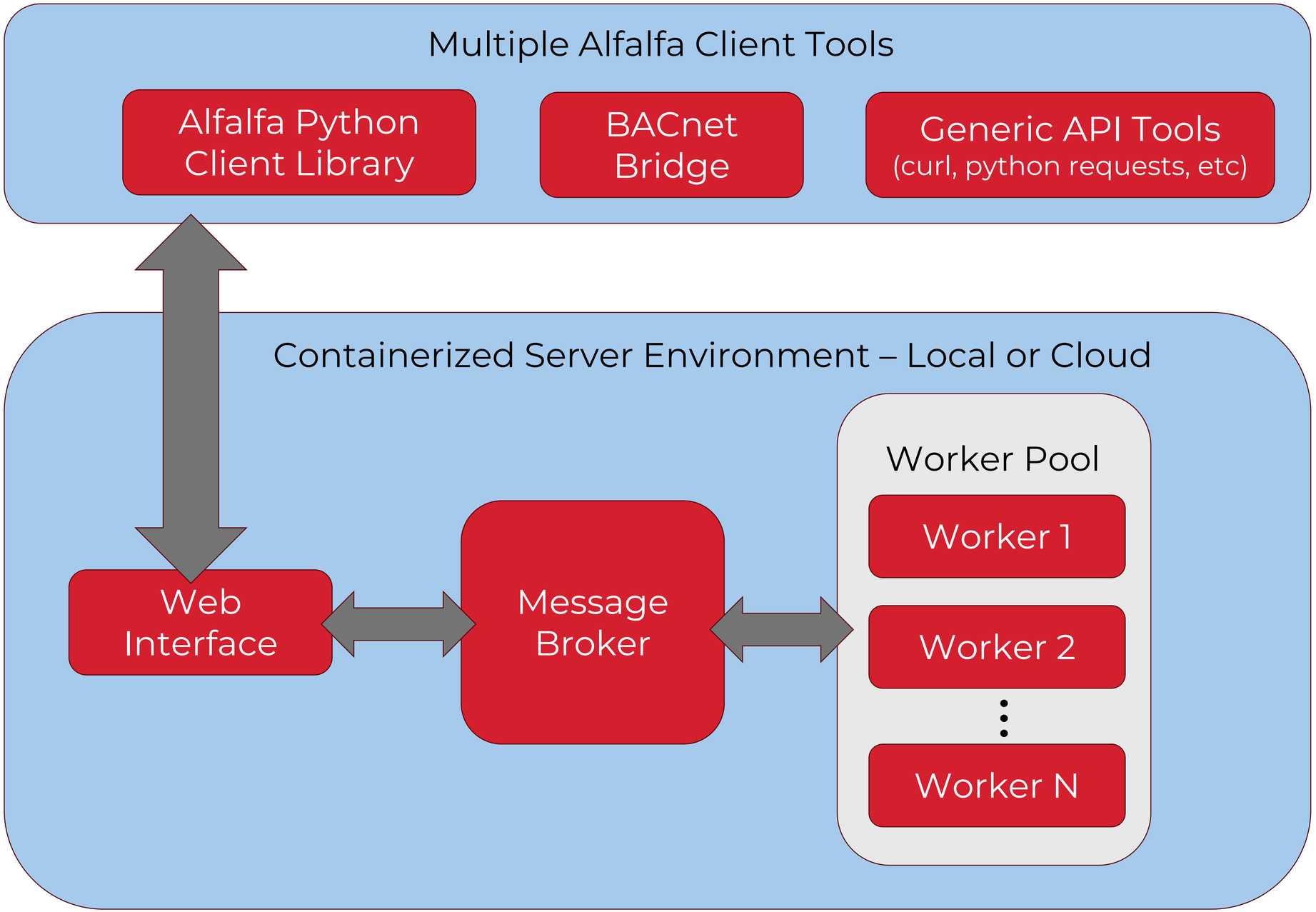
    - **Alfalfa** started as a collaboration with J2 Innovations. **“A general purpose “flight simulator” for J2’s building controls platform.”**

    - **BOPTEST** began as structured test cases for objective controls benchmarking. **”Like the EPA automobile drive course, but for buildings.”**

- **Today….**

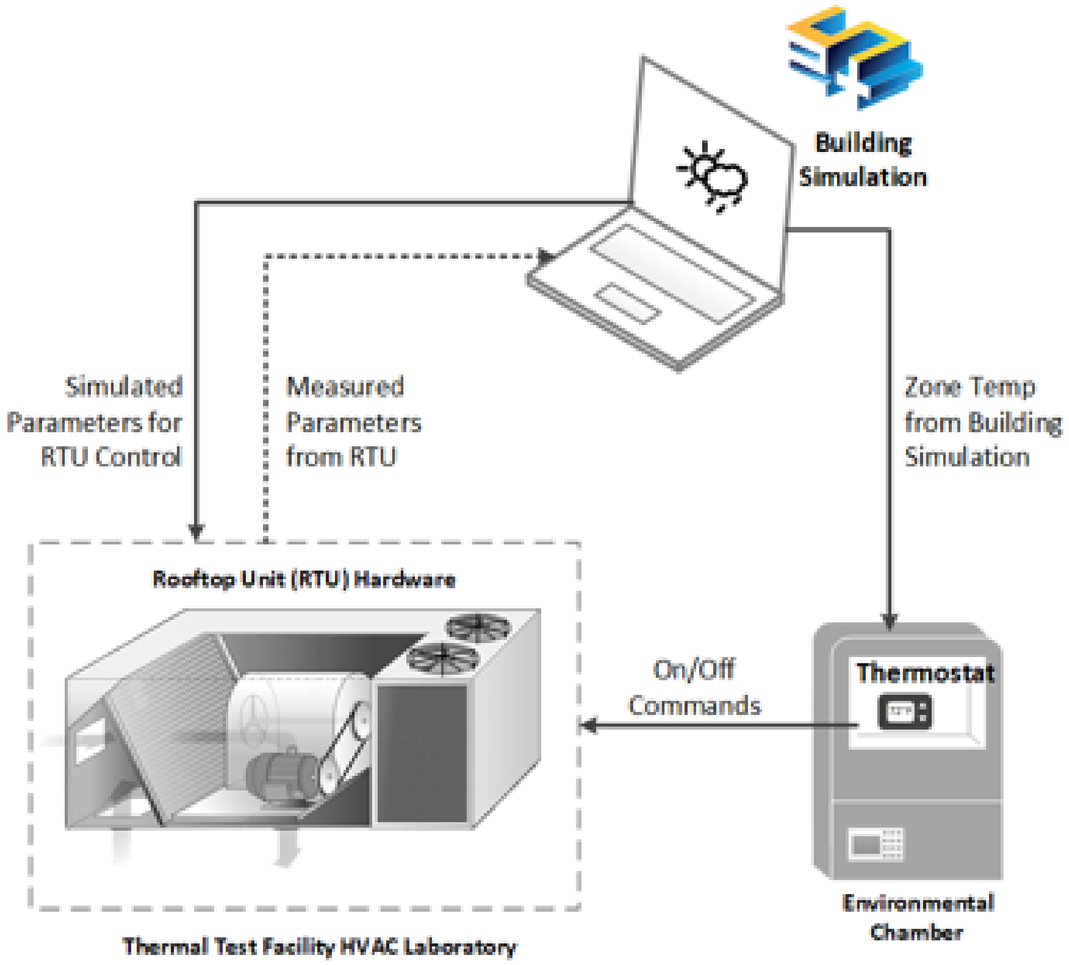
    - Teams are aligning behind a **common API**.
    - **Sharing** the Alfalfa software **architecture/approach**.
    - **Sharing code** wherever possible.

Wait!
Alfalfa sounds a lot like BOPTEST.



Alfalfa is a modern containerized architecture that is flexible and scalable

Example 1: HVAC Hardware-in-the-loop

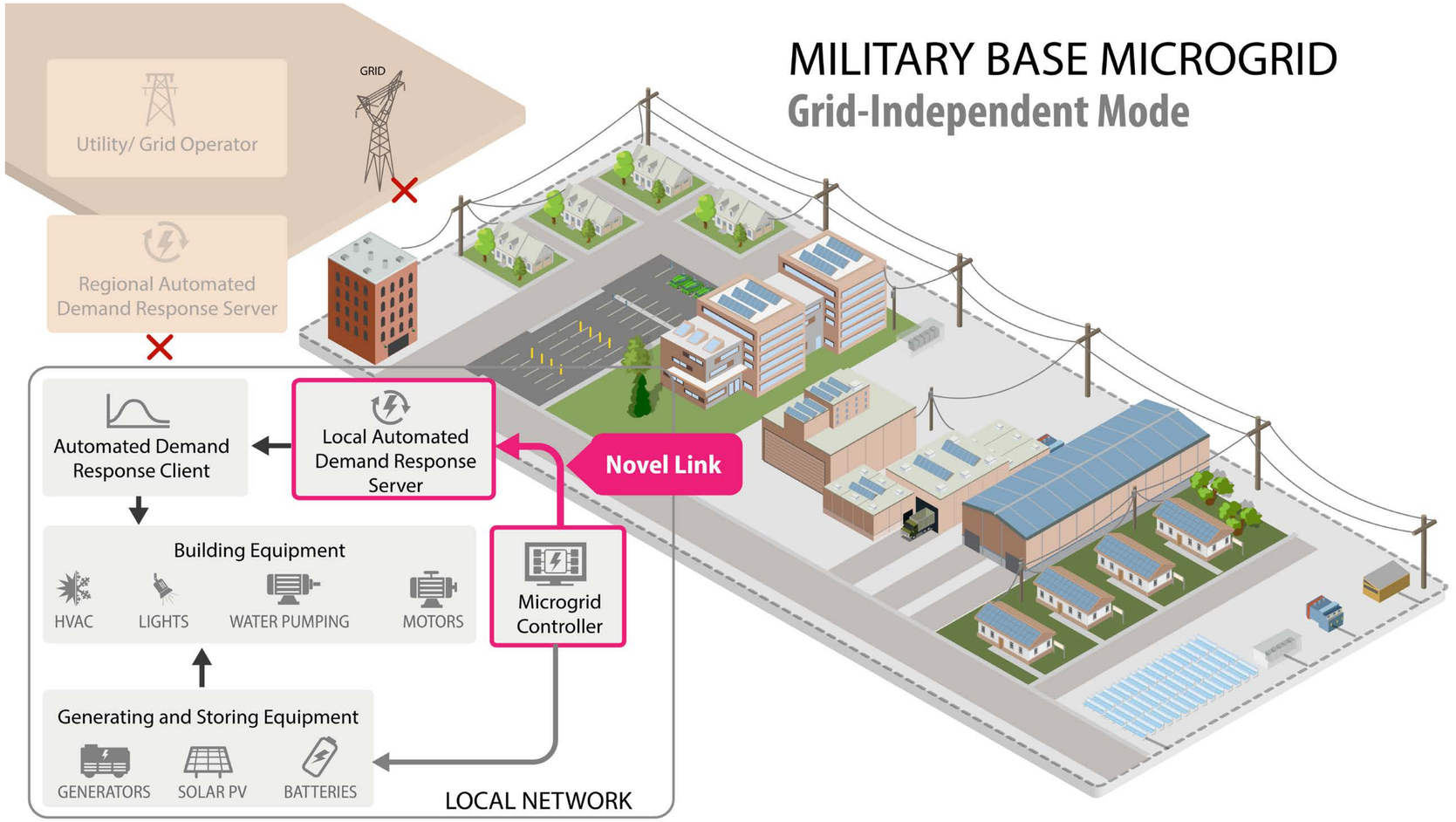


Hardware-in-the-Loop (HIL): A technique that combines hardware and simulation in a continuous feedback loop to explore complex scenarios in a controlled environment

Famous HIL example: flight simulator

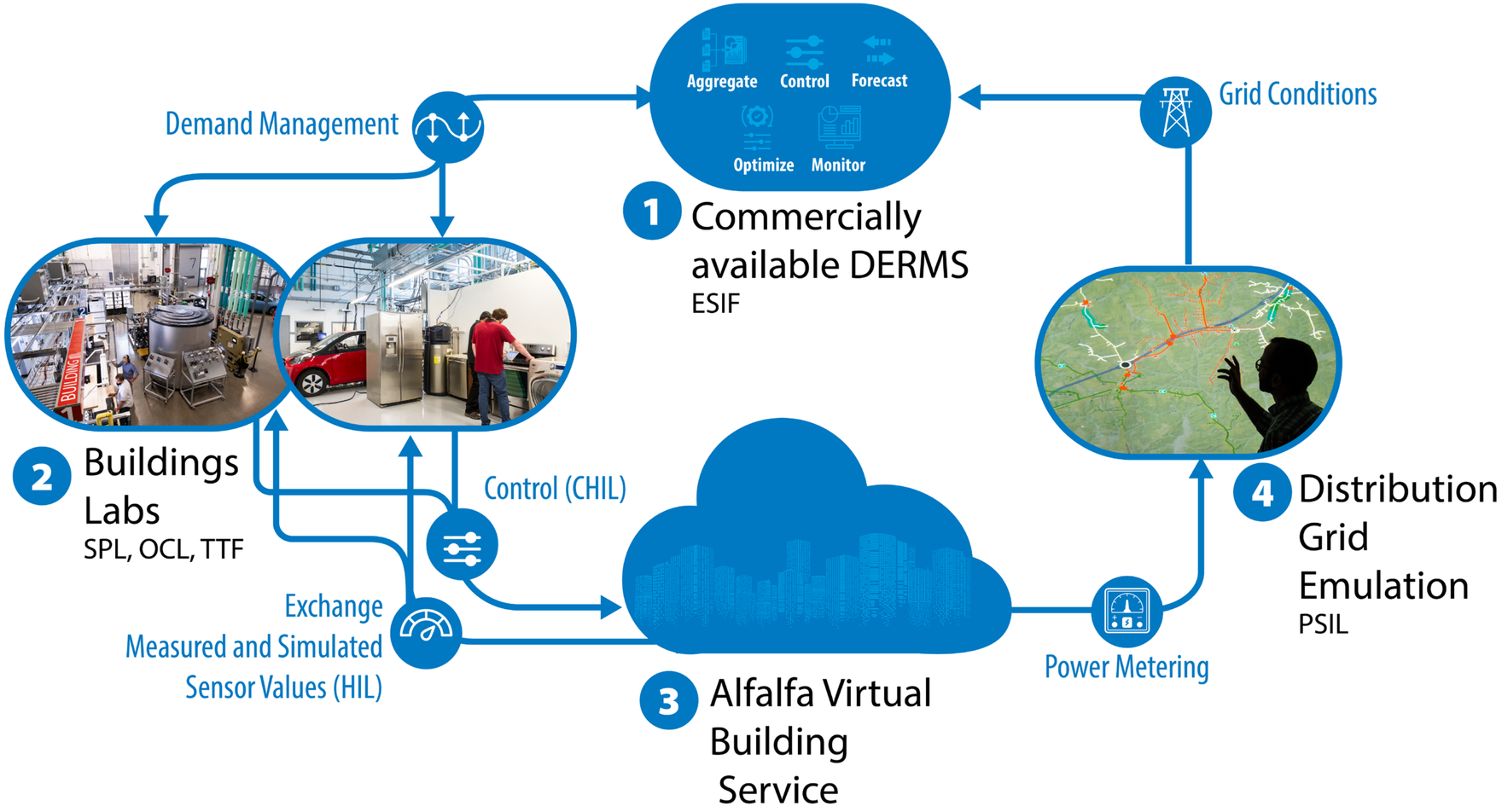
HVAC HIL example: RTU hardware operated in a realistic fashion without the need for physical building and real climate

Example 2: Demand Management for Microgrids



- Microgrid controller balances supply and demand while islanded using OpenADR
- All components are commercially available
- The technology was developed using virtual buildings in a controlled laboratory environment prior to demonstration on base

Example 3: Distributed Energy Resource Management System (DERMS) Commissioning



Note: SPL = Systems Performance Laboratory; OCL = Optimization and Control Laboratory; TTF = Thermal Test Facility; ESIF = Energy Systems Integration Facility; HIL = hardware-in-the-loop; CHIL = controller-hardware-in-the-loop; PSIL = Power Systems Integration Laboratory.

Conclusion

Alfalfa supports a wide range of applications by providing an interactive building simulation interface that is flexible, scalable, and intuitive

Kyle Benne
National Renewable Energy Laboratory
Kyle.Benne@nrel.gov

NREL/PR-5500-89738