Creating Apps and Deployment options

Marcus Åberg 2022-10-19

Modelon

Outline

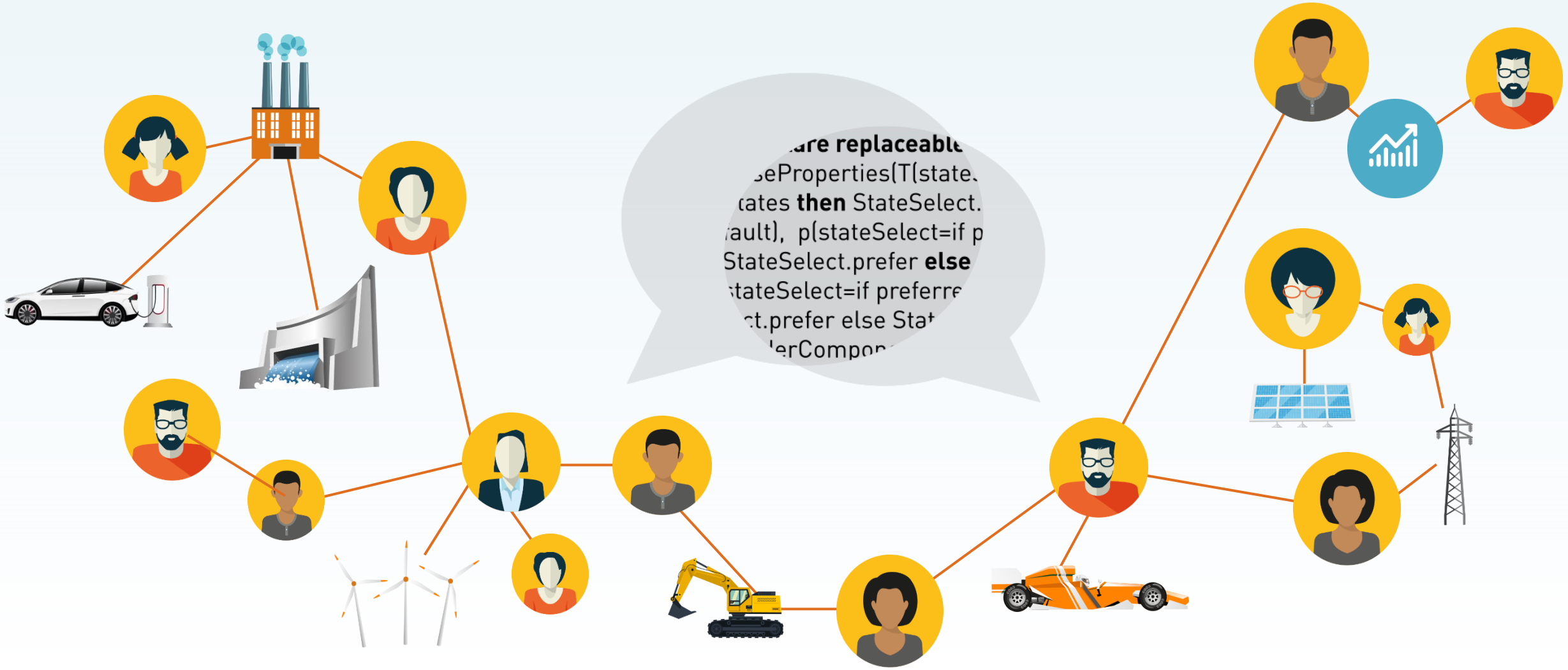
- Introduction
  - Why custom interfaces?
- Introducing - App Mode
  - Create a simplified interface for your model – fast!
- Leverage the rich Modelon Impact API for full freedom
  - Python Dashboarding
  - Traditional Web-Applications

Modelon

Introduction

Why custom interfaces?

Few simulation specialists can make their know-how available…

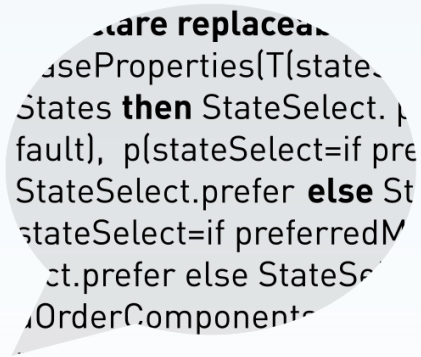


…to enhance the capability of **many** engineers!

LESS $ MORE

MODEL CREATION

= Investment cost

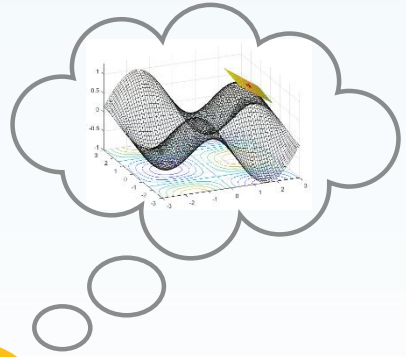


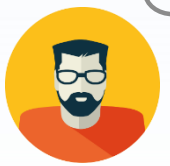
MODEL UTILIZATION

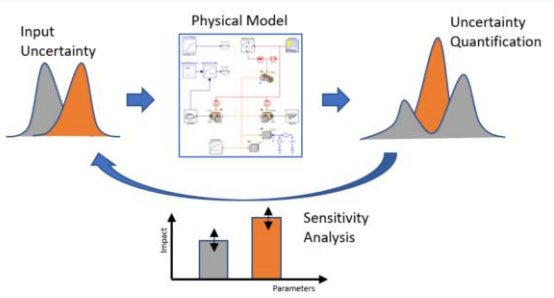
= Engineering Value

What is the optimal size of my component? What is my optimal system configuration?

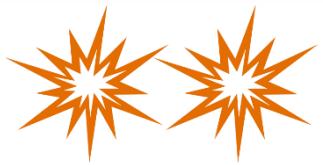
What is the impact of manufacturing variability on system performance?



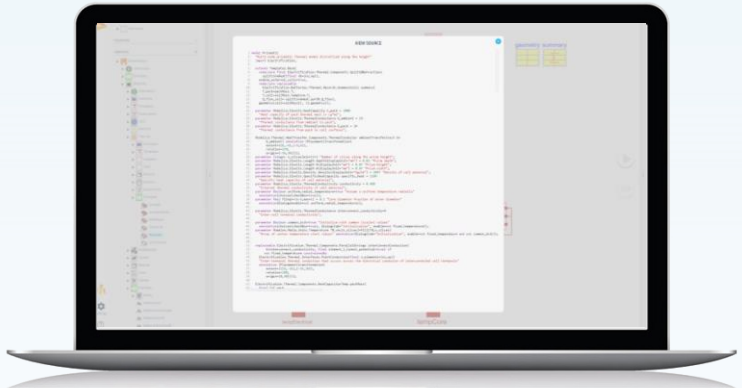


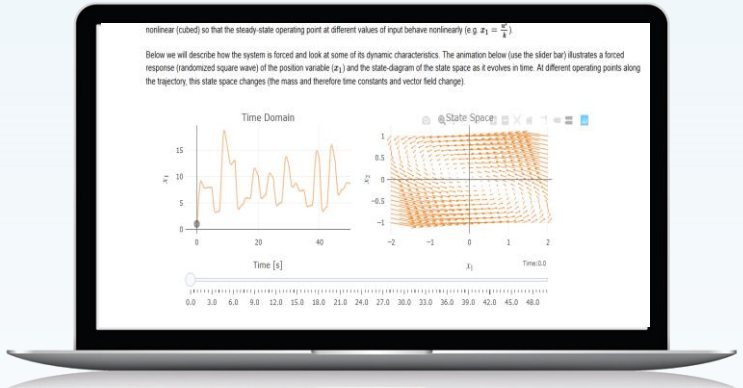


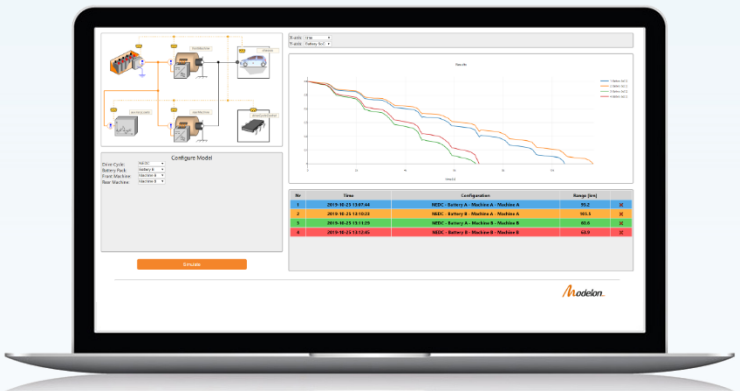
$

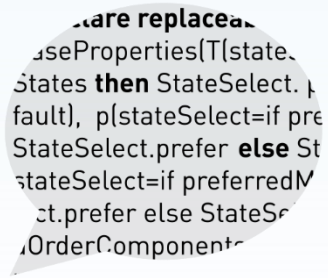


TIME TO GET PRODUCTIVE









years | months | days | minutes | seconds

Many reasons – Same concept

- Power-user enablers
  - Streamline specialized pre-or post processing tasks
    - Reports
    - Model Calibration
    - Model Verification
- Deployment workflows
  - Large audience that could benefit from information provided by model
  - Custom interfaces enables these users to fetch this information directly

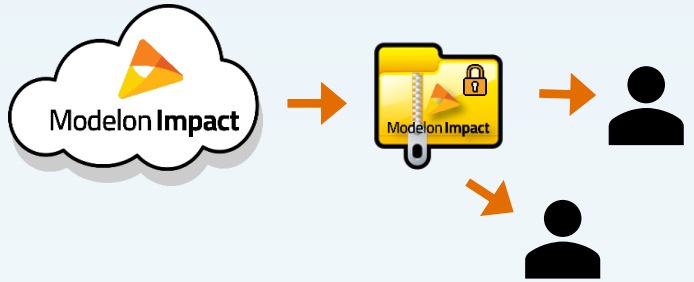
Different users benefit from different interfaces, tailored to their needs!

Modelon

Introducing - App Mode

Create a simplified interface for your model – fast!

Modelon

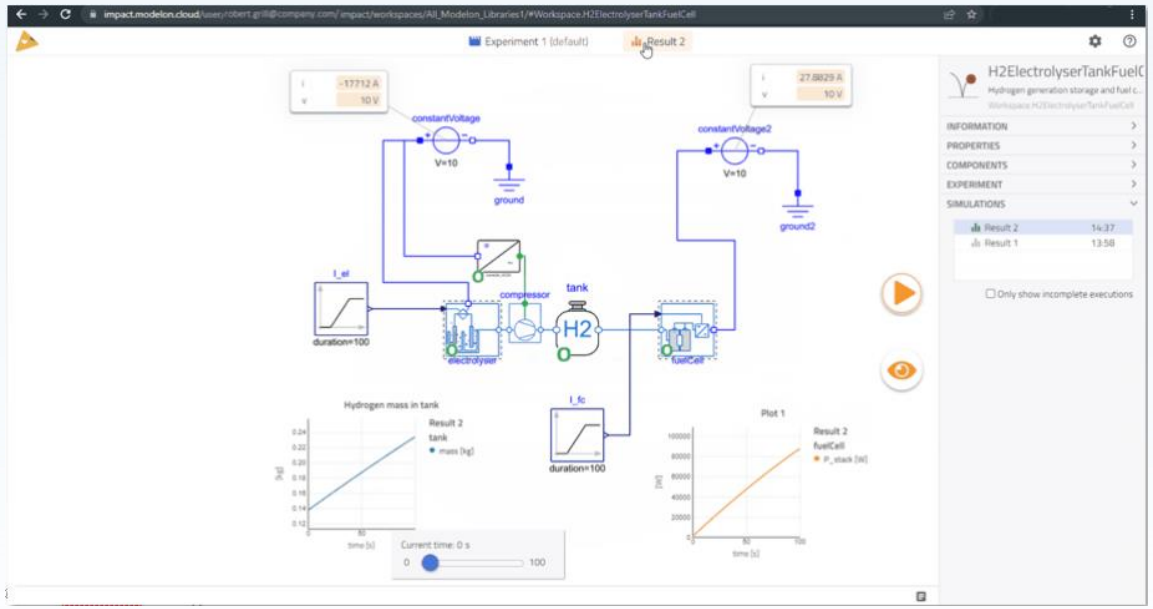


App Mode

Create a simplified interface to a model super fast!

Leveraging existing features in the MI UI:

- User interface prepared by Model author
- “Locked” mode, no editing allowed
- Interaction through stickies and views
- Portable - Share with others



Demo – App Mode

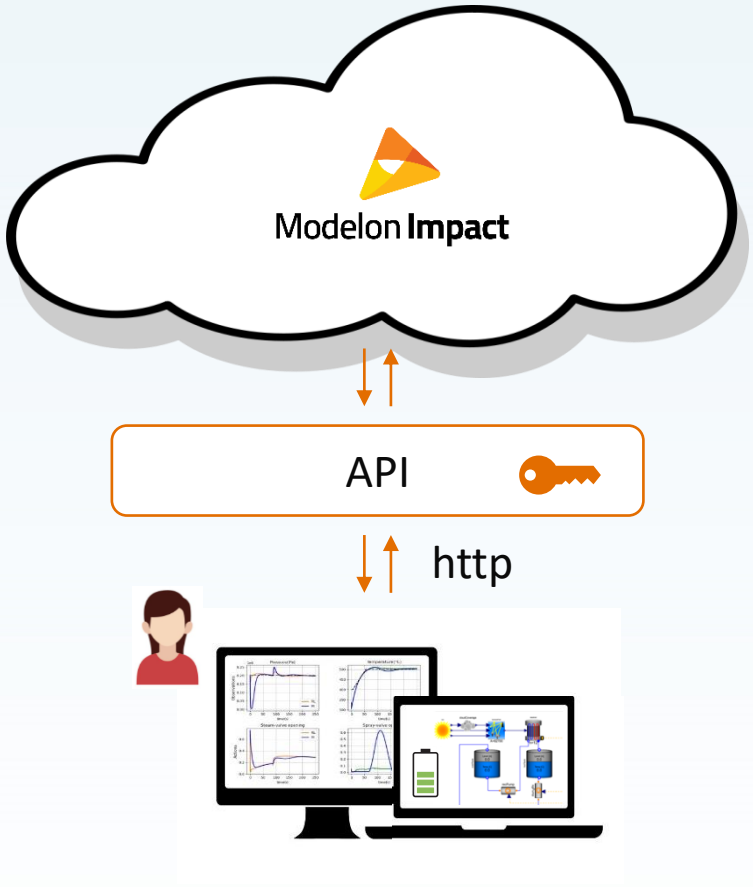
The Modelon Impact API

Leverage Modelon Impact's rich API for full freedom

Modelon

Modelon Impact (REST) API

- Modelon Impact's (REST) API enables communication with the Impact server over the Internet
- Possible to build customized interfaces on top of the API
- Implementation with a wide variety of programming languages and frameworks



https://help.modelon.com/latest/guides/rest_api/

API Overview

The API let's the developer:

- Authenticate against Modelon Impact
- Access resources, such as workspaces, models and results
- Define and execute experiments on the Impact server
- Download and access experiment results
- Upload model libraries

https://help.modelon.com/latest/guides/apidocumentation.html

Workspace Context Based

The concepts used in the API resembles normal user interaction.

Most operations in the API happens in the context of a Workspace:

- Models and dependencies
- Results
- Experiments

Everything is stored in the Workspace – can be investigated from the GUI

Modelon

How to leverage the API

In Theory:

Any environment can connect that can:

- Connect to the Impact Server
- Send receive and interpret the HTTP requests/responses

In Practice:

Specific competence required to interact with the API directly:

- REST-API programming
  - HTTP-requests/response anatomy
- Efficient debugging
- Reading REST-API documentation

Modelon

Introducing: Wrapper Libraries

Solution:

Simplified Interaction through Wrapper libraries:

- Python – general purpose/scripting
- JavaScript – for building web apps

Other benefits:

- Version handling

Modelon

Python Wrapper

- Pre-Installed in Modelon Impact-Cloud

  Open-source, can be installed with pip

- Can be used for:
  - Scripting, automating workflows
  - Creating interactive documentation and reports
  - Leveraging other packages and tools available in Python (for DoE, ML, Optimization etc.)
  - Using dashboard frameworks such as Dash, Voila, Steamlit etc.

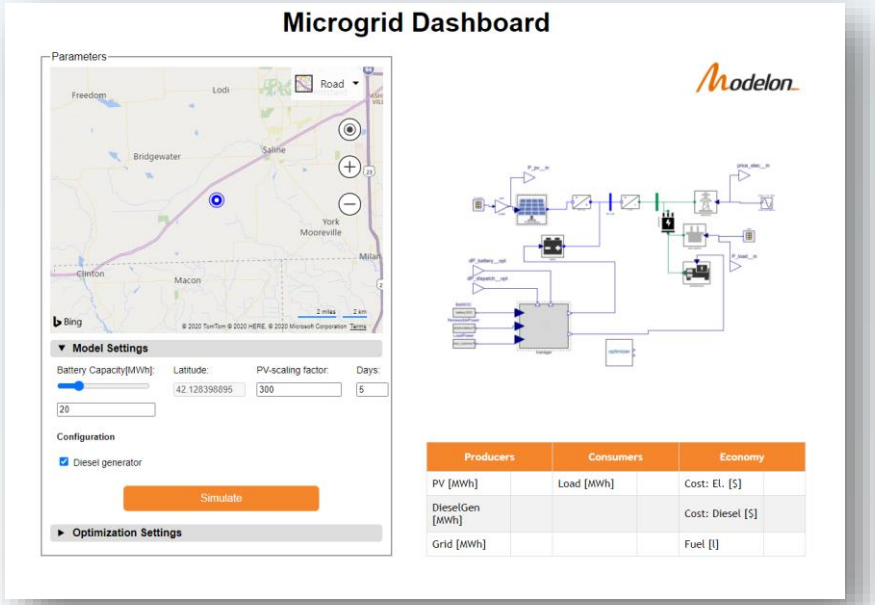
Modelon

JS-wrapper

- Leverage to create web applications

  Open-source, installed with npm

- Web apps allow:
  - Simplified or Specialized interaction with models
  - Custom post-processing (plots, reports etc.)
  - Use of external web resources, APIs or services (maps ,weather etc.)



Python Dashboarding

Streamline your workflows and automate things with Python

Modelon

Python Dashboarding Tools

Basically, what a python dashboarding tool does:

- Type some python code
- Show (interactive) visualization data
- Accessible through a browser

Some Benefits:

- Simplifies communication of results
- End user interacts and get the data they need

Modelon Impact Cloud includes: Voila, Dash

Modelon

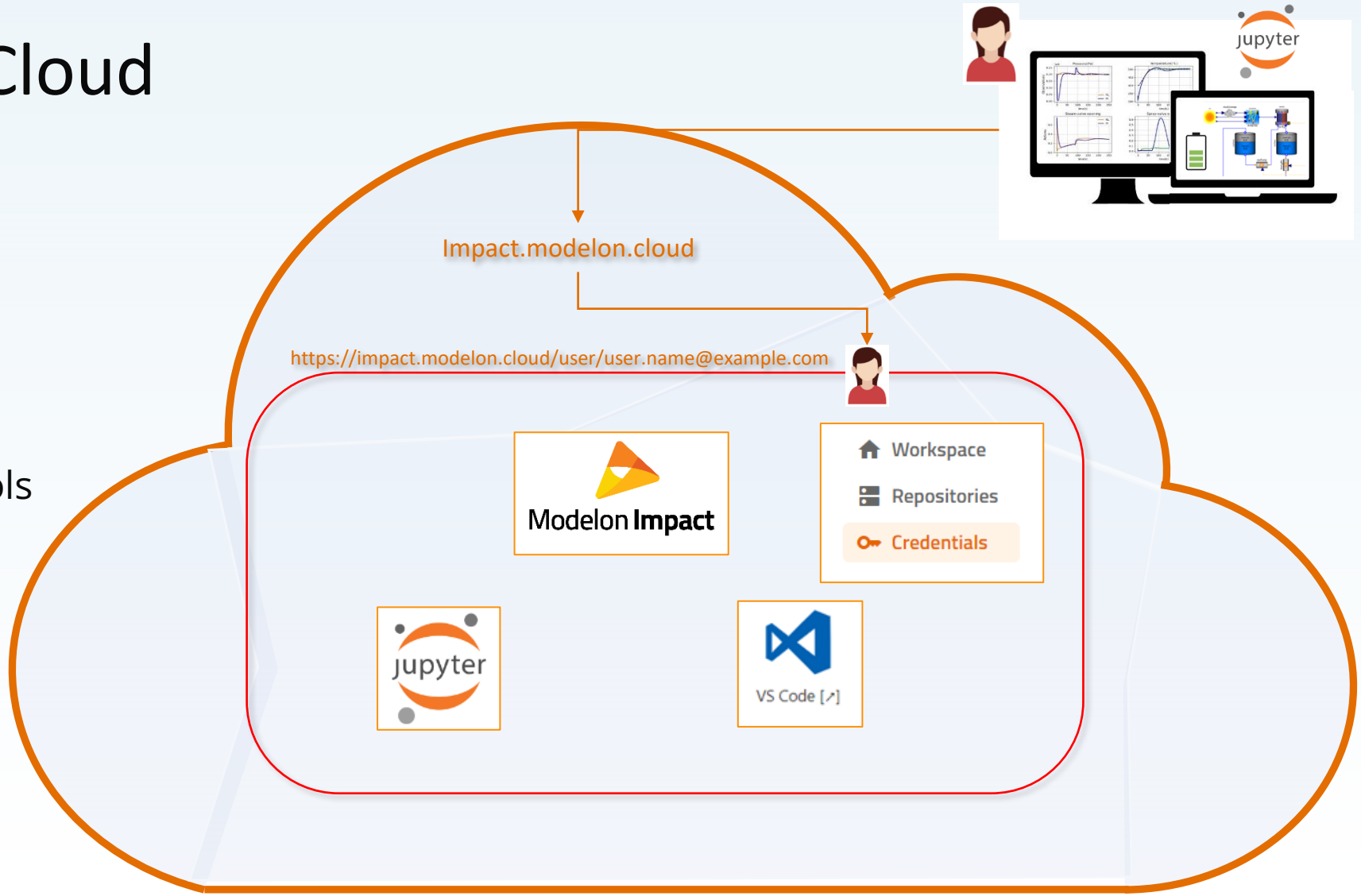
Modelon Impact Cloud

Apps available in the user profile:

- JupyterLab
- VSCode
- Workspace management Tools

Create custom apps using dashboarding tools.

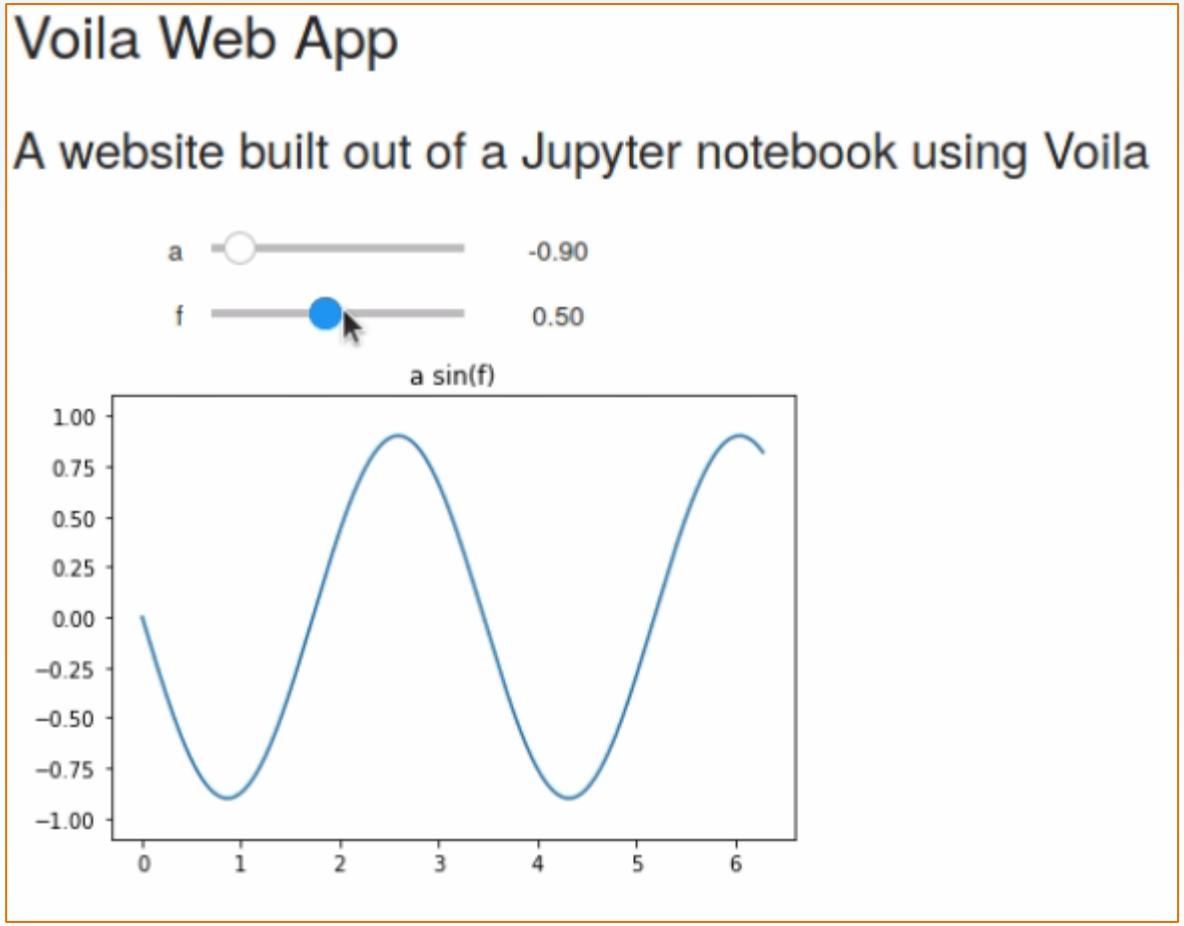
Ideal for creating and automating custom workflows!



Example: Voila app

Voilà Framework:

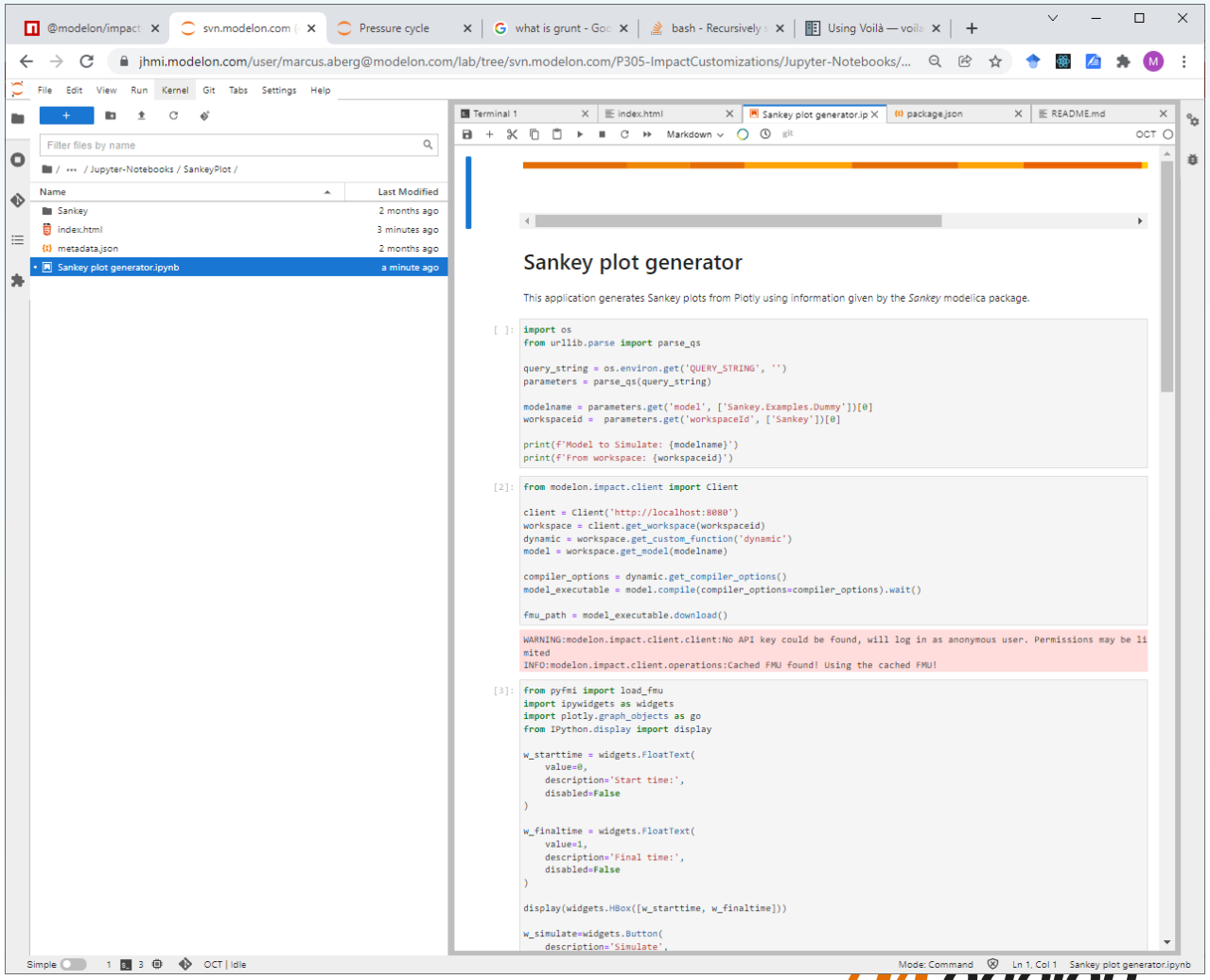
- Converts a Jupyter Notebook to a standalone app
- Removes all code
- Text, Plots and Widgets constitutes UI



Example: Voila app

Develop:

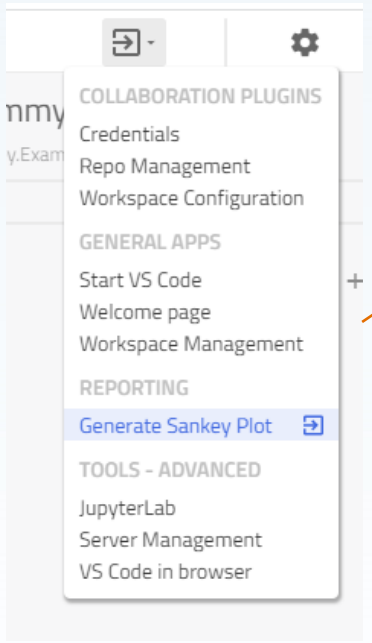
- Create a Jupyter Notebook with your custom scripted workflow
- Add widgets and plots for user-interaction



Example: Voila app

Deploy:

- Launch the app from the apps dropdown menu
- Model and Workspace information is carried over

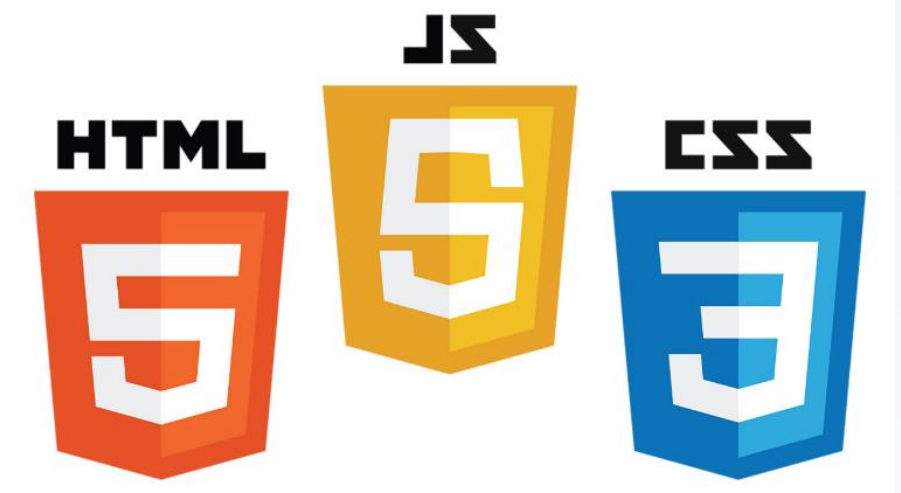


Demo – Python Dashboarding (Voila)

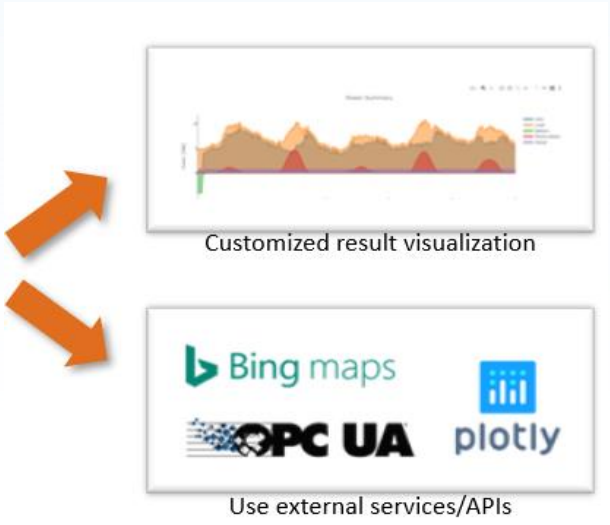
Modelon

Traditional Web Applications
(JS + HTML + CSS)

Full-blown web applications



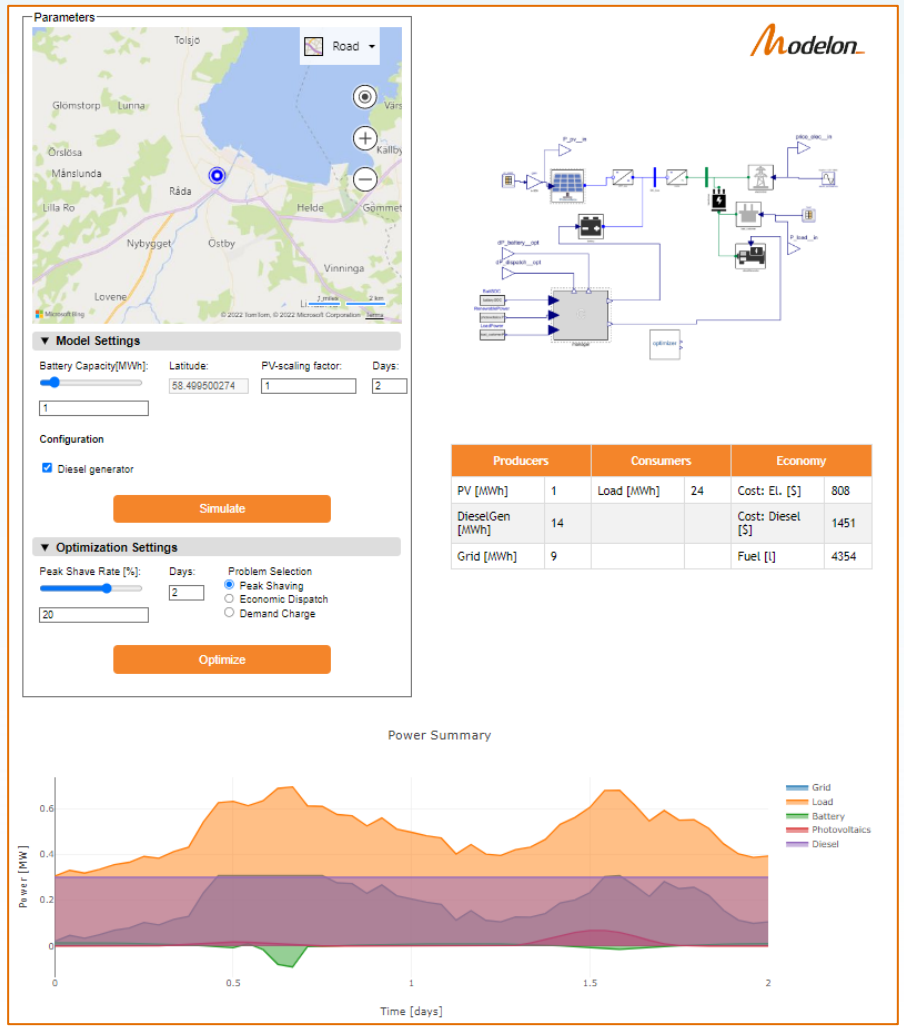
- Technology allows for creation of full-blown web applications
- Provides full flexibility
- Requires more effort and specific competence
- Example use-cases:
  - Customized/Interactive result inspection
  - Integrate with external services APIs



Modelon

Example: Microgrid Dashboard

- Map-Widget to fetch geographical data to the model
- User only presented with a relevant few parameters and settings
  - Problem mode
  - System configuration
  - Sizing information



Conclusion

- Custom interfaces have a wide range of use cases and benefits including:
  - Streamlining common workflows
  - Free up model developer resources
  - Removing the need for users to learn a new simulation tool
- Utilizing the Modelon Impact API + wrapper libraries we can create such interfaces
- Make it even simpler by utilizing App-Mode!

Modelon

Accurate Simulations. Better Decisions.