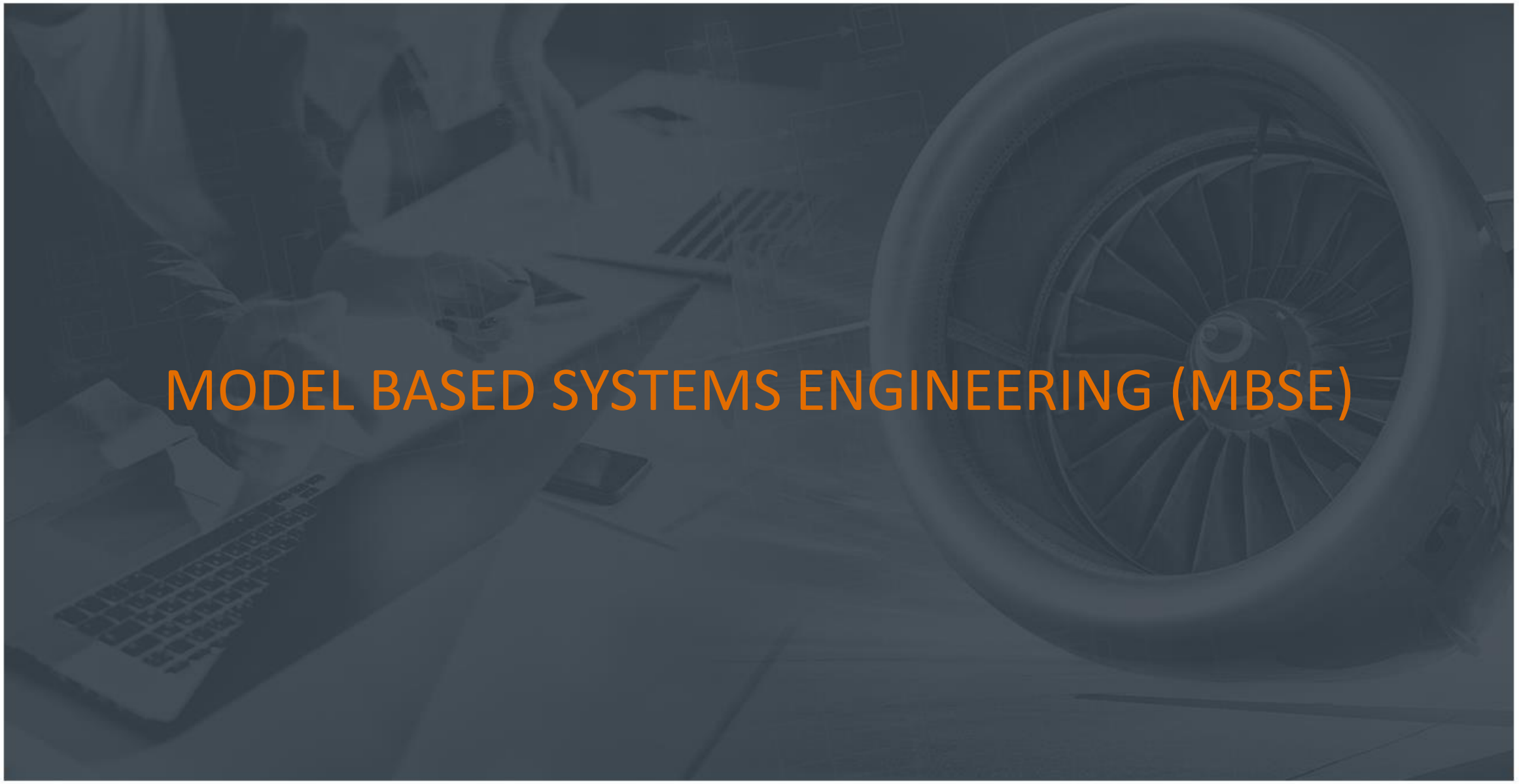
# INTRODUCTION TO MODELICA & MODELON IMPACT

Lecture 1.0

Modelon

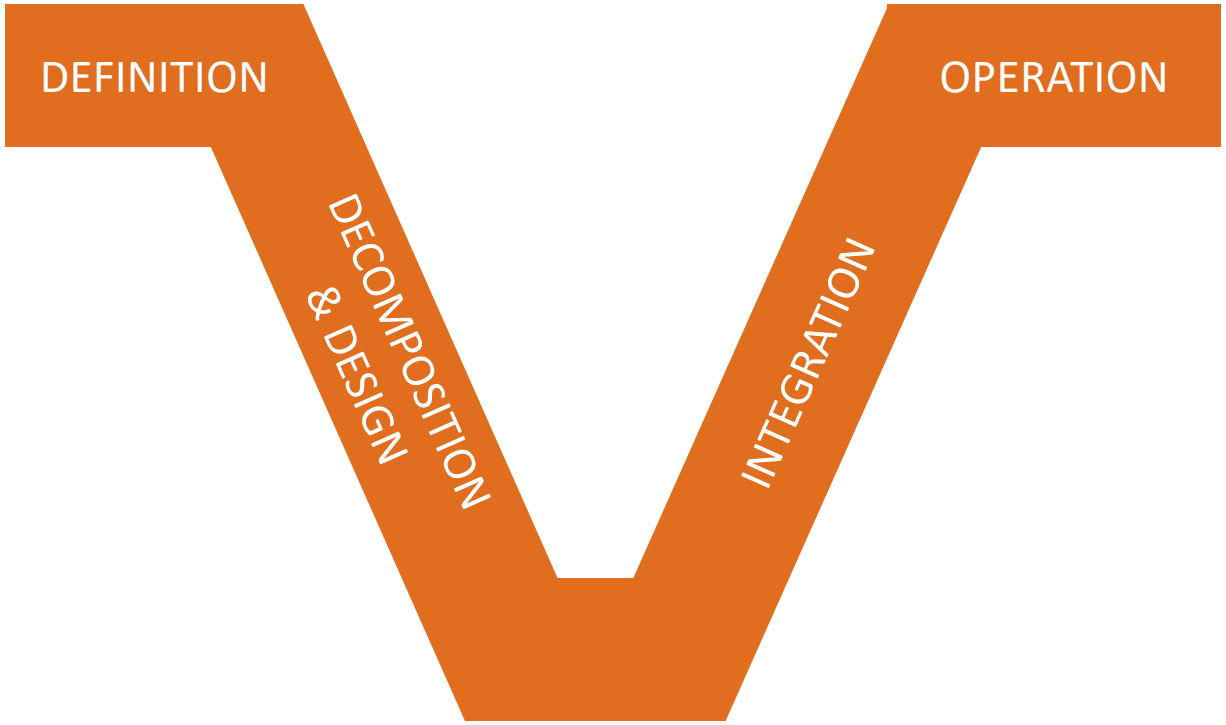
# OVERVIEW

- Model Based Systems Engineering (MBSE)
- Key features of Modelica
- Modelon Impact
- Model Libraries
- FMI technology

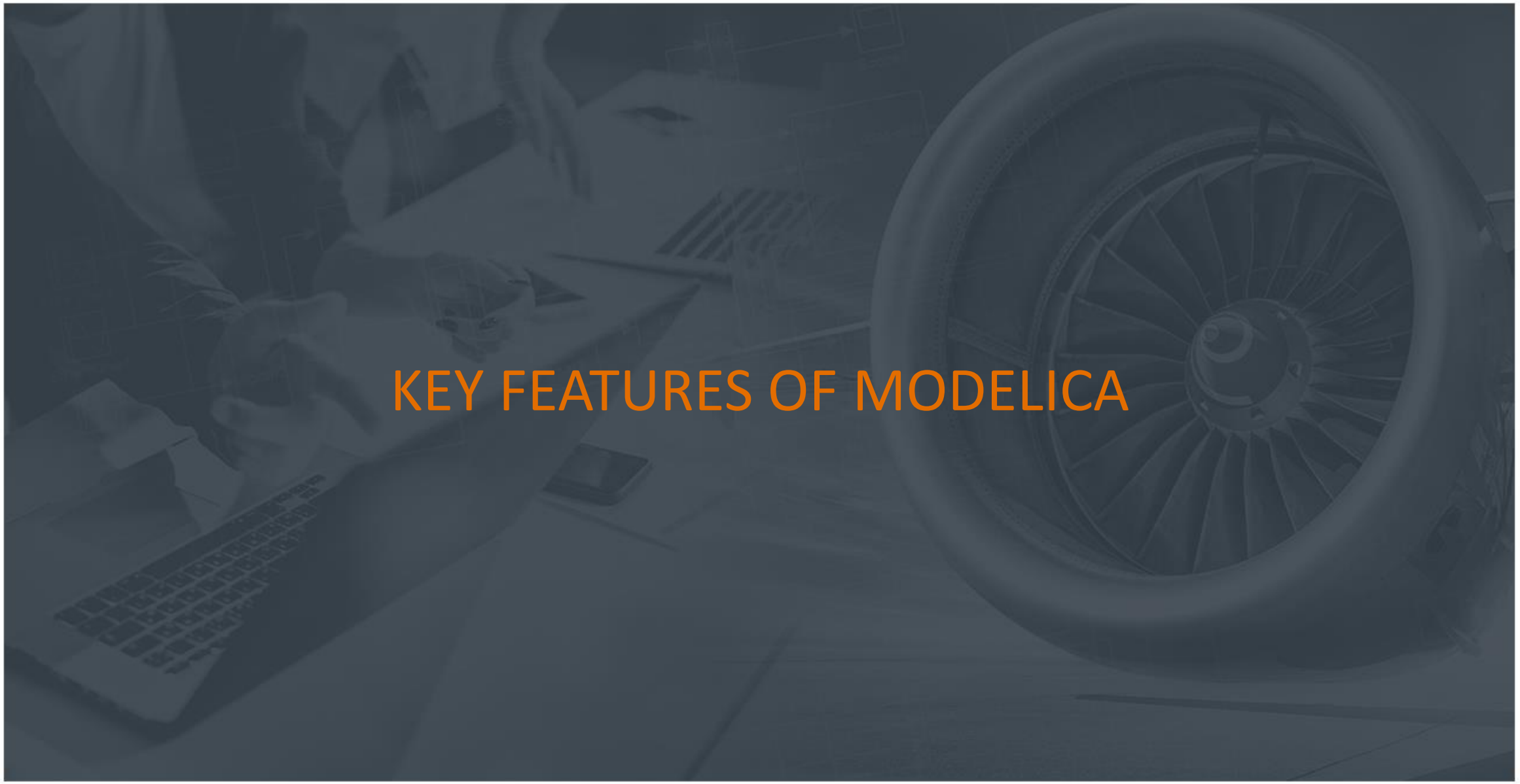


# MODEL BASED SYSTEMS ENGINEERING (MBSE)

# EFFICIENT MBSE WORKFLOW – REQUIREMENTS




- Reuse models during all stages
- Increase collaboration
- Physics based, from 1st principles
- One source of truth
- Efficient handling of design variants
- Efficient handling of multi-fidelity
- Possibility for deployment
- Running models in realtime



# KEY FEATURES OF MODELICA

# WHAT IS MODELICA?

https://modelica.org/

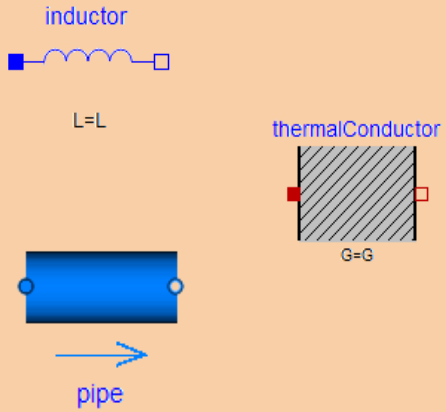
- Modelica is a non-proprietary, object-oriented, multi-domain modeling language for component-oriented modeling of complex systems.
- Developed and owned by The Modelica Association
  - Non-profit organization
  - Develops the largest, free library for multi-domain models, the Modelica Standard Library
- The Modelica language is:
  - Object-oriented → Uses classes, Inheritance
  - Acausal and equation based
  - Supports multi-domain modeling
  - Detailed specs available here

# KEY MODELICA CONCEPTS

https://modelica.org/

## Physical modeling language

- **Physics-based** equation modeling
- **Connectors** pass physical information
- Coverage of all **physical domains**
- Accurate bookkeeping of variable types and units

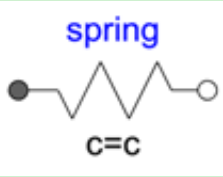



## An equivalent graphical and code interface

- Build systems rapidly from graphical components (through drag & drop or templates)
- Look at the code and change to your needs
- Documentation embedded in models

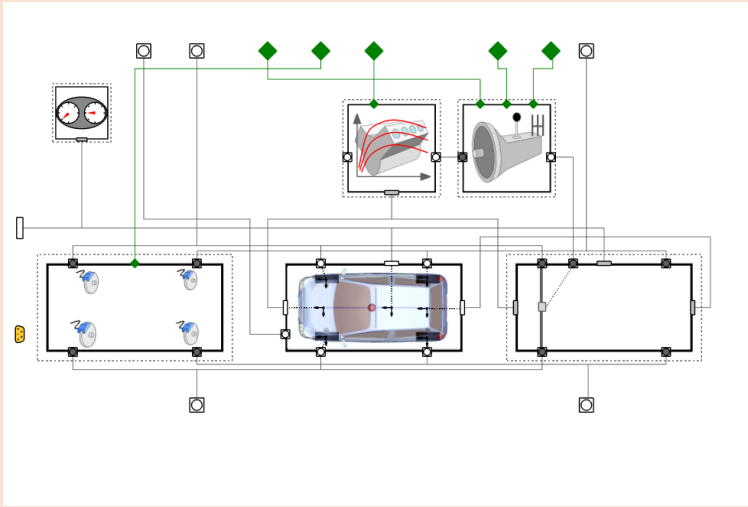
```
model Spring "Linear 1D rotational spring"
  extends Modelica.Mechanics.Rotational.Interfaces.PartialCompliant;
  parameter SI.RotationalSpringConstant c(final min=0, start=1.0e5)
    "Spring constant";
  parameter SI.Angle phi_rel0=0 "Unstretched spring angle";

equation
  tau = c*(phi_rel - phi_rel0);
  annotation (
    Documentation(info="<html>
```




## Efficient modeling

- Separation of models and algorithms
- **A-causality**
- **Architecture based modelling** for handing different levels of fidelity and variants.



# WHAT SETS MODELICA APART

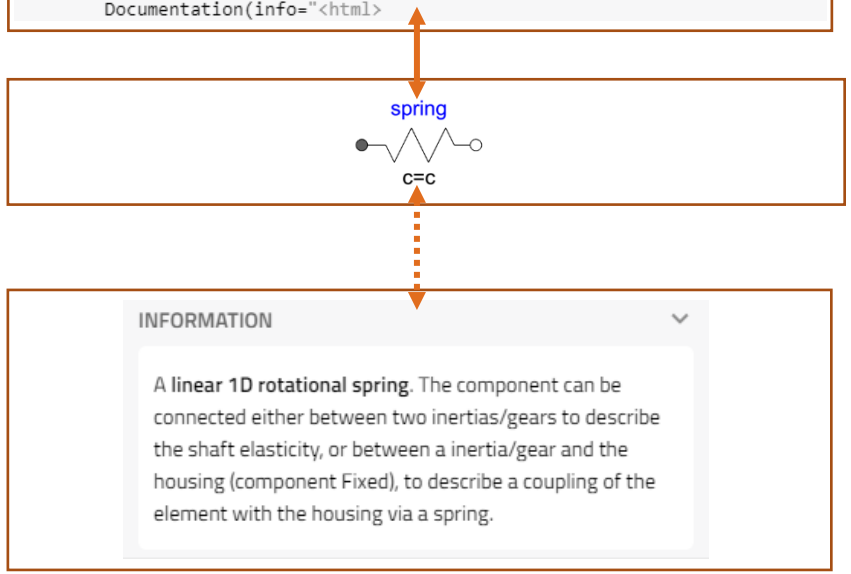
https://modelica.org/

1. Separation of models and algorithms

## Model

```
model Spring "Linear 1D rotational spring"
  extends Modelica.Mechanics.Rotational.Interfaces.PartialCompliant;
  parameter SI.RotationalSpringConstant c(final min=0, start=1.0e5)
    "Spring constant";
  parameter SI.Angle phi_rel0=0 "Unstretched spring angle";

equation
  tau = c*(phi_rel - phi_rel0);
  annotation (
    Documentation(info="<html>
```




## Algorithm

**Symbolic manipulation kernel**

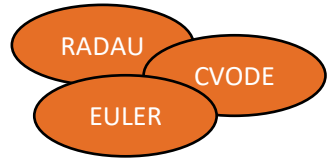
- The tool optimizes the mathematical formulation
- No risk for algebraic loops

Example: Coupling multiple inertia's does not cause any problems as the tool performs index reduction.

## Solver

**DAE solvers appropriate for multi-physics and stiff systems**

- Variable step and fixed step

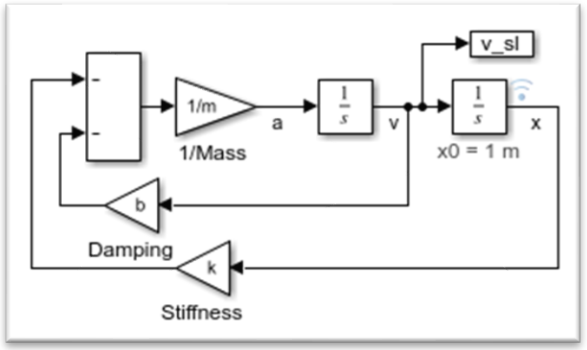



Modelon

# WHAT SETS MODELICA APART

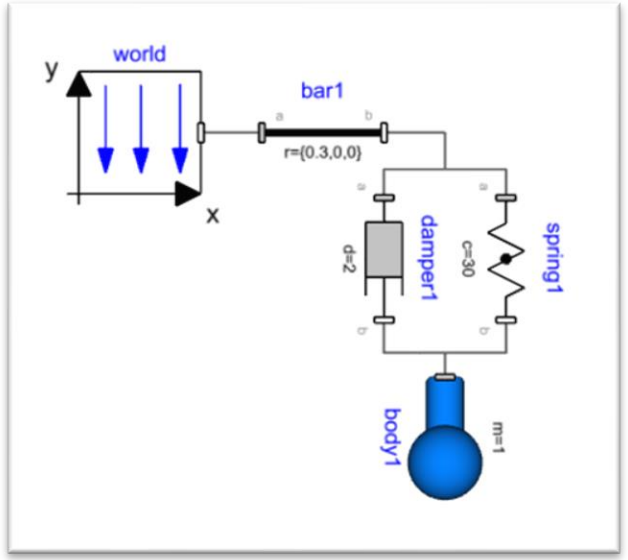
https://modelica.org/

## 2. Acausality




Block diagram

Component diagram




Example 1- Robotic arm: You can reuse the same model for calculating

- The trajectory of the robotic arm from the power supply to the actuators
- The power required to the actuators from the trajectory of the robotic arm.

Example 2 - Planetary gears: simplicity of description

```
equation
  (1 + ratio)*carrier.phi = sun.phi + ratio*ring.phi;
  ring.tau = ratio*sun.tau;
  carrier.tau = -(1 + ratio)*sun.tau;
```

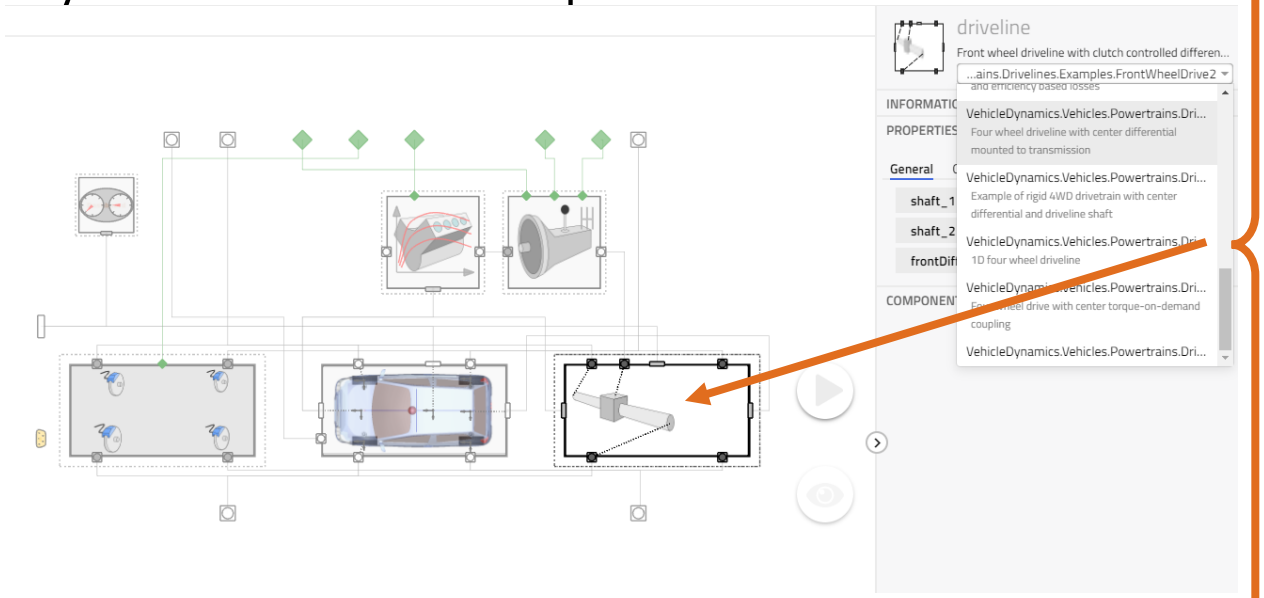
Modelon

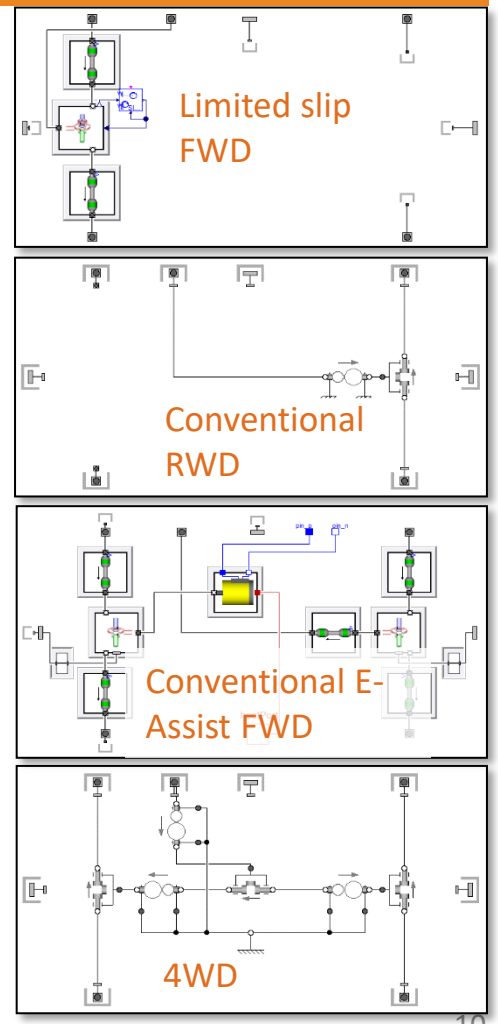
# WHAT SETS MODELICA APART

https://modelica.org/

3. Handling of variants and fidelity levels

- Plug-and-play configuration
- Context-sensitive, dynamic list of available variants
- No re-wiring, select fidelity as needed for each experiment:
  - Back of envelope
  - Detailed verification

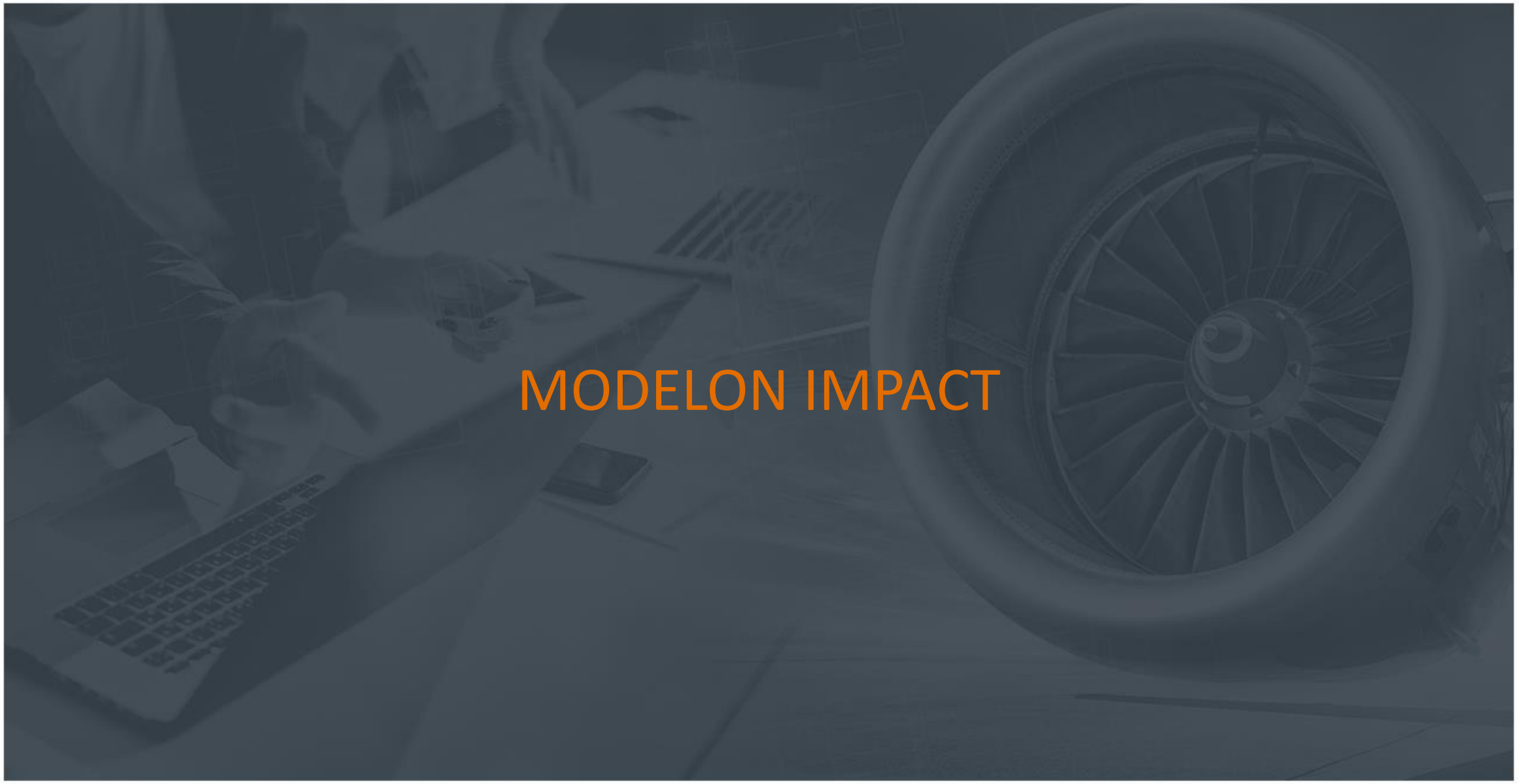


# TO SUMMARIZE..

- Model
  - Physics-based Modelica source code
- Apply boundary conditions
  - Model use under different causality conditions to solve different unknowns
- Apply analysis
  - Dynamic
  - Steady state
  - Dynamic Optimization
  - Calibration

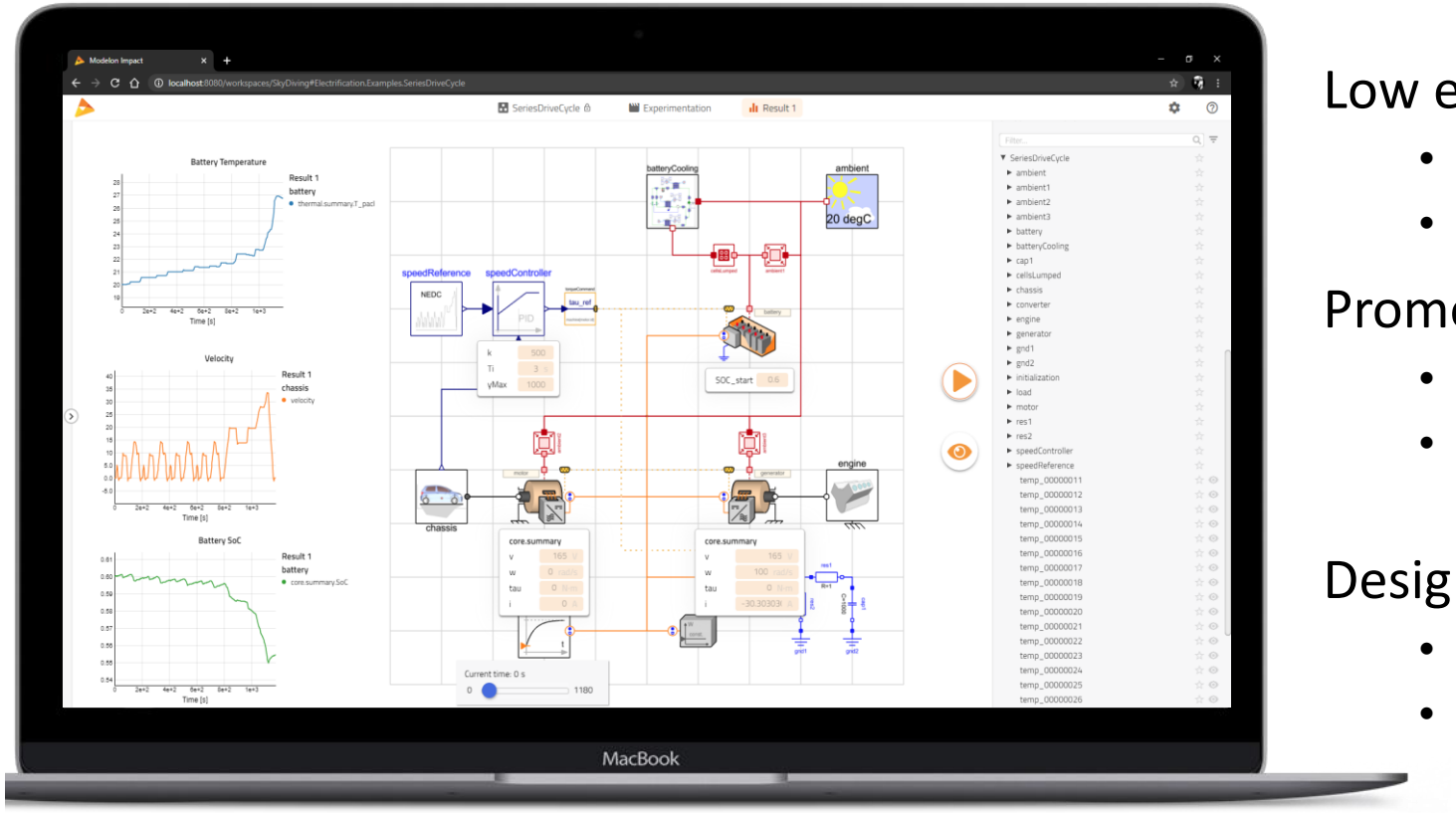
All from one code source and one tool!



# MODELON IMPACT

# MODELON IMPACT

Open Web & Modelica-based Simulation Environment




## Low effort to enter

- Easy to deploy and maintain
- Easy to access & intuitive to use

## Promoting productivity at every level

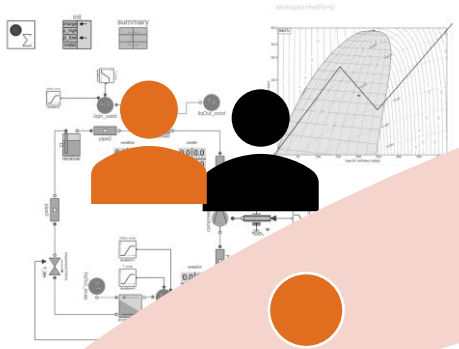
- Collaborative and workflow oriented
- Facilitates model understanding and results interpretation

## Designed to grow with you

- Native support for customization
- ... and large-scale deployment

# DESIGN PRINCIPLE

Model Democratization

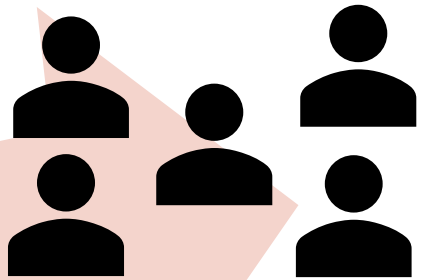


## Component creation

- A high-level, physical, a-causal language
- In-house tool capabilities
- Company specific IP

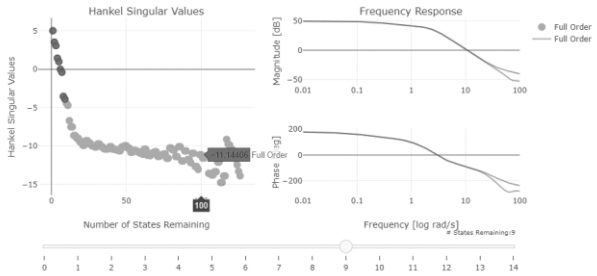
## System composition

- Drag & drop creation
- One architecture through product design
- Variants and different fidelity levels



## Deployment & value generation

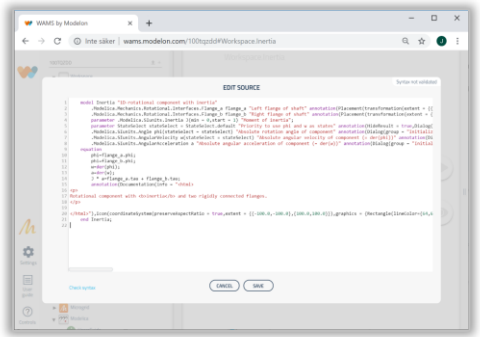
- Tool agnostic model encapsulation (FMI)
- Model Based Design workflows
- Shared engineering workspaces
- Web applications



# FOR ALL TYPES OF USERS

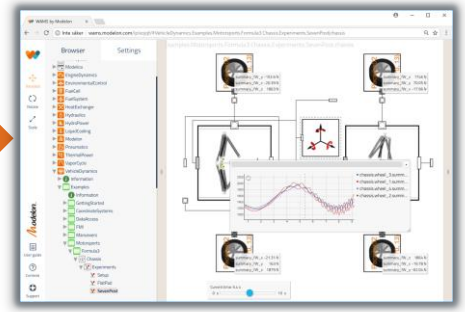
## Few experts serving several users

Component authoring in ***Modelica***

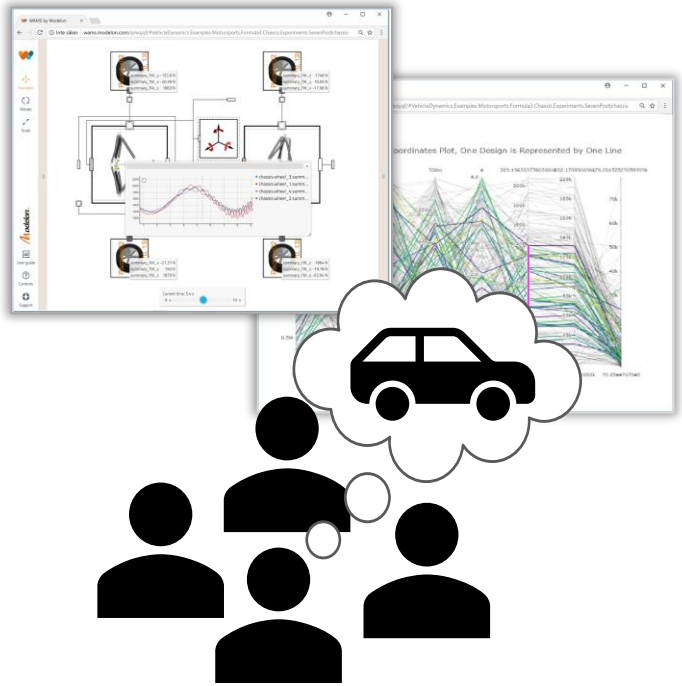


```
equation
  phi = flange_a.phi;
  phi = flange_b.phi;
  w = der(phi);
  a = der(w);
```

System composition using ***Modelica*** libraries



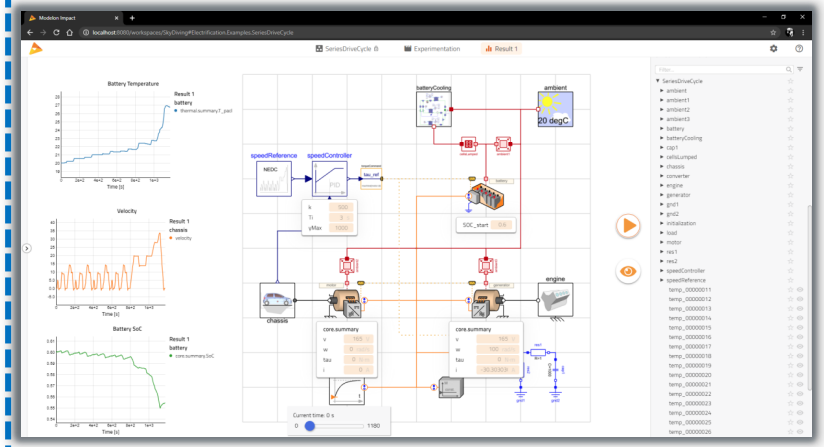
System configuration and analysis (***dynamic, steady-state, custom, optimization***)



# FOR ALL TYPES OF USERS

## Enterprise-friendly model deployment

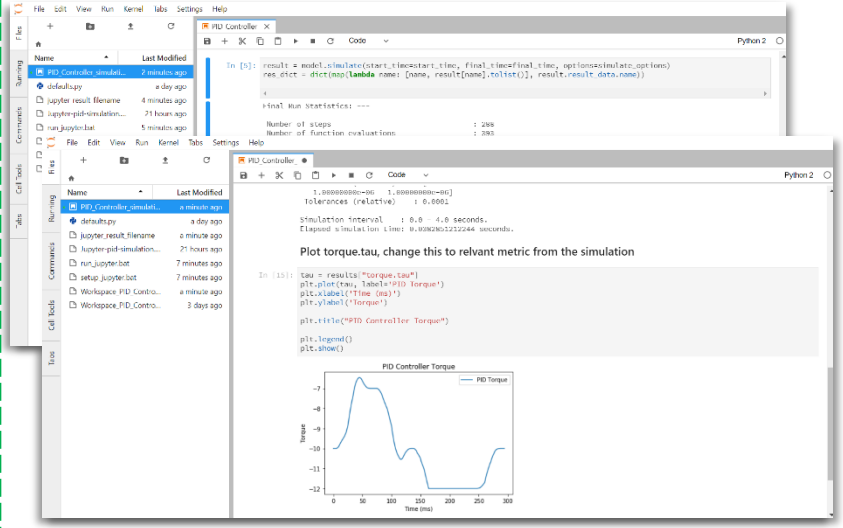
### 1. Build and simulate



**Model centric default view**

- Operation by point and click
- Build system models

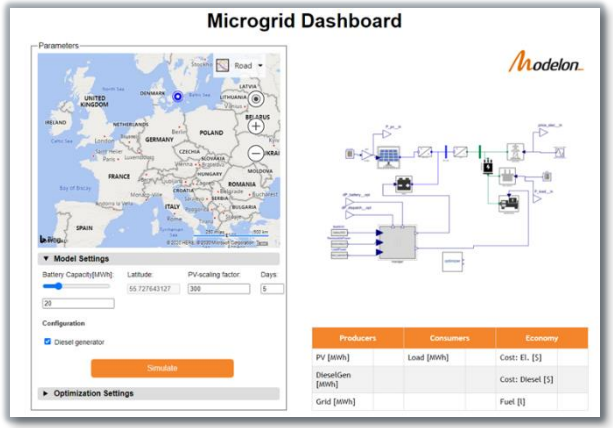
### 2. Create workflows




**Notebook view**

- Scripts
- Automatic or interactive execution

### 3. Deploy




**Dashboard view**

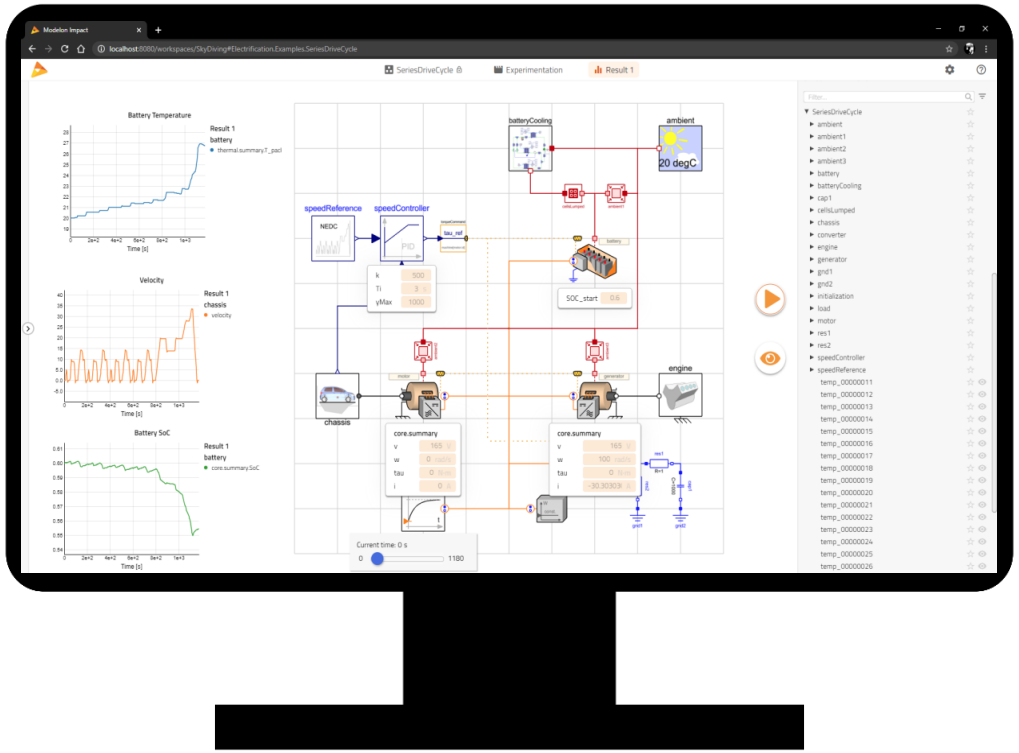
- Simplified/Specialized view
- Unlimited customization
- Compile on demand

-- UI -- API

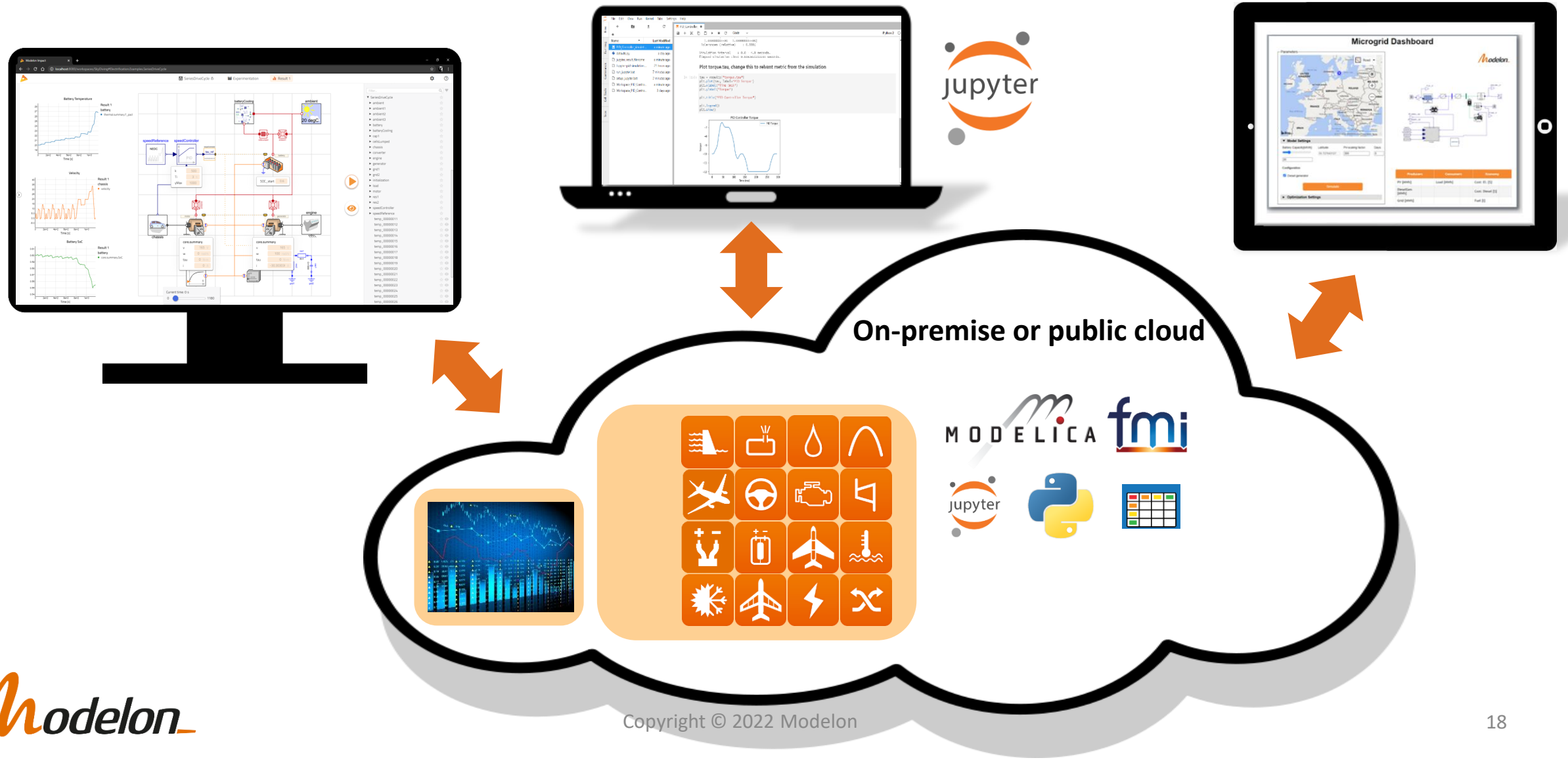
# DESIGNED FOR COLLABORATION

Workspaces contains:

- Models
- Views
- Results
- Data files

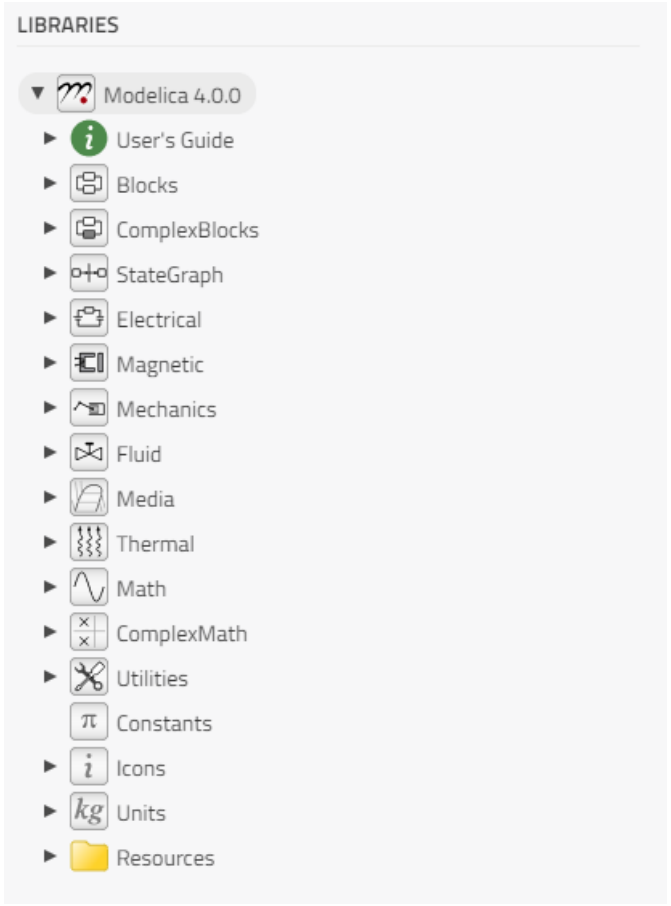


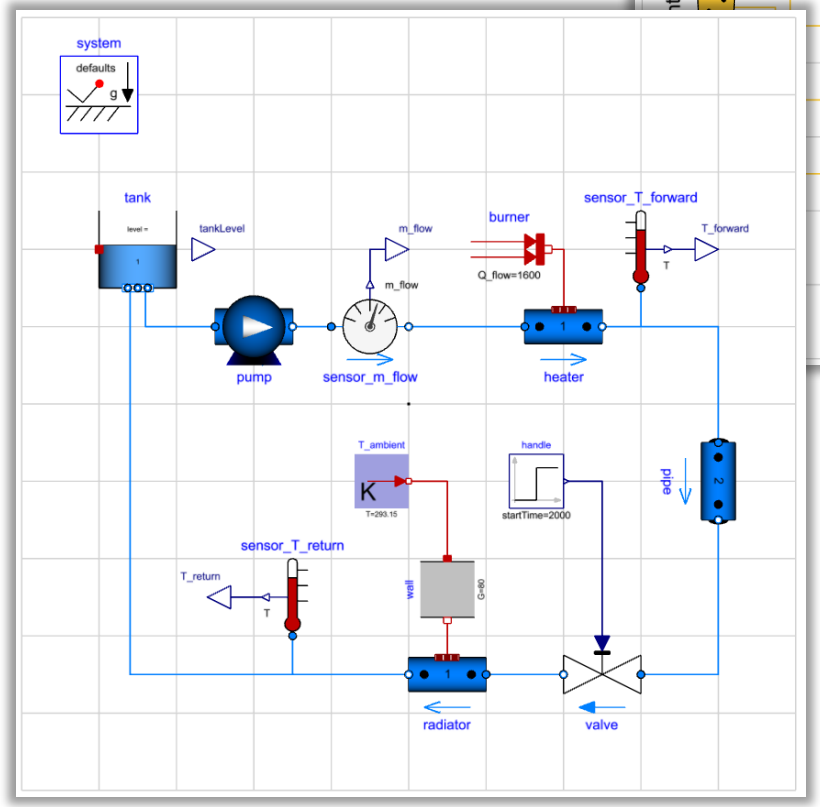
# ARCHITECTURE OVERVIEW

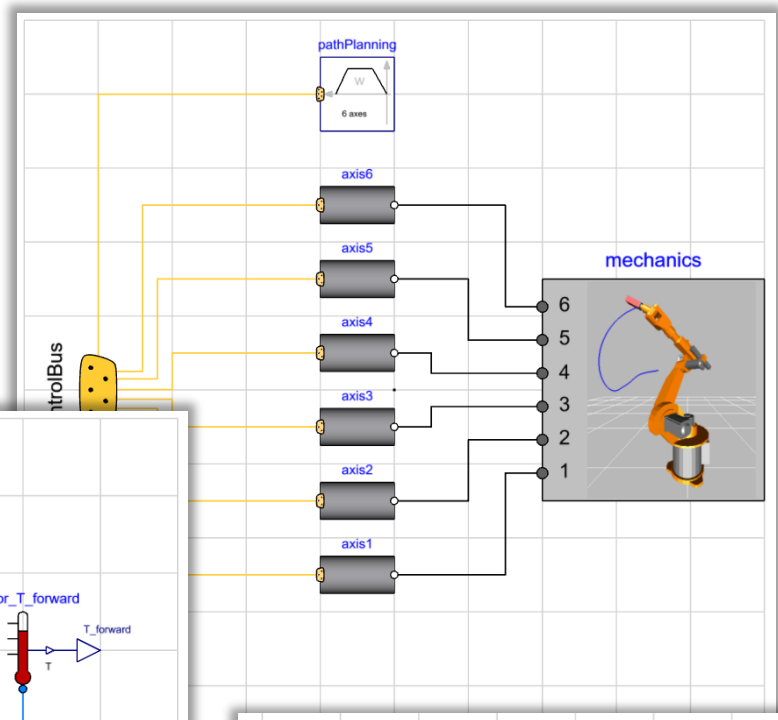


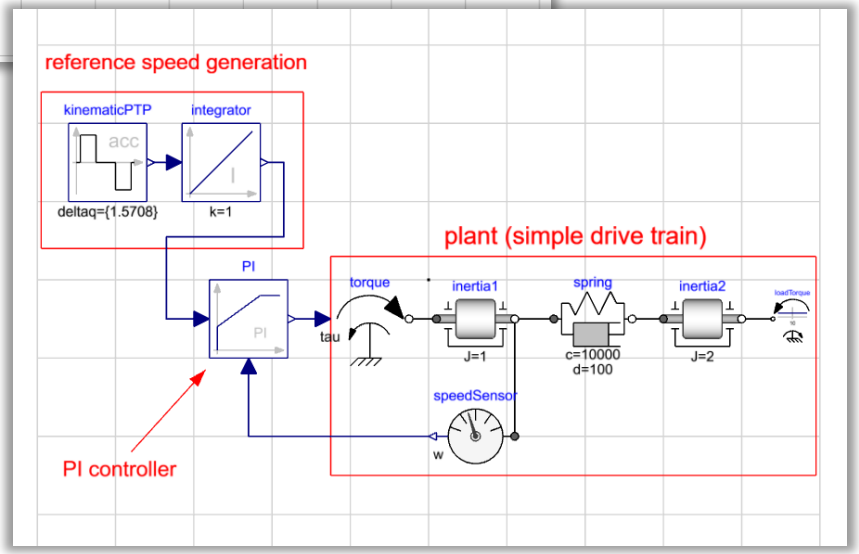
# MODEL LIBRARIES

# MODELICA STANDARD LIBRARY



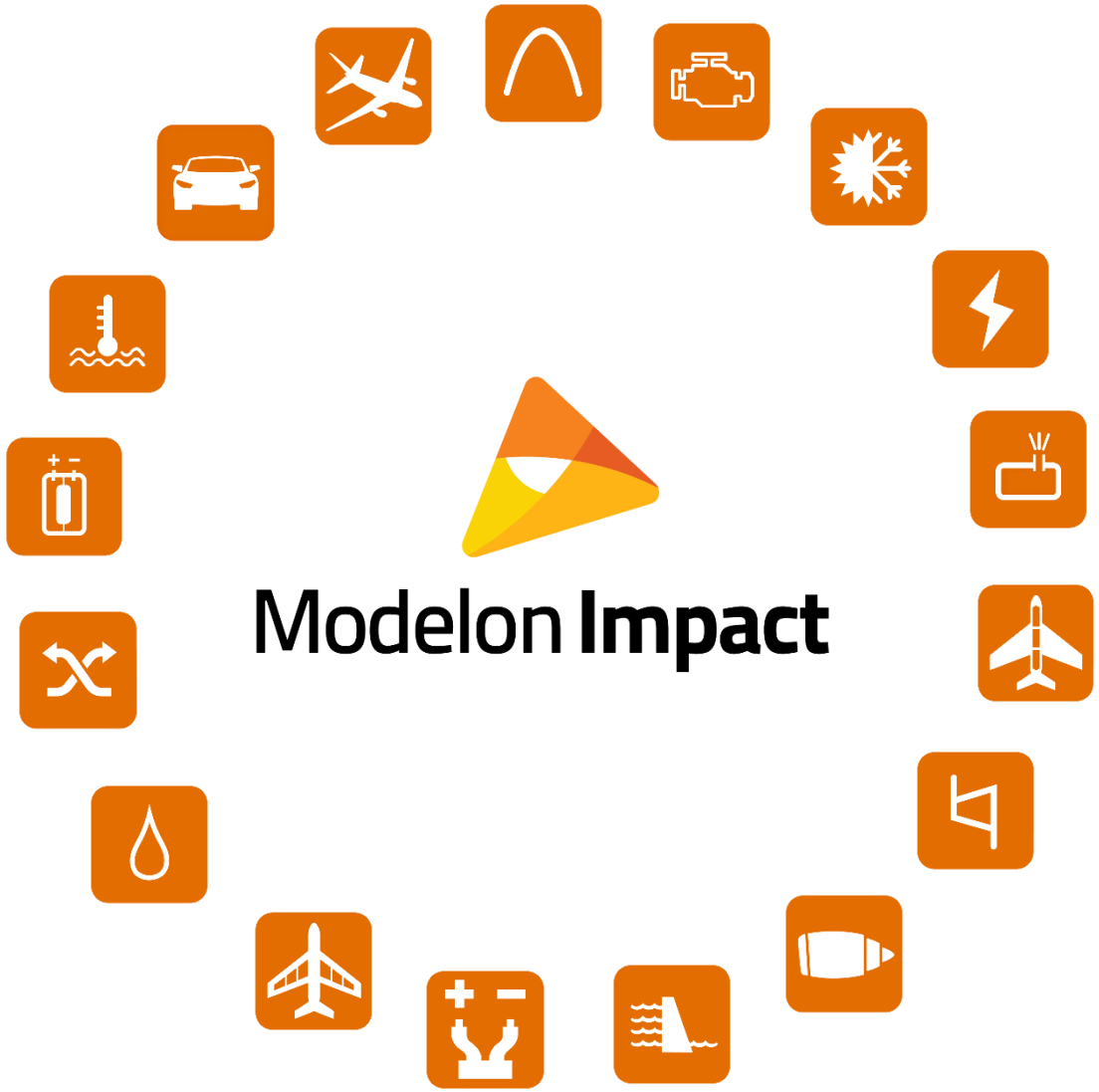



# Modelon Library Suite

*Powered by Modelica*

Our industry leading suite of libraries available in Modelon Impact are built on the Modelica standard, delivers state-of-the art system models for a wide range of industries including automotive, aerospace, industrial equipment, and energy and process.




Modelon

# FMI TECHNOLOGY

# WHAT IS FMI?

- Sister technology to Modelica
- Maintained by Modelica Association
- Standardized way for models from several tools to interact:
  - Co-simulation (each model contains its own integrator)
  - Model-exchange (models from several tools integrated by one master)
- Allows for export and import of models to and/or from simulation environments that does not support Modelica, e.g.
  - FMI toolbox for Matlab
  - Ansys Twinbuilder
  - ADAMS, Simpack, CarMaker
- See www.fmi-standard.org for further information

Modelon