BESMod - A Modelica Library providing Building Energy System Modules

Fabian Wüllhorst, Laura Maier, David Jansen,
Larissa Kühn, Dominik Hering, Dirk Müller

American Modelica Conference 2022

Motivation

Domain-coupled building sector is a foundation for integration of renewable energy

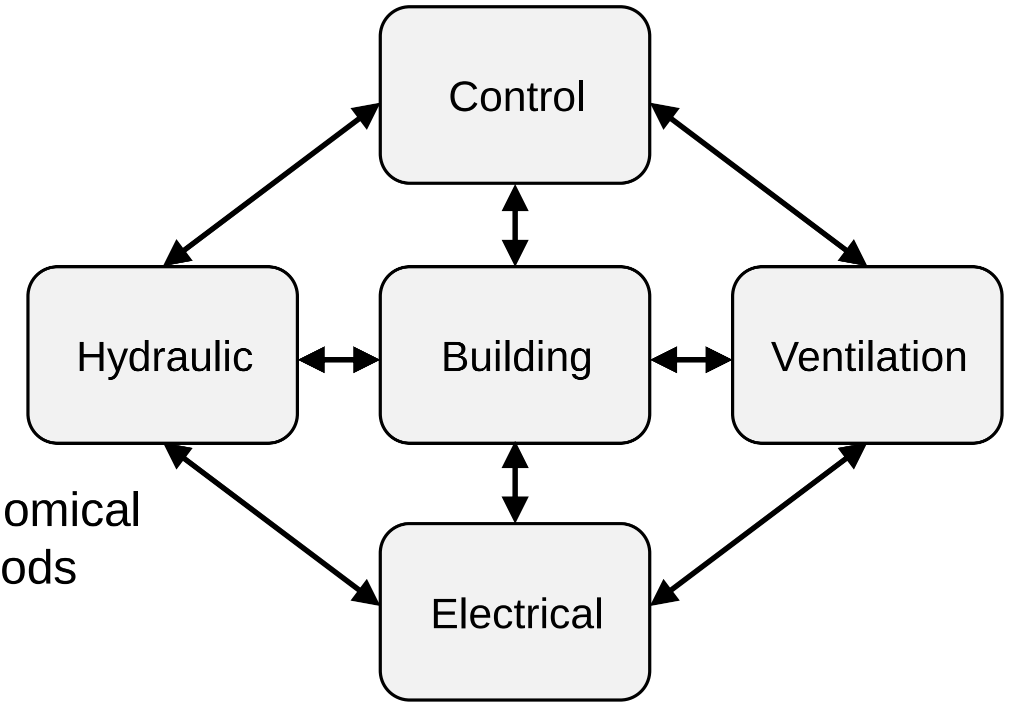
Climate Change requires installation of renewables and a shift towards a more electrified energy system

Development of new design and control methods is required

Simulation can serve as a fast and economical option to analyze and optimize new methods

Consideration of domain-coupling is vital in renewable building energy systems

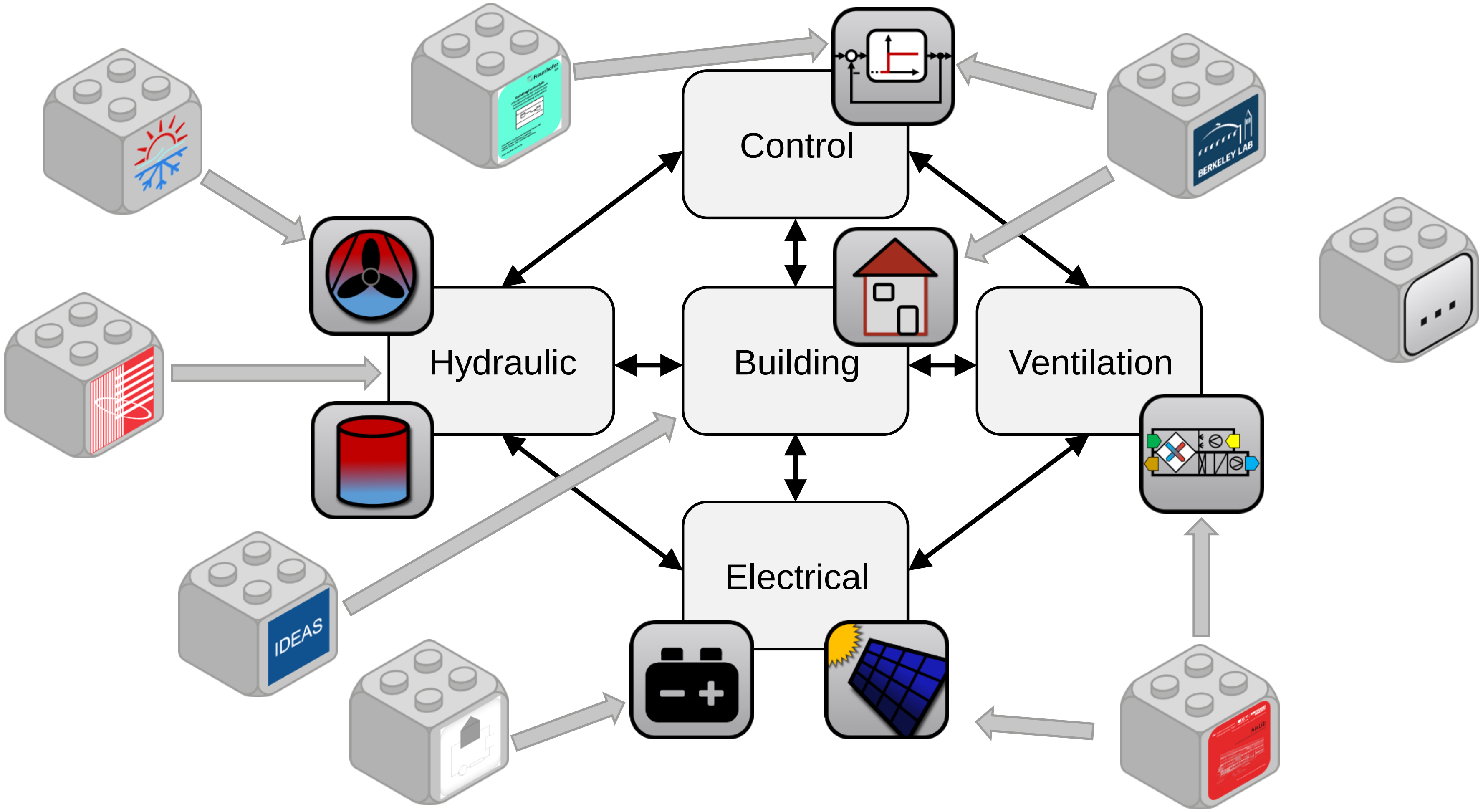
Modelica enables the modeling and coupling of different domains



Are there libraries for the modeling of domain-coupled building energy systems?

Related work

Several libraries provide component models for all relevant domains



✓ Rich pool of component models for all domains available in **twelve** libraries

Current gap

No library targets the domain-coupled simulation and analysis of build energy systems

Rich pool of component models for all domains available in **twelve** libraries

No single library provides
the “best” models for all domains

Lack of system models, uniform interfaces
and consistent parameterization

Use models from multiple libraries

Provide systems using a modular structure
and consistent parameterization

Develop **BESMod**,
a library providing
Building **E**nergy **S**ystem **Mod**ules

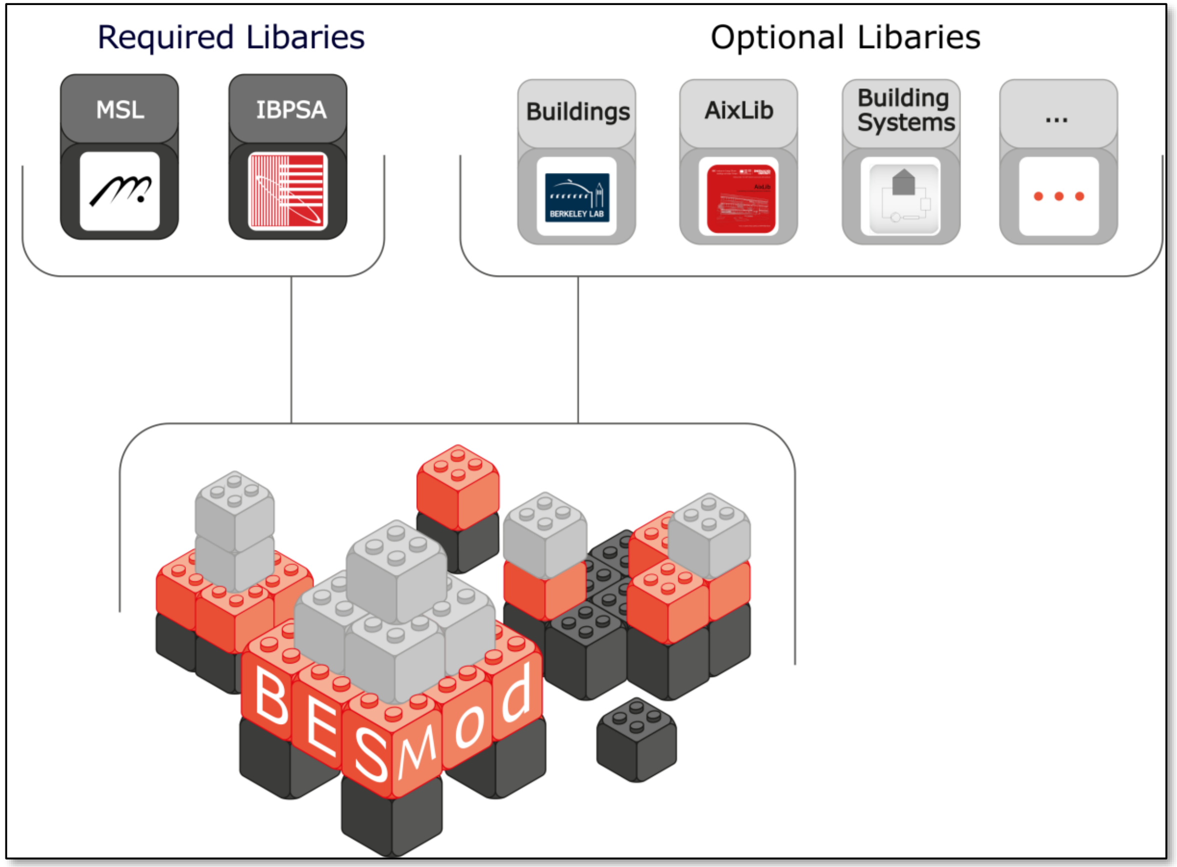
BESMod

General idea and scope of BESMod

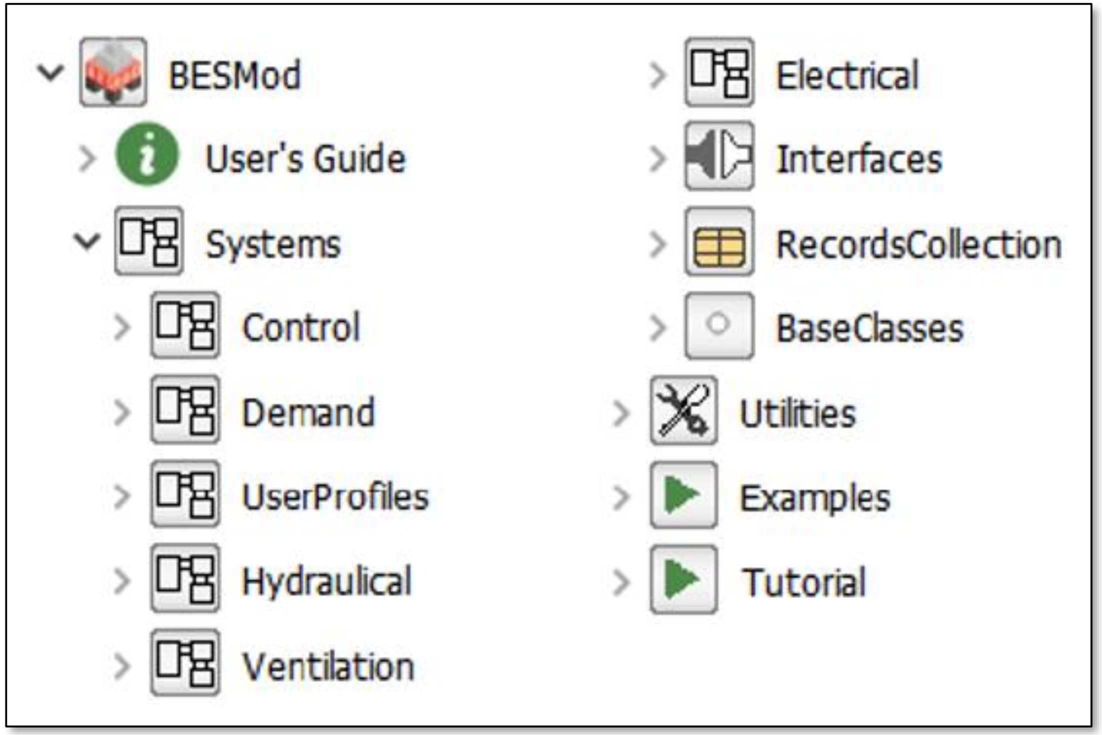
Use models from multiple libraries

Provide systems using a modular structure and consistent parameterization

- BESMod is built upon component libraries



- BESMod is fully modular
  - A (sub)-system is a module
  - Modules/subsystems for all domains



BESMod
- User's Guide
- Systems
  - Control
  - Demand
  - UserProfiles
  - Hydraulical
  - Ventilation
- Electrical
- Interfaces
- RecordsCollection
- BaseClasses
- Utilities
- Examples
- Tutorial

BESMod

Approach for a modular subsystem design

Bus connectors

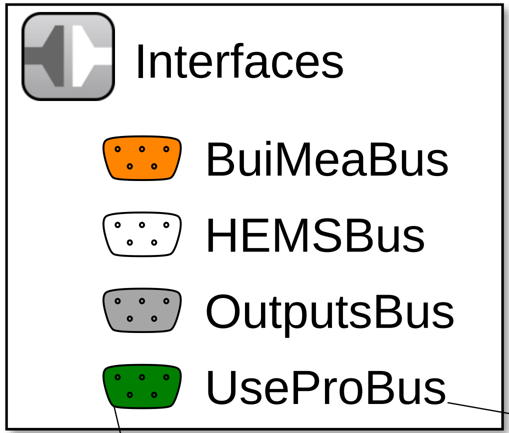
- Color-coding
- No pre-defined variables
- No usage in the uppermost system

Vector sized ports

- Multiple zones
- Multiple generation systems

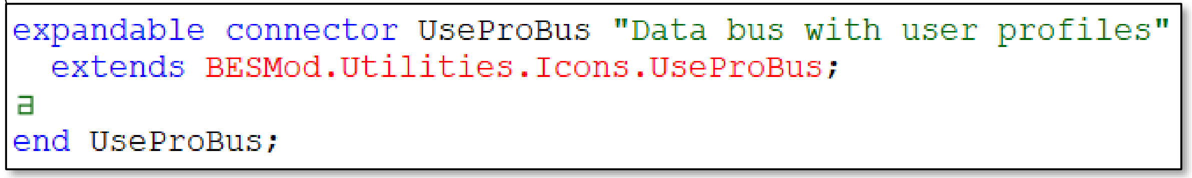
Replaceability

- All modules are *constrainedby* some *PartialModule*
- Usage of *choicesAllMatching* and *modifiers*

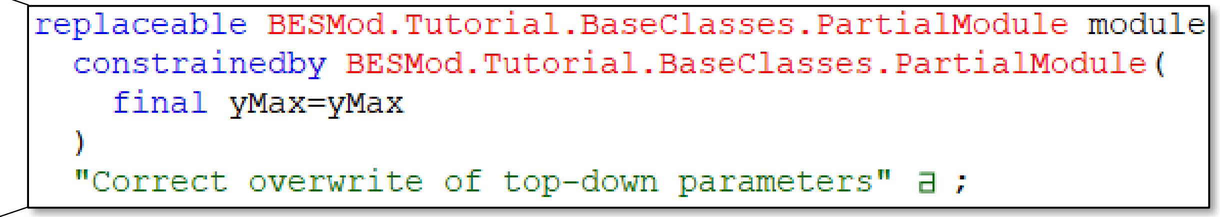


Interfaces

- 🟧 BuiMeaBus
- ⬜ HEMSBus
- ⬛ OutputsBus
- 🟩 UseProBus



```
expandable connector UseProBus "Data bus with user profiles"
  extends BESMod.Utilities.Icons.UseProBus;
ᴲ
end UseProBus;
```



```
replaceable BESMod.Tutorial.BaseClasses.PartialModule module
  constrainedby BESMod.Tutorial.BaseClasses.PartialModule(
    final yMax=yMax
  )
  "Correct overwrite of top-down parameters" ᴲ ;
```

BESMod

Uniform parameterization approach based on four principles

1. Top-Down parameters

- Given by the parent or adjacent systems
- Always *final*

2. Bottom-Up parameters

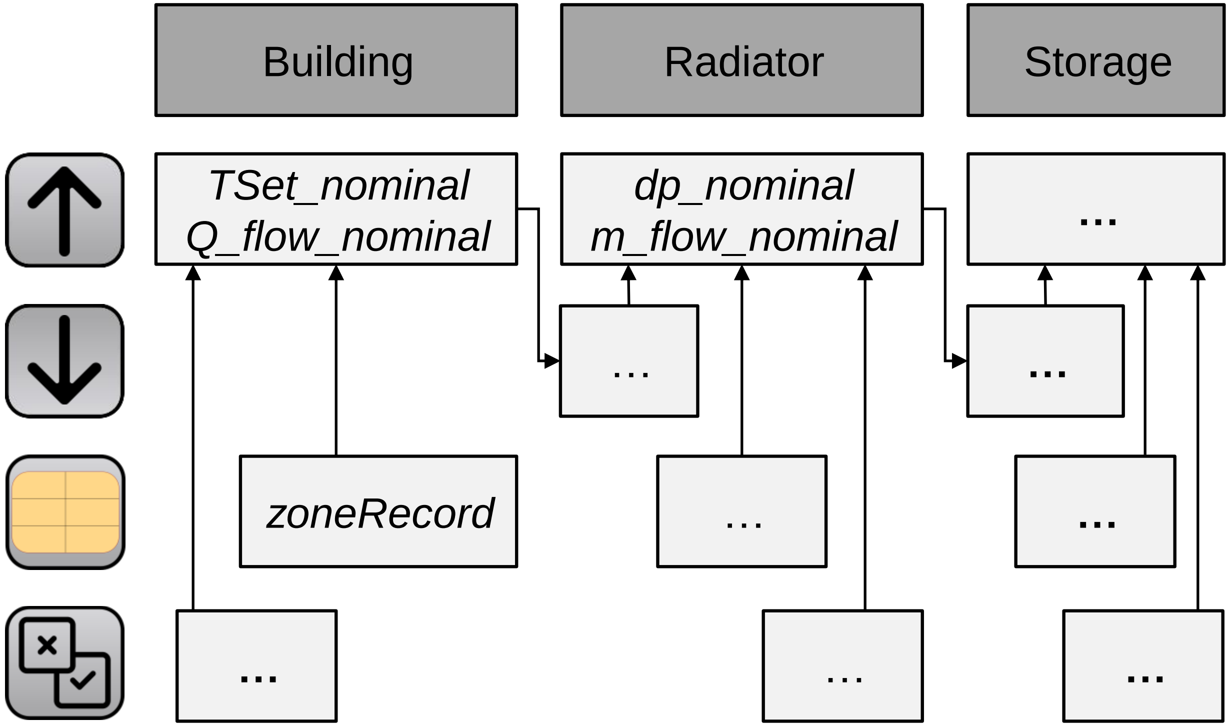
- Defined by remaining parameters
- Not *final*, enables fine-tuning

3. Records for component data

- Only component physics
- Usage of top-down parameters

4. Component choices

- E.g. use the bypass valve



BESMod

The building envelope is the core of the energy system

Layout

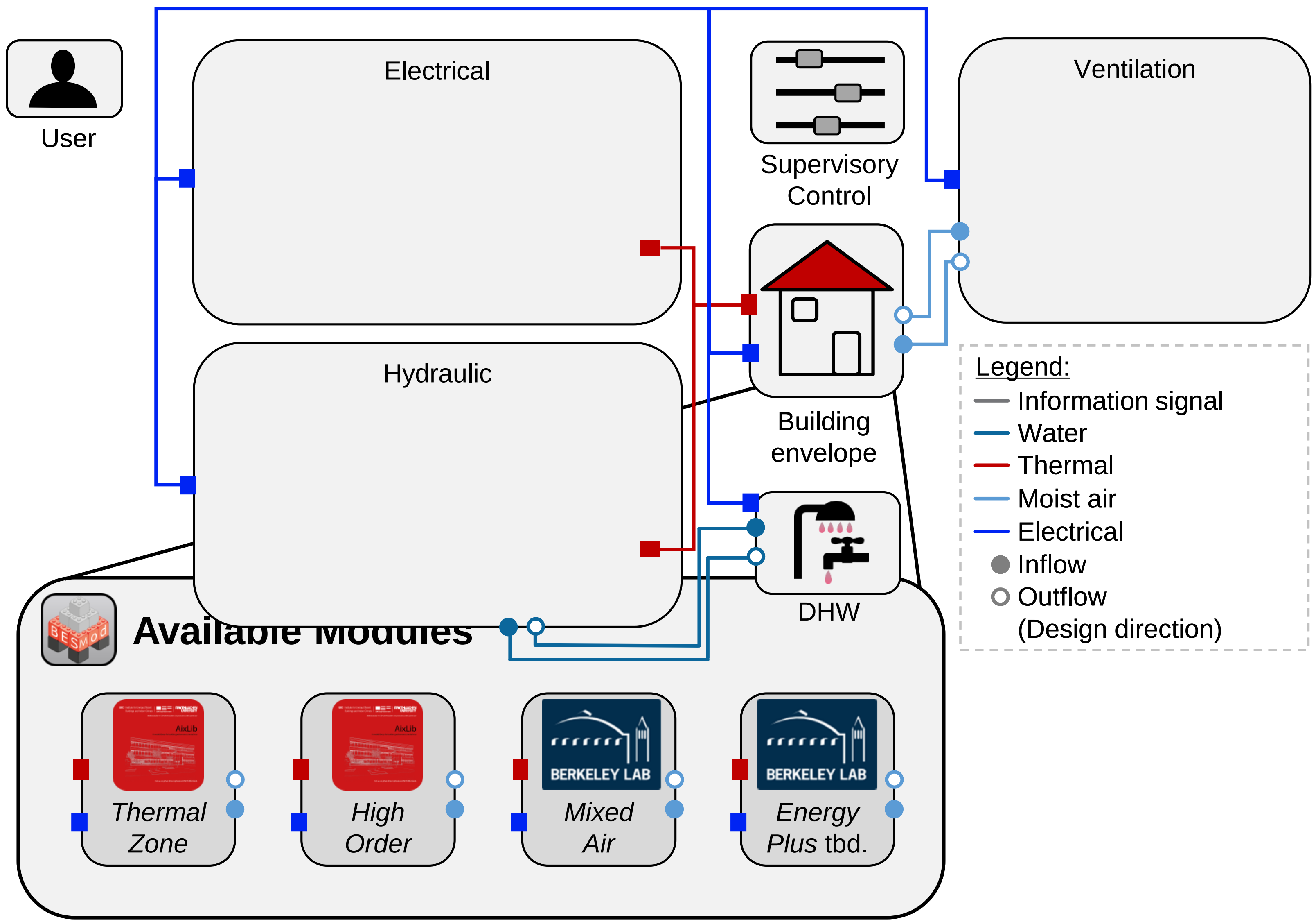
- Only building envelope
- User profiles are separated and depend on building model

Connectors

- Thermal
- Ventilation (moist air)
- Electrical

Parameterization

- Nominal T and $\dot{Q}$ as bottom-up
- Geometry as bottom-up



BESMod
Description of the hydraulic subsystem

Layout

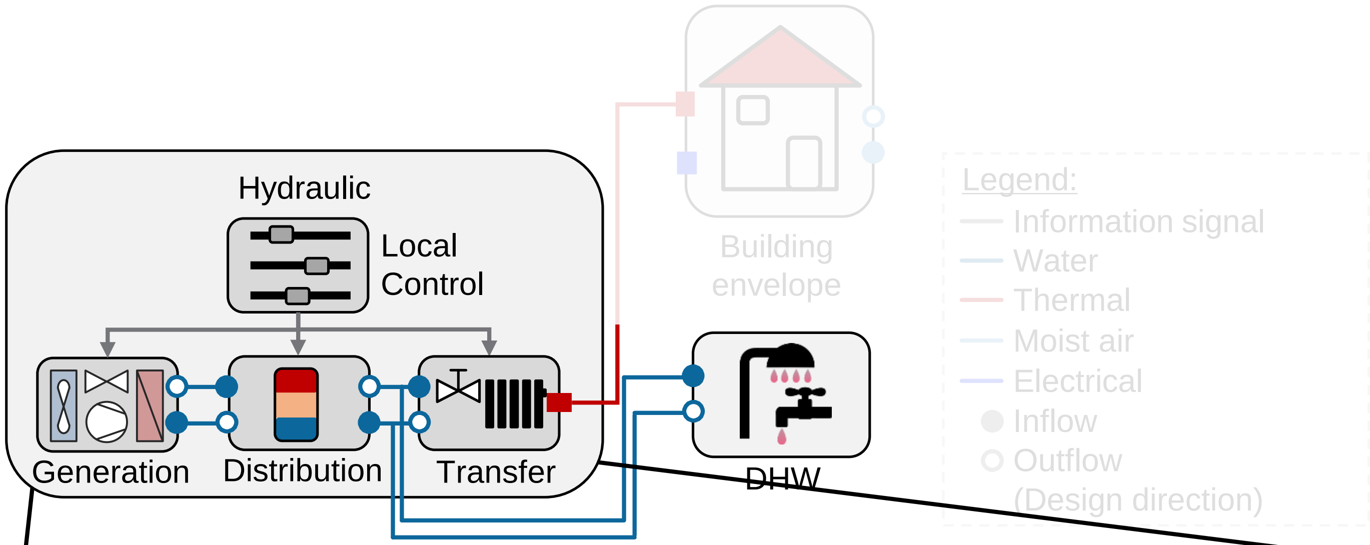
- According to EN 15316-1
- Similar setups in ASHRAE Standard 111

Connectors

- Fluid for DHW
- Heat for building

Parameterization

- Records as function of top-down parameters and constants
- Design rules presented in previous work



Available Modules

Heat pump, boiler, storages, controls radiators, UFH

SolarThermal HeatPump

Description of the electrical subsystem

Layout

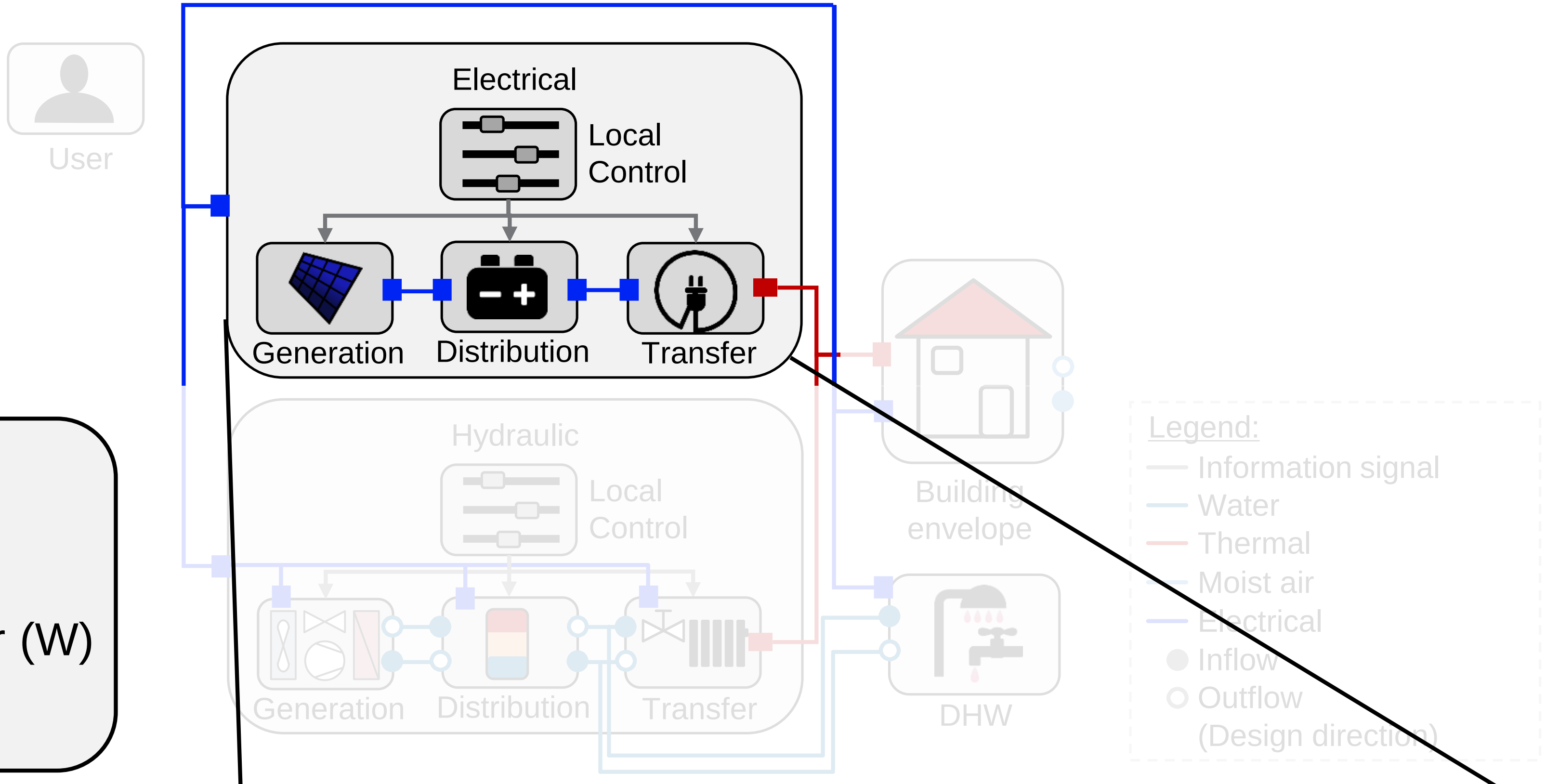
- According to EN 15316-1
- Transfer system for e.g. infrared heating

Connectors

- Component libraries use power
- No domain-coupling, use of power (W)
- No usage of voltage and current

Parameterization

- PV sizing based on the roof area (top-down parameter)



Available Modules

PVAndBattery

DirectGridConnection

BESMod
Description of the remaining subsystems

Ventilation

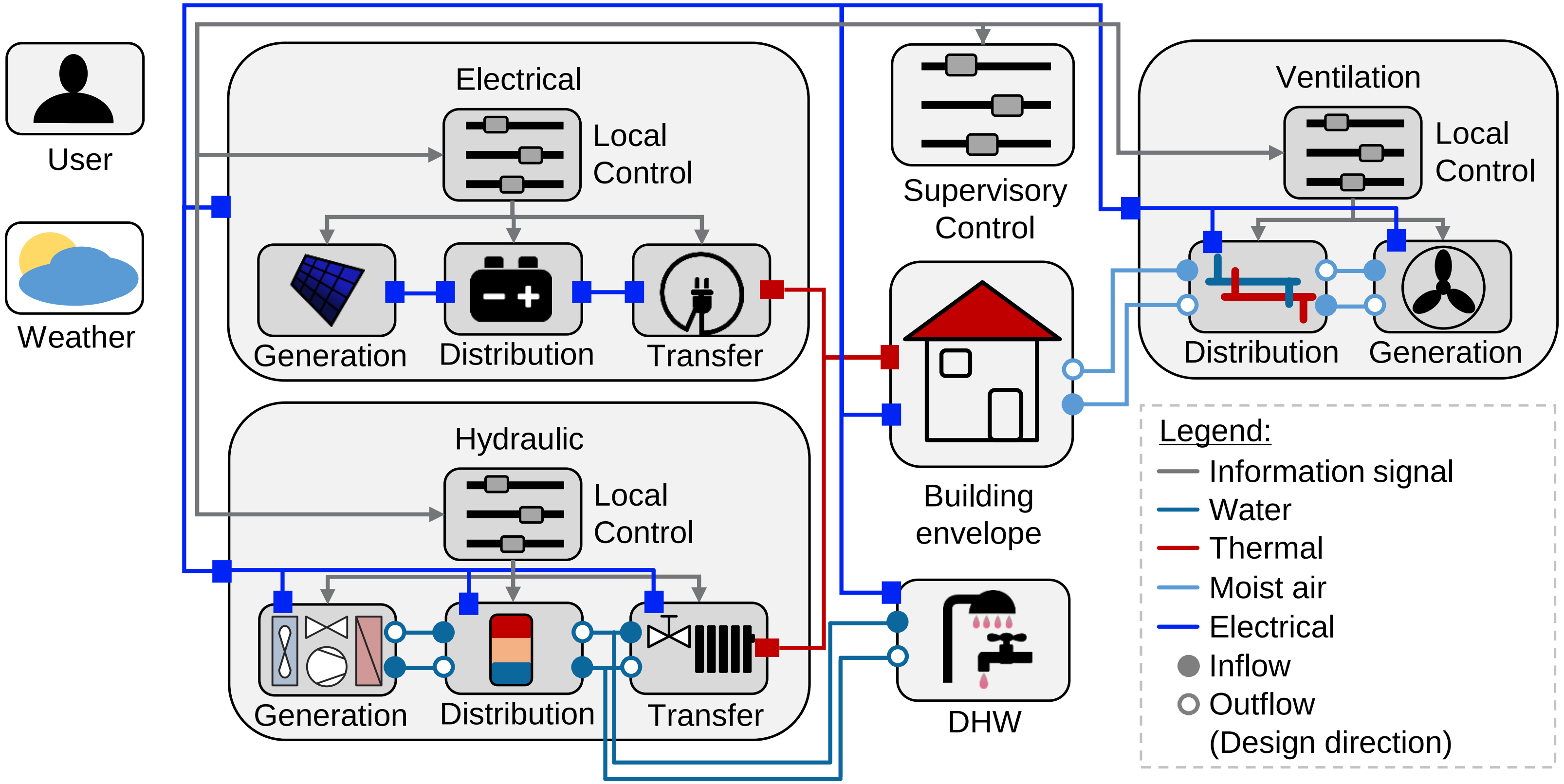
- Similar to hydraulic system
- No transfer system required

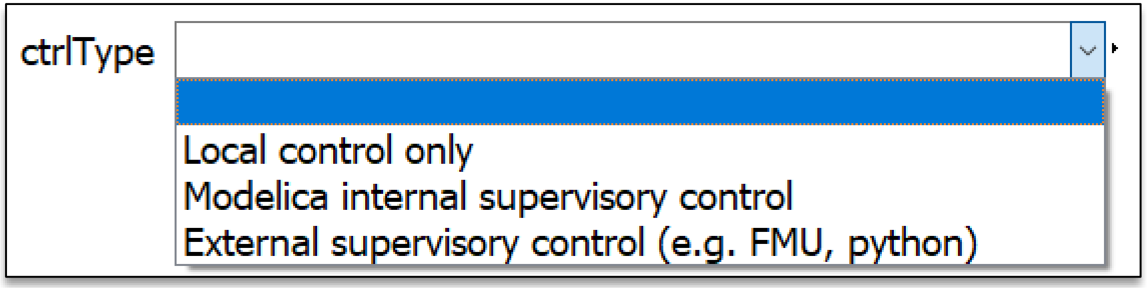
Weather

- Not replaceable
- TMY3-Reader so far works for all cases

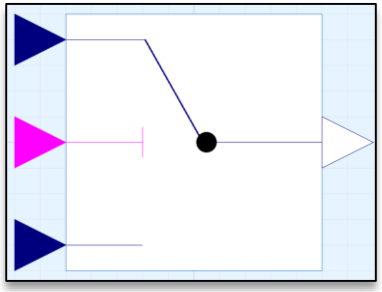
Supervisory control

- Connected to all local controls
- Type is defined in local control
- Internal supervisory control or external via e.g. BOPTEST




ctrlType

Local control only
Modelica internal supervisory control
External supervisory control (e.g. FMU, python)



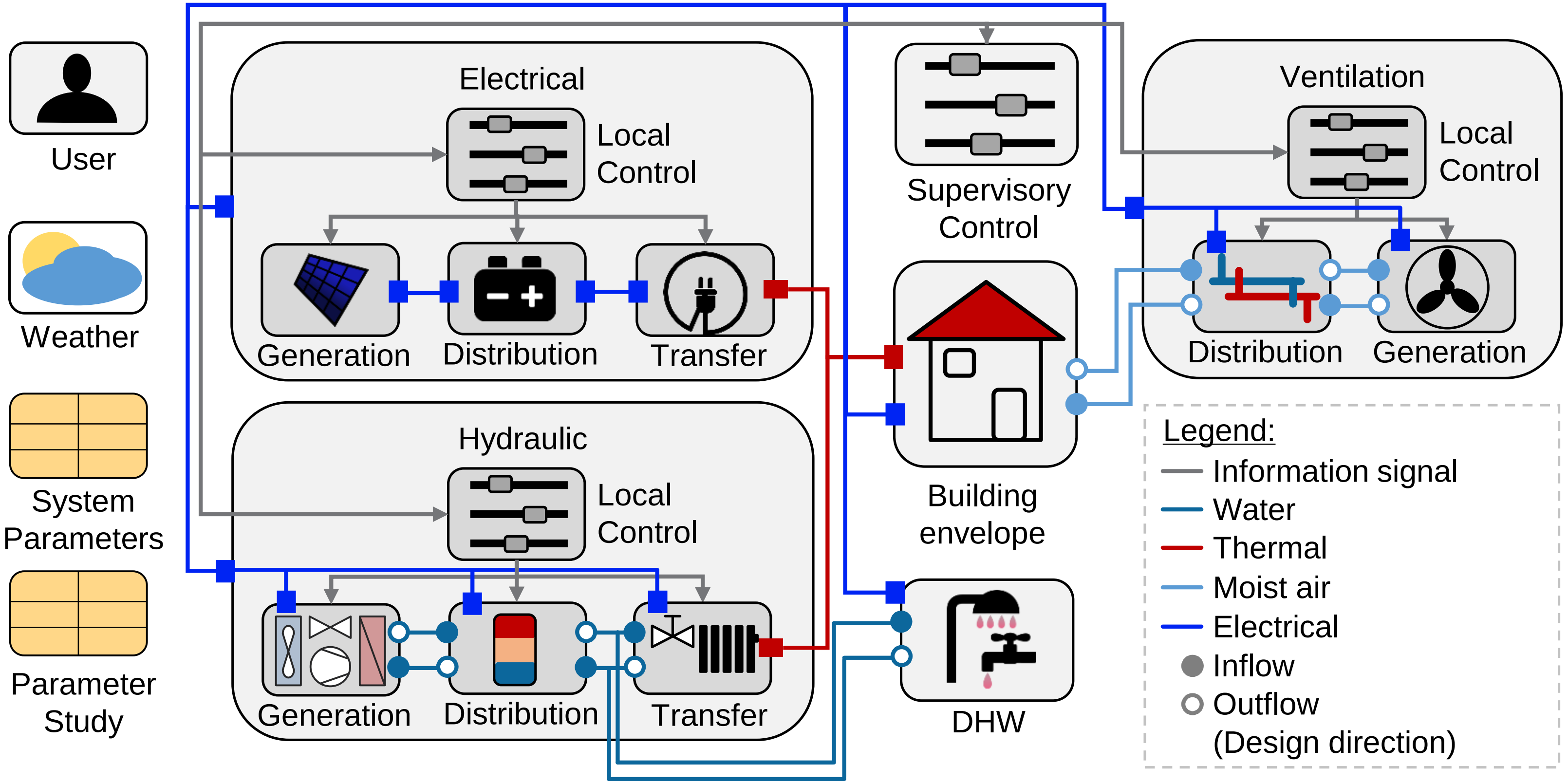
BESMod
Steps to aggregate a coupled building energy system

Simulation

- Redeclare subsystems
  - Select component choices
  - Choose component records
  - Fine-tune bottom-up parameters
- Choose weather file
- Overwrite parameters to study
- Simulate

Debugging

- Disable single subsystems
- Test new subsystems in dedicated *Tests* package



Available Systems

Bivalent heat pump, thermal storages, radiator
PV and battery
Ventilation with heat recovery
Supervisory control for DHW storage superheating

Several other Examples

Proof of concept

Comparison of two building models with the same HVAC energy system

Proof of concept

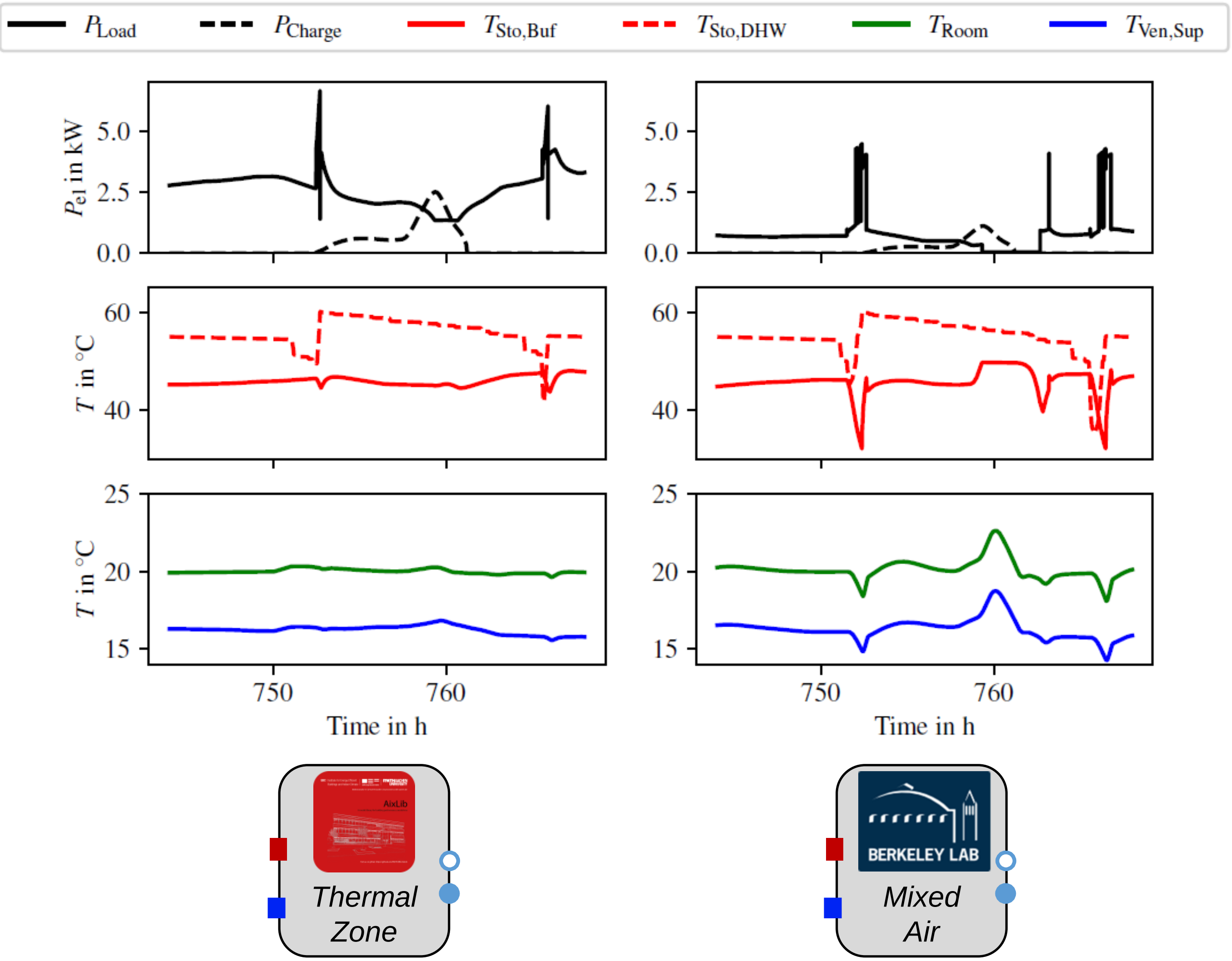
- Comparison of *ThermalZone* to *MixedAir*
- Model for *PartialUseCase*
- *User* and *building* modules not replaced

Highlights

- Only graphical interaction
- No additional connection on top-level
- Annual simulations take < 6 min

Results

- Different Building physics
- PV sizes with roof area
- *MixedAir* with higher solar gains



Conclusion

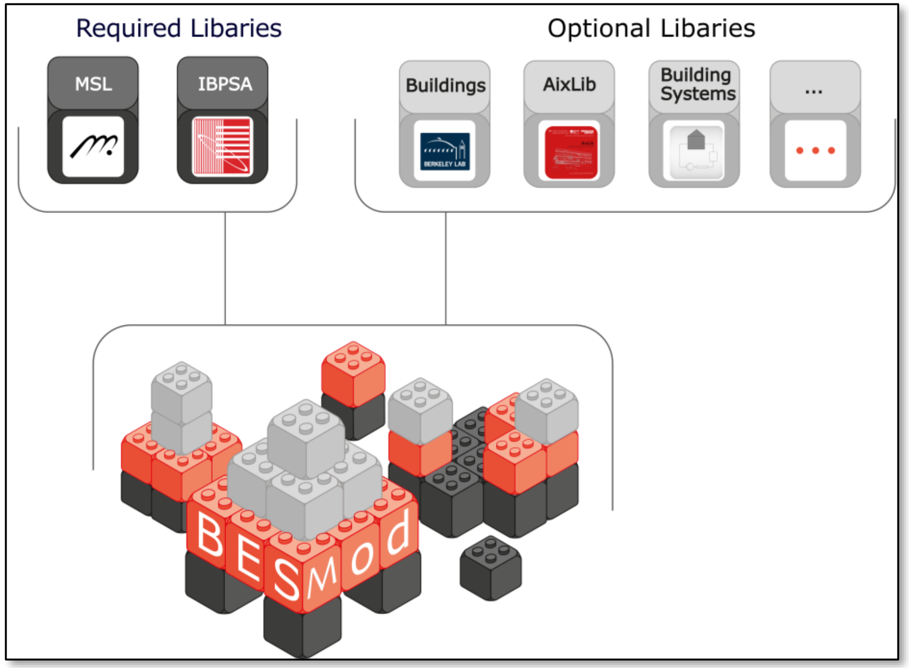
Several areas for future development exists

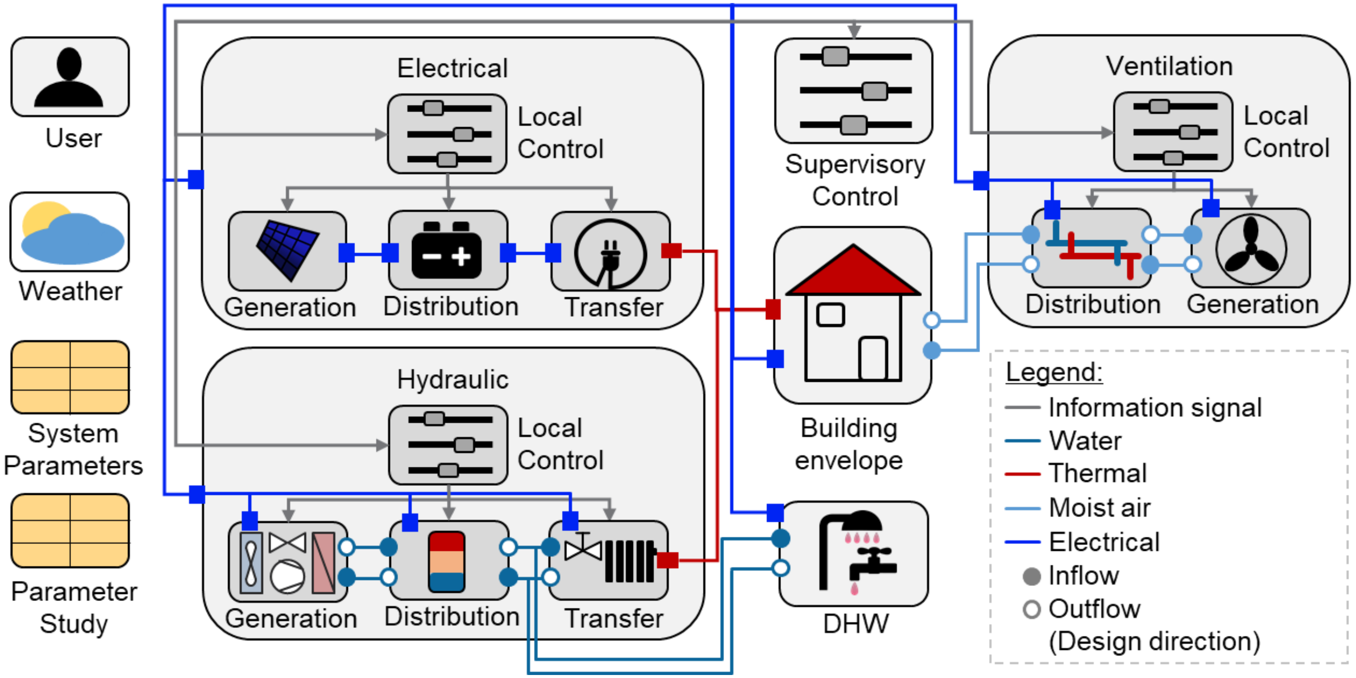
Library Development

- Electrical connectors update in IBPSA
- Validation of coupled systems
- Compatibility to OpenModelica
- Extension of Continuous Integration

Future Use-Cases

- Control development and testing
  - Development of cases for BOPTEST?
  - Coupling to Ontologies such as Brick
- Simulation based design optimizations





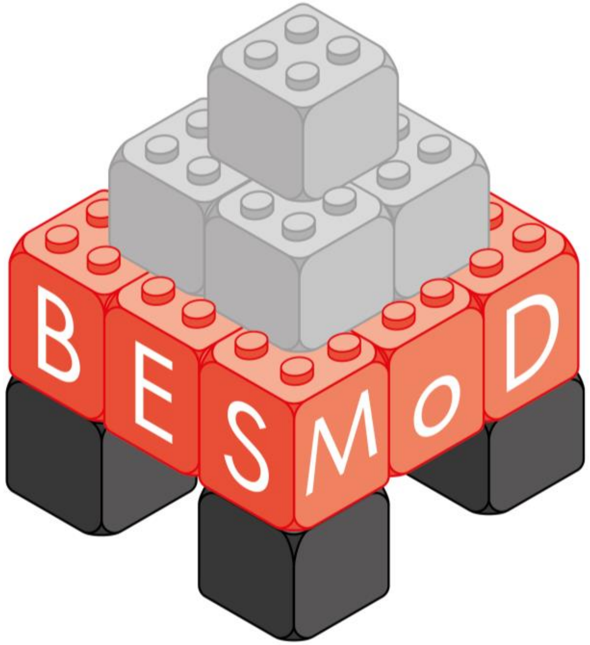
Conclusion

Already active developer community in Aachen hopes for international support

- BESMod is only as good as the component libraries and the community!
- Active user community in Aachen
  - 20 Students and 10 Ph.D. candidates
  - Currently, we perform personal workshops to explain the usage
  - Better documentation and YouTube tutorials to follow
- Invitation to use and further develop BESMod

Visit and use BESMod:

https://github.com/RWTH-EBC/BESMod



Supported by:

Federal Ministry
for Economic Affairs
and Climate Action

on the basis of a decision
by the German Bundestag

Promotional reference 03ET1495A.

International
Building
Performance
Simulation
Association

This work emerged from the IBPSA Project 1.

BESMod
Approach for a modular subsystem design

Bus connectors

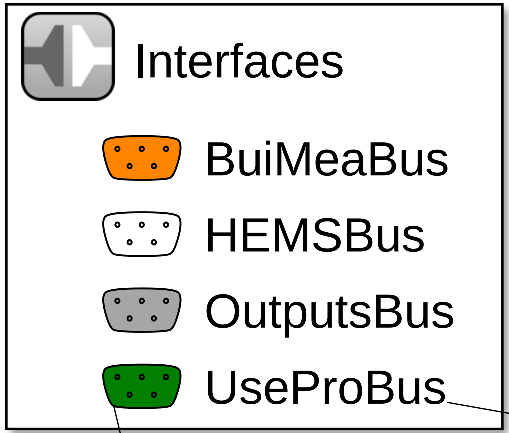
- Color-coding
- No pre-defined variables
- No usage in the uppermost system

Vector sized ports

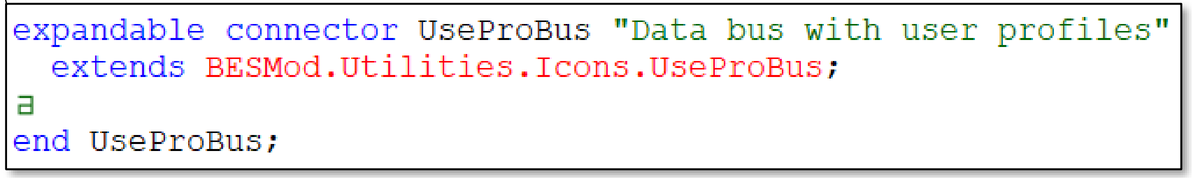
- Multiple zones
- Multiple generation systems

Replaceability

- All modules are *constrainedby* some *PartialModule*
- Usage of *choicesAllMatching* and *modifiers*


Interfaces

- BuiMeaBus
- HEMSBus
- OutputsBus
- UseProBus



```
expandable connector UseProBus "Data bus with user profiles"
  extends BESMod.Utilities.Icons.UseProBus;
a
end UseProBus;
```

Compatibility to OpenModelica

- Arrays in expandable connectors
- Expandable in expandable
- Replaceable arrays