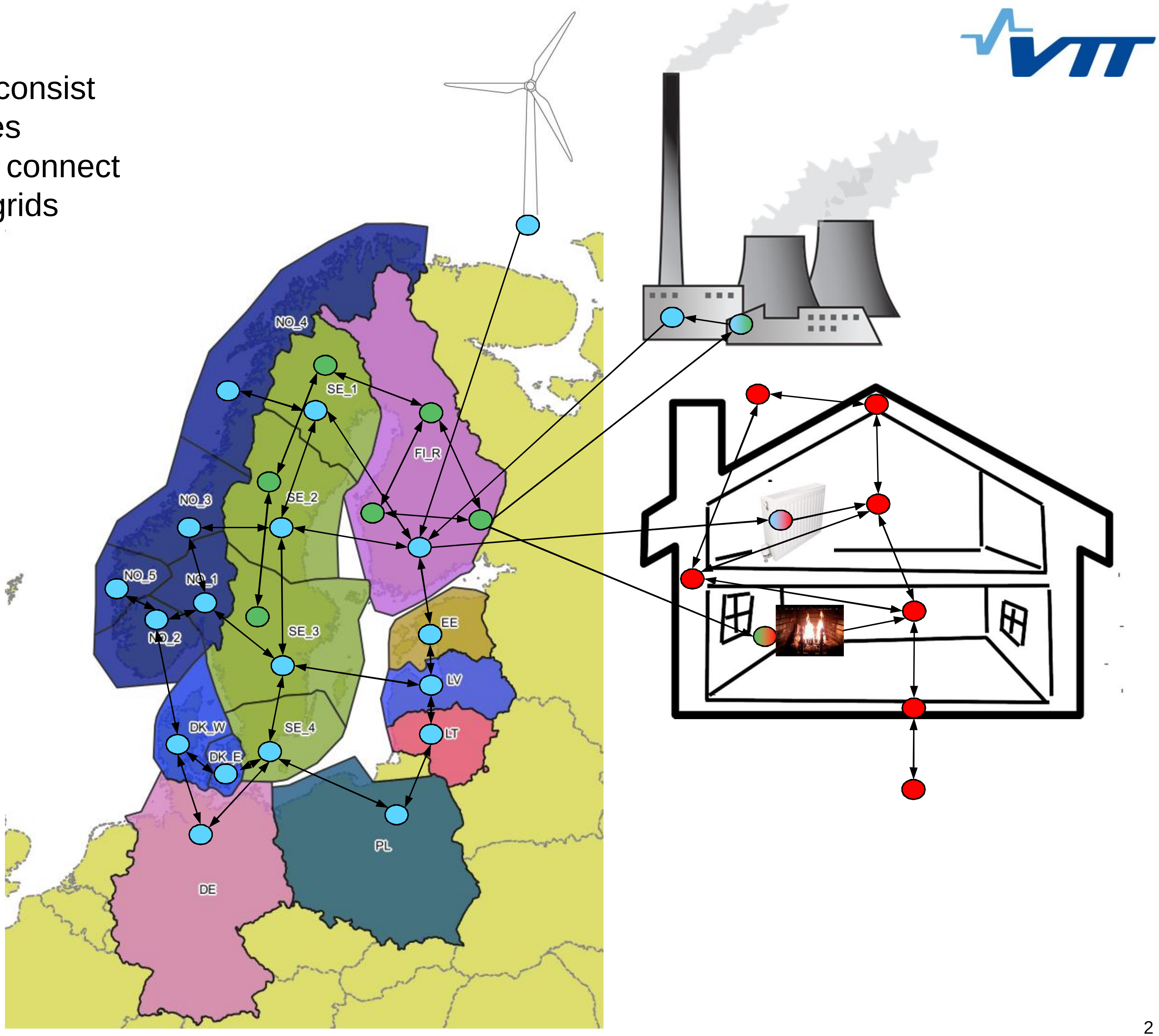
A generalized energy system model

Juha Kiviluoma

Senior scientist, VTT

Senior energy systems researcher, UCD

- Energy systems consist of grids and nodes
- Some nodes can connect different energy grids



Power, district heat, building heat, gas, river systems, fuel production, fuel resources, industrial processes…

- Multiple systems generating, storing and consuming energy
- Increasing integration through electrification
- Representing them all can lead to bunch of equations

Power and heat with investments:

```
EQUATIONS
  QOBJ                  'Objective function'
  QEEQ(RRR,S,T)         'Electricity generation equals demand'
  QHEQ(AAA,S,T)         'Heat generation equals demand'
  QECAP(RRR)            'Power generation and possible imports greater than exogenous value'
  QHCAP(AAA)            'Heat generation capacity greater than exogenous value'
  QGCBGBPR(AAA,G,S,T)   'CHP generation (back pressure) limited by Cb-line'
  QGCBGEXT(AAA,G,S,T)   'CHP generation (extraction) limited by Cb-line'
  QGCVGEXT(AAA,G,S,T)   'CHP generation (extraction) limited by Cv-line'
  QGGETOH(AAA,G,S,T)    'Electric heat generation'
  QGNCBGBPR(AAA,G,S,T)  'CHP generation (back pressure) Cb-line, new'
  QGNCBGEXT(AAA,G,S,T)  'CHP generation (extraction) Cb-line, new'
  QGNCVGEXT(AAA,G,S,T)  'CHP generation (extraction) Cv-line, new'
  QGNGETOH(AAA,G,S,T)   'Electric heat generation, new'
  QGE2LEVEL(AAA,G,S,T)  'Variation on slowly regulated technologies'
  QGEKNT(AAA,G,S,T)     'Generation on new electricity cap, limited by capacity'
  QGHKNT(AAA,G,S,T)     'Generation on new IGHH cap, limited by capacity'
  QGKNHYRR(AAA,G,S,T)   'Generation on new hydro-ror limited by capacity and water'
  QGKNWND(AAA,G,S,T)    'Generation on new windpower limited by capacity and wind'
  QGKNSOLE(AAA,G,S,T)   'Generation on new solarpower limited by capacity and sun'
  QGKNSOLH(AAA,G,S,T)   'Generation on new solarheat limited by capacity and sun'
  QHYRSSEQ(AAA,S)       'Hydropower with reservoir seasonal energy constraint'
  QHYMING(AAA,S,T)      'Regulated and unregulated hydropower production lower than capacity'
  QESTOVOLT(AAA,G,S,T)  'Electricty storage dynamic equation (MWh)'
  QESTOVOLT_CIRCULAR(AAA,G,S,T)  'Electricty storage dynamic equation (MWh)'
  QESTOVOLT_P2G(AAA,G,S,T)  'Electricty storage dynamic equation for P2G (MWh)'
  QESTOVOL_S_T(AAA,G,S,T)  'Bind S periods to each other'
  QHSTOVOLT(AAA,G,S,T)  'Heat storage dynamic equation (MWh)'
  QHSTOVOLT_CIRCULAR(AAA,G,S,T)  'Heat storage dynamic equation (MWh)'
  QHSTOUNLDTLIM(AAA,G,S,T) 'Upper limit to heat storage unloading (MW)'
  QHSTOLOADTLIM(AAA,G,S,T) 'Upper limit to heat storage loading (MW)'
  QESTOUNLDTLIM(AAA,G,S,T) 'Upper limit to electricity storage unloading (MW)'
  QESTOLOADTLIM(AAA,G,S,T) 'Upper limit to electricity storage loading (MW)'
  QHYDROSTOLOADTLIM(AAA,G,S,T) 'Upper limit to pumped hydro storage loading (MW)'
  QHSTOVOLTLIM(AAA,G,S,T)  'Heat storage capacity limit (MWh)'
  QESTOVOLTLIM(AAA,G,S,T)  'Electricity storage capacity limit (MWh)'
  QKFUELC_FIX(C,FFF)    'Total capacity using fuel FFF is fixed in a country'
  QKFUELC(C,FFF)        'Total capacity using fuel FFF is limited in country'
  QKFUELR(RRR,FFF)      'Total capacity using fuel FFF is limited in region'
  QKFUELA(AAA,FFF)      'Total capacity using fuel FFF is limited in area'
  QGMINFUELC(C,FFF)     'Minimum electricity generation by fuel per country'
  QGMAXFUELC(C,FFF)     'Maximum electricity generation by fuel per country'
  QGMINFUELR(RRR,FFF)   'Minimum electricity generation by fuel per region'
  QGMAXFUELR(RRR,FFF)   'Maximum electricity generation by fuel per region'
  QGMINFUELA(AAA,FFF)   'Minimum electricity generation by fuel per area'
  QGMAXFUELA(AAA,FFF)   'Maximum electricity generation by fuel per area'
  QGMINCF(C,FFF)        'Minimum fuel usage per country constraint'
  QGMAXCF(C,FFF)        'Maximum fuel usage per country constraint'
  QGEQCF(C,FFF)         'Required fuel usage per country constraint'
  QGMINRF(RRR,FFF)      'Minimum fuel usage per region constraint'
  QGMAXRF(RRR,FFF)      'Maximum fuel usage per region constraint'
  QGEQRF(RRR,FFF)       'Required fuel usage per region constraint'
  QGMINAF(AAA,FFF)      'Minimum fuel usage per area constraint'
  QGMAXAF(AAA,FFF)      'Maximum fuel usage per area constraint'
  QGEQAF(AAA,FFF)       'Required fuel usage per area constraint'
  QXK(IRRRE,IRRRI,S,T)  'Transmission capacity constraint'
  QXEQUAL(IRRRE,IRRRI)  'Transmission capacity is eqaul in both directions'
  QLIMCO2(C)            'Limit on annual CO2-emission'
  QLIMSO2(C)            'Limit on annual SO2 emission'
  QLIMNOX(C)            'Limit on annual NOx emission'
  QHYFXRW(RRR,S)        'Hydro with reservoir available for this region and season (MWh)'
  QHYFXCW(C,S)          'Hydro with reservoir available for this country and season (MWh)'
  QBASELOAD(C,S,T)      'Minimum production for baseload service providers (MW)'
  QRESDE(RRR,S,T)       'Reserve requirement proportional with demand (MW)'
  QRESWI(RRR,S,T)       'Reserve requirement proportional with wind power (MW)'
  QMAXINVEST(C,FFF)     'Maximal investment by country and fuel during one simulated year (MW)'
  QGMAXINVEST2(C,G)     'Maximum model generated capacity increase from one year to the next (MW)'
```

But…

- Energy conversions and energy transfer follow laws of physics
- Number of different conversions/transfers are limited
- As are the ways to express them

Backbone: Generalised model for energy systems and energy resources

- The core model offers energy conversions and energy transfers that are applicable to any conceivable energy transformation
  - Minimize equations to keep the code tractable
- Input data drives what forms of energy are actually modelled and how conversions and transfers are represented
- Allows stochastics for short-term forecasts and for long-term statistics (e.g. reservoir hydro power)
- New models are defined through model definition files: allows to build new implementations on top of the core engine as needed
- Different models can directly re-use each others results (e.g. investments and operations)

```
m('schedule') = yes;
mSettings('schedule', 't_start') = 1;  // Ord of first solve (i.e. >0)
mSettings('schedule', 't_horizon') = 8760;
mSettings('schedule', 't_jump') = 24;
mSettings('schedule', 't_forecastLength') = 8760;
mSettings('schedule', 't_end') = 25;
mSettings('schedule', 'samples') = 1;
mSettings('schedule', 'forecasts') = 0;
mSettings('schedule', 't_aggregate') = 72;
mSettingsEff('schedule', 'level1') = 1;
mSettingsEff('schedule', 'level2') = 48;
mSettingsEff('schedule', 'level3') = 96;
mSettingsEff('schedule', 'level4') = 168;
mf('schedule', f)$[ord(f)-1 <= mSettings('schedule', 'forecasts')] = yes;
mInterval('schedule', 'intervalLength', 'c000') = 1;
mInterval('schedule', 'intervalEnd', 'c000') = 96;
mInterval('schedule', 'intervalLength', 'c001') = 3;
mInterval('schedule', 'intervalEnd', 'c001') = 168;
mInterval('schedule', 'intervalLength', 'c002') = 6;
mInterval('schedule', 'intervalEnd', 'c002') = 336;
mInterval('schedule', 'intervalLength', 'c003') = 24;
mInterval('schedule', 'intervalEnd', 'c003') = 4392;
mInterval('schedule', 'intervalLength', 'c004') = 168;
mInterval('schedule', 'intervalEnd', 'c004') = 8760;
```

q_obj "Objective function"
q_balance(grid, node, mType, f, t)
q_boundState(grid, node, node
q_maxStateSlack(grid, node, mType, f, t) connected reserves exceeding desired/permitted state limits"
q_minStateSlack(grid, node, mType, f, t) "Slack variables keep track of state variables and connected reserves under desired/permitted state limits"
q_conversionDirectInputOutput(effSelector, unit, f, t, effSelector) "Direct conversion of inputs to outputs (no piece-wise linear part-load efficiencies)"
q_conversionSOS1InputIntermed of input to intermediates restricted by piece-wise linear part-load efficiency represented with SOS1
q_conversionSOS1Constraint(eff e linear intermediate variables"
q_conversionSOS1IntermediateOutput n of intermediates to output"
q_conversionSOS2InputIntermediate(effSelector, unit, f, t) "Intermediate output when using SOS2 variable based part-load piece-wise linearization"
q_conversionSOS2Constraint(effSelector, unit, f, t) "Sum of v_sos2 has to equal v_online"
q_conversionSOS2IntermediateOutput(effSelector, unit, f, t) "Output is forced equal with v_sos2 output"
q_outputRatioFixed(grid, node, grid, node, unit, f, t) "Force fixed ratio between two energy outputs into different energy grids"
q_outputRatioConstrained(grid, node, grid, node, unit, f, t) "Constrained ratio between two grids of energy output; e.g. electricity generation is greater than cV times unit_heat generatio
q_transferLimit(grid, node, node, f, t) "Trans ns are less than the transfer capacity"
q_resDemand(restype, resdirection, node, f, t) ter than demand"
q_maxDownward(grid, node, unit, f, t) "Downw lant minimum load constraints or maximum elec. consumption"
q_maxUpward(grid, node, unit, f, t) "Upward con le capacity or consumed power"
q_startup(unit, f, t) "Capacity started up is greate in the previous time step"
q_bindOnline(unit, mType, f, t) "Couple online va ble time periods"

Balance with dynamic state variable and state equations

Conversion equations
- From simple and fast
- To accurate and slow

Transfer constraints

Power specific constraints

Startup constraints

Constraints related to temporal structure of the model

Work in progress…

- Each new energy form may require changes to the core equations or new constraints
- E.g. balance equation got state variables in order to model heat transfer in buildings
- DC power flow will require separate constraints
- Still, the attempt is to minimize equations by representing generic principles instead of particular processes

Sceleton: Python interface for any model (including Backbone)

- Keeps input data in order
  - Stores things only once, project-based
  - Can handle multiple input data sources: Excel, databases, binary files, etc.
- Calls data conversion tools and models
  - Possible to chain input data, tools and models
- Shows what scenarios have been run
- Can display and process results
- Fast and reliable creation of sensitivities with a spreadsheet program
  - Possible to use ’recipes’ where one can do Cartesian products or other set theory based scenario combinations
  - Sceleton will distribute the scenario tasks to available computation units (in future)