



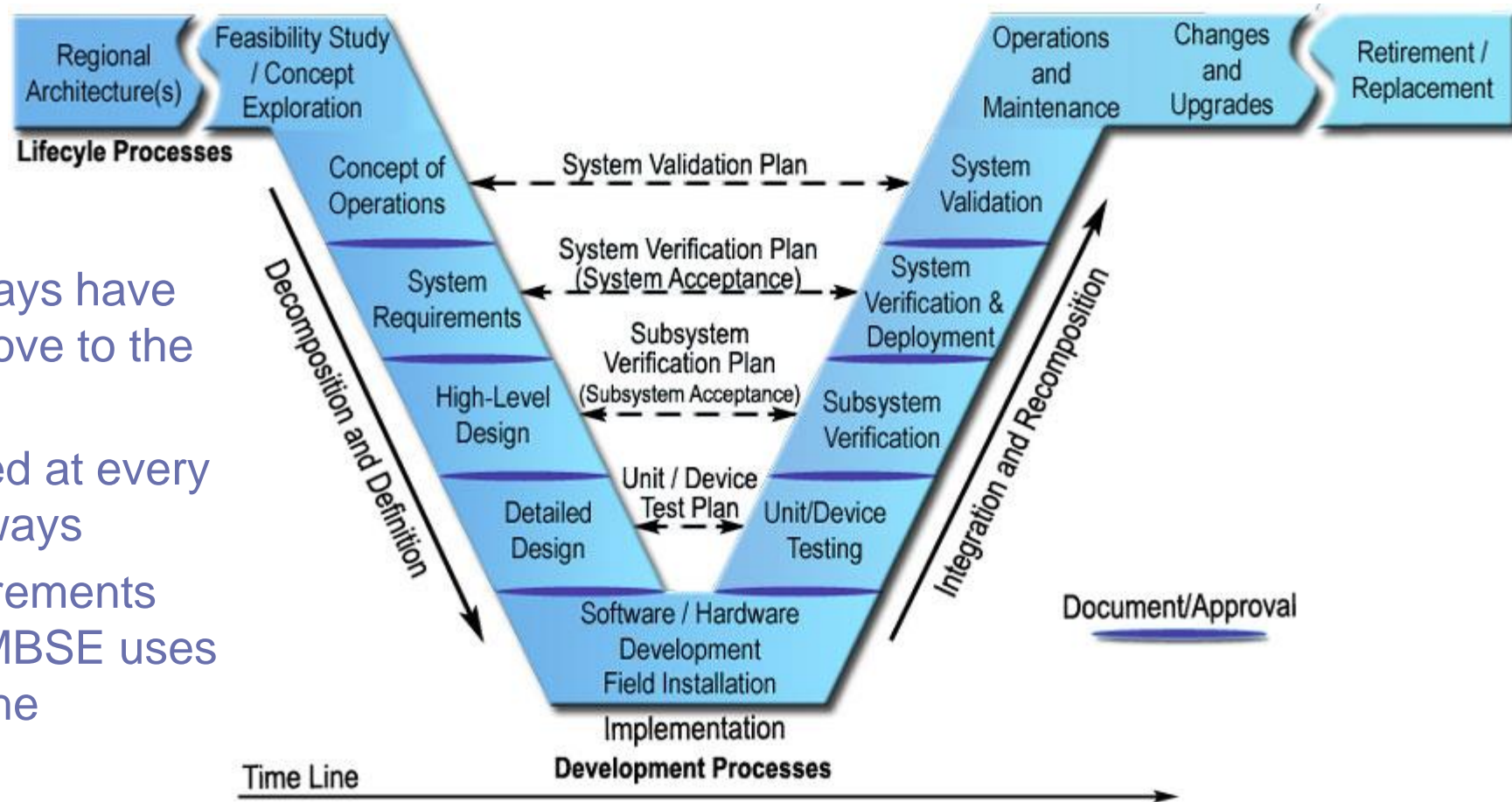
Mapping the CAE Tools for the development of EVs

Systems Engineering

Systems Engineering

- Systems engineering is an interdisciplinary field of engineering and engineering management that focuses on how to design and manage complex systems over their life cycles
- In the CAE context, this is usually described with one of the following terms:
 - Model Based Systems Engineering (MBSE)
 - Model Based Design (MBD)
 - Model Driven Design (MDD)

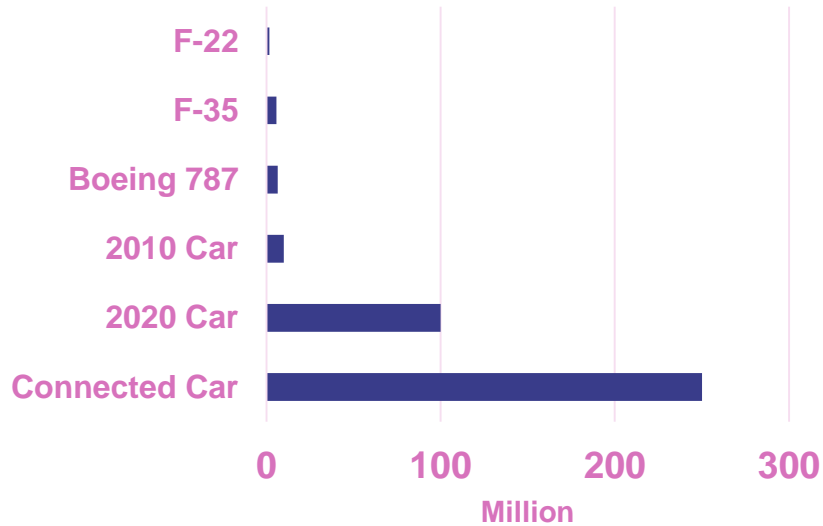
Systems Engineering – V diagram



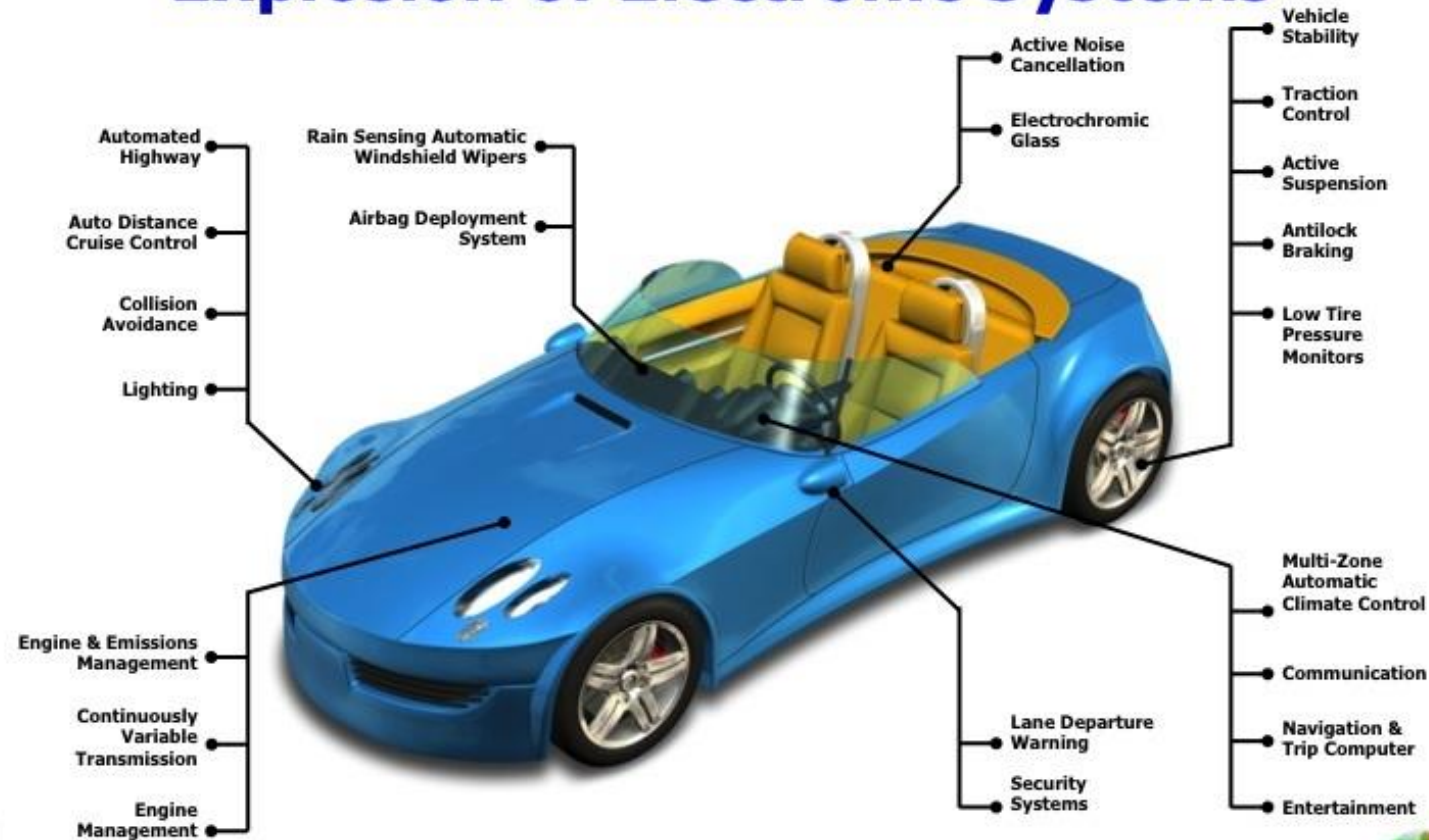
- Milestones/Gateways have to be passed to move to the next stage
- CAE can be applied at every stage in different ways
- Traditionally requirements were purely text, MBSE uses models to define the requirements

Evolution of Automotive Systems Engineering

Software lines of code



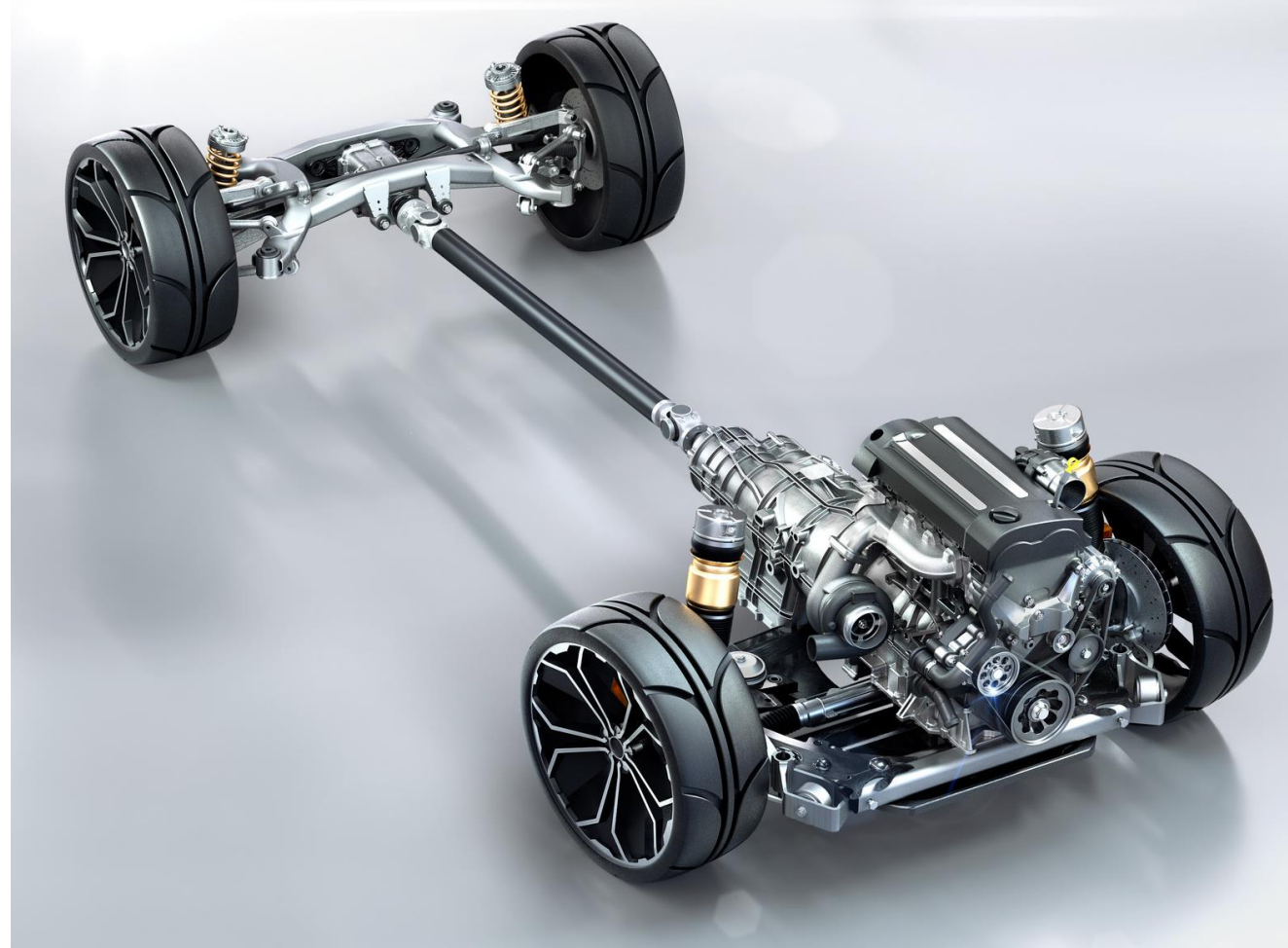
Explosion of Electronic Systems



>100 ECU's in current car

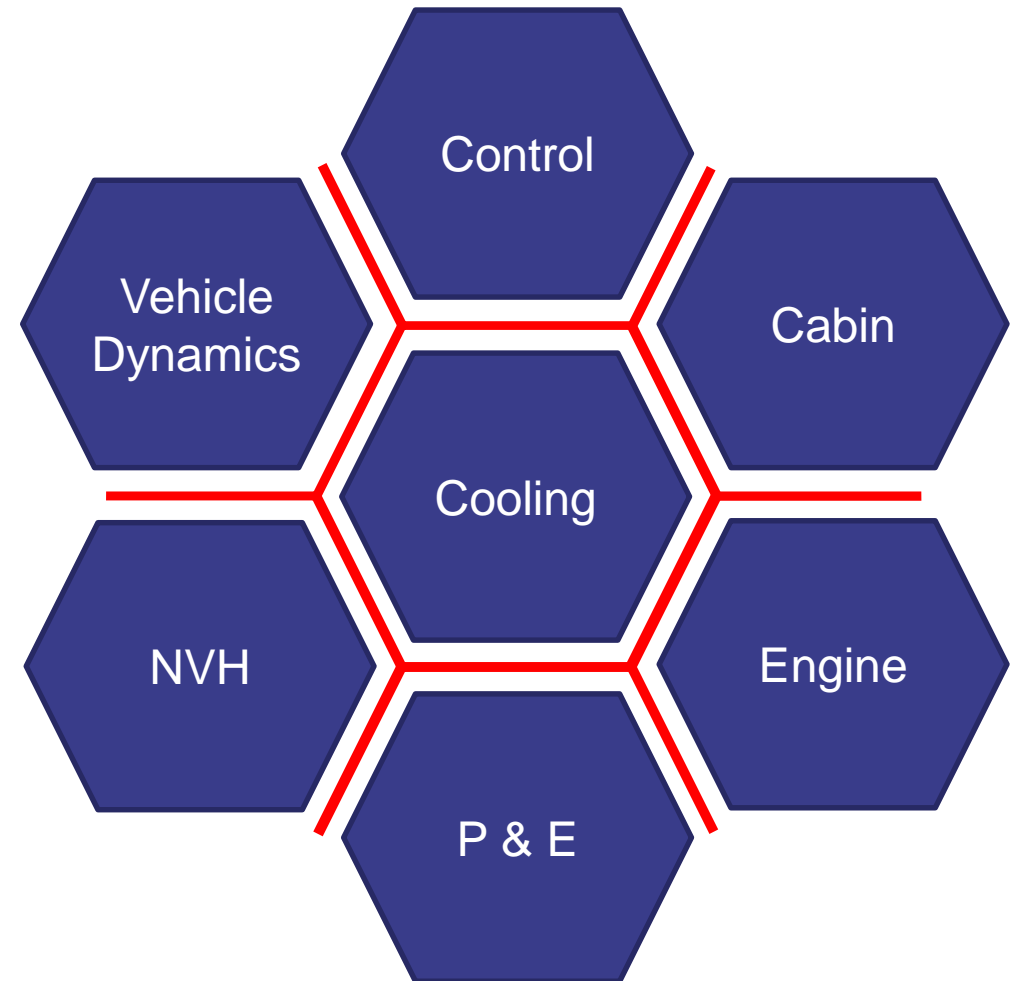
Conventional Vehicle Architecture

- Conventional ICE vehicle is largely a mechanical system
 - Engine, driving gearbox, through driveline to the wheels
- Cooling for the engine
- Heating/cooling for the occupants
- Electrical systems
 - lights, infotainment, control systems
- Integration of systems possible late in the development process when prototypes are available



CAE Tools

- Typical OEM uses a lot of different CAE tools to simulate different aspects of the system
- These are usually disconnected from one another and use proprietary data formats
 - Within each of these areas, there could be a number of different tools used still with this problem
- Typically an OEM's organisation also reflects this
 - Different departments operate in their own silo, disconnected from one another



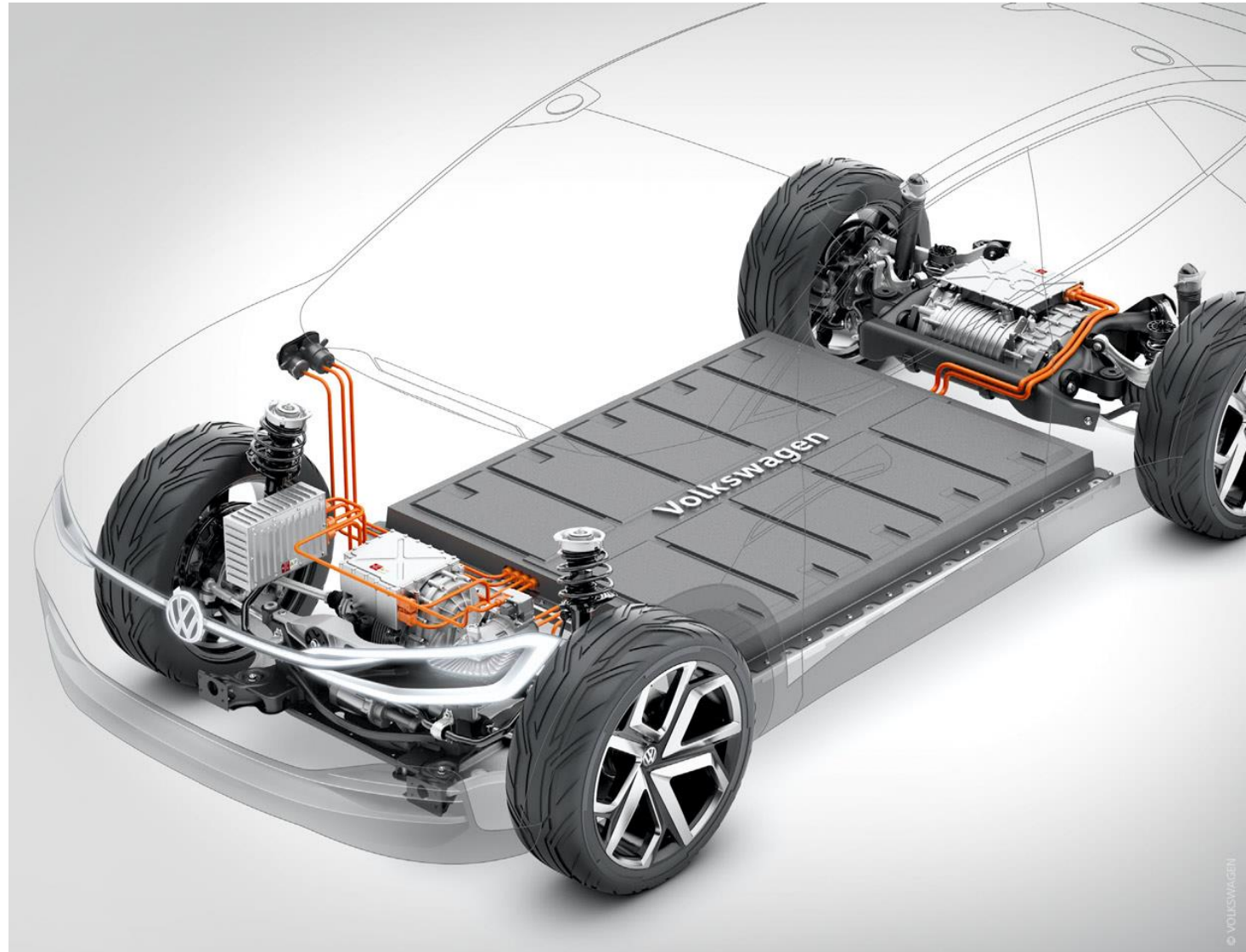
Automotive Trends - ACES

- Autonomous
 - Billions being invested in the development of autonomous vehicles
 - Vast array of sensors to “see” the world around the car coupled to massive compute requirements
- Connected
 - Providing more services to the vehicle user as well as sending data back to the manufacturer
- Electrified
 - Main issues holding them back is range anxiety and the cost/weight of the battery
 - Zero emissions at the point of use will help ease pollution in cities
- Shared
 - If the vehicle is autonomous and can be summoned when needed, do we really still need/want to own the vehicle?

Taken together this means massive increases in the complexity of the systems and uncertainty about which technologies are the right ones to back

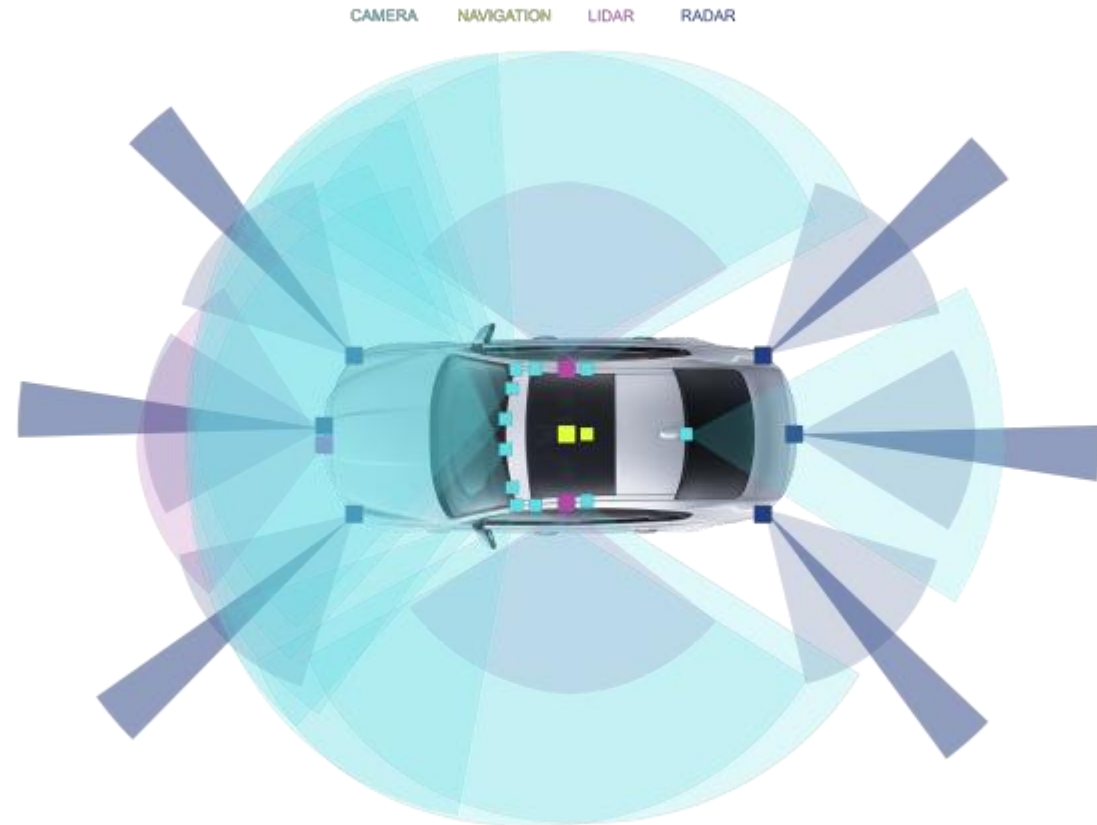
ICE to EV powertrain

- The EV of tomorrow:
 - Multiple electrical motors
 - Power electronics
 - Battery
 - Mechanically much simpler
 - Heating/cooling for the motors, electronics, battery, occupants
 - Electrical systems
 - lights, infotainment, control systems
- Very tightly coupled systems
- Multiple physical domains
 - Electrical, Thermal, Fluids, Mechanical, Electronics



EV + Autonomous

- A discussion about EV's is almost always coupled with autonomy
 - Might still be classed as ADAS
 - Level 1-3 autonomy
- Increases system complexity even further
 - Large number of sensors of different types to perceive the environment
 - Huge compute requirement to process all the data
 - AI to interpret what the sensors “see”, plan what to do next and then execute a manoeuvre
 - How all this works seamlessly together requires extensive testing – virtual testing

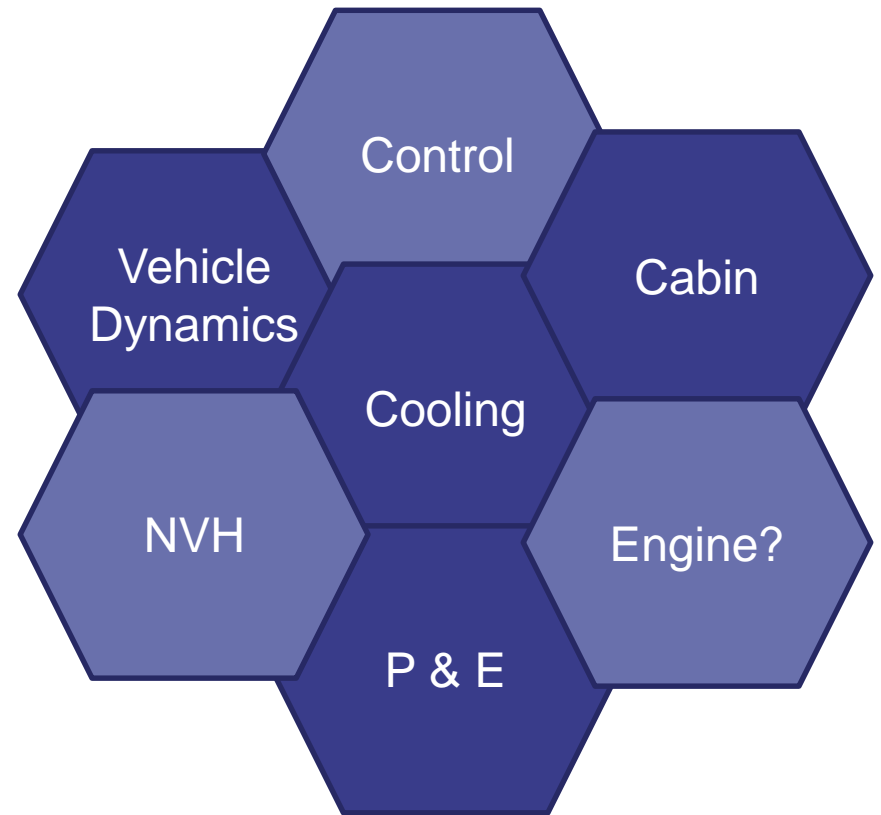


CAE Trends

- All major tool vendors are working on ways to connect their tools and ease the movement of models/data between them
 - Dassault Systemes 3DEXPERIENCE platform
 - Siemens Teamcenter
 - Mathworks have added physical modelling toolboxes to Simulink
- For systems engineering, one significant trend is the adoption of Modelica
 - Dassault Systemes acquired Dymola in 2006
 - Siemens acquired Amesim in 2012 and added improved Modelica support in 2018
 - Altair added Modelica support with Altair Activate in 2018
 - Ansys added Modelica support in 2014
 - ESI acquired CyDesign in 2013 and Simulation-X in 2016
 - Ricardo, Maplesoft, Wolfram all offer tools that support Modelica
 - Open source projects include OpenModelica, Jmodelica.org

CAE Tools – Digital Twin

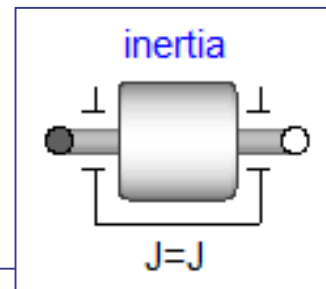
- EV's have tighter coupling between different systems
- Need to break down the barriers between the different disciplines
- We need models and data to move easily between different tools
 - No-one tool can do it all
 - Modelica and FMI are two technologies that enable this



Modelica Modelling Language

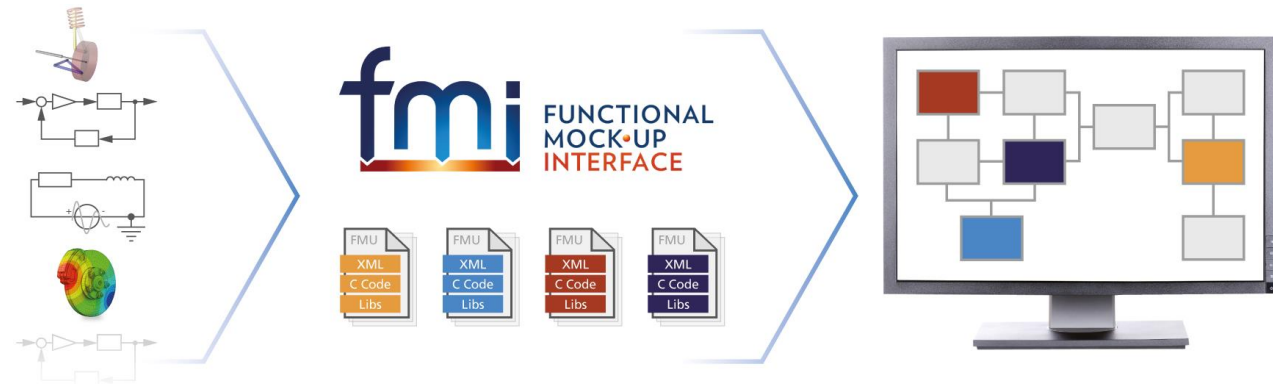
- Modelica is a object-oriented, equation based language to conveniently model complex physical systems
 - e.g. mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents
- Open source project managed by the Modelica Association
 - Collaborators from universities, research institutes, tool vendors and end users
- Modelica Standard Library provides basic models across many domains
 - 1D thermofluids, MultiBody mechanics, electrical, control, magnetics, etc.
- Development started in 1996, currently version 3.4

```
model Inertia
  import SI = Modelica.SIunits;
  Rotational.Interfaces.Flange_a flange_a "Left flange of shaft" a;
  Rotational.Interfaces.Flange_b flange_b "Right flange of shaft" a;
  parameter SI.Inertia J(min=0, start=1) "Moment of Inertia" a;
  parameter StateSelect stateSelect=StateSelect.default "Priority
    to use phi and w as states" a;
  SI.Angle phi(stateSelect=stateSelect) "Absolute rotation angle
    of component" a;
  SI.AngularVelocity w(stateSelect=stateSelect) "Absolute angular
    velocity of component (= der(phi))" a;
  SI.AngularAcceleration a "Absolute angular acceleration of
    component (= der(w))" a;
equation
  phi = flange_a.phi;
  phi = flange_b.phi;
  w = der(phi);
  a = der(w);
  J*a = flange_a.tau + flange_b.tau;
a;
end Inertia;
```

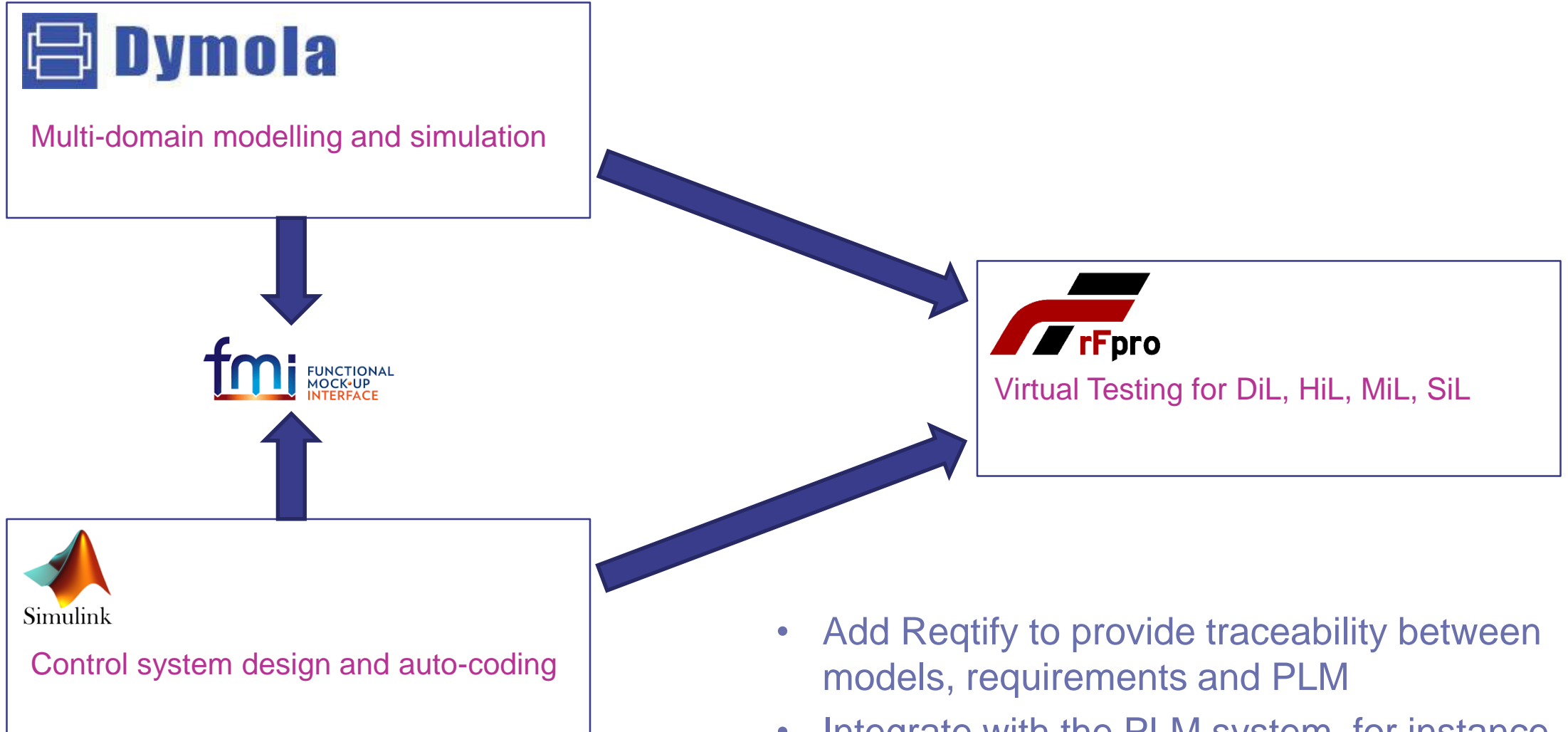


FMI Standard

- Functional Mock-up Interface (FMI) is a tool independent standard to support both model exchange and co-simulation of dynamic models
 - The first version, FMI 1.0, was published in 2010, followed by FMI 2.0 in July 2014
 - The FMI development was initiated by Daimler AG with the goal to improve the exchange of simulation models between suppliers and OEMs
- FMI is supported by over 100 tools and is used by automotive and non-automotive organizations throughout Europe, Asia and North America
- FMI is an open standard developed by the Modelica Association
- A model compatible with the FMI standard is referred to as an FMU
 - Functional Mock-up Unit



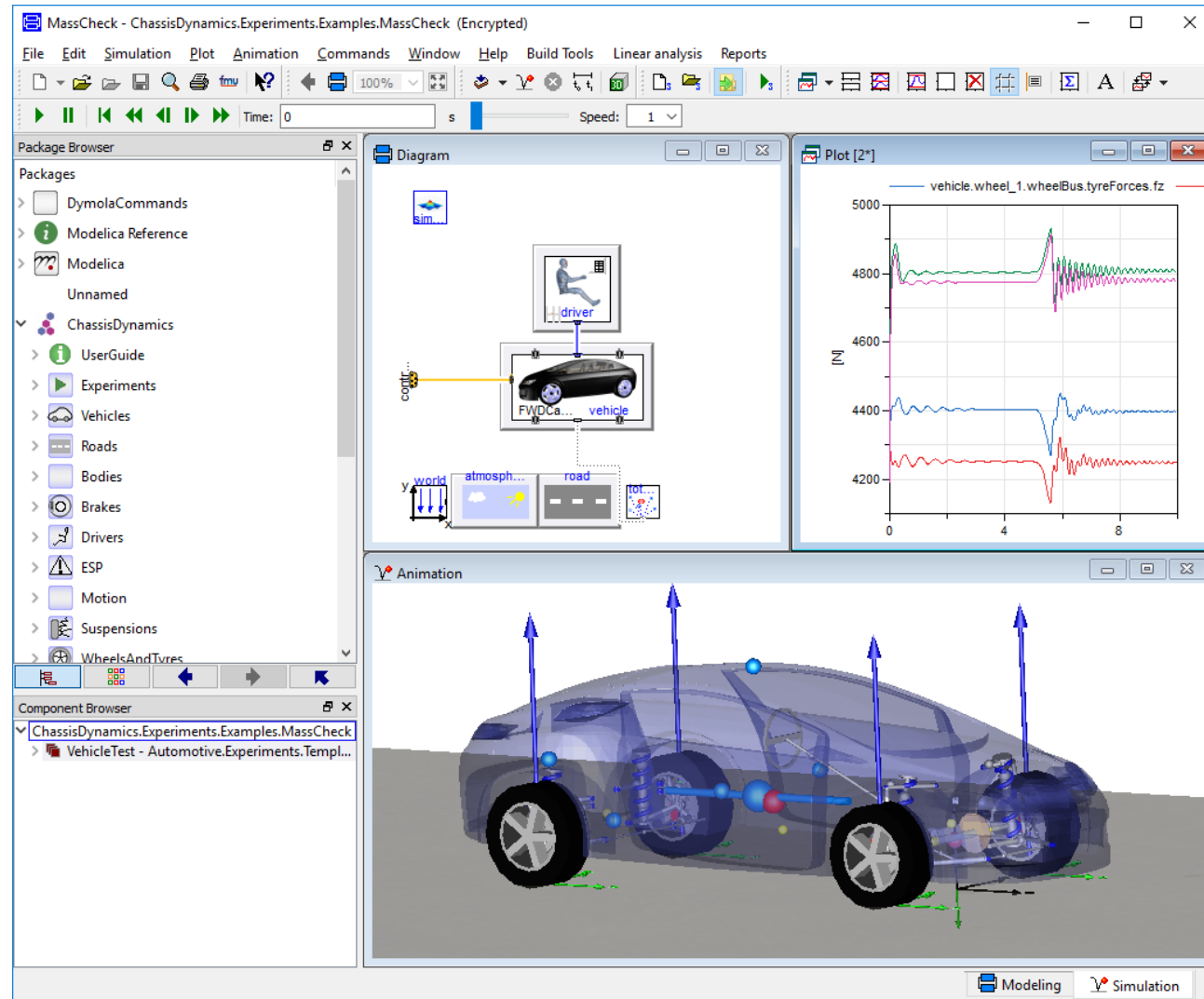
Example Toolchain



- Add Reqtify to provide traceability between models, requirements and PLM
- Integrate with the PLM system, for instance Dymola is available on 3DEXPERIENCE

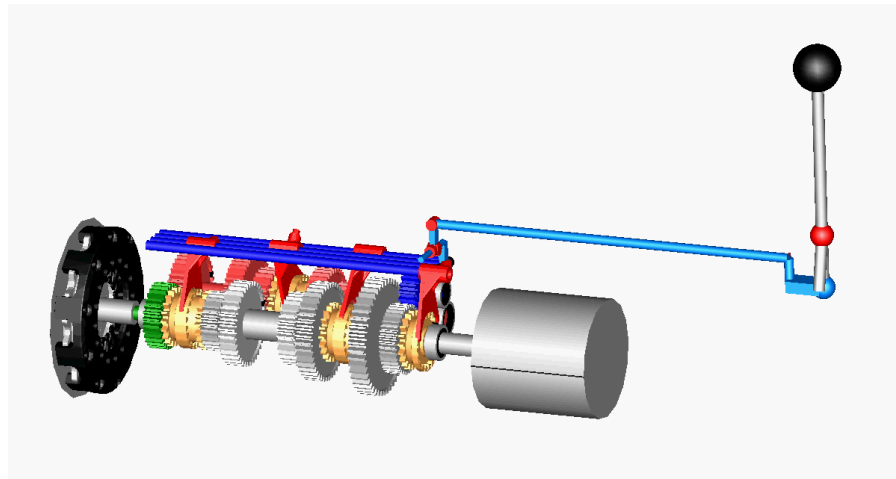
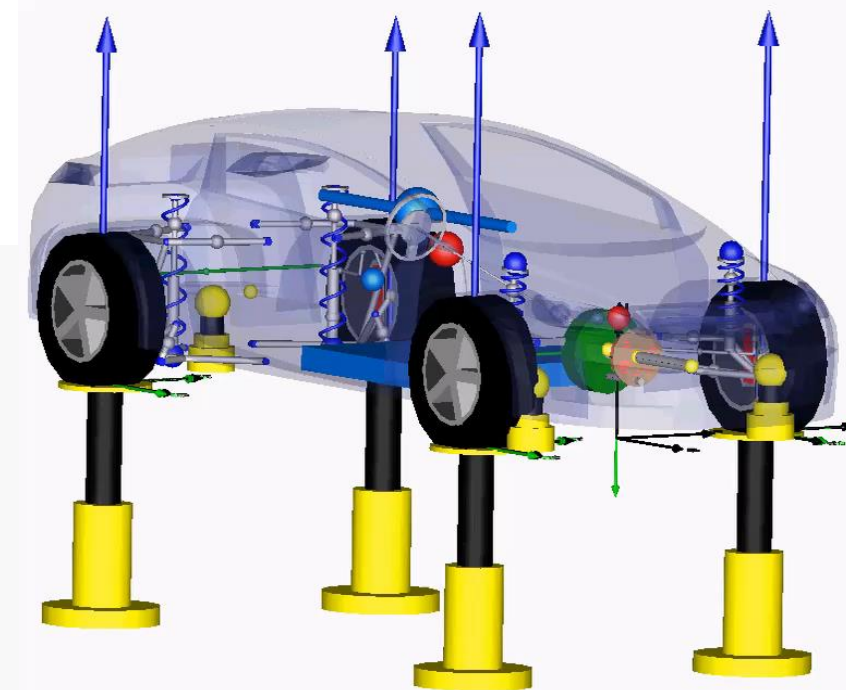
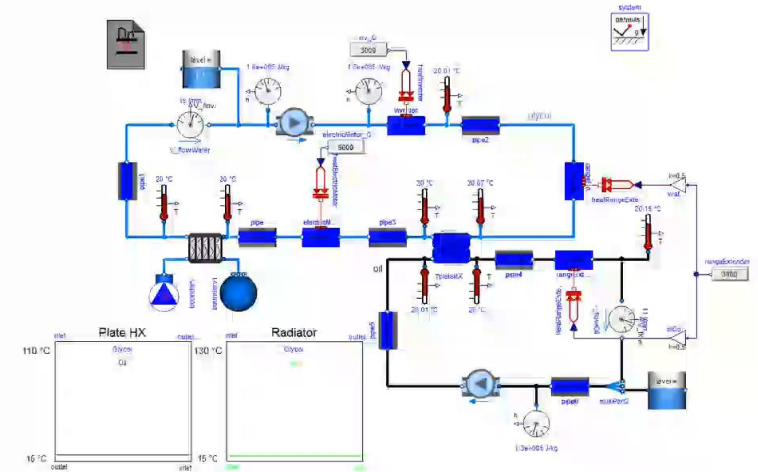
Dymola

- A component orientated multi-domain tool for modelling and simulation
- Multi-domain modelling of complex dynamic systems
 - Mechanical, Electrical, Hydraulic, Pneumatic, ThermoFluids, Thermal, Control
 - Uses Modelica language
- Enables a Model Based Design and Development process
- Developed by Dassault Systemes
 - Available standalone or integrated in 3DEXPERIENCE



VeSyMA

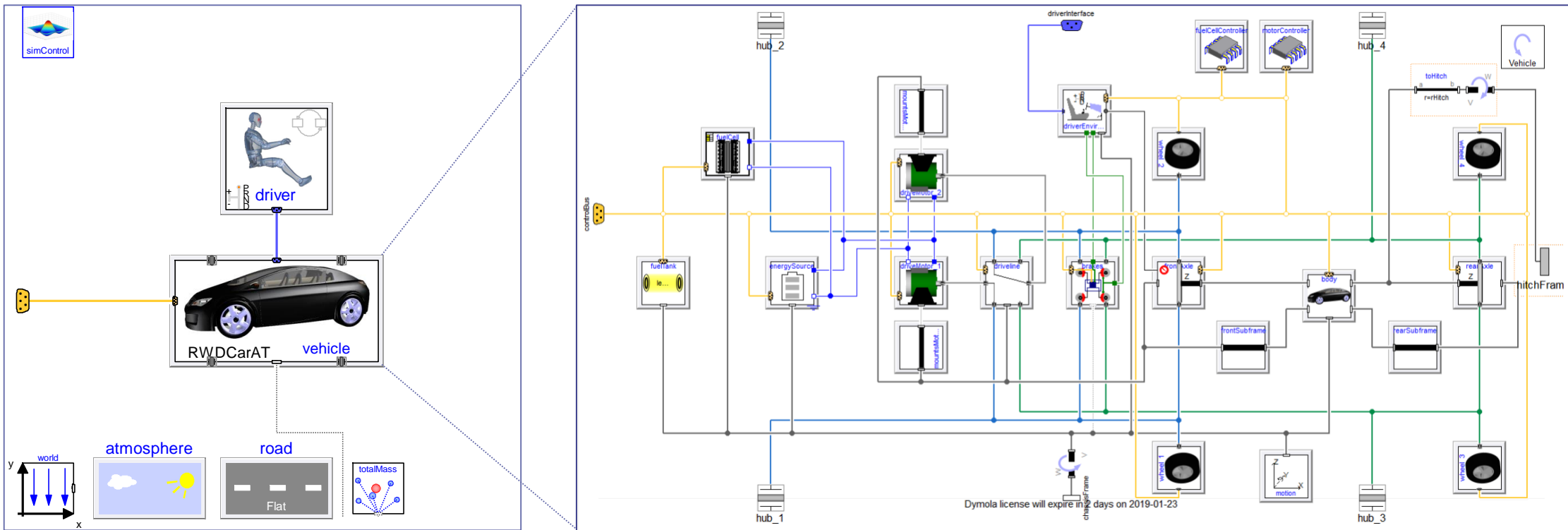
- Vehicle Systems Modelling and Analysis
 - Suite of Modelica Libraries
- Compatible with other Modelica libraries for battery, motors, human comfort and many other applications
- Developed by Claytex, distributed by Dassault Systemes with Dymola



VeSyMA

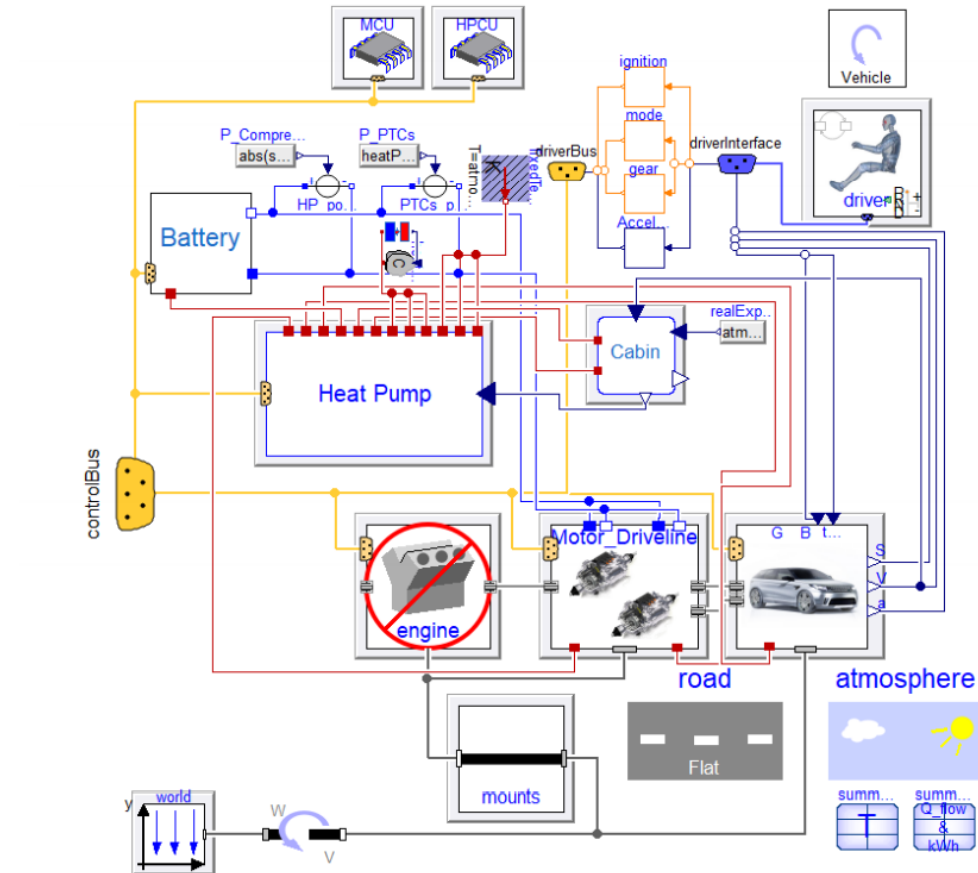
- **Driver-in-the-Loop**
- **Engines**
- **Fluids**
- **Kinematics**
- **Motorsports**
- **Powertrain**
- **Suspensions**
- **Terrain Server**

- Performance, Fuel Economy and Energy usage prediction for ground vehicles
- Main purpose is to provide a platform for the design and analysis of vehicle systems and components



JLR use case

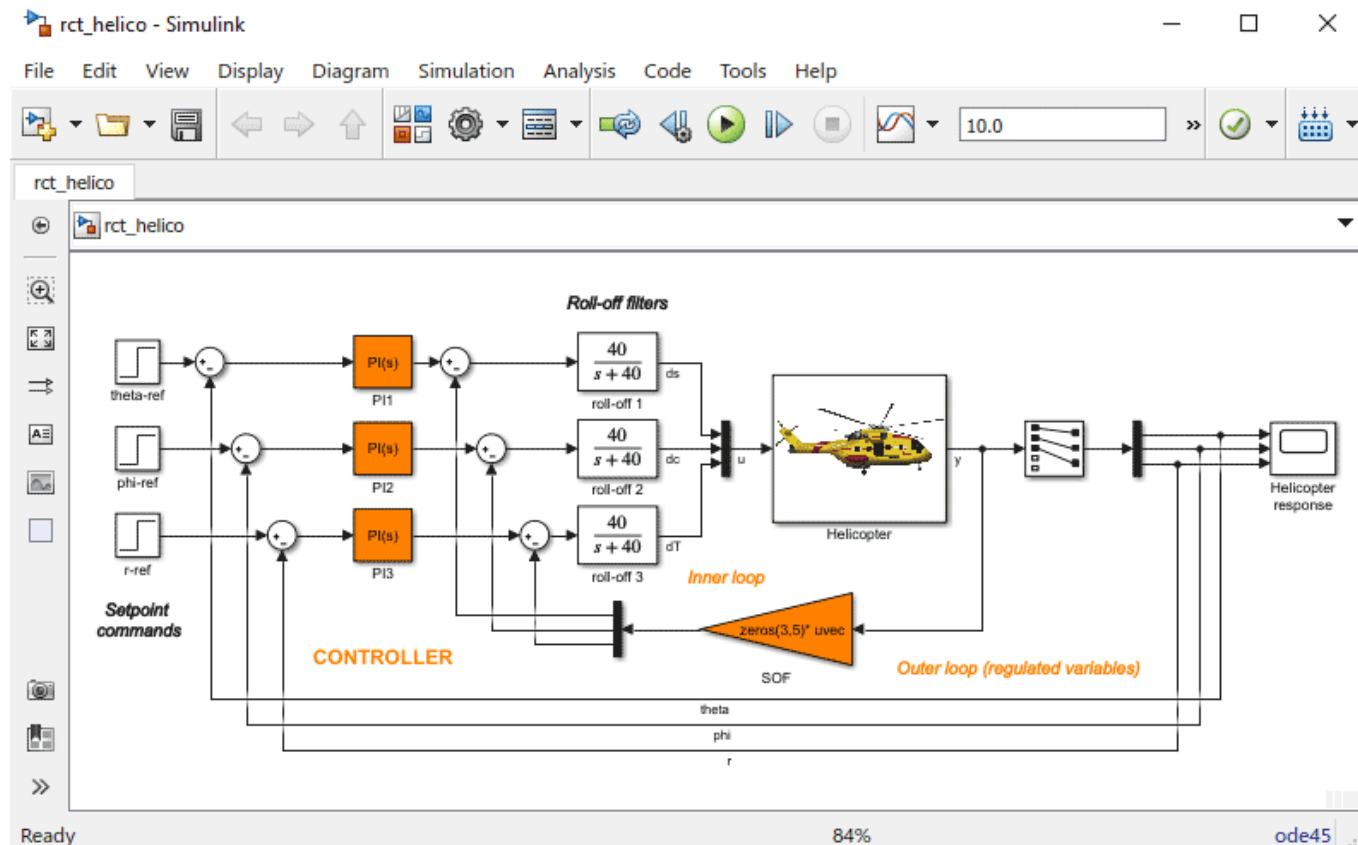
- Conference paper from JLR, WMG and Claytex
 - Complex Heat Pump Operational Mode Identification and Comparison for Use in Electric Vehicles
 - <https://doi.org/10.3390/en11082000>
- Research study looking at the use of 5 heat pumps in an electric vehicle
- The results presented suggest:
 - the motor, a thermal storage device, and cabin exhaust extraction should be used >80% of the time
 - while battery heating and transmission heat extraction should be used subject to conditions on the ambient temperature and drive cycle
- Only possible to evaluate using models that couple all the different physics together



Simulink



- Developed by The Mathworks, builds on Matlab
- Control system design using block diagram modelling approach
- Design and simulate your system in Simulink before moving to hardware
- Explore and implement designs that you wouldn't otherwise consider – without having to write C, C++, or HDL code
- Automatically generate production-quality C and HDL code that behaves the same way as the model you created in Simulink



Dymola + Simulink with FMI

- Dymola and Simulink both support import and export of FMI compliant models
 - Control engineers can work in Simulink with physics models
 - Hardware engineers can work in Dymola with controller model
- Dymola support for FMI is a standard feature
- Simulink support for FMI comes from 3rd party tools
 - Dassault Systemes provide tools to import and export FMU's from Simulink
 - Claytex, and many others, provide tools to import FMU's into Simulink
- Co-simulation FMU means the FMU includes a solver provide by the tool used to export the FMU
- Model exchange FMU means the FMU uses a solver from the host system



- Solution for the testing and development of ADAS and autonomous systems
- Origins as a Driver-in-the-Loop simulation environment

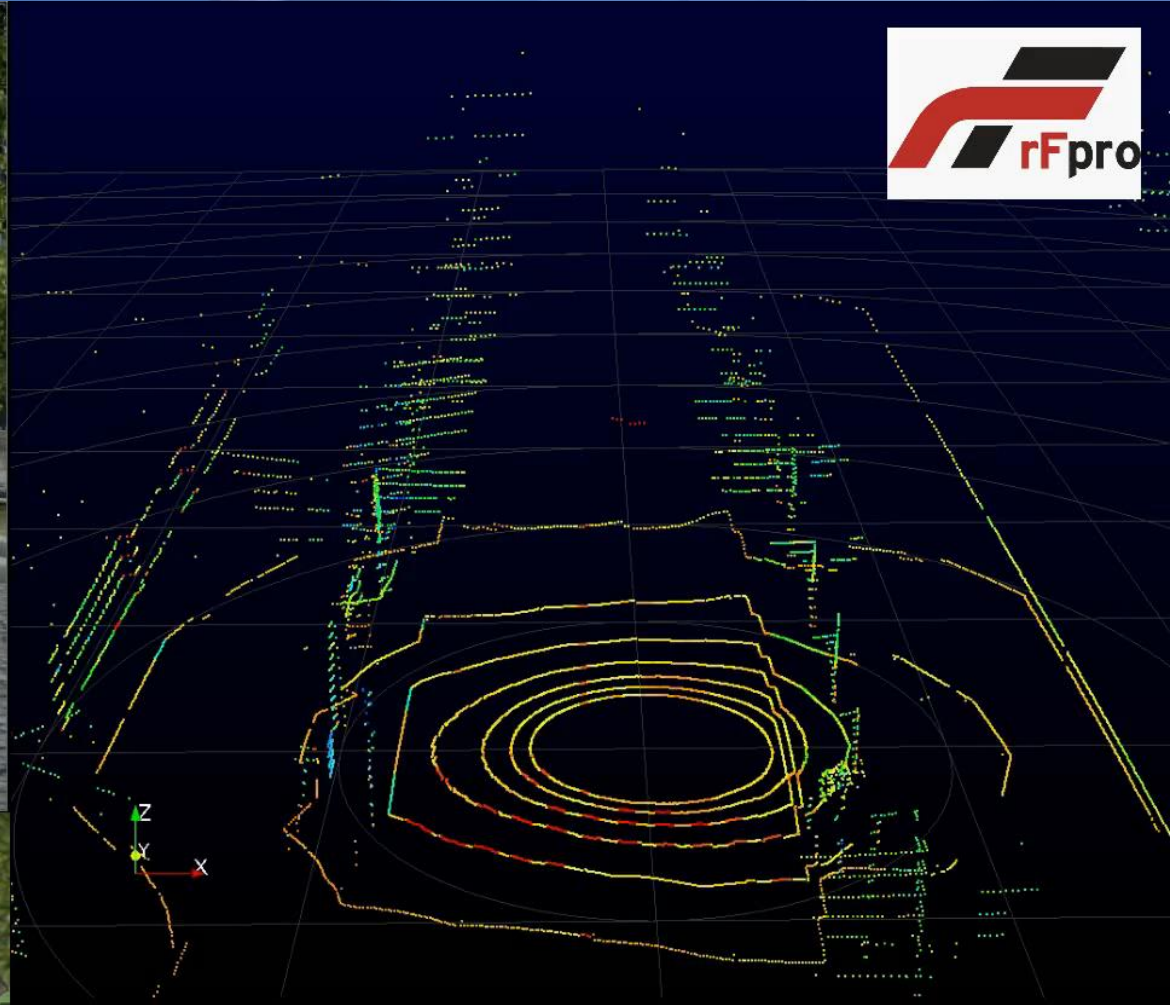
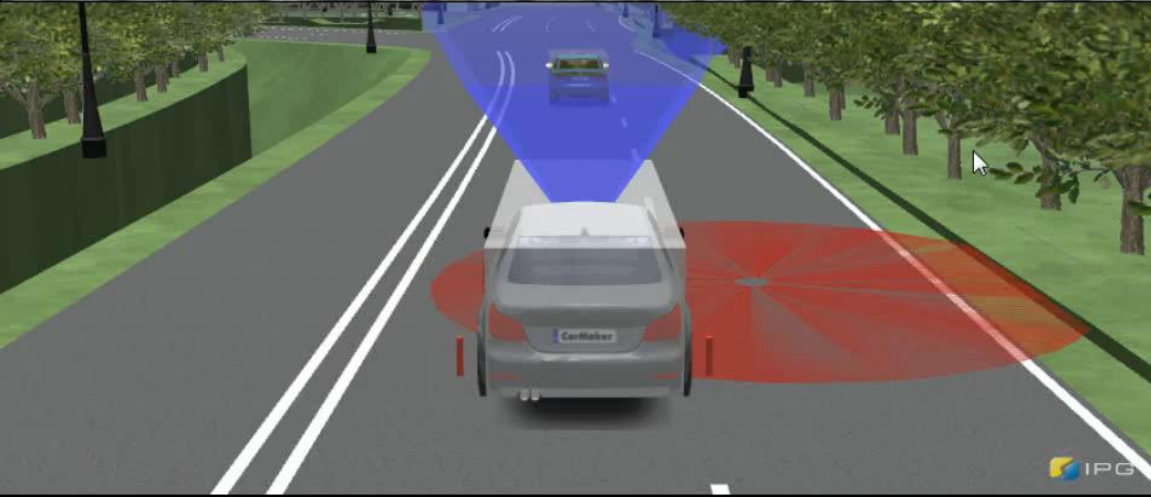


rFpro – Key features

- 100km's of real world locations are available
 - Proving grounds: Idiada, Millbrook, Mcity, SMLL
 - Race tracks: Nordschleife, full Formula 1 and NASCAR calendars, Formula E calendar and many other tracks worldwide
 - Public roads including locations in:
 - France, Germany, USA, Canada, Italy, China
- Integrates with many different vehicle modelling tools
- Include traffic, pedestrians, cyclists, etc.
- Supports real-time simulation for DiL, HiL, SiL, MiL
- Plugin our vehicle physics from Dymola
- Plugin, or connect, our control systems from Simulink



Sensor modelling for Autonomous Vehicles



rFpro ParisStreets digital road model
rFpro hosted LiDAR sensor model
Real-time data stream visualised in native software
Traffic objects controlled by IPG CarMaker

Enabling further use of CAE

- Cloud simulation platforms supporting FMI standard enable models to be deployed to non-experts
 - Also possible to embed FMU's in Excel and other office tools
- Test automation tools allow automated workflows to be built and wrap around these models
- Requirements traceability tools can track how models are linked to requirements in text documents or databases
- Automatic generation of mechanical models from CAD data



Discuss