A Case Study of Application Development and Production Code Generation for a Telematics ECU with Full Unified Diagnostics Services

turnkey software solutions

Plan

- A little about Embed and our Ethos
- Description of the telematics module Embed worked in partnership with Auto-txt Ltd to deliver.
- The process by which the software was developed.
  - Automotive SPICE® level 3 (ISO/IEC 15504)
- The software architecture
  - Device Drivers:- CAN, GPS, GSM, Bluetooth
  - Application as libraries enabling extensive unit testing
- Focus on Unified Diagnostic Services (ISO 14229) within the telematics module, over CAN and GSM
- How diagnostics are usually developed compared to how they can be better developed with the application within Simulink.
  - Advantages of the Embed Unified Diagnostic Services (ISO 14229) Blockset

Embed Ethos

New Technologies

Image Processing
HMI Rapid-Prototyping
Advanced Driver Assistance Systems (ADAS)

Faster Results

Model, Simulate, Execute, Debug, Test...
Then code...

Off-the-shelf ECUs

Embed partner with New Eagle to offer MotoHawk ECUs

New Eagle™

Higher Quality

Automotive SPICE® Level 3 Process.
(ISO/IEC 15504)

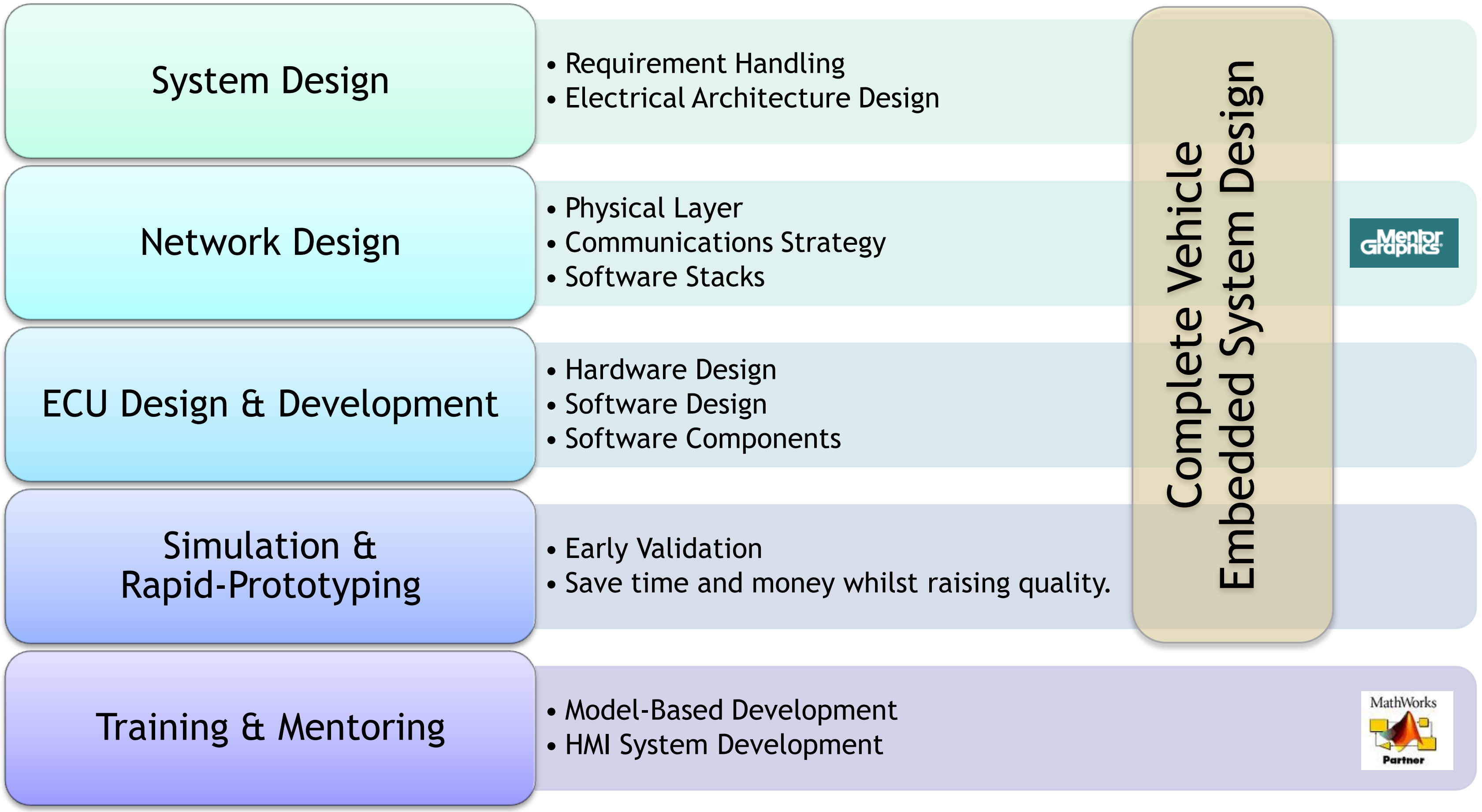
Model Based Development

Off-the-shelf software components

embed
UDS, CAN, CAN-TP, Simulink Blocksets for production ECUs.

embed
turnkey software solutions

Embed Offers



System Design	• Requirement Handling • Electrical Architecture Design
Network Design	• Physical Layer • Communications Strategy • Software Stacks
ECU Design & Development	• Hardware Design • Software Design • Software Components
Simulation & Rapid-Prototyping	• Early Validation • Save time and money whilst raising quality.
Training & Mentoring	• Model-Based Development • HMI System Development

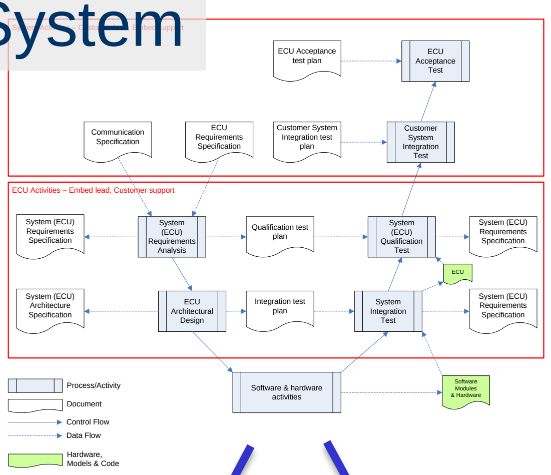
turnkey software solutions

Auto-txt Advanced Telematics Module

- Advanced telematics module that can offer remote connectivity and communication to any vehicle (12V & 24V)
  - Stolen Vehicle Tracking, Fleet Management, Remote Diagnostics, Remote Logging & Control
- GSM, GPS, Bluetooth, Proprietary RF, WiFi, CAN
- Fully expandable via daughter boards
  - Soon adding LIN & RS232 support.
- Full automotive grade product
  - Line-fit for Aston Martin, dealer fit for Jaguar and Land Rover
- Bootloader for CAN or GSM reprogramming
  - Complete over the air reprogramming

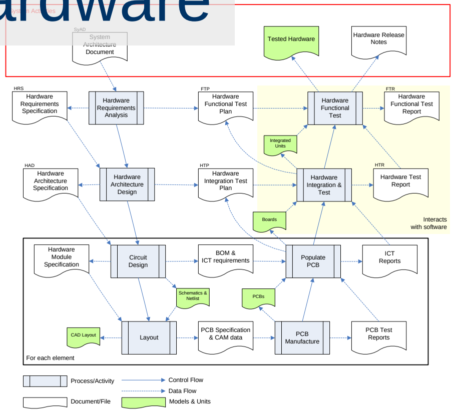
Development Process

System

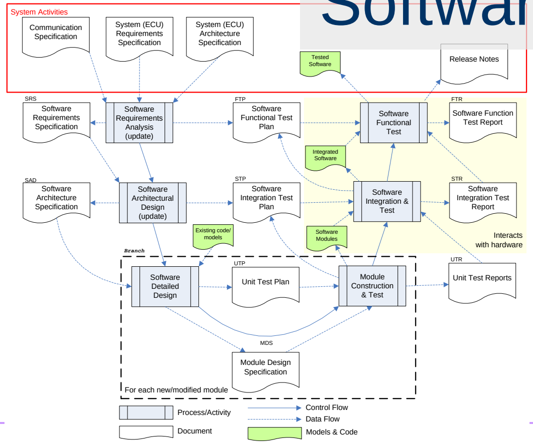

ECU Acceptance test plan
ECU Acceptance Test
Communication Specification
ECU Requirements Specification
Customer System Integration test plan
Customer System Integration Test
ECU Activities – Embed lead, Customer support
System (ECU) Requirements Specification
System (ECU) Requirements Analysis
Qualification test plan
System (ECU) Qualification Test
System (ECU) Requirements Specification
ECU
System (ECU) Architecture Specification
ECU Architectural Design
Integration test plan
System Integration Test
System (ECU) Requirements Specification
Software & hardware activities
Software Modules & Hardware
Process/Activity
Document
Control Flow
Data Flow
Hardware, Models & Code

Automotive SPICE® Level 3 Process. (ISO/IEC 15504)

Hardware



Software


System Activities
Communication Specification
System (ECU) Requirements Specification
System (ECU) Architecture Specification
Tested Software
Release Notes
SRS
Software Requirements Specification
Software Requirements Analysis (update)
FTP
Software Functional Test Plan
Software Functional Test
FTR
Software Function Test Report
Integrated Software
SAD
Software Architecture Specification
Software Architectural Design (update)
STP
Software Integration Test Plan
Software Integration & Test
STR
Software Integration Test Report
Existing code/models
Software Modules
Interacts with hardware
Branch
UTP
UTR
Software Detailed Design
Unit Test Plan
Module Construction & Test
Unit Test Reports
MDS
Module Design Specification
For each new/modified module
Process/Activity
Control Flow
Data Flow
Document
Models & Code

embed
turnkey software solutions

autotxt™

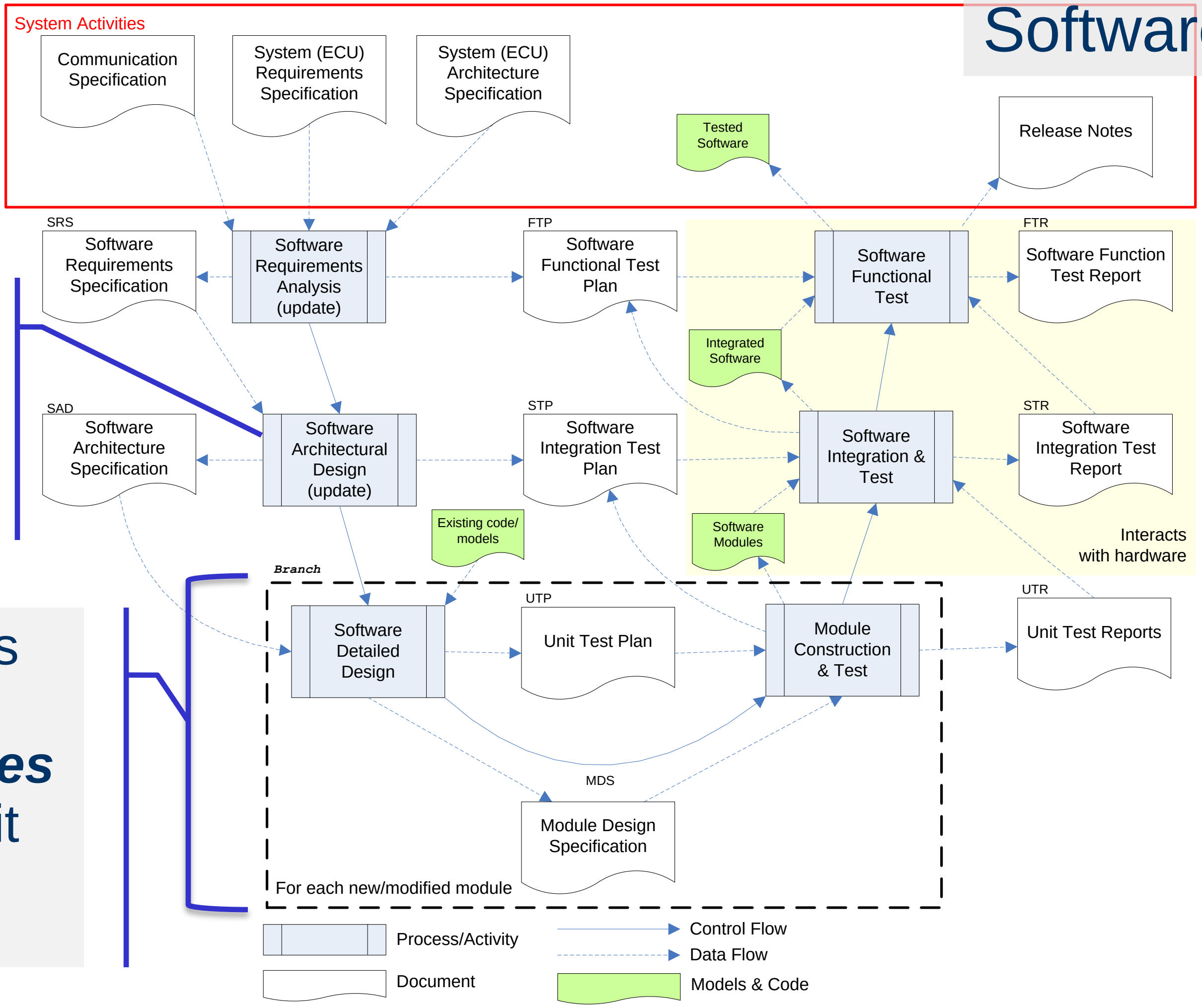
The product which is the subject of this presentation is the proprietary technology and intellectual property of Auto-txt Limited. Details are reproduced with the permission of Auto-txt Limited. "Autotxt" is the trade mark of Auto-txt Limited.

Development Process

Software

Application architecture defined in Simulink model

Software modules designed as ***Simulink Libraries*** complete with unit tests and documentation


System Activities
Communication Specification
System (ECU) Requirements Specification
System (ECU) Architecture Specification
Tested Software
Release Notes
SRS
Software Requirements Specification
Software Requirements Analysis (update)
FTP
Software Functional Test Plan
Software Functional Test
FTR
Software Function Test Report
Integrated Software
SAD
Software Architecture Specification
Software Architectural Design (update)
STP
Software Integration Test Plan
Software Integration & Test
STR
Software Integration Test Report
Existing code/ models
Software Modules
Interacts with hardware
Branch
Software Detailed Design
UTP
Unit Test Plan
Module Construction & Test
UTR
Unit Test Reports
MDS
Module Design Specification
For each new/modified module
Process/Activity
Document
Control Flow
Data Flow
Models & Code

embed turnkey software solutions

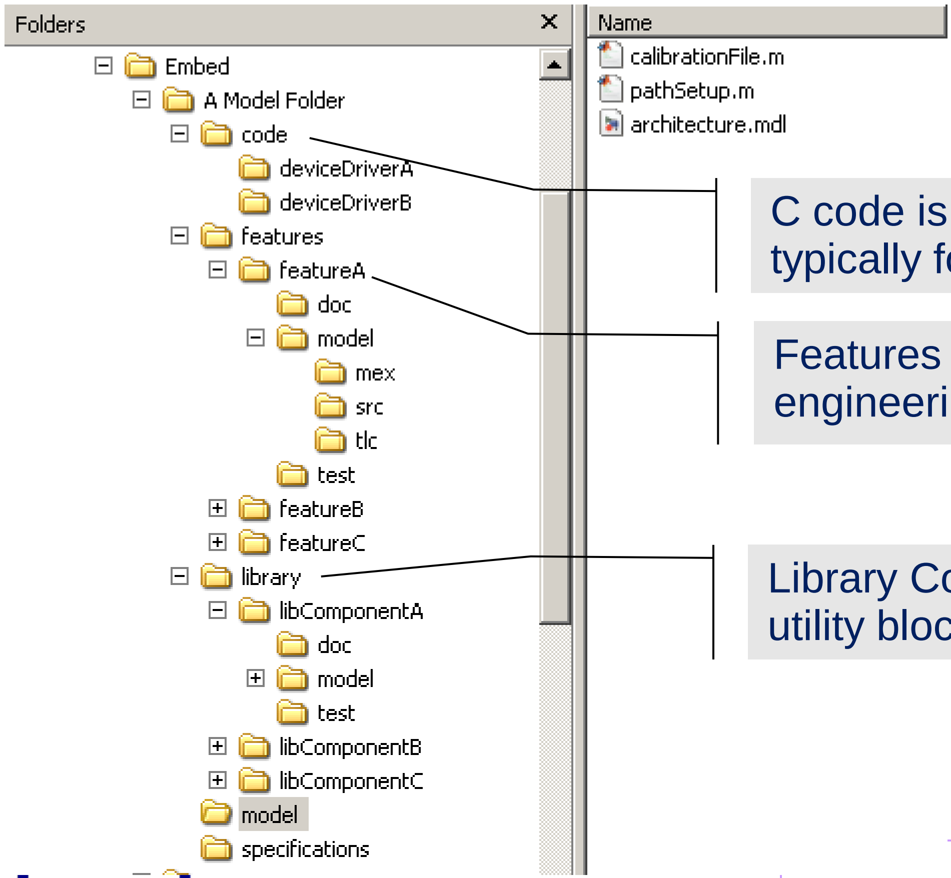
AUTOTXT™

Developing using Simulink Libraries

- Simulink Libraries enable many advantages.
  - Version Control
    - Library bocks live as separate files
  - Unit Test
    - Test harness models
  - Multiple Developers working on the same project
    - Clearly defined bus interfaces
  - Reuse the blocks across may projects
    - Tested modules ready to deploy

Developing using Simulink Libraries

- Embed Standard Model Organisation



C code is required in all embedded projects, typically for device drivers

Features are libraries for process, concurrent engineering and version control

Library Components are true libraries and are utility blocks that are reused many times

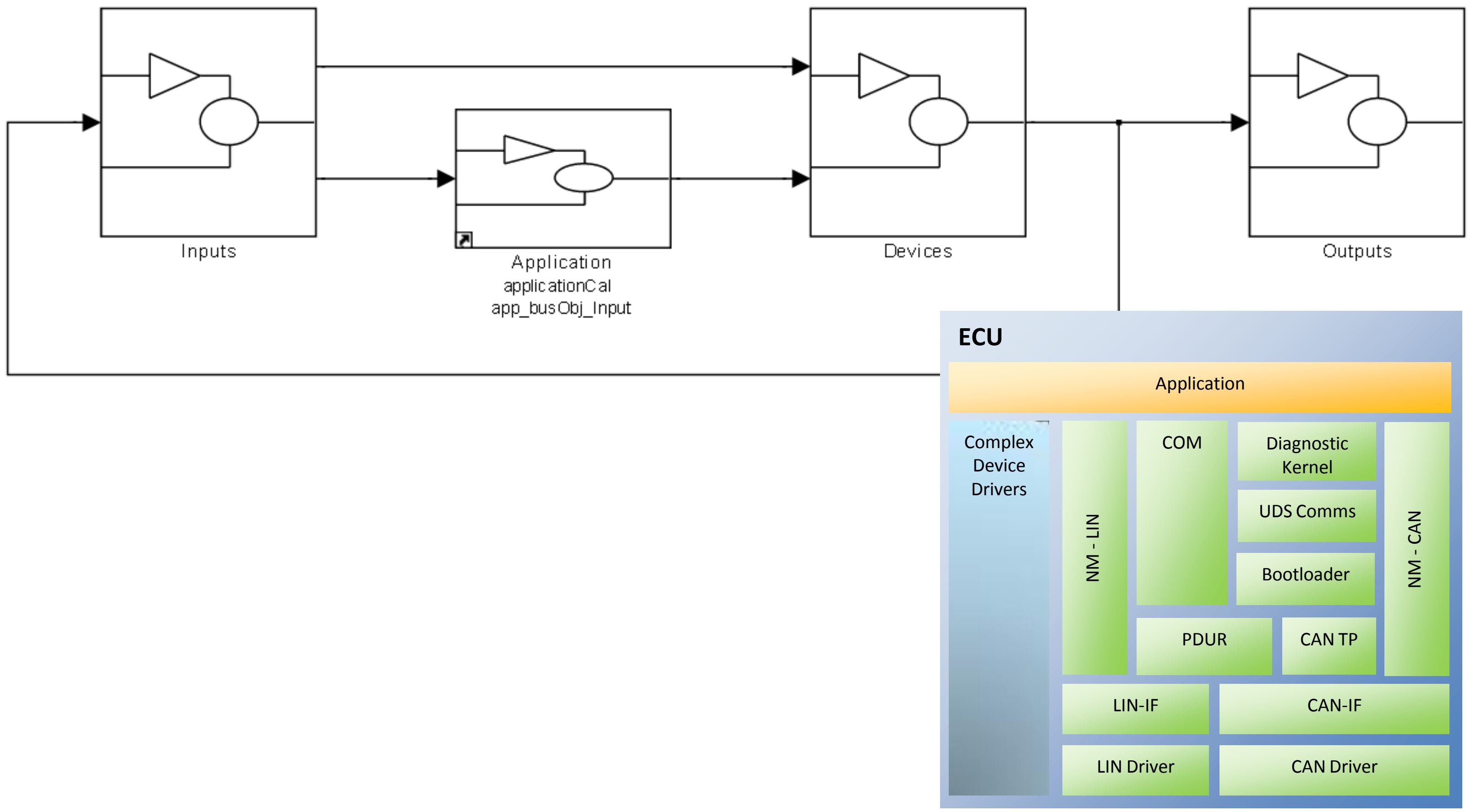
embed
turnkey software solutions

AUTOTXT™

Software Details

- Developed as a Simulink Model
- Device Drivers wrapped with Simulink Blocksets
- Code-Generation via Real Time Workshop Embedded Coder
- AUTOSAR ‘like’ where applicable
- Same application runs on previous hardware as well as new hardware.
  - Completely different processor (ARM7 -> MPC55xx)

Software Architecture

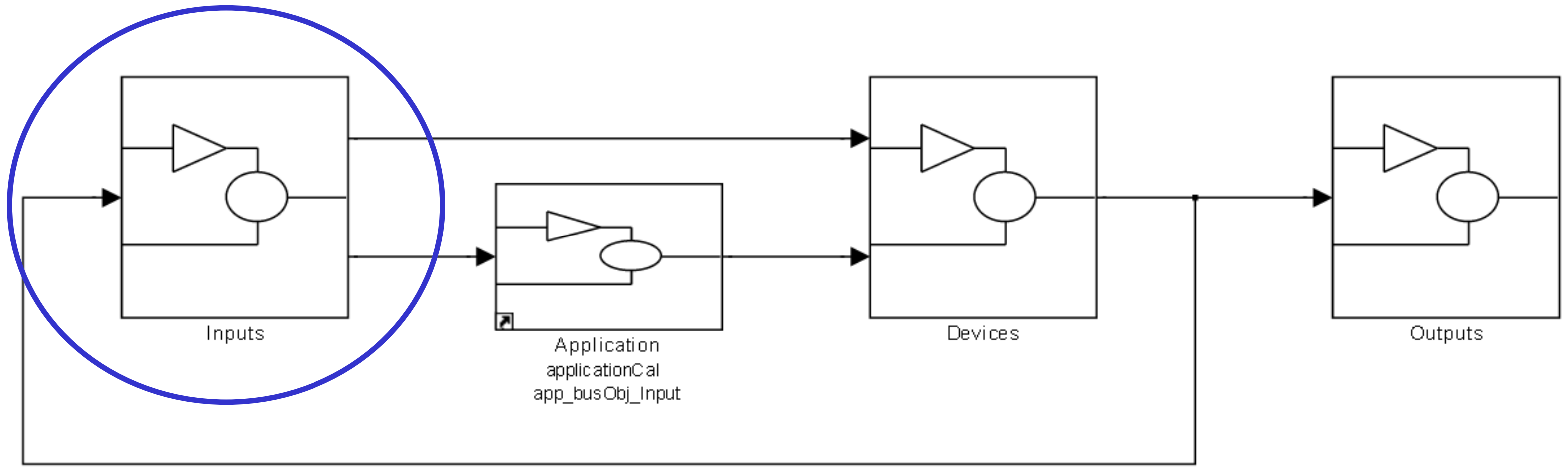


embed
turnkey software solutions

AUTOTXT™

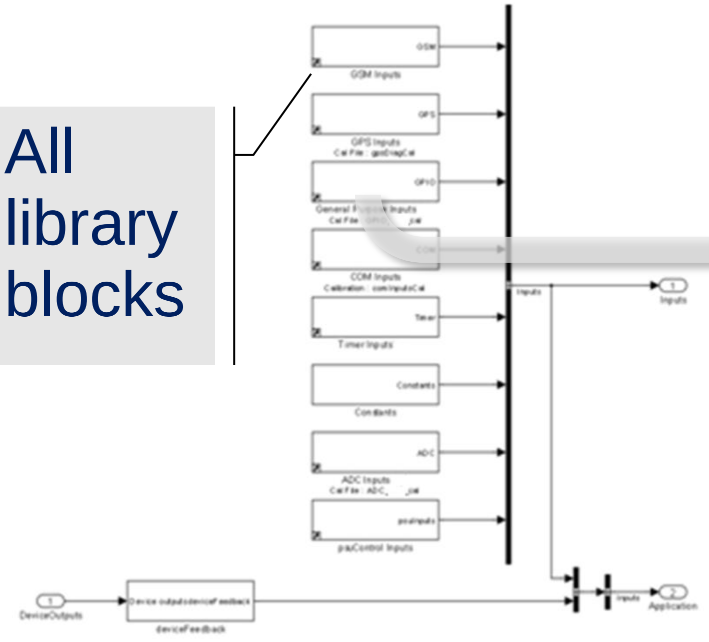
Inputs

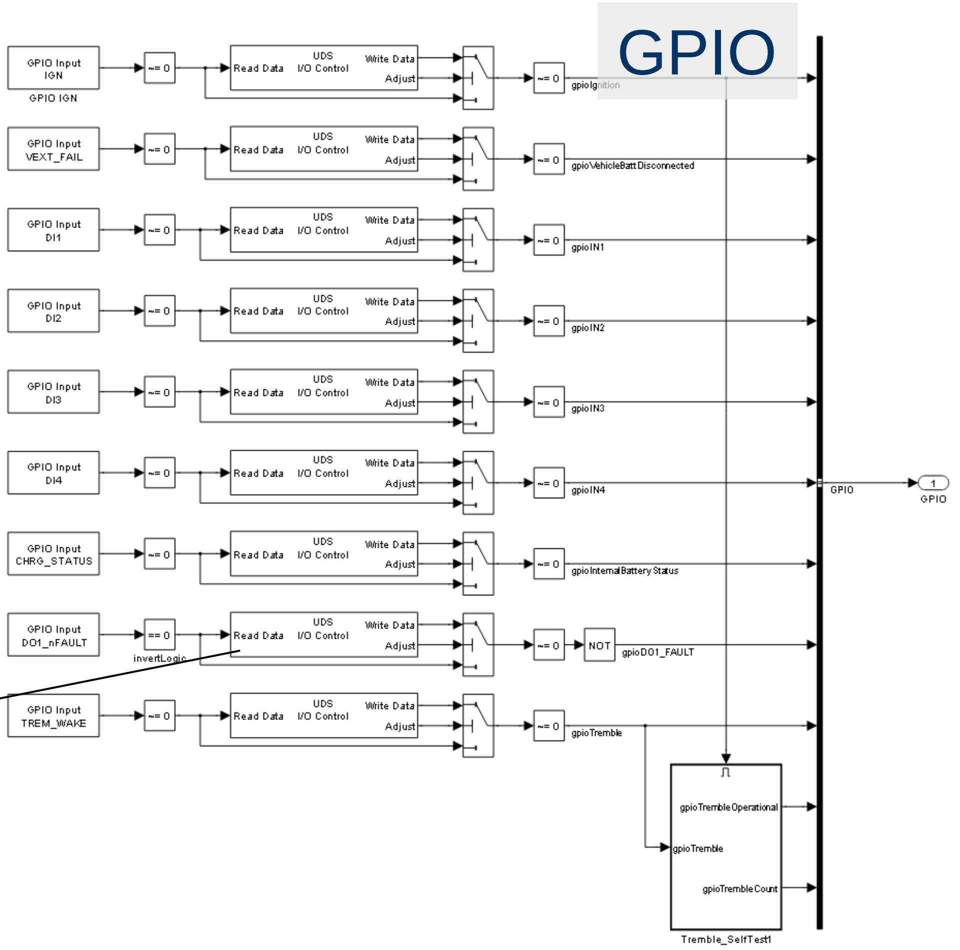
- Inputs are hardware specific and are separate from the application



Inputs

All library blocks





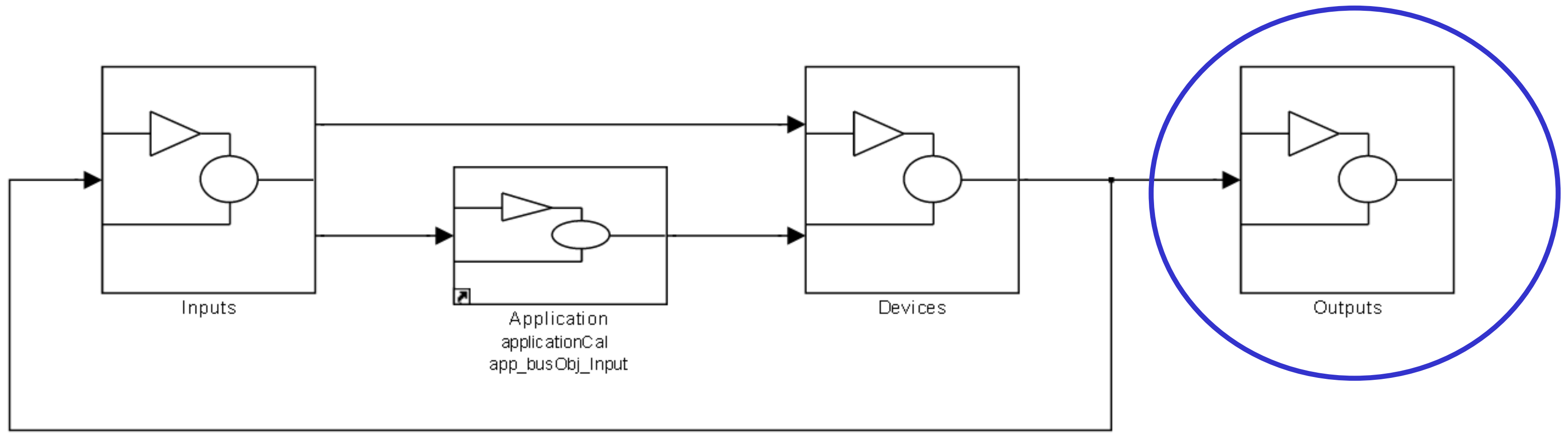
All inputs have UDS IO control, defined in the model

embed turnkey software solutions

AUTOTXT™

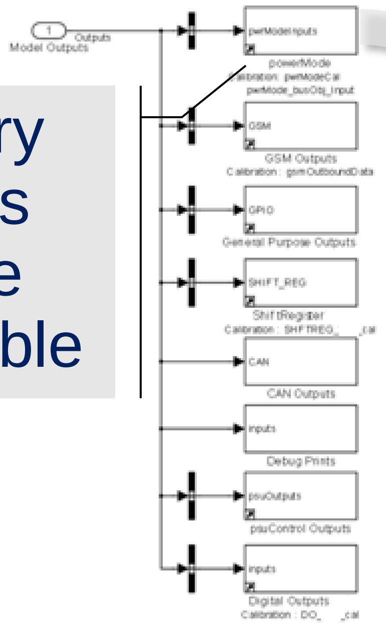
Outputs

- Outputs are hardware specific and are separate from the application



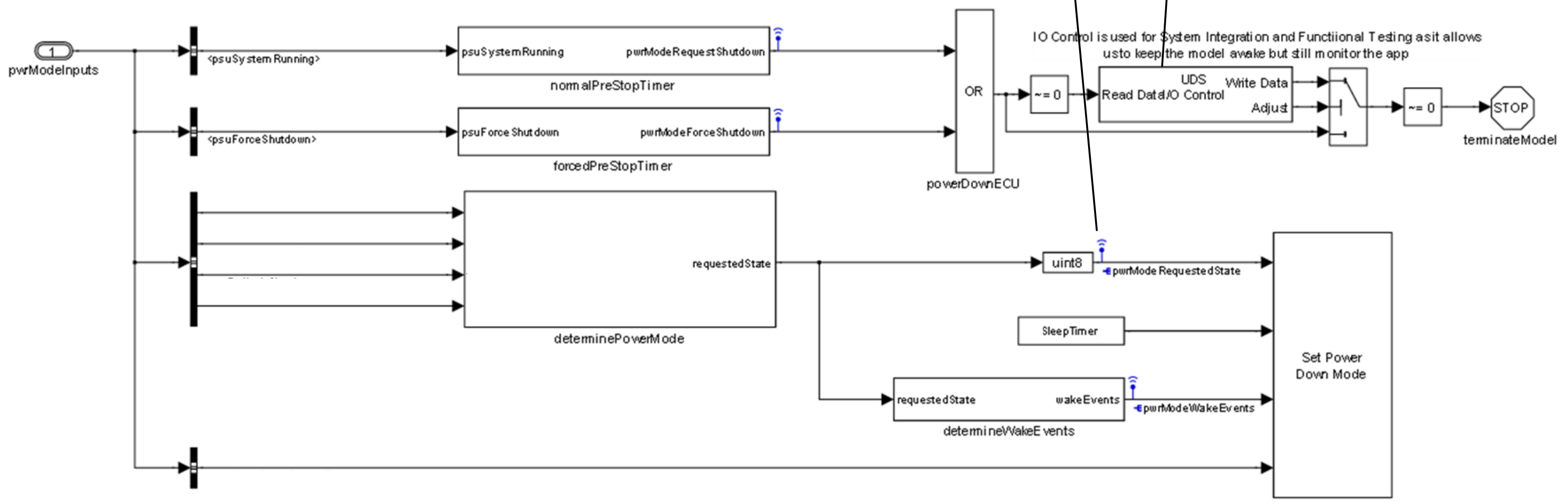
Outputs

Library blocks where possible



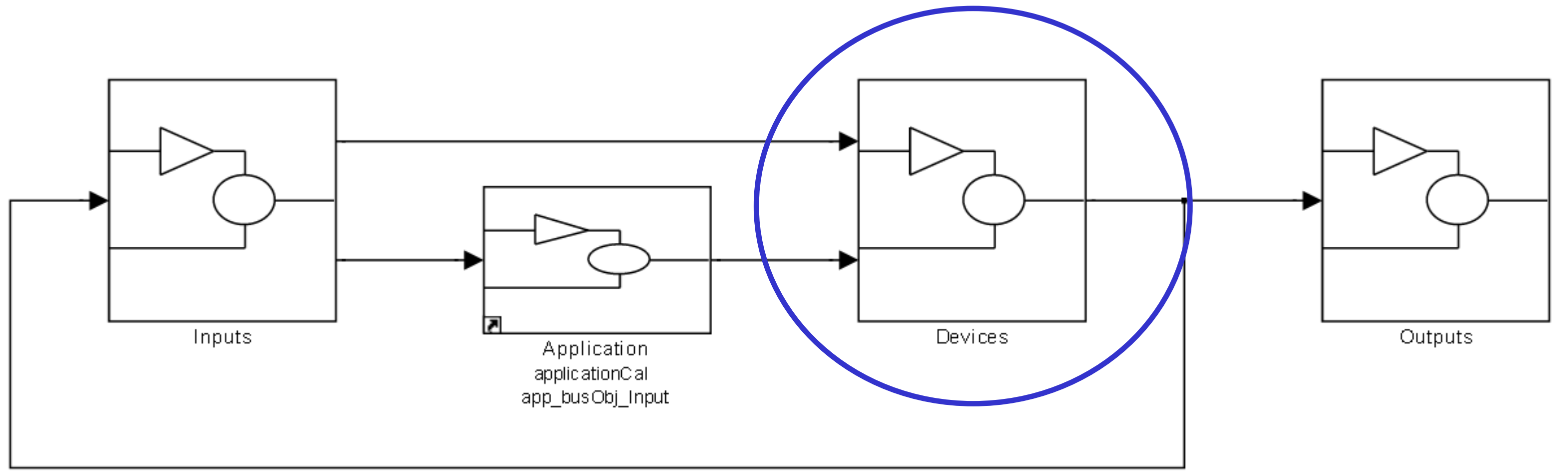
UDS Parameters to read internal data. Implemented as signals resolved to UDS objects

IO Control on all outputs



Device Drivers

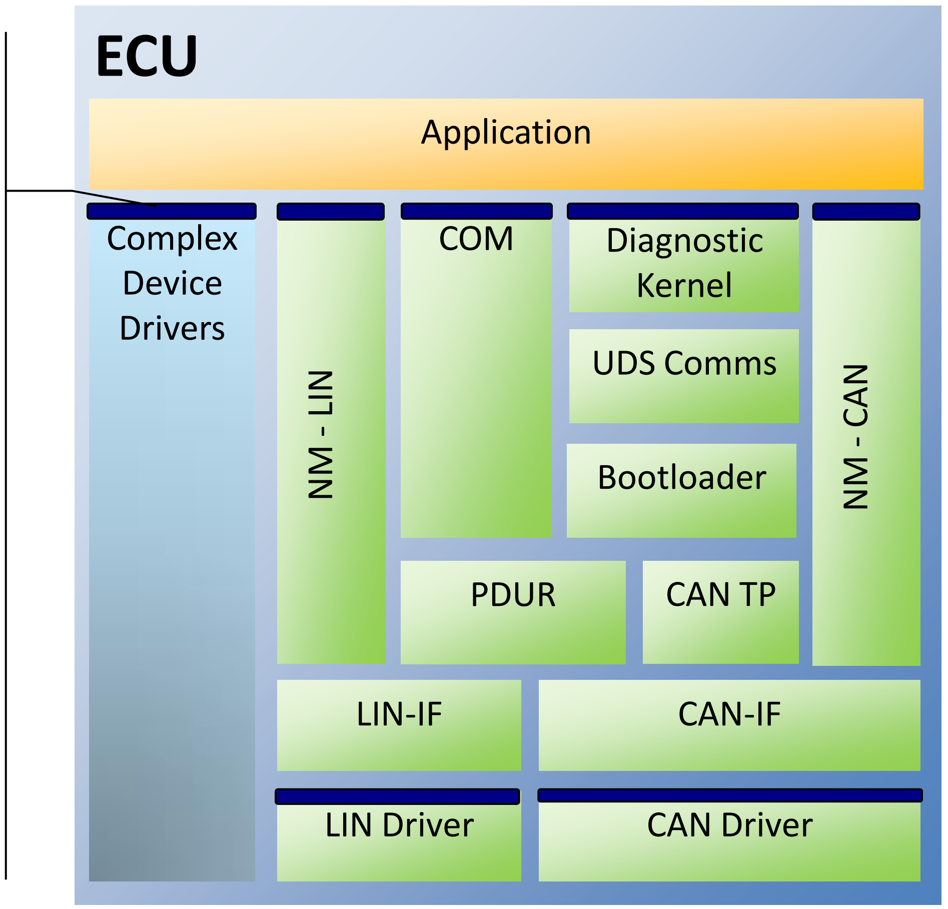
- Device Drivers are hardware specific and are separate from the application



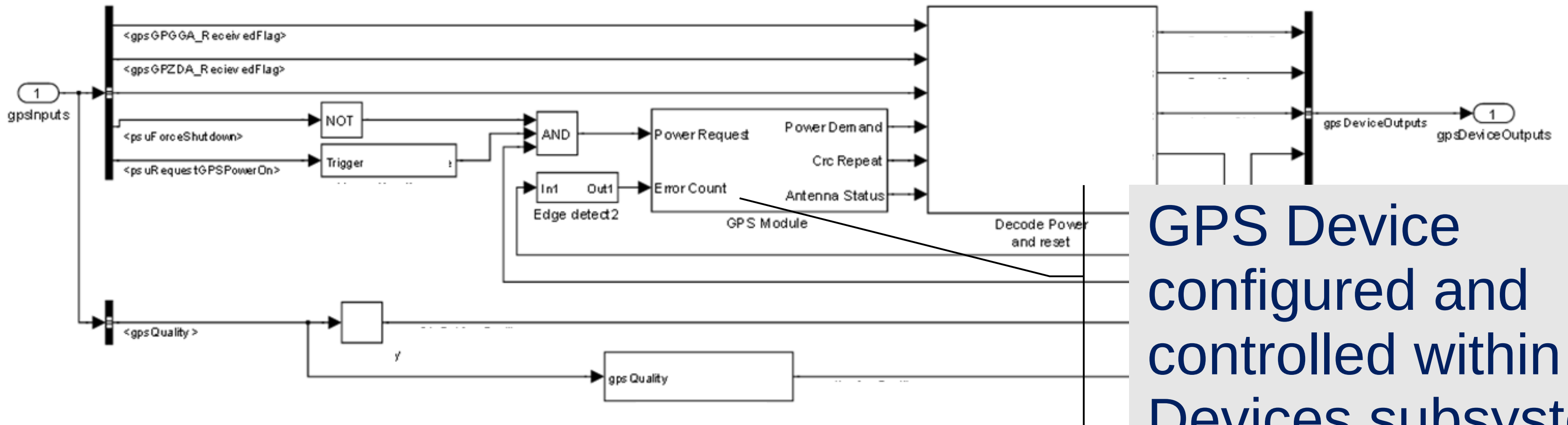
Device Drivers

Simulink Blocksets

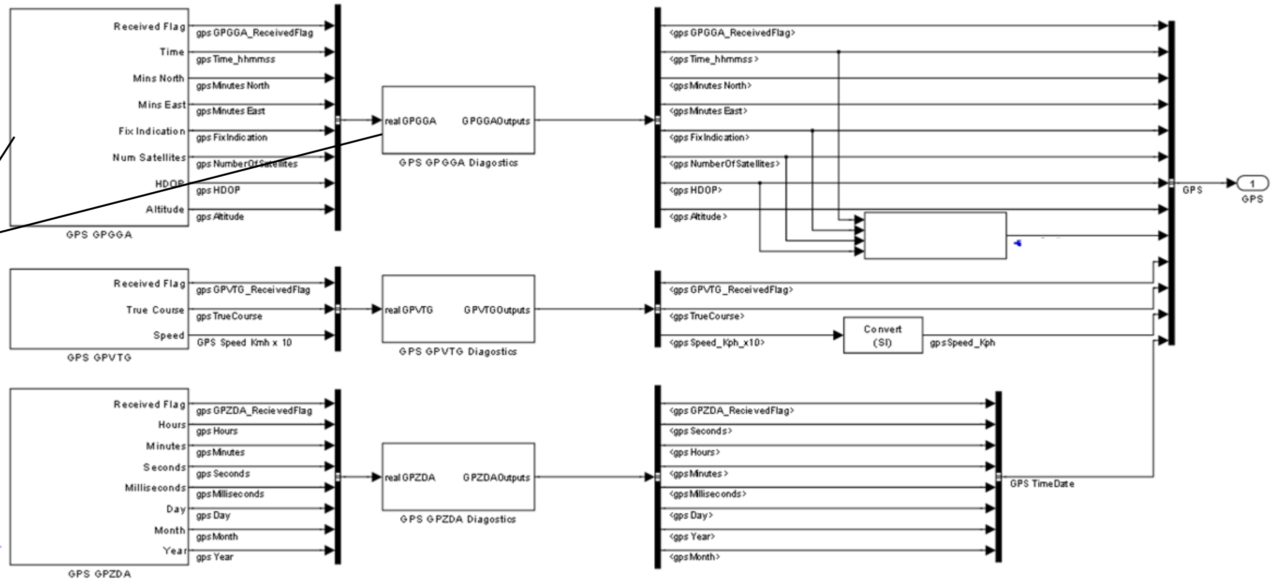
- CAN
- COM
- UDS
- Network Management
- GSM
- GPS
- Bluetooth
- GPIO
- Timers
- Power Management



Device Driver Blocksets (GPS)



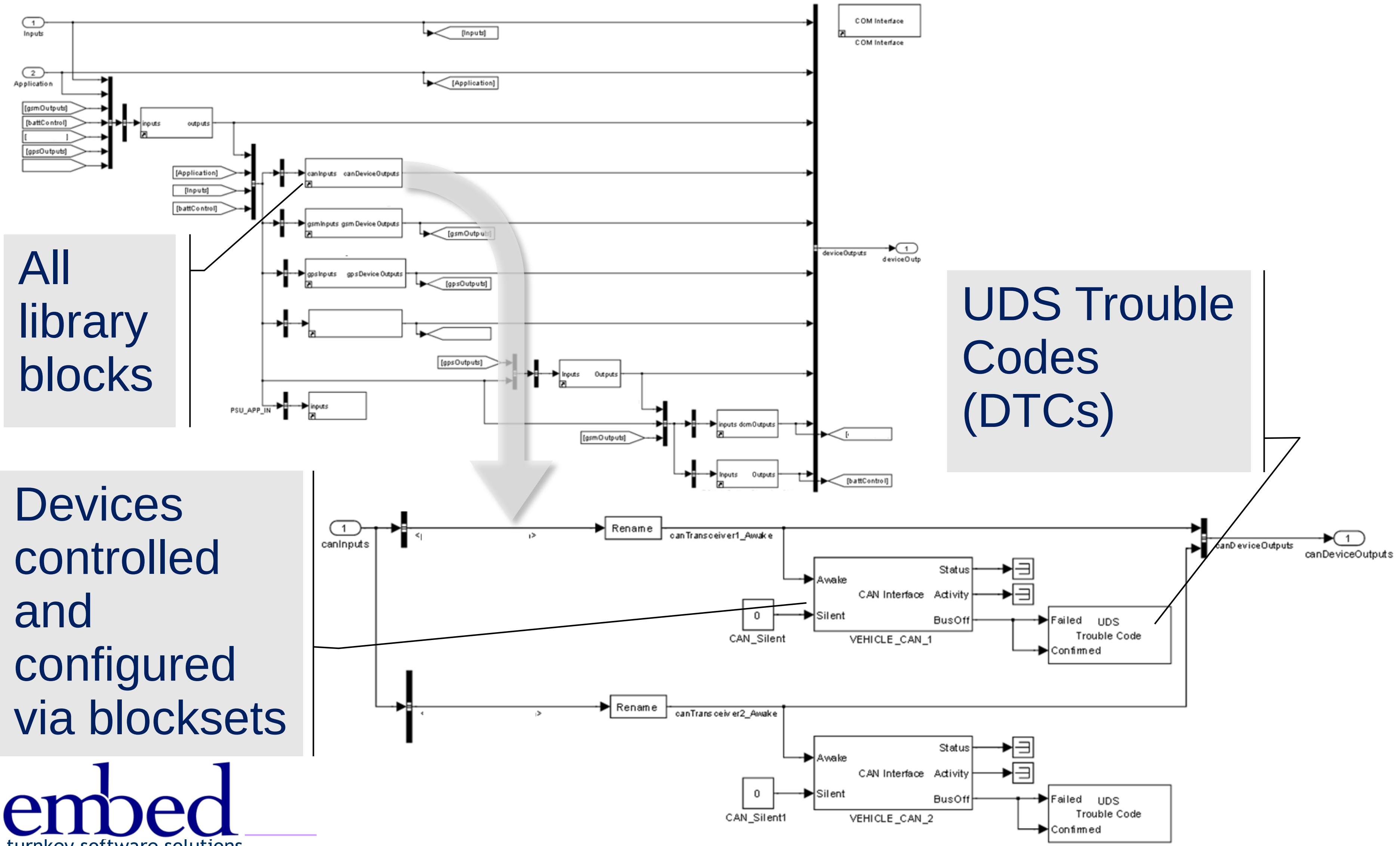
GPS Device configured and controlled within Devices subsystem



Data received in the Inputs subsystem complete with UDS IO Control

embed
turnkey software solutions

Device Driver Blocksets (CAN)



All library blocks

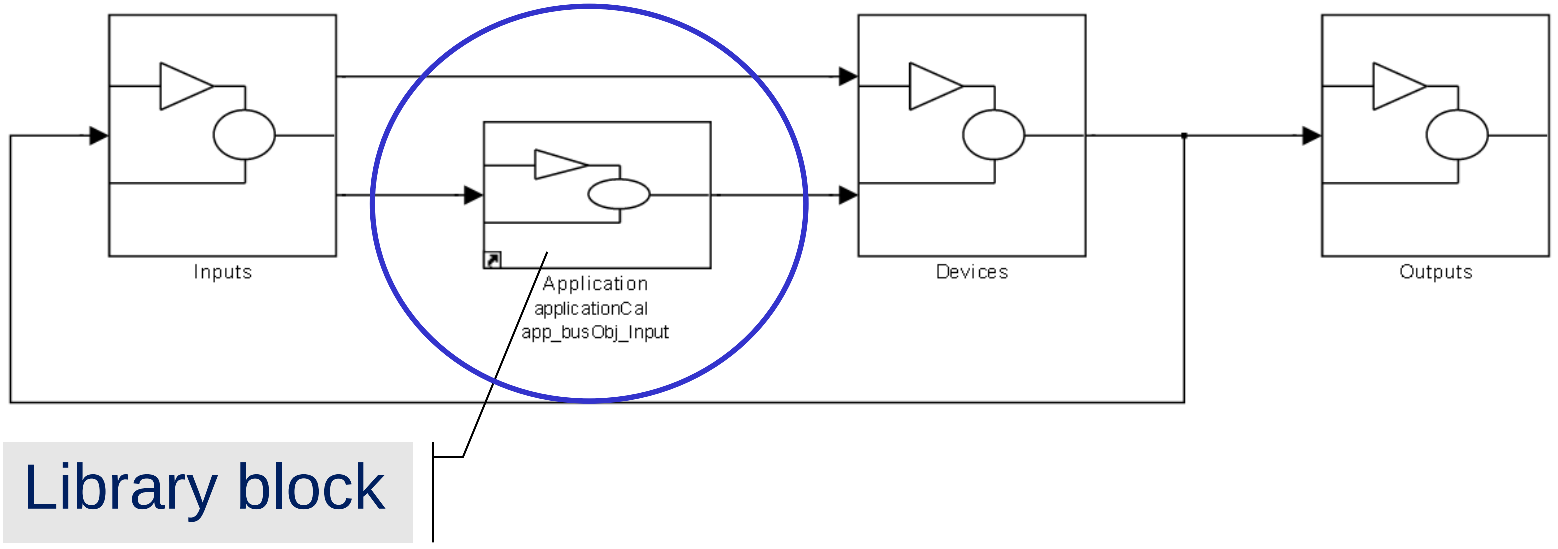
UDS Trouble Codes (DTCs)

Devices controlled and configured via blocksets

embed
turnkey software solutions

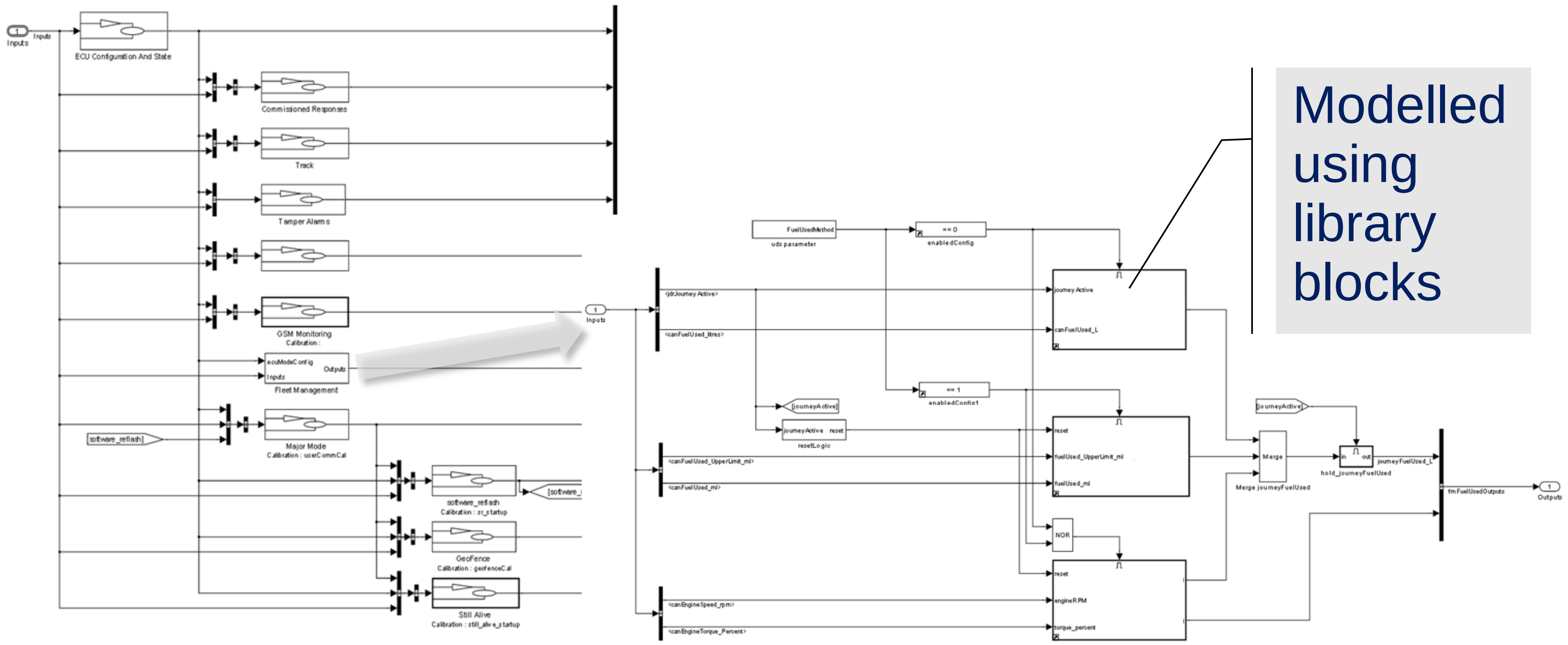
Application

- All hardware specifics architected out
- Easily ported to any hardware



Application

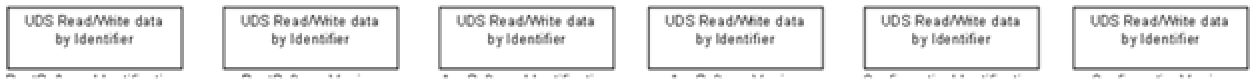
- Application further decomposed and Architected within Simulink

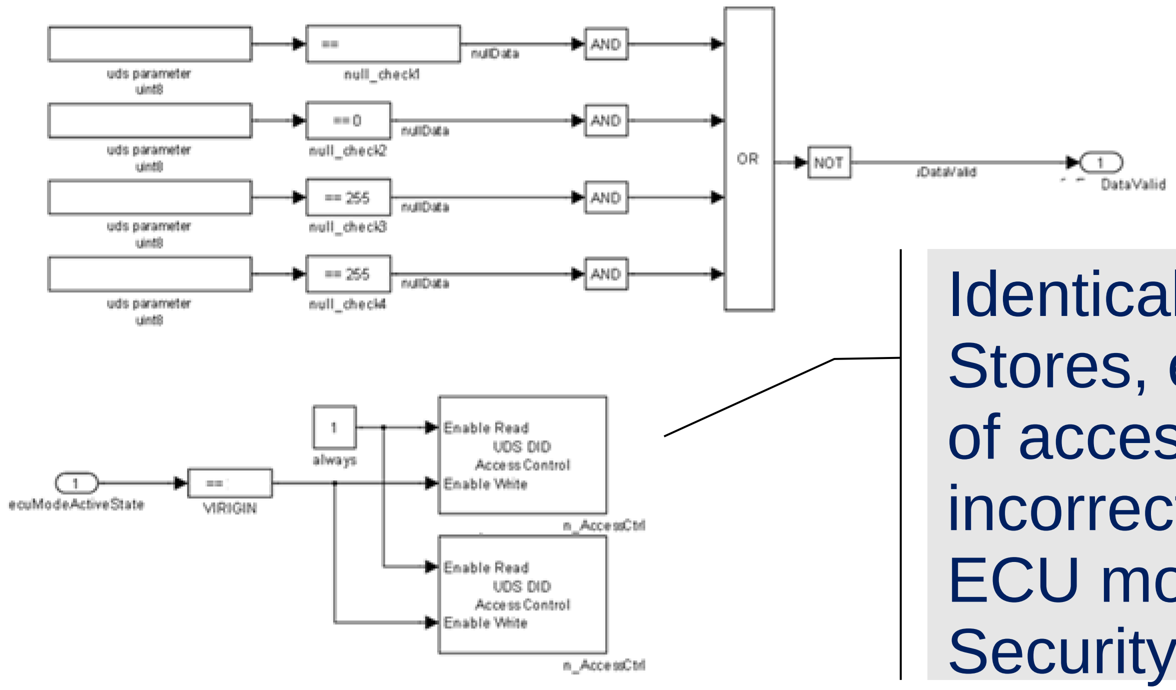


Modelled using library blocks

Application EOL Programming

- UDS Parameters (PIDs and DIDs)


UDS Read/Write data by Identifier
UDS Read/Write data by Identifier
UDS Read/Write data by Identifier
UDS Read/Write data by Identifier
UDS Read/Write data by Identifier
UDS Read/Write data by Identifier



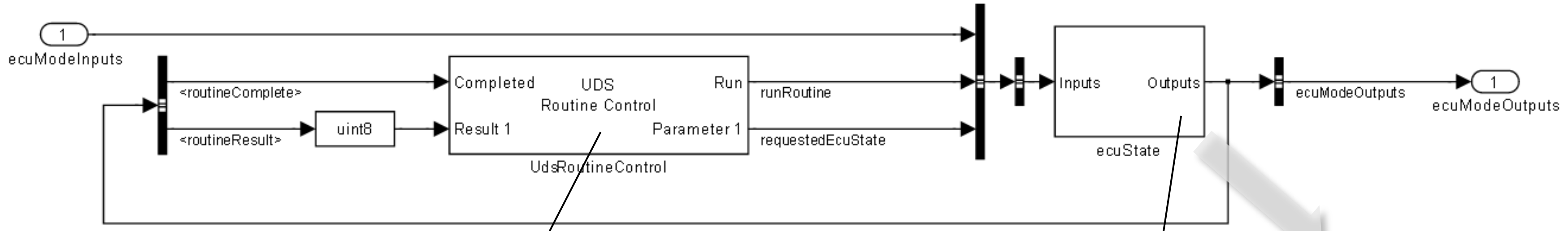
Identical usage to Data-Stores, except the addition of access control to inhibit incorrect usage due to ECU mode or wrong Security Access Level

embed
turnkey software solutions

AUTOTXT™

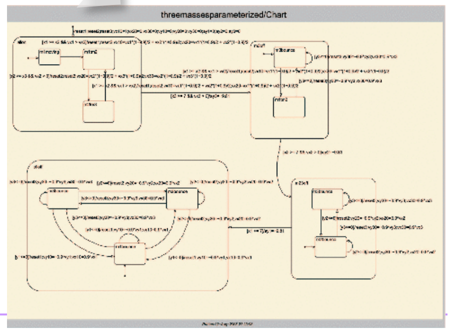
Application Commissioning

- Commissioning the Telematics ECU achieved using UDS Routine Control
  - Accessible only via Security Access



Controls the UDS Routine and UDS communications

Model the logic within Simulink, test and debug at the model level instead of in code



turnkey software solutions

Telematics ECU Development Summary

- Up to 7 developers worked on the software concurrently
  - Auto-txt and Embed Engineers
- Developed using a recognised Quality Management System
  - Automotive SPICE® Level 3
- Fully auto generated code from Simulink and Stateflow using RTW-EC
- Application runs on two very different micro-processors
- New developers quickly became productive
- Low defect development
- New features being added to the ECU every month

Diagnostics Services Development: How Model Based Development can make big improvements

embed
turnkey software solutions

AUTOTXT™

UDS the typical picture

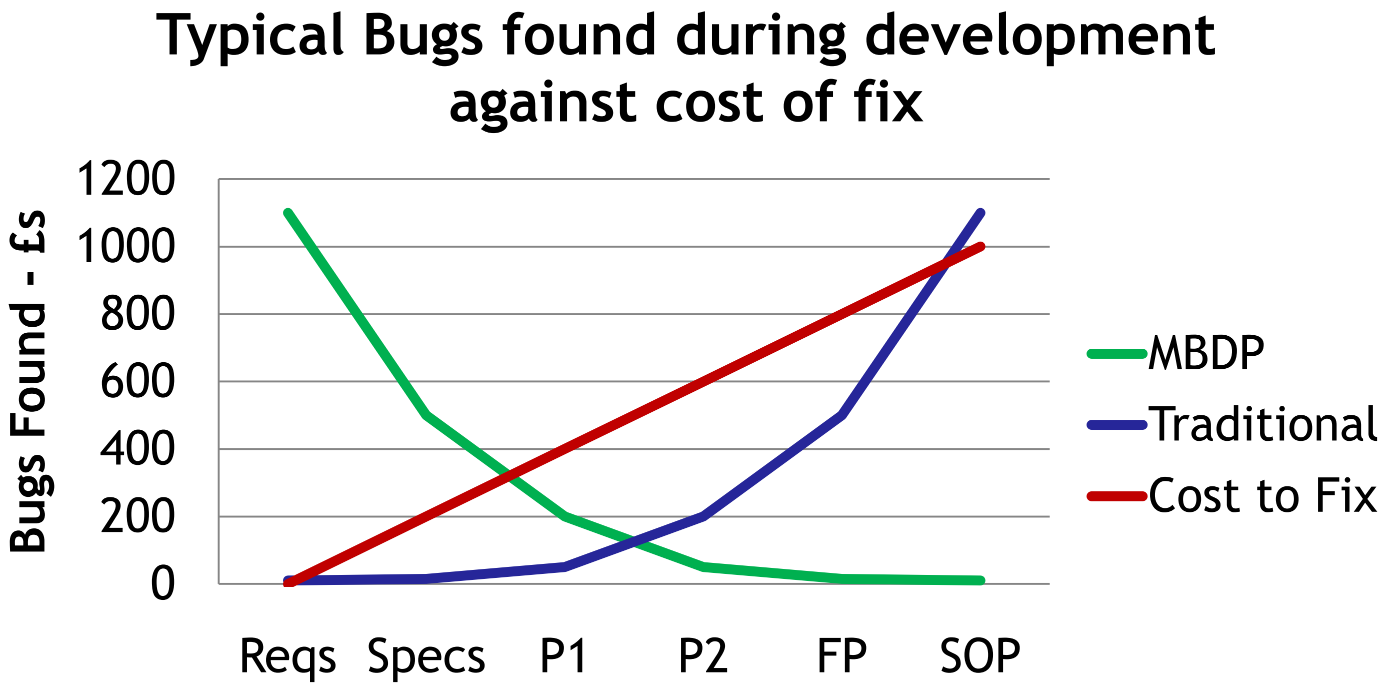
- Delivered last...
  - Everyone is focused on features and functionality.
- Delivered late...
  - The feature or function owner rarely truly addresses the diagnostics requirements.
- Delivered wrong...
  - Functionality related to diagnostics isn't often captured by feature designer
    - Add/delete key fob, match PCM/IMMO, etc
  - The diagnostics team are a little detached from the rest of the development team and are usually the last to be informed of any changes.
  - Need to speak and think hex-codes, not English.

UDS by Model Based Development

- Model Based Development has been proven to increase quality and speed time to market
  - Correctly define the requirements
  - Solve problems early
  - Remove translation errors
  - Iteration loops at the ‘cheap’ stage of development
  - Code generation from the models

UDS by Model Based Development

- Enables the Diagnostics to be addressed early on in the development cycle



embed
turnkey software solutions

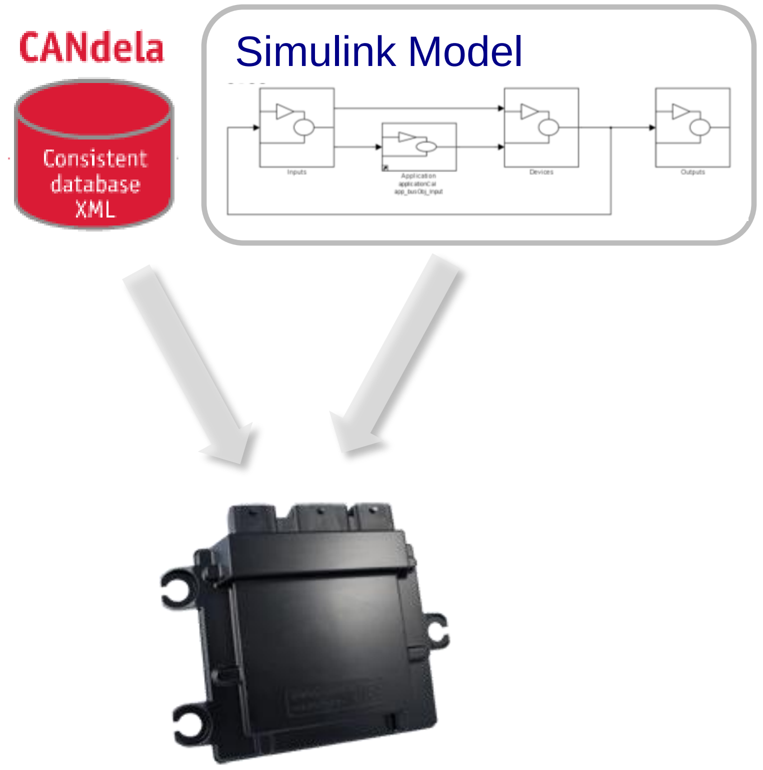
AUTOTXT™

UDS by Model Based Development

- Enables the Diagnostics to be designed in Simulink
  - Focusing on the functionality not on the code
  - Testable in a friendly environment
  - Testable before the ECUs are available
  - Most important for routine control
    - Self tests and calibration (stepper motor end stops etc)
    - Programming keys and ECUs in Immobilisation Ring

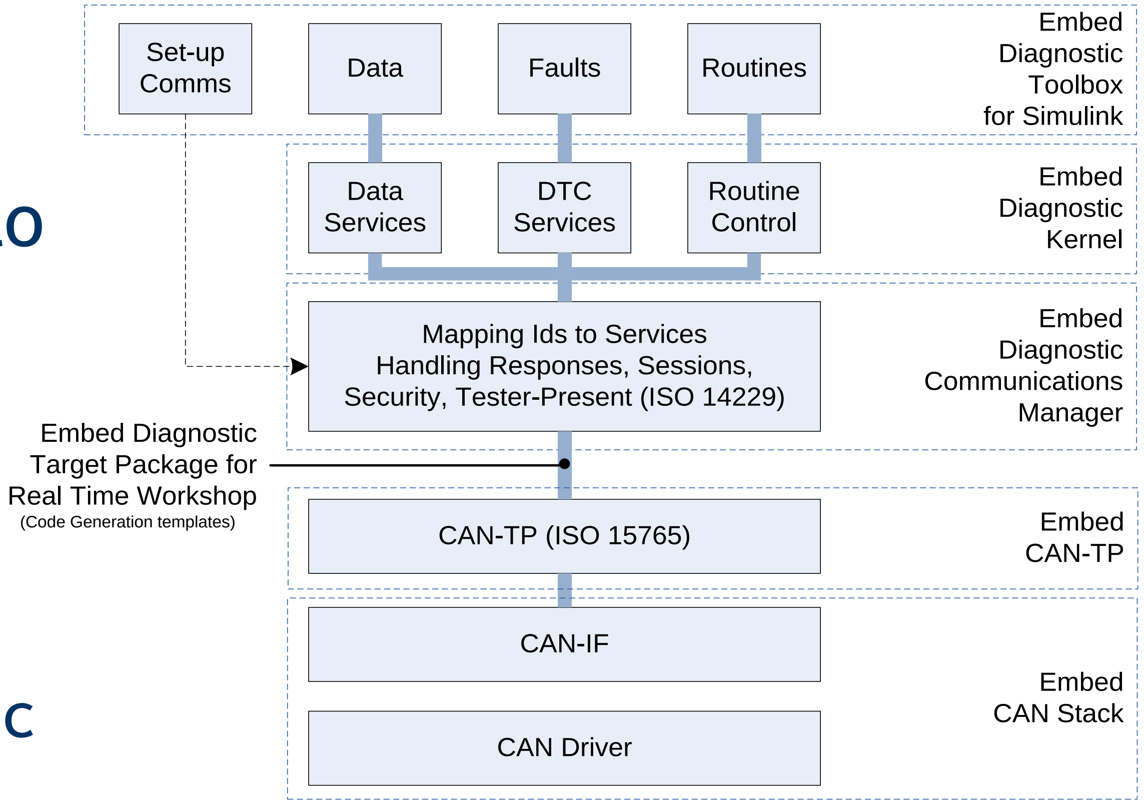
UDS by Model Based Development

- Detaches the Hex Codes from the Diagnostics
  - Enables the diagnostics functionality to be ported from one ECU to another
  - Diagnostics live with the features where they are used
  - Diagnostics database configures the feature for the ECU / Vehicle for deployment



UDS by Model Based Development

- Off-the-shelf Simulink Blockset
- Clear interfaces to enable any CAN stack to be used
- ASNII-C MISRA compliant code
  - Hardware agnostic


Set-up Comms
Data
Faults
Routines
Embed Diagnostic Toolbox for Simulink
Data Services
DTC Services
Routine Control
Embed Diagnostic Kernel
Mapping Ids to Services Handling Responses, Sessions, Security, Tester-Present (ISO 14229)
Embed Diagnostic Communications Manager
Embed Diagnostic Target Package for Real Time Workshop
(Code Generation templates)
CAN-TP (ISO 15765)
Embed CAN-TP
CAN-IF
CAN Driver
Embed CAN Stack

embed
turnkey software solutions

AUTOTXT™

UDS Blockset Summary (ISO 14229)

- There are clear and large advantages of developing diagnostics services using a Model Based Development Process.
- The Embed UDS Blockset can be placed on top of any CAN stack
- Please get in touch if you need more information

ivan.wilson@embeduk.com

Thank you

embed
turnkey software solutions

AUTOTXT™