# FROM SHORTAGE TO SOLUTION: RAPID PROTOTYING STRATEGIES FOR AUTMOTIVE EMBEDDED SYSTEMS
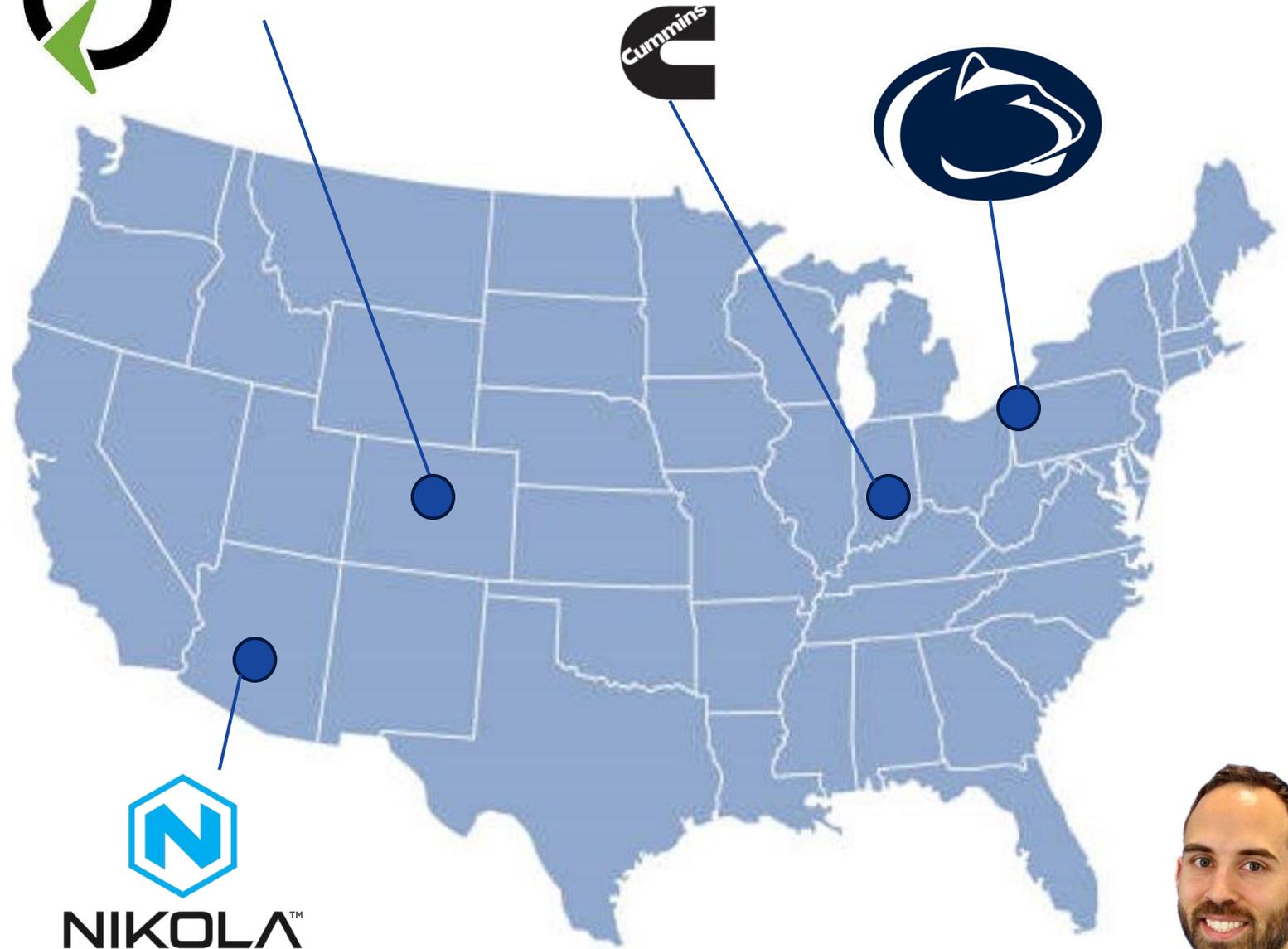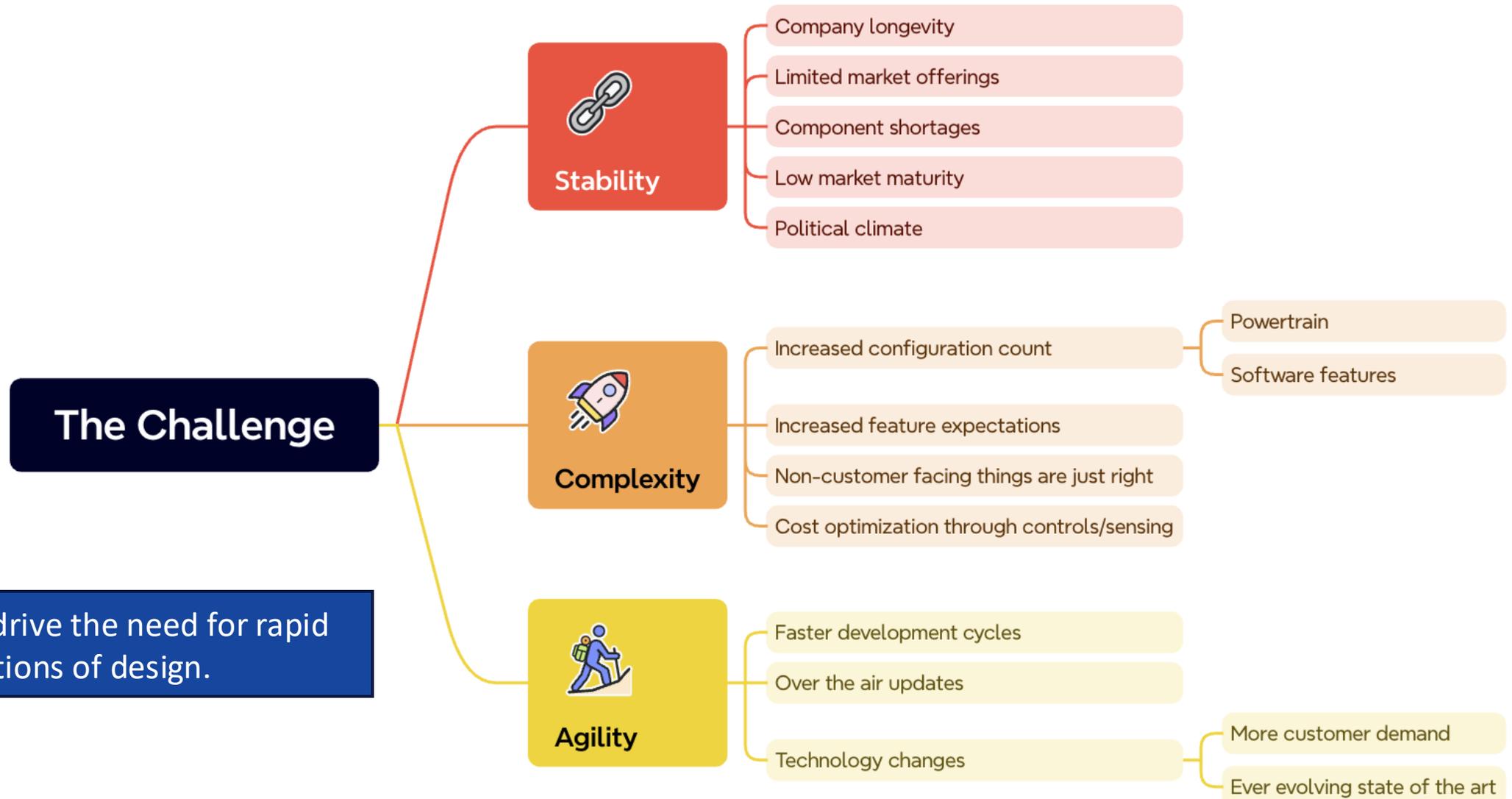
MICHAEL CORONA

9/22/25

# BACKGROUND

- Michael Corona – Eng Director of Powertrain and Control Systems @ GILLIG

- 13 years automotive embedded controls experience

- Significant involvement in 8 vehicle rapid prototypes – 3 made it to full production

- Involvement in countless component obsolescence and replacement projects. Worked with 7 different HV batteries suppliers.

- From CAN driver development to writing HV battery thermal management algorithms to full vehicle electrical architect

# THE CHALLENGE



**The Challenge**

**Stability**
- Company longevity
- Limited market offerings
- Component shortages
- Low market maturity
- Political climate

**Complexity**
- Increased configuration count
  - Powertrain
  - Software features
- Increased feature expectations
- Non-customer facing things are just right
- Cost optimization through controls/sensing

**Agility**
- Faster development cycles
- Over the air updates
- Technology changes
  - More customer demand
  - Ever evolving state of the art

These items drive the need for rapid iterations of design.
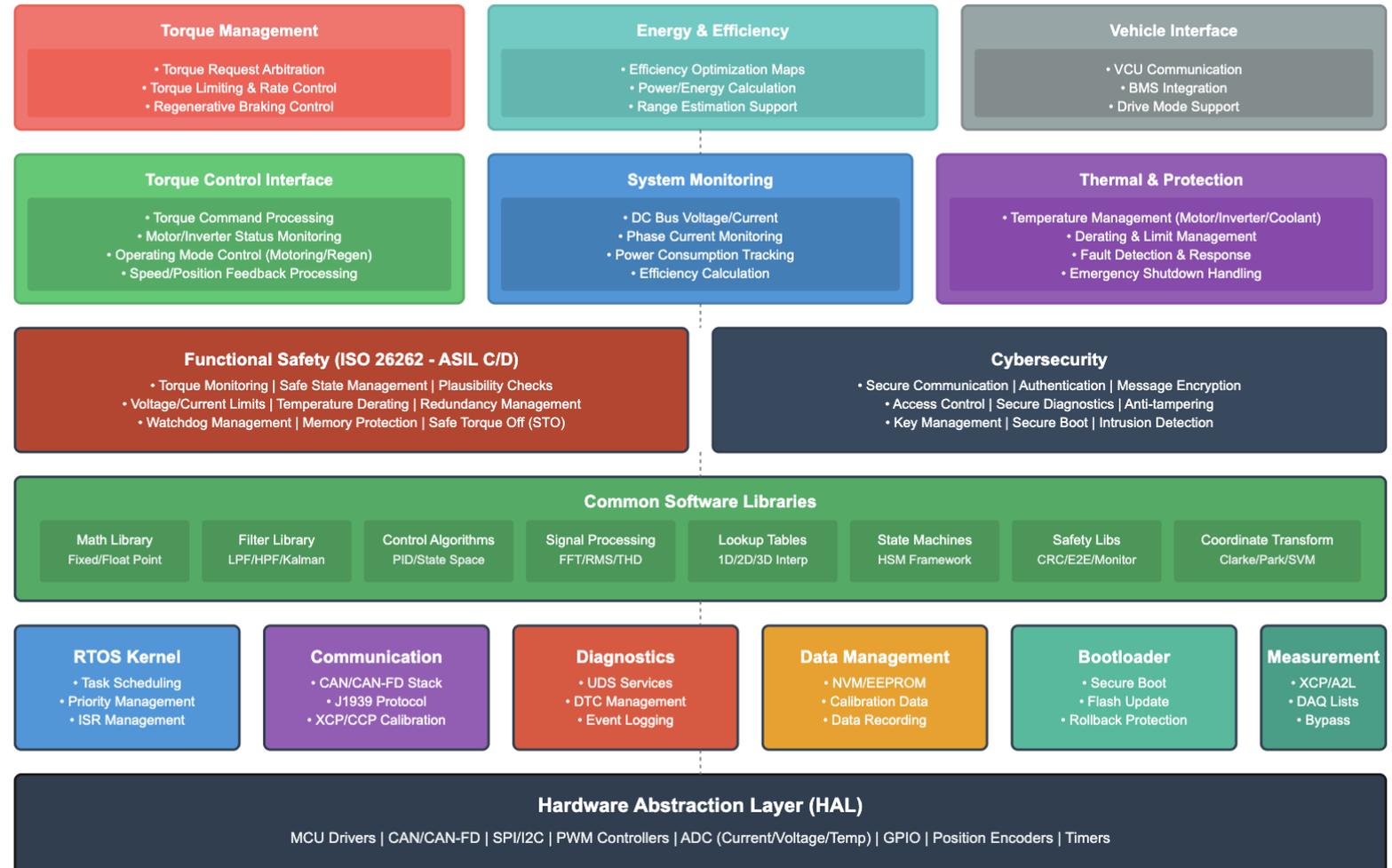
# SCOPE TO YOUR FOCUS

- What are the areas you want to develop in?
  - What provides significant value to your business?
  - Where do you want to be unique?
  - Where do you want to invest and innovate?
- Focus on stability in areas with high re-use – Where are you most in control?
- Focus on modularity in areas with known instability

## Open Systems Interconnection (OSI) Model

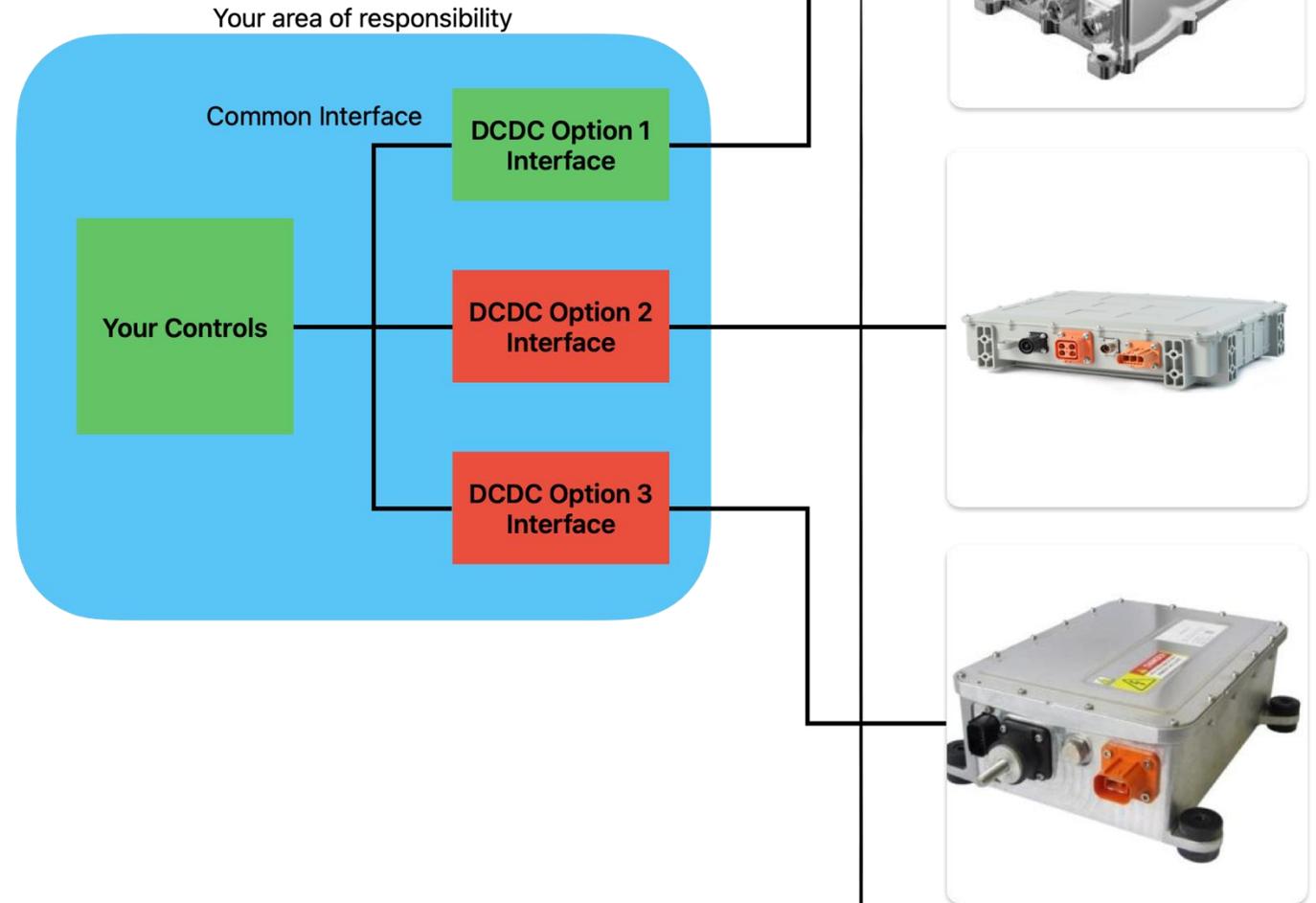| | | | |
|---|---|---|---|
| Application | Torque Map | Interlocks | Visualizations |
| Presentation | Scaling | | J1939 |
| Session | UDS | | XCP |
| Transport | | Multi-Packet Messaging | |
| Network | Addressing | | IDs |
| Data-Link | | Packet structure | |
| Physical | CAN | | Ethernet |

# ENABLE A REUSABLE FLEXIBLE ARCHITECTURE

- Break up your architecture into manageable parts that have interfaces that make sense.

- Maintain interfaces and repositories for the items you need to be modular.

- For components that you need "plug and play" on, create standard interfaces.



**EV Powertrain Control Module - Modular Architecture**

**Torque Management**
- Torque Request Arbitration
- Torque Limiting & Rate Control
- Regenerative Braking Control

**Energy & Efficiency**
- Efficiency Optimization Maps
- Power/Energy Calculation
- Range Estimation Support

**Vehicle Interface**
- VCU Communication
- BMS Integration
- Drive Mode Support

**Torque Control Interface**
- Torque Command Processing
- Motor/Inverter Status Monitoring
- Operating Mode Control (Motoring/Regen)
- Speed/Position Feedback Processing

**System Monitoring**
- DC Bus Voltage/Current
- Phase Current Monitoring
- Power Consumption Tracking
- Efficiency Calculation

**Thermal & Protection**
- Temperature Management (Motor/Inverter/Coolant)
- Derating & Limit Management
- Fault Detection & Response
- Emergency Shutdown Handling

**Functional Safety (ISO 26262 - ASIL C/D)**
- Torque Monitoring | Safe State Management | Plausibility Checks
- Voltage/Current Limits | Temperature Derating | Redundancy Management
- Watchdog Management | Memory Protection | Safe Torque Off (STO)

**Cybersecurity**
- Secure Communication | Authentication | Message Encryption
- Access Control | Secure Diagnostics | Anti-tampering
- Key Management | Secure Boot | Intrusion Detection

**Common Software Libraries**

| Math Library | Filter Library | Control Algorithms | Signal Processing | Lookup Tables | State Machines | Safety Libs | Coordinate Transform |
|---|---|---|---|---|---|---|---|
| Fixed/Float Point | LPF/HPF/Kalman | PID/State Space | FFT/RMS/THD | 1D/2D/3D Interp | HSM Framework | CRC/E2E/Monitor | Clarke/Park/SVM |

**RTOS Kernel**
- Task Scheduling
- Priority Management
- ISR Management

**Communication**
- CAN/CAN-FD Stack
- J1939 Protocol
- XCP/CCP Calibration

**Diagnostics**
- UDS Services
- DTC Management
- Event Logging

**Data Management**
- NVM/EEPROM
- Calibration Data
- Data Recording

**Bootloader**
- Secure Boot
- Flash Update
- Rollback Protection

**Measurement**
- XCP/A2L
- DAQ Lists
- Bypass

**Hardware Abstraction Layer (HAL)**

MCU Drivers | CAN/CAN-FD | SPI/I2C | PWM Controllers | ADC (Current/Voltage/Temp) | GPIO | Position Encoders | Timers

# EXECUTE

- Develop rigid interfaces between different software components

- Spend the most time verifying and validating areas with high re-use.

- Continuously monitor the components that are *unstable*



Components serving the same purpose

Your area of responsibility

Common Interface

Your Controls

DCDC Option 1 Interface

DCDC Option 2 Interface

DCDC Option 3 Interface

# THE BOTTOM LINE

**Rapid Prototyping
and Replacement**

Software architecture needs to be well thought out up front to enable plug and play modularity

Relatively higher effort on reusable portions of software

Component replacement development effort can be minimized through a modular software approach

Trial feasibility studies on component replacement and determine impacted areas

GILLIG

# THANK YOU

GILLIG