APPENDIX I

Technical Specifications

Table of contents

1 Introduction

For 2026 the FIA intends to introduce a new Standard Electronic Control Unit (SECU) to be used by all competitors in the Formula One championship.

The core requirements remain unchanged compared to the current SECU in use in the Formula One Championship. In fact, the uttermost priorities are to ensure that cars always run safely, reliably and in conformity with the governing rules. This notably requires the SECU to interface with the FIA on-car safety systems, including the Accident Data Recorder (ADR) and the FIA F1 Marshalling System (F1MS) as well as their ancillaries. Another critical requirement for the SECU and its associated toolset is to empower FIA technical delegates with enough data intelligence to assess competitors' compliance with the Formula One Technical and Sporting regulations.

Besides those essential requirements, the FIA and the Teams also pursue objectives toward cost control and technology rationalisation that will allow Formula One to remain at the forefront of technology while guaranteeing a fair competition, an entertaining show and a sustainable environment for the Sport and its stakeholders.

The SECU is primarily required to control the hybrid Power Unit, Gearbox, Clutch and Differential that constitute the current Formula One powertrains, as well as the Adjustable Bodywork, Brake System and Tyre Monitoring System on the car.

The renewed digital architecture should guarantee a smooth integration with teams', Power Unit suppliers' and FIA's current off-car hardware and software while maintaining or exceeding the current level of data protection.

This document provides an overview of the hardware and software functionalities required for the SECU and the associated digital architecture in order to meet these requirements.

All proposals must include maintenance and repair requirements and costs, software update support for the duration of the contract in addition to initial development, integration and track support for all teams, PU suppliers and FIA for all competitions and testing as set out in the contract.

2 Definitions

Standard: designed according to an FIA specification. Manufactured and supplied by a single provider.

Prescribed: designed according to an FIA specification. May be manufactured and supplied by multiple providers.

Proprietary: developed by a team or by a third party.

ECU: Electronic Control Unit

SECU: FIA F1 Standard ECU

ADR: FIA Accident Data Recorder

F1MS: FIA F1 Marshalling system

PU: Power Unit

API: Application Programming Interface

REST: REpresentational State Transfer

Provider: the manufacturer and supplier selected by this invitation to tender

Team: the racing teams that have been accepted by the FIA to take part in the championship

PU supplier: supplier of a power unit to one or more teams

Entity: any one of: SECU provider, FIA, team, PU supplier

Competition: any race forming part of the championship and entered on the International Sporting Calendar of the FIA

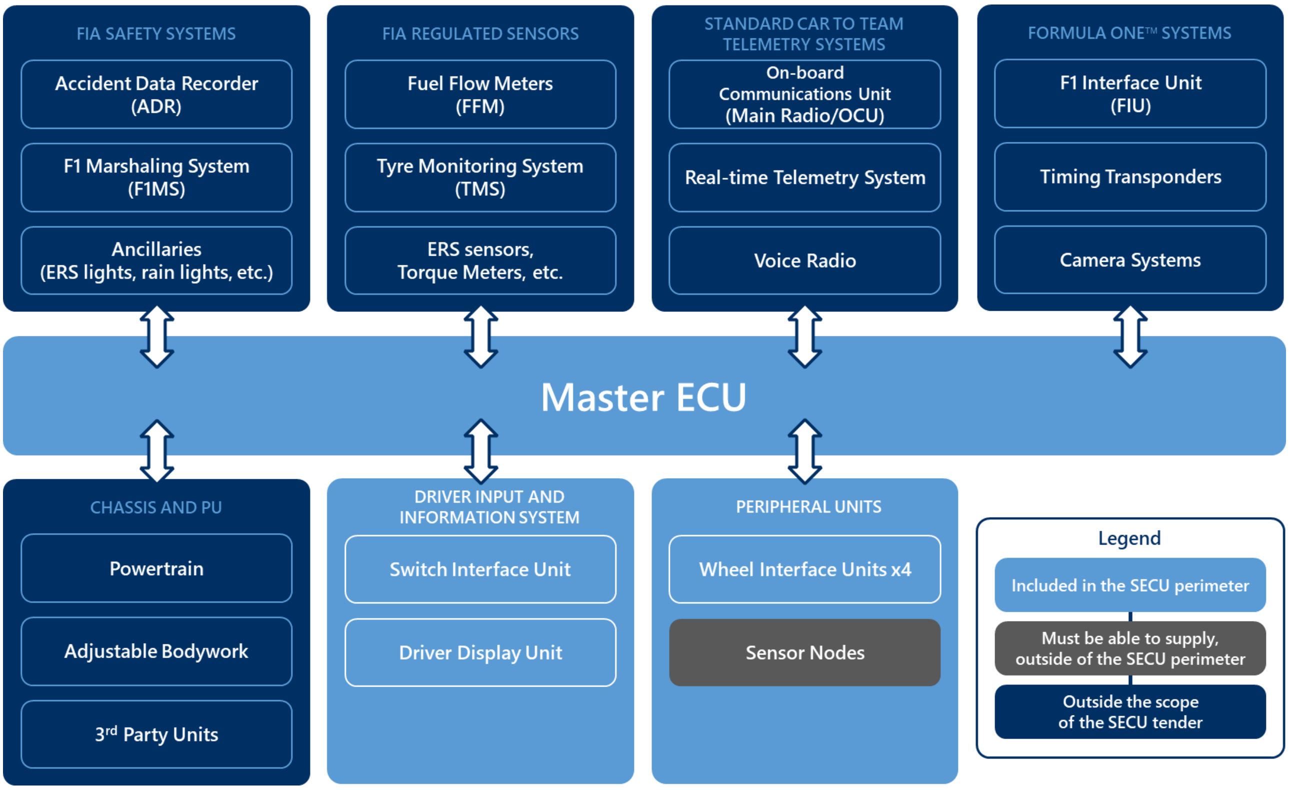
3 Overview

3.1 On-Car Systems

The SECU encompasses the following on-car hardware components:

- a Master Electronic Control Unit (Master ECU), including embedded control software and hardware resources and a configurable data logger;
- a Driver Input and Information System, including a switch interface and a driver display (screen + LEDs);
- a set of peripheral units including Wheel Interface Units and optional sensor nodes.

The on-car electronic architecture naturally structures itself around the Master ECU as outlined below.



Together, the on-car SECU hardware and software must be able to support all electro-mechanical systems running on a car designed according to the Formula One regulations. These include the hybrid Power Unit, Gearbox, Clutch and Differential that constitute current Formula One powertrains as well as the Adjustable Bodywork, Brake System and Tyre Monitoring System (TMS) of the car.

In practice, this requires providing the following key features, including but not limited to:

a) Closed-loop controllers for:

- a 6-cylinder spark-ignited engine, including maximum fuel mass flow limitation, maximum crankshaft speed limitation, throttle actuation, pneumatic valve air pressure, lambda sensors, fuel supply system as well as communication with any FIA sensors and/or peripheral actuator used for direct and indirect injection and for ignition;
- an Energy Recovery System (ERS), including kinetic energy recovery (MGU-K), exhaust energy recovery (MGU-H), power and energy monitoring, as well as communication with any FIA sensors and/or peripheral ERS actuator;
- a hydraulic Brake-By-Wire (BBW) system, including brake balance and pressure control;
- a PU torque arbitrator, coordinator and controller;

- a 8-speed semi-automatic gearbox, including 2 actuators, ratchet and closed loop types, gear engagement synchronisation and freewheel actuator;
- a hydraulic multi-plate clutch, including position, pressure and torque control;
- a hydraulic differential;
- adjustable bodywork sections;
- a number of digital controlled output including reverse gear selection, chassis-mounted lights and drink pump.

b) The electronics for a sophisticated driver interface including steering wheel and cockpit switches along with a driver display that may combine a set of coloured LEDs and a configurable high contrast and brightness screen.

c) Interface to ancillaries such as wheel interface units.

3.2 Off-Car Software Tool Suite

The SECU shall also include a PC Software Tool Suite and associated set of APIs that allow managing the SECU configurations as well as collecting and distributing the live and historical SECU and third-party data.

Data viewer and analysis tools are not included in the SECU perimeter but the provider must be able to supply a data viewer suitable for Formula One with the SECU on offer.

Typical use cases include but are not limited to:

a) SECU configuration management:

- Reprogramming embedded software code, defining data logging and CAN configurations,
- Setting applications parameters for control strategies and/or calibration of sensors and actuators,
- Setting application parameters on the fly, commonly referred to as “Live tuning”, i.e. without having to save any file on the disk or reprogramme the ECU,
- Importing, viewing, editing, comparing and/or merging two or more SECU applications parameter sets,
- Exporting part or the complete SECU configuration to an open data format,
- Sending specific commands required to manage and/or configure SECU functionalities.

b) Data management:

- Collecting and distributing all the live telemetry data sent by the SECU through the dedicated communication link e.g. remote telemetry or cable-link,
- Offloading, converting and archiving all the data logged by the SECU internal logging functionality, to an open data format.

c) Security and privacy:

- Preventing access to unauthorised users to a predefined set of SECU applications parameters, e.g. between the team and the PU supplier,
- Preventing access to unauthorised users to live telemetry and offloaded SECU data,
- Allowing the FIA unrestricted access to all SECU and third-party units data and configurations.

d) Auditing:

- Providing and exporting a certifiable audit trail of the SECU accesses and configuration changes in a human readable format,

- Verifying and locking part or the complete code and configuration areas of the SECU by an authorised FIA user.

4 Core Requirements

4.1 Reliability and safety

The SECU must provide the highest level of reliability and safety across all its software and hardware components.

It is recommended to follow international quality and safety standards such as ISO 61508 and ISO 26262 for the on-board electrical, electronic and software components. Unless such certification is proven, the proposal should explicitly demonstrate how the PRODUCT matches or exceeds the rules, guidelines and/or characteristics provided in the aforementioned standards.

A key facet of the SECU and its associated tools will be its ability to provide high-level live diagnostics that make it intuitive to use and straightforward to diagnose faults or deficiencies with the on and off-cars systems. This should include but not be limited to fault detection for all inputs, outputs, power supplies, communication lines, control strategies and embedded resources as well as a comprehensive set of KPIs to evaluate the health of all hardware and software components.

The SECU provider must prove that it has completed and that it will maintain an exhaustive FMEA (Failure Mode and Effects Analysis) and FTA (Fault Tree Analysis) for all supplied electronic systems, hardware and software, with a special emphasis on torque generation paths through the internal combustion engine, energy recovery system and brake by wire.

Particular attention should be given to the following elements:

- hardware quality processes, including EMI (Electro-Magnetic Interference), vibration, thermal testing and manufacturing end of line testing,
- compliance with air freight regulations (e.g. for batteries),
- software development process, tools and validation, including formal methods, process and tools used to validate the embedded software (BIOS included) before it is released to customers,
- watchdogs including inter-processor, inter-units, torque controller,
- fault detection and failure mitigation for inputs, outputs, power supplies, communication lines and control strategies,
- monitoring of embedded resources including processor load metrics, memory utilisation, task usage metrics,
- functional redundancies,
- torque and acceleration monitoring,
- stuck throttle safety,
- driver interfaces and warnings,
- formal methods, algorithm and tools to avoid data corruption including protection against division by zero, propagation of NaN (Not-A-Number) signals.

4.2 FIA safety systems

The SECU should manage data communication to and from the FIA safety devices mounted on the cars, including the ADR and F1MS.

This should cover various authenticated data flows between those units and the SECU, including but not be limited to feedback signals that acknowledge the physical activation of marshalling lights on the driver display.

4.3 Regulations compliance

The SECU software will run a comprehensive set of FIA functions used to enforce and monitor the compliance of the car systems with the Formula One regulations. The SECU systems must enable the FIA to monitor those functions in real-time during any track session.

The SECU must not include strategies considered as driver aids. This includes but is not limited to traction control, braking control, launch control, auto clutch, automated gearshift, gear pre-selection and multiple gearshifts sequencing.

The SECU must provide the ability to authenticate or encrypt selected data exchanges between the SECU and peripheral units or sensors.

The SECU must incorporate a data logger with sufficient memory, throughput and triggering capabilities to log key system and strategy parameters in real-time for the duration of a race.

At any time, the FIA must be able to extract an exhaustive view of the SECU state (code, configurations, communications, data).

4.4 Off-car tools

The SECU provider must supply a standard Software Tool Suite that allows configuring the unit exhaustively including the privacy settings, offloading logged data, acquiring and distributing real-time telemetry data and managing all the auditing functionalities.

The SECU provider must also provide a fully documented set of APIs that allows replicating all the functionalities of the standard Software Tool Suite mentioned above as well as visualizing telemetry data in real-time. These APIs may either interact directly with the Master ECU or via a headless service.

Real-time telemetry data viewer and analysis PC Software are not included in the SECU perimeter, but the provider must be able to supply a data viewer suitable for standard use in the Formula One Championship with the SECU on offer.

If the Software Tool Suite and the data viewer do not use the aforementioned set of APIs, the provider should supply an exhaustive list of differences in functionalities and performance.

4.5 Interoperability

The SECU must provide additional expansion capability for third-party units including but not limited to powertrain actuation units.

The SECU must provide additional expansion capability for data acquisition units including but not limited to aero logging and FIA sensors (e.g. fuel flow meter, ERS sensors, tyre pressure sensors and torque meters).

The SECU must provide the ability to authenticate or encrypt selected data exchanges between the SECU and peripheral units or sensors.

5 Hardware

Through the duration of the contract, the provider will manufacture only one hardware build.

Each unit will have a unique serial number marked externally and will be sealed and have its identity tracked throughout its entire life cycle.

Each unit must have suitable provision to prevent tampering. Offers should include anti-tampering solution proposals for each SECU unit, preferably without requiring sticker seals.

Offers should include an overview diagram showing all units of the SECU system, their inter-connections as well as any other connections to external units, sensors and actuators. Teams will build their own control looms based on an approved FIA electrical specification.

When applicable, the provider must supply the following additional documentation for each SECU unit that is part of its proposal:

5.1.1.1 HARDWARE

- **mechanical characteristics**, including dimensions, weight, case material, connector types and complete pinout,
- **installation characteristics**, including mounting points and guidelines, heat sinks, anti-vibration mounts,
- **electrical characteristics**, including supply voltage and current consumption (including in-rush currents peak values), supply protection,
- **environmental characteristics**, including storage temperature, operating temperatures (for processors and circuit boards), operating thermal shock, fluid ingress protection, vibration profile, electromagnetic compatibility,
- **thermal characteristics**, including heat rejection capability and cooling requirements,

5.1.1.2 PROCESSING AND MEMORY

- **processing units characteristics,** including but not limited to microprocessor, microcontroller, FPGAs,
- **memory characteristics**, including but not limited to RAM, Flash, EEPROM,
- **operational characteristics**, including time to boot, reprogram, download calibrations, offload logged data.

5.1.1.3 COMMUNICATION and DATA ACQUISITION

- **communications bus and interface characteristics,** including an exhaustive communication topology diagram between the ECU system and any other on-car and off-car system,
- **internal data acquisition characteristics**, including capacity, maximum number of channels, data throughput, triggering modes, logging modes, typical and maximum data offload speed.

5.1.1.4 INPUTS

- **analogue inputs characteristics**, including type, quantity, circuitry, range, impedance, resolution, accuracy, filtering, sampling rate, protection, diagnostics, configuration options,
- **digital inputs characteristics,** including type, quantity, circuitry, voltage range, impedance, frequency range, resolution, switching/trigger threshold and hysteresis, sampling rate, protection, diagnostics,
- **UEGO inputs characteristics**, including type, quantity, circuitry, range, impedance, resolution, accuracy, current control, filtering, sampling rate, protection, diagnostics, configuration options.

5.1.1.5 **OUTPUTS**

- **analogue outputs characteristics**, including type, quantity, circuitry, range, resolution, impedance, accuracy, control rate, maximum voltage and current, protection, diagnostics,
- **digital outputs characteristics**, including type, quantity, circuitry, voltage and frequency ranges, impedance, current limit, protection, diagnostics,
- **High and Low Side Drivers (HSD and LSD) characteristics**, including type, quantity, circuitry, voltage, frequency range, current limit, protection, diagnostics,
- diagnostic outputs characteristics,
- display characteristics,
- state of all outputs during boot time,
- proposed allocation of inputs and outputs to control and monitoring functions managed by the SECU.

5.1.1.6 **INTERNAL DIAGNOSTICS**

- a list of all internal diagnostics available, including temperatures, accelerations, voltages, currents and the associated error-detection mechanism,
- service intervals and details, life period, quality control.

5.1.1.7 **ELECTROMAGNETIC COMPATIBILITY**

It is recommended to follow international Electromagnetic Compatibility (EMC) Directive 2014/30/EU for all on-board units. Unless the directive is strictly applied including providing all associated documentation, the proposal should explicitly demonstrate how the PRODUCT matches or exceeds the rules, guidelines and/or characteristics provided in the aforementioned directive.

In all cases, providers shall supply a comprehensive documentation of their risk assessment and analysis related to both:

- emissions and,
- susceptibility/immunity.

The former refers to the generation of unwanted electromagnetic energy that may cause disruption to other equipment. The latter will aim at demonstrating that the practises and list of standards fully or partially applied during the design process ensure that each unit will remain fully functional in the harsh electromagnetic environment of a car designed according to the Formula One regulations. This includes, but is not limited to, running in close proximity to an Energy Recovery System operating at high electrical power and high switching frequency.

When the first prototypes are available, they should be duly EMC tested to representative operating conditions that may be defined in conjunction with the teams and PU suppliers.

5.2 Master ECU

The table in Appendix 1 lists a set of target requirements for the Master ECU hardware. Providers are requested to fill in the “Offer specification” and “Reasons for delta to target specification” (if any) columns.

5.2.1 Hardware Interface

The Master ECU shall provide hardware circuitry for:

- generic inputs used on cars designed according to the Formula One regulations,
- generic outputs including but not limited to servo-valve, HSD (High Side Drive), LSD (Low Side Drive),
- spare inputs for additional third-party data acquisition systems.

The Master ECU shall also provide generic power supplies and communication lines to interface the SECU peripheral units, including:

- the Driver Input and Information System comprising the Switch Interface and the Driver Display,
- up to four Wheel Interface Units and,
- some optional Sensor Acquisition units.

The Master ECU shall additionally provide generic power supplies and communication lines to interface the following mandatory external systems:

- FIA Safety Systems such as ADR, F1MS as well as various ancillaries (e.g. ERS, rear impact structure light and rear wing end plate lights),
- FIA Regulated Sensors such as Fuel Flow Meters, Tyre Pressure Monitoring System, ERS sensors and Torque Meters,
- F1 Standard Communication System including the real-time telemetry and voice units,
- FOM on-board camera units, F1 Interface Unit (FIU) and associated transponders.

The Master ECU shall incorporate a data logger with sufficient memory, throughput and triggering capabilities to log key system and strategy parameters in real-time for the duration of a race.

The Master ECU will also act as a gateway to connect to the off-car software suite for SECU and third party units configuration, data acquisition and FIA auditing.

5.2.2 Processing Resources

The Master ECU shall provide enough processing power and memory to be able to run the software required to run cars and power units designed according to the Formula One regulations. The provider shall also provision extra computing power and memory for future evolutions of the regulation as well as cope with the development pace of the teams, the PU suppliers and the FIA.

In order to provide a ground for comparison between the different proposals, all offers shall include the performance of the Master ECU processing architecture at the EEMBC CoreMark® benchmark:

www.eembc.org/coremark

This will provide a more complete and consistent view of the overall performance compared to the somehow too simplistic approach of evaluating MIPS count. It includes but is not limited to the cache hierarchy and the volatile memory. Results at a common benchmark will also allow levelling the field in the case of heterogeneous multi-processor architectures where communication solutions are often not negligible in terms of performance.

Along with an exhaustive description of the SECU processing architecture including the CoreMark® performance, offers shall include an analysis of the strengths and weaknesses of its architecture and associated low-level software, including but not limited to the following key timings:

- Interrupt response,
- High priority task (e.g. 1ms) jitter
- Context switching,
- Task-to-task communication latencies.

5.3 Driver Input and Information System

The Driver Input and Information System is an advanced driver interface aimed at providing information to the driver as commanded by the Master ECU, by visual and audio signals. The system also allows the driver to control configurable inputs to the control strategies via a set of switches, toggles, etc. usually designed by the teams onto the steering wheel. Finally, the Driver Input and Information System should also provide a two-way radio interface used to exchange information between the team garage and the driver.

To provide those functionalities, the system should include the following separate modules:

- a switch interface,
- a driver display that will combine a high resolution and high contrast screen with a set of coloured LEDs,
- Press-To-Talk and tone generation demands, managed from the Master ECU.

5.3.1 Switch Interface Unit

The provider shall supply the Switch Interface Unit as a separate part. The unit must be compact and light so that the teams can integrate it into their steering wheel. The module may be supplied as a raw PCB, assuming the presence of a coating providing reasonable level of mechanical and moisture ingress protection. Only a single instance of this unit will be used in each car.

The Switch Interface Unit should continuously stream its raw signals and internal diagnostics to the Master ECU for use by the control strategies and for logging by the datalogger. Those data must be authenticated to provide a certifiable real-time representation of the driver's actions.

The table in Appendix 1 provides the target requirements for the Switch Interface Unit as well as optional features. Providers are requested to fill in the "Offer specification" and "Reasons for delta to target specification" (if any) columns.

5.3.2 Driver Display Unit

The provider shall supply the Driver Display Unit as a separate part. It shall be a lightweight unit designed to operate in the harsh motorsport environment of an open cockpit car. The unit may be installed into the steering wheel or on the cockpit. It will combine a screen of at least 4.3" diagonal, a set of marshalling reserved colored LEDs (as defined in the Formula One regulations) and a set of configurable colored LEDs. All information should remain fully visible under direct sunlight and at night, with or without artificial lights.

The Driver Display Unit will provide information to the driver, exclusively as commanded by the Master ECU via CAN, either on the screen or via the sets of colored LEDs. The user interface layout of the screen must be fully user-configurable via a licence-free PC software supplied with the SECU. Typical user interface layout used by teams may include text and graphics.

The Master ECU should receive feedback signals to acknowledge the physical activation of each marshalling LED. The master ECU should also continuously log the signals sent to the Driver Display Unit and the diagnostics received from the Driver Display Unit. Those data must be authenticated to provide a certifiable real-time representation of the data sent to and received from the Driver Display Unit.

The Driver Display also generally have high media visibility and as such, providers are encouraged to make use of state-of-the-art display technology.

The table in Appendix 1 provides the target set of requirements for the Driver Display Unit. Providers are requested to fill in the "Offer specification" and "Reasons for delta to target specification" (if any) columns.

5.4 Wheel Interface Unit

The Wheel Interface Unit will act as a remote, compact and lightweight sensor node, equipped with a microprocessor. This tailored sensor node will handle the signal acquisition from a variety of sensors at the wheels and brakes. The data will be packaged and sent to the Master ECU via CAN.

The Wheel Interface Units are likely to be located close to or within the wheel hubs and as such, must be able to withstand the associated harsh environmental conditions (temperature, vibrations) both when the car is running and standing still after a run. The FIA may mandate up to four Wheel Interface Units per car to acquire data from control sensors.

The table in Appendix 1 provides the target requirements for the Wheel Interface Unit as well as optional features. Providers are requested to fill in the "Offer specification" and "Reasons for delta to target specification" (if any) columns.

5.5 Optional peripheral units

5.5.1 LVDT interface unit

A dedicated LVDT (Linear Variable Differential Transformer) interface unit is not included in the SECU perimeter but the provider must be able to supply a LVDT interface unit compatible with the SECU on offer.

The SECU should be able to receive data from a LVDT interface unit via analogue inputs and/or authenticated CAN messages.

5.5.2 Sensor nodes

Sensor nodes are not included in the SECU perimeter but the provider must be able to supply sensor nodes compatible with the SECU on offer.

Sensor nodes will be used as remote, compact and lightweight signal acquisition units. Equipped with a microprocessor the units shall be capable of handling the signal acquisition from a variety of sensors, including CAN sensors. The data will be packaged and sent efficiently to the Master ECU via CAN. Altogether, sensor nodes may be used to reduce the load on the Master ECU CAN buses as well as provide robustness to potential unit failures.

5.6 SECU-in-a-box

The provider shall supply the FIA with a SECU packaged in a box for transportation to competitions and with the following features to allow testing of key functionalities:

- Power supply with cable connection to typical mains supply
- Power on/off switch
- Ignition on/off switch
- D-type connectors for connection to a subset of CAN buses
- RJ45 connector for ethernet connection

The SECU-in-a-box will be available for FIA and for the provider's track support team to validate configurations, procedures and to assist with investigations.

6 Embedded software

The Master ECU will provide real-time algorithms that interface via its electrical input and outputs to control the functions of a Formula One Car and its Power Unit, both designed according to the Formula One regulations.

In practice, the Master ECU will host:

- a base input and output software developed by the provider
- a set of standard control applications developed by the provider
- a set of customer control applications developed by the teams and PU suppliers.

The control applications will be built using the software development toolset supplied by the provider along with the Master ECU.

Along with the source code of the standard control applications, the provider shall supply a complete documentation for all the standard algorithms implemented. Under the direction of the FIA, the provider shall design the standard control applications in a way that achieves the necessary functionality with the minimum complexity.

The embedded software running in the Master ECU shall provide four types of configurable parameters:

- Parameters exclusively configurable by the FIA,
- Parameters exclusively configurable by the teams and FIA,
- Parameters exclusively configurable by the PU suppliers and FIA,
- Parameters that can be configured by all parties.

The SECU should provide all necessary mechanisms to ensure the integrity of its programmed code including but not limited to:

- A mechanism to guarantee that only FIA approved software versions can be run on any of the SECU components and,
- A fool-proof process to ensure that all software contained within the SECU is identical to that validated and approved for use.

The Master ECU should finally provide all necessary mechanisms to ensure the protection and integrity of its logged data, including but not limited to:

- The ability for the FIA to restrict the data logger logging capability and configurability and,
- The ability for the FIA to prevent data erasure during race weekends.

6.1 Application code starting point

From the first release of the SECU application code, the provider will track the current SECU releases and provide development version no later than 1 month after the current SECU release.

In particular, from January 2026 onwards, all official releases of the SECU application code shall cover or exceed the functionalities of the last SECU software released during the course of the 2025 season

The selected provider should seek access to the latest Models from the current SECU provider and shall bare any associated cost. The conditions of access to the application code models by the selected provider are subject to the terms of the agreement between the FIA and the current SECU provider, which will permit a provider contracted by the FIA to:

- Make unlimited copies of the Models;
- Install the Models on an unlimited number of computers;
- Create derivative versions of the Models;

- Interface the Models with alternative hardware platforms;
- Make the Models or derivatives of them available to Competitors.

Licensing of the associated libraries and development tools will be available at the current provider's normal commercial rate.

6.2 Coding Standards

6.2.1 Terminology

In order to provide a common agreement on terms to use, the standard terminology currently in use shall remain. The terms are as much as possible compliant with ASAM standard and MathWorks® Real-Time Workshop for example:

- Parameters such as the gain of a PID are identified as either "parameters" or "CHARACTERISTICS".
- Input/Output (I/O) control variables such as a measured battery voltage or an inverter duty cycle are identified as either "signals" or "MEASUREMENTS".

6.2.2 Naming Convention and Units

The entire parameters set and data dictionary shall comply with the latest version of the FIA-approved naming convention. During the course of the contract, the provider will be in charge of maintaining and extending the standard naming convention document. Changes shall be approved by the FIA before released.

The naming convention notably implies that all signals at the application interface must be defined in functional terms rather than in I/O channel terms. This means using rThrottlePedal in % rather than VRawAnalog001 in Volts. The supplier shall also ensure that all data at the application interface level are expressed in SI or engineering units.

6.3 Master ECU BIOS

6.3.1 Prerequisites

- The Master ECU will support the presence of a BIOS (Basic Input/Output System) and a number of control applications, including but not limited to an FIA application.
- The BIOS must provide memory area protection on a per application basis. In particular, by no means should it be possible for an application to write into any applications' code area. This includes its own data area or that of another application.
- The BIOS shall ensure that applications complete tasks within time boundaries and it shall provide in real-time, comprehensive execution time diagnostics.
- The boot code must be capable of preventing to run unauthorised code or unauthorised parameter data sets. This shall at least be configurable on a per application basis, including the BIOS itself.
- Rebuilding or changing any application should not require rebuilding of other applications (ignoring requirements due to functional changes). This may be achieved by runtime mapping of inter-application parameters and signals.
- The BIOS must include such access control scheme that only permits the FIA to execute a configurable set of restricted commands.

6.3.2 BIOS data access control

The Master ECU must prevent any application from directly accessing I/O signals (MEASUREMENTS). Access to BIOS data and shared data between applications will be controlled in conjunction between the BIOS and the FIA application.

The signals import/export concept shall allow the FIA application to control the flow of data between applications whilst removing the task of moving data between applications and processors from the applications' developers.

For each control application, whether inside or outside the Master ECU, the FIA application will supply the lists of I/O signals that the control application is allowed to access. To supplement this mechanism, the Master ECU should also provide the necessary medium for the FIA application to limit the maximum access rate for each I/O signals.

The BIOS will resolve all the imports and exports according to the FIA application's lists before running the control application initialisation. The BIOS will search the applications in order to find the required signals. The request shall include all the desired attributes such that all attributes should match in order to allow access to the data. In the case where the BIOS does not find a perfect match, the application code will not be executed.

Customer "MATHS" applications will typically be used to post-process data for logging or monitoring purposes. Such applications are only allowed to export signals to the data logger, and for that reason the BIOS and FIA application may grant them access to all other signals.

The BIOS will populate the registered data with the signals required by each control application at the refresh rate given by the minimum between the access rate limit set in the FIA application and the rate required in the control application.

Parameters (CHARACTERISTICS) should be propagated in the background.

If a parameter or signal is exported from a different processor to that on which it is needed, then the BIOS must take responsibility for making it available on any other processor where it may be used.

An application could request to use a parameter or signal that it generates. This would allow the code for that application to be refactored across processors with the BIOS handling the propagation of the parameter or signal values.

6.3.3 BIOS calls

The standard and customer control applications' codes need to be able to call the BIOS to make use of its available services. These entry points should be defined as macros, including but not limited to:

- Configuration for FIA-authorised communications,
- Receive for FIA-authorised communications,
- Transmit for FIA-authorised communications,
- Reset the Master ECU,
- Logging control system events i.e. the combination of a unique identifier, a name, a description, a set of metadata and a priority level, all associated with an accurate date and timestamp.

6.3.4 BIOS services

The BIOS needs to be able to make a number of calls into the application code in order to ask it to perform any tasks. The exact method by which the BIOS makes these calls will be platform dependent. The supplier shall provide the necessary linker files along with the executable image post-processing tools to allow the BIOS to identify the location of the entry points for each application.

The set of application calls include but is not limited to:

- Initialisation,
- Individual task rate calls (1ms, 2ms, 5ms, 10ms, 20ms, 50ms, 100ms, 200ms, 500ms, 1s and an optional 500us),
- Actuator output scheduling with a resolution of 100us,
- Background task,
- Engine synchronous task, always with higher priority than timed tasks.

6.4 Master ECU Resource Sharing and Entities

The current requirements make no assumption regarding the number of processors that will be available in the Master ECU, their interconnections and resources or on the ability of each processor to run application code or BIOS code.

Yet, the Master ECU shall provide a configurable means to share processing and memory resources between the four entities that will require running code on the Master ECU:

- The SECU provider,
- The FIA,
- The team and,
- The PU supplier.

The objectives of such resource-sharing concept on the Master ECU shall include but may not be limited to:

- Increasing the level of flexibility for the development of customer applications,
- Simplifying the resources allocation between the four entities,
- Achieving separation of embedded resources between the four entities.

Ideally, each entity should have access to defined independent resources and as a result should not need to compete for resources when building their applications.

6.5 Master ECU Software Architecture

6.5.1 Assumptions

The current list of requirements assumes that the application code created for the Master ECU will be generated from Simulink models using automatic code generation. Yet it should also be possible to integrate hand coded C if desired.

6.5.2 Requirements

The provider shall specify a single compiler version for building all the applications regardless of the processor they will run on.

Each entity should have the ability to develop its own set of applications. This includes a set of standard applications for the provider and the FIA (BIOS, FIA, Chassis and Arbitrator) and a set of proprietary applications for the teams and PU suppliers.

For the teams and the PU suppliers to be able to integrate the provider's embedded software build process into their own development environment, the provider shall supply build scripts and configurations that are non-obfuscated and fully documented.

Altogether, the Master ECU and its associated toolset should be able to manage at least ten applications, shared between the four entities according to the example below. Three additional control applications could be used to rationalise the code architecture even further or to adapt to regulatory changes.

6.5.3 Virtualization Technology

The governing principle of the SECU architecture aims at simplifying the application code development, deployment and coordination between the four entities. In that context, it is clear that virtualization technology is a key enabler for the next generation of embedded software architectures.

6.5.4 Memory Areas

The provider shall define the implementation details of the memory areas of the application code, providing that the following constraints are respected:

- The code area must be write-protected at runtime,
- Selected code areas may be read and write-protected for non-FIA users,
- The parameter area of an application must be read-only to that application,
- Memory areas of other applications must not be accessible for reading or writing and,
- The Master ECU must support non-volatile memory, i.e. have the ability to retrieve data after having being power cycled.

6.5.5 Customisation definitions

The provider shall declare all C code definitions in header files.

The provider shall supply a header file that fully defines all types and macros and that teams and PU suppliers may include in their custom code. This header file should be robust to multiple inclusions. Nesting of header files will be allowed.

The provider shall supply any additional functions required to support the compiled code in the form of C source code or library files.

6.5.6 Hardware abstraction

6.5.6.1 INPUTS

The Master ECU BIOS shall be responsible for mapping all the hardware inputs to their corresponding functional signals.

The provider shall supply a standard input processing algorithm to handle basic functionalities. These include but are not limited to calibration, zeroing, failure detection and filtering. This should both cover conventional inputs and inbound data from communication lines.

All configurable parameters for input allocation and filter frequency should be in the FIA parameter area.

All configurable parameters for scaling, zeroing and failure detection should be in the team or PU Supplier parameter area.

6.5.6.2 OUTPUTS

The Master ECU BIOS shall be responsible for mapping all the functional signals to their corresponding hardware outputs. This should include but not be limited to conventional outputs and outbound data to communication lines.

All configurable parameters for output allocation shall be in the FIA application parameter area.

Communication lines used for the sole purpose of data acquisition will be unrestricted outbound of the Master ECU. Any inbound data flow on such communication lines will exclusively connect to the Master ECU data logger.

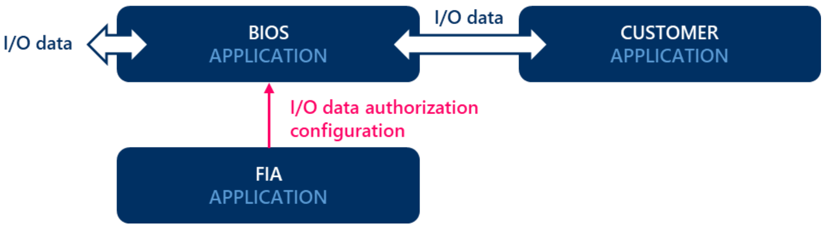
6.5.7 Internal Data Flow Restrictions

6.5.7.1 BIOS ACCESS

The FIA application shall have unrestricted access to all BIOS application data and services.

Conceptually, the FIA application operates as a gateway between the customer applications and the BIOS. In particular, all the customer control applications I/O activities to and from the BIOS must be approved by the FIA application. This shall provide an unforgeable mean to monitor and control data access.

In practice, the FIA application hosts the configurations used by the BIOS to allow or restrict access to its data and services, to the customer control applications.



6.5.7.2 WHITELIST CONCEPT

The Master ECU shall provide a mechanic for the FIA to control the data discovery and exchange between all the different control applications. To achieve that, the provider must supply an intuitive toolset to setup and monitor a parameter whitelist that will be configured in the Master ECU on a competition per competition basis.

The data flow controlled by such whitelist include but may not be limited to:

- Data exchanges between customer apps of different entities,
- Data exchanges between customer apps within a given entity,
- Data exchanges between customer apps and the BIOS.

At any point during a competition, the FIA should be able to monitor in live and in an unequivocal fashion, the content of the whitelist in use in a given Master ECU.

The FIA shall also be the only entity capable of updating the whitelist through the course of a competition.

6.5.8 Simulink libraries and code generation

In order to streamline development processes for the teams, the SECU libraries should be compatible with the MathWorks tool suite including but not limited to advanced model hierarchy (model referencing, variant systems), dictionaries and Simulink Project Management.

Rebuilding or changing any application should not require rebuilding other applications (ignoring requirements due to functional changes). The proposal should include an efficient method to rebuild a subset of applications.

6.5.8.1 COMPATIBILITY WITH MATHWORKS SIMULINK

For the development environment that will be used to build embedded software for the SECU, the supplier shall:

- Maximise the use of Simulink standard libraries,
- Build upon Simulink standard blocks as much as possible for any additional tools that may be required and,
- Make the use of proprietary elements only to interface the BIOS or BIOS-related functionalities.

The Simulink libraries shall be made fully compatible no later than one year after any annual MathWorks releases. As such, the SECU libraries used in year N will use the latest release of year N-1.

- It shall be possible to simulate the source code in Simulink with no modifications,
- The provider shall supply the necessary software in order to run the Master ECU control application on a PC, using historical data or against live data. The application programme image shall be the exact one that is downloaded to the Master ECU.
- It should be possible to compile the applications to be able to run on a PC and simulate ECU use when provided with suitable plant models. This system should produce data and be calibrated in the same way as the SECU. In order to provide a sufficient number of applications for teams to develop plant models the number of applications supported should be at least 50% more than the ECU.
- It shall be possible to browse the source models with Simulink (e.g. xml format).

6.5.8.2 PROVIDER LIBRARY

The provider's Simulink library will cover the following areas:

- Access to BIOS Input and Output signals,
- Signal and parameter imports from other applications,
- Signal exports to other applications,
- Signal exports for logging and live monitoring purposes,
- Commonly used blocks requiring configuration parameters,
- Access to BIOS functions including but not limited to events logging, task calls, ECU reset and handling of non-control communication lines,
- List of signals allowed to be imported by an application, including the maximum import rate,
- List of signals allowed to be exported by an application, including the maximum export rate.

Versions and releases of the SECU libraries will require approval of the FIA.

6.6 FIA Specific Requirements

6.6.1 FIA Application

The provider shall develop the FIA application as a Simulink model.

The FIA application shall include a set of algorithms designed to enforce and monitor the compliance of the cars with the Formula One Technical and Sporting regulations and to support the operation of FIA safety systems.

The FIA application will notably be responsible for monitoring or controlling the following systems:

- Internal Combustion Engine,
- Energy Recovery Systems,
- gearbox,
- differential,
- clutch,
- Brake-By-Wire systems,
- fuel system,
- adjustable bodywork,
- driver interfaces including switches and displays,
- safety and virtual safety car periods,
- F1MS,
- ADR.

Power/actuation and peripheral units will transfer signals to the Master ECU as required by the FIA to allow legality monitoring.

The provider must supply the source code and documentation of any new FIA Application version prior to its release. The FIA will approve the release upon presentation of the software validation documentation.

6.6.2 FIA Scrutineering

To support the FIA scrutineering process, the provider will design the SECU so that the FIA can obtain, at any time throughout a competition, a comprehensive, exhaustive and indisputable run-through all the SECU data and configuration.

In particular, before, during and after any session, the FIA should be capable of obtaining:

- The complete configuration of each SECU application,
- All real-time telemetry data, diagnostics and events (including those of compatible third-party units),
- All data, diagnostics and events (including those of compatible third-party units) logged on the Master ECU.

During a competition, the FIA must have the ability to connect to the Master ECU via a jump battery and an external laptop at all-time.

The provider shall implement a robust mechanism to ensure that, during a competition, the Master ECU memory containing logged data, diagnostics and events may only be cleared by the FIA.

6.6.3 Software Version Verification

All programmable devices involved in the SECU systems must have a mechanism that allows indisputably identifying the software versions in use.

Typical examples include but are not limited to “program image” verification, upload of “program image”, secure hash algorithm, e.g. MD5, SHA-256 of the “program image”. Note that the FIA would not deem sufficient mechanisms such as a simple version string communicated by the device.

The supplier shall select and implement an appropriate combination of mechanisms that may include one or a combination of the following approaches:

(A) Boot code programming of any electronic unit provided to the teams should only be possible using a production process from the supplier and require opening the unit.

(B) A secure hash logging methodology may be suitable; subject to using an FIA supplied random filler sequence. Any microprocessor with access to non-code memory of sufficient size to store additional program images may not be considered suitable for this method.

All units that can be reprogrammed via an external connector and/or that are connected to an Electronic Control Unit, as defined by the FIA, are expected to provide this mechanism.

(C) Reprogramming of electronic units must be restricted by a homologated mechanism that has been approved before the electronic unit is first used. The following is a non-exhaustive list of possible techniques:

- Depending on the design, FIA sticker or stickers may act as a seal e.g. where the reprogramming is via an internal connector or where the sticker seals some hatch that gives access to reprogramming,
- Password protection,
- Application list restriction: typically, a list of secure hashes, e.g. MD5s, SHA-256s, of all the allowed program versions that the unit can run. Updating the list must require knowledge of an FIA secret password. Any application code area should have a fixed size for each version.
- Signed application code. An FIA private encryption key must be securely stored in each unit. The code will not run unless it has been digitally signed by the FIA.

The provider shall not implement any security 'fuses' or other such mechanisms that prevents access to stored 'images'.

7 SECU Data Logger

Along with the Master ECU, the provider shall supply a data logger responsible for logging in real-time all data transiting on the unit, all events generated by the application code (including BIOS) and all configurable parameters. The Master ECU shall also be able to seamlessly manage logging from third-party units connected to the data logger. No separate partition between the data and events of each entity and/or unit is required.

The SECU software suite shall provide intuitive and comprehensive configuration capabilities as well as support individual logging configurations from each of the entities, both for data generated by the SECU and by third-party units connected to the SECU.

In order to provide sufficient level of data scrutineering, some of the data logger functionalities shall only be enabled with FIA privileges.

The table in Appendix 1 lists target requirements for the Master ECU data logger as well as optional features. Providers are requested to fill in the "Offer specification" and "Reasons for delta to target specification" (if any) columns.

7.1 Data logger configuration

All logging configurations shall be stored in an open and intelligible data format including but not limited to XML or JSON.

The SECU shall make available in real-time and via the SECU software tool suite, a comprehensive set of diagnostics for the logger itself and each individual entity's logging configuration.

The SECU shall provide a mechanism that is either open source or based on open standards to manage third-party units such that a modification of the logging configuration does not require any code change in the SECU or in the third-party unit.

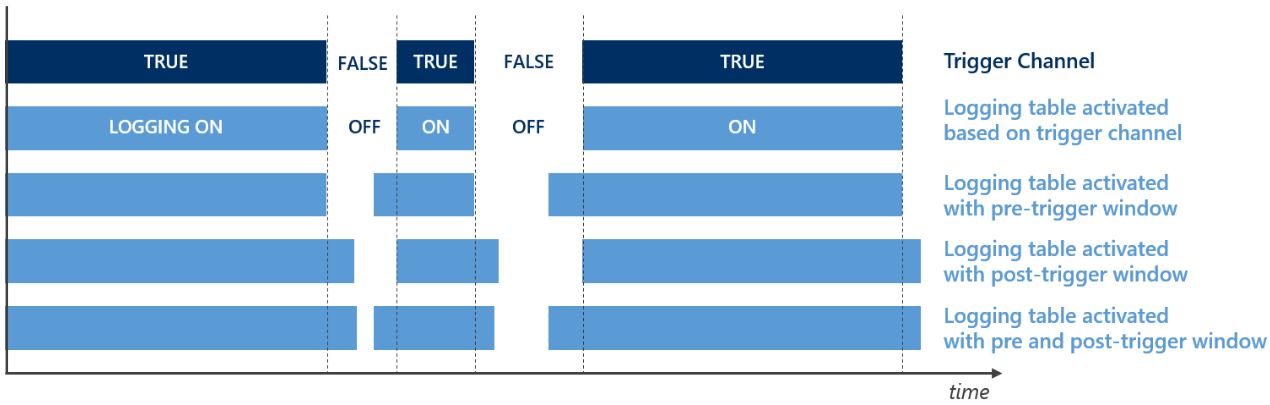
Setting up the logging configuration of third-party units shall be possible using the SECU Software Tool Suite or its programming interface (see section 9). Such third-party unit logging configurations shall be transmitted to compatible third-party units via the Master ECU.

7.2 Logging tables

The data logger and associated software tool suite shall provide the concept of logging tables that specifies for each individual signal the logging rate for both live telemetry and on-board logging.

The data logger and associated software tool suite shall be able to:

- Manage a set of separate logging tables for each entity. In case of parameter duplication across logging tables, the highest logging rate should supersede,
- Support the concept of trigger to activate or deactivate each individual logging tables,
- Support the concept of logging configuration managed directly by the application software,
- Support triggers based on the different states of the SECU including but not limited to "Power On", "Power Off", "Lap trigger detected" or any event triggered by the application code.
- Support triggers based on watching a combination of internal or virtual channels or events. For that purpose, the Master ECU shall support in excess of 32 such triggers e.g., "when {vCar>5kph}" or "when {rThrottlePedal>20%} AND {nEngine>9,000 rpm}",
- Provide a "post-trigger" mechanism i.e., upon detection of a falling edge of a trigger, the associated logging table remains active for a configurable amount of time after the trigger becomes inactive,
- Provide a "pre-trigger" mechanism so that upon detection of a rising edge of a trigger, the logging table is retrospectively taken into account for a given amount of time before the trigger became active.



7.3 Time Synchronization

The goal here will be to facilitate merging of data sets from diverse on and off-car sources. For this to work well it is essential that each data source is time synchronized.

In particular, the Master ECU should support Network Time Protocol (NTP) capable of achieving at least 1ms synchronisation against absolute time provided by an external time server.

7.4 Data logger performance

- The data logger shall be capable of logging, at a configurable high rate, in excess of 4,000 data channels besides all configurable parameters,
- Regardless of any configuration table, the data logger shall log all channels at a minimum configurable rate of 0.2Hz, defined by the FIA,
- All data (low and high rate) should have an accurate 64bit timestamp.
- Offers should detail the logging processes, including timestamping, synchronisation relative to other data and task execution.
- The data logger memory should be configured to wrap around by default. The FIA, via its application, should be able to limit the usable memory.
- When a connection to the Master ECU is available, the SECU software tool suite should automatically supplement the data received via telemetry with the data available in the logger memory to fill in any missing data.

7.5 Scrutineering

- When locked by the FIA, the Master ECU logger data area should only be cleared using FIA privileges,
- When the unit is locked, the FIA shall be the only entity capable of clearing the Master ECU events. The events should be stored in a cyclic buffer in the Master ECU until manually cleared. Note that this shall not prevent the user or the FIA to clear the events list displayed in the PC software tool suite, yet such action shall not clear the associated memory in the unit.

8 Interoperability with peripheral systems

Communication with the peripheral systems will use a combination of Ethernet T1, CAN 2.0 and CAN FD. Refer to Appendix 1 for the associated requirements.

8.1 FIA Safety Systems: ADR and F1MS

The SECU shall provide a communication interface to and from the FIA Safety Systems units. This includes the Accident Data Recorder (ADR) and the F1 Marshalling System (F1MS) units as well as their ancillaries.

- The SECU shall provide a mechanism to configure a set of dedicated CAN lines for connecting the Master ECU to the ADR, the F1MS and their ancillaries,
- The CAN messages to the ADR and F1MS unit shall only be enabled with FIA privileges,
- The SECU shall provide a mechanism, also exclusively accessible with FIA privileges, to lock the associated CAN configurations,
- The Master ECU shall have the processing capacity to view all CAN messages without overrun on CAN buses loaded up to 90%,
- All communication messages to and from FIA safety systems should be authenticated using a method approved by the FIA.

8.2 FIA Regulated Sensors

FIA sensors used for regulatory purposes may include, but are not limited to, Fuel Flow Meters, ERS sensors, Tyre Pressure Measurement System, pressure sensors, temperature sensors and torque meters.

- The SECU must provide an exhaustive yet configurable interface to and from regulatory sensing devices. This may include but not limited to communication bus, analogue and digital inputs and outputs and power supplies.
- The provider shall supply a mechanism so that it is possible to lock the configuration of this interface, using FIA privileges.
- Communications to and from selected FIA sensors should be authenticated and, in some cases, fully encrypted for privacy, using methods approved by the FIA.

8.3 Standard Car to Team Telemetry System

The SECU must provide a communication interface to and from the Standard Car to Team Telemetry System. This includes but is not limited to CAN and Ethernet communications to and from the units of the Standard Car to Team Telemetry System.

8.4 Formula One™ Systems

The SECU must provide a communication interface to and from the Formula One™ Systems. This include but not limited to CAN communication to and from the F1 Interface Unit (FIU).

The SECU should support at least two independent CAN configurations provided by:

- the FIA for CAN messages used in control strategies (e.g. timing loop identifiers) and,
- the Teams for outbound CAN messages used by the FOM broadcast network.

The provider shall provide a mechanism so that it is possible to lock the configuration of this interface, using FIA privileges.

Communications to and from the Formula One™ Systems may be authenticated and, in some cases, fully encrypted for privacy, using methods approved by the FIA.

8.5 Third-Party Units

The SECU shall interface with third-party units including but not limited to team data loggers, sensor nodes and proprietary Electronic Control Units. The aim here shall be to provide a seamless interoperability between the third-party units and the SECU calibration system, via the Master ECU.

Upon compatibility of the third-party unit to an agreed communication protocol that is either open source or based on open standards, it should be possible, directly through the SECU software tool suite, to dynamically:

- programme, verify and read third party units embedded code images,
- programme, verify and read third party units parameters configurations,
- live-tune third party units configurable parameters,
- programme, monitor and configure third-party units logging configurations,
- monitor and display third party data in real-time.

The Master ECU shall support the associated full-duplex data flow.

As part of the tender proposal, potential providers shall supply a complete documentation regarding the selected protocol and its implementation in the SECU paradigm. The provider shall also identify a list of strengths and weaknesses along with clear arguments for how this proposal may comply with the following key requirements:

- scalability of features,
- minimal resource utilisation on the SECU and third-party units,
- efficient communication,
- plug-and-play configuration with minimal number of parameters,
- compatibility with existing standards,
- ability to synchronise timestamps across units for data acquisition.

8.6 Configurable CAN

In order to maximise the versatility of the SECU and its interoperability with other ECUs that run on a Formula 1 car, the Master ECU shall provide a CAN interface that is configurable at runtime.

In particular, it shall be possible to construct a CAN message from one of the signals available on the SECU and send it over a selected CAN bus without rebuilding any application (ignoring requirements due to functional changes).

It shall also be possible to configure the reception of a particular CAN message on a selected CAN bus in order to create the associated signals. Those signals shall then become available in the Master ECU as any other signals but must be limited for logging purpose only. Similarly, this shall be available without rebuilding any application (ignoring requirements due to functional changes).

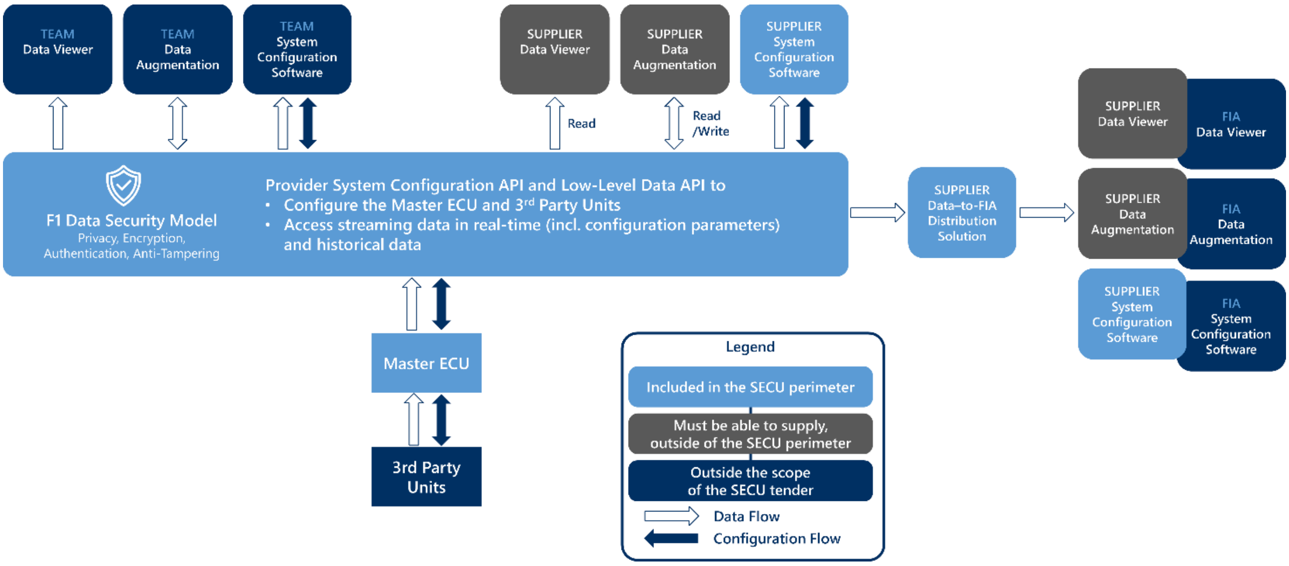
The configuration of such an interface shall be possible using the SECU Software Tool Suite or its programming interface (see section 9). As for to the logging configuration, these CAN configurations shall be stored in an open and intelligible data format including but not limited to XML or JSON.

9 SECU Software Tool Suite

9.1 Concept and Objectives

It is well known that the challenging standards of Formula 1 is achieved through detailed analysis of data and streamlined processes that make the engineering tasks as efficient as possible, both at the track or on a test bench, dyno, rig. Therefore, it is crucial that the onboard systems are supported with software that have the necessary functionalities, versatility, scalability and performance.

Equally, while Formula 1 may lead in its proximity to ultimate performance, it also has to be able to adopt developments in data technologies that are evolving through many other industries. The last decade has notably seen significant growth in the distribution, storage, and analysis of time series data, both on premise and in the Cloud. The SECU should enable the various stakeholders to adopt these technologies by streamlining the access to its data, thus supporting efforts aimed at harnessing the latest innovations in the field of data visualization and large-scale data analysis to drive performance.



Streamlining the access to data should be achieved by providing a set of Applications Programming Interfaces (APIs). The above diagram shows how these APIs may be integrated within a typical trackside Formula 1 data platform. Note that the various software applications and tools interfacing with the Master ECU through this set of APIs are expected to be either supplied by the provider, developed in-house by the various stakeholders or subcontracted to third-parties.

To support in-house or third-party development, it is essential that the APIs supplied by the provider are exhaustive, documented and publicly available. The APIs implementation must be adequately documented including technical notes, tutorials and sample code aimed at popularizing their usage.

The APIs should provide a programmatic access to all the telemetry data streamed or stored on the Master ECU as well as the configuration parameters, streamed or stored on the Master ECU. Where applicable, the APIs should provide a real-time access to the data stream. Through this set of APIs, it should also be possible to configure the SECU exhaustively.

In all cases, it is a key requirement all the APIs mentioned above have no dependencies with other PC application software. In the case where this would not be possible - for example would a dedicated gateway service be required to interface with the SECU - such service shall be headless and an integral part of the offer. Beyond the term of the contract, it should be supplied through a perpetual licence, at no extra charge.

9.2 SECU Software Tool Suite Components

The SECU Software Tool Suite includes, within the perimeter of this tender, the following elements:

- A System Configuration Software in the form of a traditional Microsoft Windows PC application with a Graphical User Interface that allows exhaustively configuring the SECU,
- A System Configuration Service and its API that exhaustively replicates the commands available in the System Configuration Software PC application. Such service may be hosted directly on the Master ECU allowing a direct connection. If not hosted in the Master ECU itself, the service should be headless,
- One or more Data Server Services capable of broadcasting the Master ECU telemetry stream either received via Ethernet or at the Wireless Telemetry Team End Point (TEP). It should be possible to simultaneously broadcast in real-time, the data stream on a local network (typically at the track) and to *at least* 2 remote locations accessible over the internet. Typical use-cases of the latter are Teams and PU suppliers so-called Mission Control Rooms. It should be possible to daisy-chain the services when covering multiple remote locations, to minimize bandwidth usage when at the track. It is also likely that one of this Data Server Services will be responsible for exposing the Low-Level Data API described below.
- A Low-Level Data API providing real-time access to all configuration parameters and timeseries data contained in the telemetry stream available via Ethernet, at the Wireless Telemetry TEP or stored in the Master ECU. This will enable in-house or third-parties to develop custom software applications for managing the configuration as well as obtaining, recording, analysing and displaying both live and historical data for both the SECU and third-party units.

Outside the perimeter of the SECU tender, the provider shall also be able to supply:

- A state-of-the-art telemetry Data Viewer in the form of a standalone Microsoft Windows PC application including a Graphical User Interface.

In all cases, would a specific hardware and/or software data infrastructure be required to support any of the above items, the provider shall make available, as part of its offer, a detailed specification that the FIA, teams and PU supplier can implement.

9.3 Scenarios

The SECU Software Tool Suite must cover a wide variety of use cases both in terms of PC Application software as well as the various associated APIs. Suppliers are requested to explain how the following key use-cases are managed:

- The Software Tool Suite and its APIs shall allow a direct connection from a PC to the Master ECU via an Ethernet connection. This scenario will notably be used for debugging/testing at the track or by the FIA in order perform its scrutineering operations including but not limited to locking/unlock units and offloading configuration and timeseries telemetry data stored in the Master ECU.
- The Software Tool Suite and its APIs shall also be compatible with scenarios where multiple users interact with multiple SECU systems on the networks, either for configuring or monitoring purposes.

Note that both scenarios may depict standard operations at the dyno, on a test bench or at the track, all for which the SECU Software Tool Suite and its APIs shall be equally suitable.

A more exhaustive inventory of use cases, along with the associated interface available, can be found in the table below. Suppliers are requested to detail for each use case, what will be the hardware and software data infrastructure required to access the full functionality of the PC software applications (the supplier System Configuration Software and Data Viewer) as well as the associated set of APIs (System Configuration API and Low-Level Data API).

	Scenario	Interface available
Live data	Team at the track	Wireless Telemetry End Point (TEP), Ethernet or both
	FIA at the track	Wireless Telemetry End Point (TEP), Ethernet or both
	Team Remotely	Supplier Data Server Services
	FIA Remotely	Supplier Data Server Services
	Hardware in the Loop (HiL)	Ethernet local network
	Driver in the Loop (DiL)	Ethernet local network
	Software in the Loop (SiL)	Ethernet local network
	Bench	Ethernet local network
	Dynos	Ethernet local network
	Laptop to Master ECU	Ethernet local network

9.4 Environment

The SECU Software Tool Suite will typically be used in a Motorsport environment by the FIA, the teams and PU suppliers, at the track, while travelling or back at their respective base. Altogether this is an eclectic collection of IT resources and infrastructures, PC hardware and software which with the SECU Software Tool Suite must be designed to operate, without compromising the User Experience (UX).

This may include but not be limited to PC laptops with limited processing power and memory, no discrete GPU, limited or non-existent internet bandwidth and high-latency network. On the other side of the spectrum the SECU Software Tool Suite shall also make the best possible use of contemporary hardware which may include but not be limited to displays with high pixel density, multi-screen display configuration and touch screen monitors.

9.5 SECU Software Tool Suite Functionalities

The provider shall supply a standard Software Tool Suite that allows managing the configuration as well as monitoring and analysing both live and historical data both for the SECU and for third-party units using a compatible interface.

The typical use cases involved in the day-to-day activities of the FIA, the teams and the PU suppliers include but are not limited to:

- Editing, viewing, comparing and archiving parameter datasets for applications,
- Splitting parameter datasets for team, PU supplier and FIA parameters, including configuring privacy requirements,
- Managing multi-user sub-system parameter datasets, including each other's privacy requirements,
- Recording configuration changes in intelligible log files,
- Calibrating sensors and actuators with routine pre-defined semi-automated processes,
- Loading application code and configuration for all or individual applications,
- Verifying and locking all or a subset of the code and configuration areas, using FIA privileges,
- Monitoring in real-time all the channels generated in the SECU and third-party units via cable link to the Master ECU, within the limits of the privacy configuration,
- Setting up the logging configuration of channels generated in the SECU and third-party units that will be sent via telemetry, including configuring privacy requirements,

- Setting up the logging configuration of channels generated in the SECU and third-party units that will be logged in the internal memory of the Master ECU data logger, including configuring privacy requirements,
- Offloading all or a subset of the data logged in the internal memory of the Master ECU data logger, via cable link to the Master ECU,
- Converting all or a subset of historical data to an FIA format,
- Communication with dynamometer “Slew Box”.

The proposed standard Software Tool Suite must provide an API that will allow the teams, the PU suppliers and the FIA to plug bespoke and/or third-party software tools for the units’ configuration, data visualisation, data analysis and/or data augmentation.

Providers are encouraged to use the exact same APIs, throughout the Software Tool Suite, in order to guarantee full support and an uncompromised consistency in terms of raw performance, feature set, security and privacy. Whenever the aforementioned set of APIs is not used, the provider should supply an exhaustive list of differences in functionalities and performance.

The provider shall also distribute the necessary updates to the associated APIs whenever the Software Tool Suite is updated, throughout the duration of the contract.

9.6 Development and testing

From the track to the design offices and the dynos (where the largest amount of data is generated), the extensive list of scenarios requires a continuous software development roadmap, supported by thorough testing capabilities and processes. Potential providers are thus invited to detail the quality processes that will be put in place as well as provide strong evidence of their effectiveness.

9.7 Releases

Versions and releases of the SECU Software Tool Suite and its APIs will require approval of the FIA.

9.8 Privacy

The SECU Software Tool Suite and its APIs shall be able to setup and enforce restrictions to prevent entities from accessing all or a subset of the data, configurations or program of the SECU and/or third-party units.

In particular, the Software Tool Suite shall allow managing the following use cases:

- Preventing access to unauthorised entities to a predefined set of configuration parameters, e.g. in the particular case of a team and PU supplier protecting their respective IP while running their respective customer applications on the same hardware,
- Preventing access to unauthorised users to all or a subset of live telemetry and/or offloaded ECU data,
- Allowing the FIA unrestricted access to all SECU and third-party units data and configurations.

Using the same authentication process, the SECU Software Tool Suite and its APIs shall also provide the ability to setup and enforce restrictions to access certain functionalities. This may include but is not limited to preventing access or the use of certain specific SECU ad hoc commands to a specific user or group of users.

In all cases, the FIA shall have unrestricted access to all the SECU Software Tool Suite functionalities. The access to some of these functionalities will be disable by default to all entities but the FIA. This includes but may not be limited to locking ECU applications, clearing data and event memory in the Master ECU and data logger.

9.9 System Configuration Software

The provider shall supply a standard software to interface with the SECU, its data logger and third-party units, namely the System Configuration Software.

9.9.1 Complexity abstraction

In all cases, both the System Configuration Software and the associated API shall abstract the complexity of the underlying electronic architecture for its user. Typical hidden details of implementation include but are not limited to:

- The fact that communications to the electronic units might be serialized and exclusive shall be abstracted by a sense of parallelism where the requests are all accepted and queued,
- The fact that there are several electronic units with their own memory implementation details and communication protocols.

9.9.2 Main functionalities

For the SECU and any third-party unit using a compatible interface, the typical use cases for the System Configuration Software and API include but are not limited to:

- Creating, loading all or a subset of the project configuration,
- Querying, visualizing and exporting status information about the units including but not limited to the state of the connection to the unit and the firmware and software code versions,
- Querying, visualizing, exporting a list of measurements and/or parameters along with their associated characteristics, available in the Master ECU, the data logger and/or third-party units,
- Querying, visualizing, exporting the values of selected measurement or parameters,
- Querying, visualizing, modifying, importing, exporting or comparing of parameters against a historical configuration,
- Importing and exporting parameter data-sets using an open file format e.g. MATLAB m-file, XML, JSON,
- Querying, visualizing status, events, error or miscellaneous diagnosis,
- Creating virtual channels that subsequently act as any other channel generated in the application code,
- Sending ad hoc commands to enable specific functionalities, notably switching or executing test, routines and calibration or zeroing procedures,
- Uploading, downloading firmware and software,
- Reading log files,
- Locking and unlocking sets of measurements and parameters to disable or enable the read or write access to other entities,
- Implementing a RBAC (Rules Base Access Control) to police the System Configuration API usage with the use of an authentication mechanism based on entities,
- Creating a backup or an archive capable of restoring the complete configuration of the system on a car or on a bench, notably to quickly swap the hardware while preserving the software configuration,
- Managing logging tables for the SECU and third-party units, including but not limited to loading, saving, comparing, editing, querying from the unit and downloading to the unit,
- Overriding logging triggers to force logging on or off,
- Managing logged data and events memory area on the Master ECU and the data logger using FIA privileges when appropriate,
- Verifying and locking all or a subset of the code and configuration areas using FIA privileges.

9.9.3 System Configuration Software User Interface

Within the System Configuration Software the user shall be able to create a set of custom live dashboards that gather both parameters and measurements from the Master ECU, the data logger and/or third-party units. The parameters shall be editable and by default, display the current configuration of units. There should be a clear mechanism that highlights changes that have not yet been downloaded to the units. The measurements shall display their current value in real-time.

An effective User Interface shall let the users customise the data they see on their dashboard, along with the layout. The User Experience (UX) design shall focus on allowing users to interact with the data displayed in the dashboard. In particular, to construct the dashboards layout, the users shall have access to a large variety of UI (User Interface) elements within a library of controls, informational components and *containers.*

9.9.4 System Configuration Service and its Application Programming Interface

The FIA, Teams and PU Suppliers would like to program and configure the master ECU and its third-party companion units via a dedicated Application Programming Interface (API) which goal should be to:

- Facilitate all ECU management and configuration operations required for racing , dyno and FIA scrutineering operation via a headless service API,
- Easy to integrate with 3rd party systems,
- Compliant with the Formula One data security and privacy requirements described above.

Once the API is available and documented, Teams, PU Suppliers and the FIA will be responsible for interfacing their own software to the provider's interface.

9.9.4.1 API Environment

The API must support remote access via technologies such as REST (REpresentational State Transfer) APIs and allow for multiple users to access different parts of an ECU simultaneously.

It is highly likely that the API will need to be backed up by a stateful service that manages the orderly configuration tasks of the SECU and requests via the API. Such service should be headless.

The API should support two modes of operation:

1) Offline where the configuration for an app is first built up and then manually downloaded to the car or,

2) Online where each configuration change is downloaded and takes effect immediately.

9.9.4.2 API Functionalities

Below is a list of likely essential functionalities required to manage and configure the Master ECU.

Many of the methods need to pass large amounts of data back and forth, in these cases an open standard such as JSON should be used.

9.9.4.2.1 Establishing a Connection

There needs to be a method for opening a connection to the Master ECU that ensures only users with the correct credentials are allowed to access the Master ECU and any associated resources.

9.9.4.2.2 Get/Set Connection Status and Run State

There should be methods to provide the current status of the ECUs and to reset or clear any alerts that might require acknowledgement.

9.9.4.2.3 Project Management

The complete configuration of the SECU comprises of many documents (e.g. application code, configuration parameters and CAN configuration), methods are required to create, add, remove, delete, upload and download (program) each of them to/from the SECU.

9.9.4.2.4 Compare and Merge

Each document type (configuration parameter values, logging configuration, CAN definition…) should be supported with merge and compare capabilities. Where merge means taking a data version from one program and applying it or a subset to another program version possibly with different configuration parameters. in the eventuality where there are failure of a merge, the delta of what was achieved and what was requested should be explicitly returned.

9.9.4.2.5 List Parameter Names and Properties

Methods to list all measurement and configuration parameters and their properties.

9.9.4.2.6 Get/Set Configuration parameter values

Methods should be provided to read and write the values of configuration parameters in engineering units. Where parameters are multidimensional the associated axes breakpoints should also be get/settable.

9.9.4.2.7 Program code and data by application or everything

There should be methods to program individual parts of the SECU or to fully reprogram everything necessary to bring the units into a runnable state.

9.9.4.2.8 Create/download/upload/unpack CAN configuration

The data expected to be received or sent on any customizable on-car comms interface such as CAN must be definable through the API. The API should provide methods to build and decode such configuration or use an open standard.

9.9.4.2.9 Define Virtual Channels

Virtual Channels are user defined channels typically to characterize a guest sensor or create a new parameter based on existing parameters. There should be methods for defining and managing virtual channels so that Virtual channels can be added to the logging configuration.

9.9.4.2.10 Get/Set Parameter Unlock List for applications

The API should support configuration of which parameters are visible to other entities. This might be achieved by making the default for every parameter to be hidden, then only the parameters listed in a configurable unlock list are made to be visible to the relevant entity.

9.9.4.2.11 Create/download/upload/unpack Logging configuration

The logger within the Master ECU should be fully configurable via the API. This includes creating a logging configuration, adding parameters, setting their rates and the trigger conditions for the logging tables. The API should provide methods to build and decode such configuration or use an open standard.

9.9.4.2.12 Unit Lock/Unlock

Assuming the application has sufficient rights methods, the API should allow the unit to be locked or unlocked.

9.9.4.2.13 Special Message

If the ECU has featured controlled by special messages, there should be an interface to send these messages and return any feedback. Feedback should include the results of the command as well as the success of delivery.

9.9.5 Project and session

The System Configuration Software and its API shall provide the concept of “SECU Project” which allows saving the complete configuration of an ECU in an open and intelligible format. The SECU Project repository will only contain a reference to the different information that altogether constitute an exhaustive configuration of the SECU and third-party units. The SECU Project should notably reference the program versions of all the Master ECU and third-party units applications, all the associated parameters configurations, the logging configurations, the CAN configurations and the virtual parameters settings.

The System Configuration Software and its API shall provide the concept of "SECU Project Export" which encapsulates the SECU Project along with all the actual information it references into a single archive.

Finally, the System Configuration Software and its API shall also provide the concept of "Session" which allows saving the state of the software in an open and intelligible format. A session will gather all the required references to the information required to restore the state of the software including the SECU Project, the user settings and the User Interface layout.

Proposals should detail the intended format(s) and types of repositories supported, e.g. file system, database.

9.10 Data Management

The provider shall supply an API to manage the SECU and third-party units data, including obtaining and distributing real-time data streams as well as storing and reading historical data:

- Live data may be obtained in real-time either via a direct Ethernet connection or downstream of the Standard Car to Team Telemetry System. The latter is responsible for transporting the data from the on-car systems to the teams' garages where a Telemetry Entry Point (TEP) allows to receive the telemetry data stream. Whatever software, gateway services or API library may be required to connect, record or distribute this live data stream to multiple listeners, it shall be an integral part of the offer.
- Historical data may either be stored in an open or a proprietary data format. While an open file format is encouraged, if the provider elects to use a proprietary format, the teams, PU suppliers and the FIA shall be able to use the same API to save/load historical data to/from such a proprietary format.

Upon being received, live data should be recorded into the historical data format at regular interval. This interval should be configurable and have the ability to be triggered, among others, by a condition built up from the telemetry data channels themselves. Typical use cases would be to make the historical file available once a sector or a lap is completed, when a particular condition has been met or after a configurable duration (e.g. every minute). This will naturally give birth to a partitioned data set that can be ingested by quasi-real-time data analytics pipeline.

Built on top of this API, yet not included in the SECU perimeter, the provider shall also be able to supply a state-of-the-art Data Viewer solution to display and analyse the SECU and third-party units real-time and historical data. Except for its Graphical User Interface, the Data Viewer is encouraged to exclusively rely on the use of the Low-Level Data APIs that the provider supplies as part of its offer. If that is not the case, the provider should supply an exhaustive list of differences in functionalities and performance.

9.10.1 Provider data viewing and storage

Despite the supply of an API allowing the development of alternative solutions, in the context of this tender, the provider data viewing and storage solutions shall be considered as the de facto solutions to view, analyse and store both live and historical data.

Consequently, and even though such data viewing and storage solutions are outside of the SECU perimeter, the proposal shall include a comprehensive documentation of such solutions. This may include but not be limited to the elements highlighted in the following sub-sections.

9.10.1.1 Data Viewer

The Data Viewer shall take the form of a standalone Microsoft Windows PC application capable of displaying the SECU and third-party units real-time and historical data recorded in the provider format. The software shall also support typical motorsport data analysis. Such analysis may include but not be limited to basic statistical calculations, lap-based data aggregation and comparative data overlay.

In all use cases, robustness, flexibility and speed are on the top of the list for expert data analysts operating either at the track, in a design office, on dynos or on a bench. For those users, the Data Viewer shall:

- Help streamline their workflow to strengthen their operational efficiency and,
- Facilitate the access to data to carry out advanced data analysis on an ad hoc basis.

The Data Viewer shall also satisfy the needs of another group of less intensive users, distinguished by a smaller frequency of use of this PC application. Nevertheless, enabling them to access and work with the data is key for the teams, the PU suppliers and the FIA operations success. Thus, when necessary, appropriate documentation, dedicated tools or training shall be included in the proposal to:

- Speed up PRODUCT adoption for novice users and,
- Empower intermittent users to obtain the information they need efficiently.

Combining the pure performance requirements of expert data analysts operating at the forefront with the accessibility required when serving a large and varied population of users, shall be at the core of the design philosophy. Potential providers are thus invited to detail how their proposal satisfies this design philosophy, providing both qualitative and quantitative information, in the context of a motorsport environment.

9.10.1.2 <u>Historical data storage</u>

As part of its proposal, the provider shall supply a complete documentation for access to the data format and data storage solution it elected to store and access historical telemetry data. Requirements regarding the infrastructure required to support such a solution shall be clearly detailed. The provision of this infrastructure is outside of the perimeter of the SECU and will be the responsibility of the Teams, PU suppliers and FIA.

The provider shall also supply a list of strengths and weaknesses along with clear arguments for how this proposal, combined with the Data Viewer aforementioned, may comply with the following key requirements:

- Scalability,
- Reliability,
- Security,
- Versatility and support from programming languages,
- Ability to be replicated between sites easily, efficiently and rapidly,
- Availability of the data offline.

A particular attention shall be given to “Scalability”, an often-overlooked requirement when it comes to dyno applications. Yet the amount of data generated off-track is, in practice, significantly larger than the amount of data acquired during a competition. It is also equally fundamental when considering the FIA use cases where the SECU and third-party units’ data of all the cars of all the teams need to be accessed, processed and analysed. Finally, “Scalability” is also essential with regards to the overall objective that is developing a sustainable end-to-end data platform.

Additionally, the information regarding “Reliability” shall consider how the data integrity is preserved at all times, particularly in the case where a program modifying the data terminates abnormally.

Suppliers are encouraged to evaluate a file-base historical data storage solution and in particular modern file formats such as Parquet. In fact, combining its open-source and columnar nature, it theoretically has the ability to 38fulfil all the requirements above while also being natively supported by most contemporary programming languages including MATLAB and all state-of-the-art on-premises and Cloud data platforms.

Whatever historical data storage solution the suppliers elect to use, they are requested to detail for each use case below, what will be the hardware and software data infrastructure required to read/write historical data.

	Scenario	Interface available
Historical data	Team at the track	Ethernet local network
	FIA at the track	Ethernet local network
	Team at the track	Ethernet local network
	FIA at the track	Ethernet local network
	Team Remotely	VPN or internet
	FIA Remotely	VPN or internet
	Team offline	PC local storage
	FIA offline	PC local storage

9.10.2 Software extensions

Beside all the basic functionalities of a data viewing and storage solution, the provider shall make the following software extensions available in order to supplement its PRODUCT:

- A module to create local ad hoc data processors i.e. local data channels calculated in real-time from a combination of several other data channels or data processors,
- A module to augment the data stream in real-time with simulation data ran in MATLAB Simulink and/or a compiled version of the models,

9.10.3 Low-Level Data Application Programming Interface

Suppliers are required to provide an Application Programming Interface in order to enable live and logged data to be ingested in real-time into various data analysis and storage tools that are used or will be developed in the coming years by the Teams, PU Supplier and the FIA. The goals of this API are:

- Easy to integrate with 3rd party systems,
- Compatible with FIA anti-tampering requirement where required,
- Compliant with Formula One data security and privacy requirements,
- 64bit,
- Usable from low level languages such as C or C++ and C# (via NuGet packages for .NET, .NET Core, GRPC),
- Built in a modular fashion so that it is possible to reference only the relevant packages,
- Compatible with high level languages such as Python and MATLAB,
- Compatible with Microsoft Windows and Linux,
- Demonstrate a low memory footprint,
- Demonstrate a low CPU usage,
- Scalable.

Once the API is available and documented, Teams, PU Suppliers and the FIA will be responsible for interfacing their own data analytics software to the provider's interface.

9.10.3.1 API Environment

It is assumed that the API will be run from an application with local network access to the Master ECU and /or TEP (Telemetry End Point) and capable of managing the handover from one to the other seamlessly. Note that although such API might be able to run from the Cloud, at the track the API interfacing is expected to be done on-premises, upstream of any Cloud processing.

9.10.3.2 API Functionalities

Below is a list of likely essential functionalities for the creation of an agent to stream telemetry data in real-time and decoded as engineering values.

For the sake of the proposal below, it is assumed that the data will be read in small sample blocks of parameter identifier, timestamps and data values.

9.10.3.2.1 Time

Where times are specified, they should align to the Master ECU time which may or may not be synchronised with PC/UTC time. Time and date resolution should be precise to 1µs.

9.10.3.2.2 Establishing a Connection

There needs to be a method for opening a connection to the streamed data that ensures only applications with the correct credentials are allowed to receive the data.

Connection should be by car, plant or rig name which should group the various data routes from the Master ECU together i.e. the physical link from the car (typically Ethernet) and the telemetry link from the car. It should be possible to restrict which links are to be listened to.

9.10.3.2.3 Data Stream Contents

There are multiple sub streams that may be required to be supported:

- Telemetry data – new samples that have just been captured and transmitted
- Logged data – stored data being offloaded
- Configurable parameters data
- [App] Events – messages triggered by onboard applications
- [Operational] Events – ECU reset, logging config change, logging started/stopped etc
- Telemetry backfill – data resent to fill gaps in telemetry data
- Missing data offload – data that was either logged to memory only or was lost in transmission.
- Message logs – e.g. messages sent to Driver Display Unit required off car to drive a virtual dash

The solution on offer should be flexible to support potential additional future sub streams such as video, audio or other complex sensors.

9.10.3.2.4 Parameter Metadata

All necessary metadata for each measurement and tune parameter must be available before or as soon as the first data sample is made available. Metadata should include but not be limited to:

- Identifier <String>
- Description <String>
- ID <number> if used to link data samples with metadata
- Format <String>
- Units <string>

If data needs to be converted to engineering values

- Raw data type
- Value and Error masks
- Conversion to engineering value rules
- Display format <string>
- Min/Max display values <number>

Additional properties for maps

- Dimensions and size
- Default parameter identifiers for each axis

9.10.3.2.5 Conversion to Engineering Values

Methods are required to convert all measurement and tune data to engineering values as only engineering values are expected to be used downstream of this interface. Virtual channels or any other formula-based parameters need to be resolved at this stage as well.

If the Virtual depends on other channels, there is a risk that those other channels might not be available at the time of calculation. To avoid this risk, calculations should either be done onboard, or all input channels must be transmitted in the same atomic message.

9.10.3.2.6 Updates to the Set of Parameters

The set of parameters, measurement and tune, may change when units are reprogrammed, logging tables are updated, or virtual channels are configured. Notification is required when such changes take place.

9.10.3.2.7 Visibility of Gaps in the Data

Despite improvements in telemetry coverage the problem of dealing with gaps in the data still needs to be managed. Typically, gaps are backfilled in near real-time or when there is a physical connection. The API should provide visibility of any gaps in the data due to network losses, in particular for variable rate data, so that systems downstream can know if the data is complete.

9.10.3.3 API Performance

The performance of the API is important and must be scalable as the logging rates and number of channels increases. Average telemetry rates are currently of the order of 200 ksamples/s. Peak rates may be higher e.g. after a pre-trigger has occurred.

9.10.3.3.1 Latency

Latency of telemetry data being available via the API should be < 200ms for any data logged at 100Hz or higher.

9.10.3.3.2 Errors and Debug

If any function cannot achieve the desired functionality clear error reporting should guide the developer or user to the cause of the problem.

9.10.3.3.3 Start of Logging

The logger is expected to stream data at a configurable background rate whenever it is powered, but this data may not be stored onboard.

When the car is on track or an important task is in operation, data may start to be written to onboard storage. This will be referred to as “start of logging”. Logging is often triggered by engine cranking where seeing data quickly is important. Note that the start of logging should not cause any delay to exceed the latency requirement above.

9.10.4 API Sample Program and Documentation

The APIs implementation must be adequately documented including technical notes, tutorials and sample code aimed at popularizing their usage. Samples programs should be provided to demonstrate the functionality and performance of the API. Examples implemented in modern programming language such as C#, Python or JavaScript are expected along with an implementation in the native language of the API e.g. C++.

9.10.5 Data distribution to the FIA

Within the perimeter of the tender, the provider shall supply a standalone solution to transfer to the FIA servers in real-time, all the live telemetry data of all the teams, available at their respective Telemetry Entry

Point (TEP). The FIA shall then be able to use either the provider data viewing and storage solution and/or the API to obtain, display, analyse and store those data in real-time.

During Events the FIA typically analyse all cars telemetry data in real-time as well as post-process the historical data, offloaded from the car after the sessions. To do so successfully, the FIA requires some data viewing capabilities beyond those of a typical Team or PU Supplier in particular:

- Ability to change the car on view quickly (of order 1 second, not 10s of seconds), including backfilling the telemetry data back to the point of "start of logging",
- Ability to compare live any car with any other car,
- Ability to have an overview via a customizable dashboard with configurable data channels and aggregation functions. Such aggregation function included but are not limited to current, lap, min, max value.

10 Licensing

Offers should include a copy of the terms of any licence associated with the PRODUCT.

Providers should also detail how the terms of the licences may evolve to safeguard the use of the PRODUCT purchased in the context of this contract and its functionalities beyond the term of the contract. Typical examples include supporting running historical cars for Teams.

Providers are requested to fill-in the "GENERAL" sheet in Appendix 1.

11 Support System

Along with the provision of the SECU hardware and software throughout the duration of the contract, the provider shall also put in place a first-class support system aimed at continuously improving the intersection between the customer experience and the PRODUCT. This means delivering high-quality support when and where the users will need it most.

The system shall be prepared to the fact that requests may grow unpredictably, especially at the start of the supply contract or during certain part of the Formula 1 season or in relation to changes to the regulations. Yet, while scalability shall be built-in and the system designed to handle surges, consistency of support shall remain a priority, both across time and across customers.

A 24/7 provision is not required and the focus shall really only be on a couple of high-efficiency support channels. Yet what will lack in availability shall be made up in quality with a dedicated customer support team, knowledgeable and efficient, operating both at base and at the track. The provider shall also actively collect feedback to make improvements to the platform thanks to what is commonly seen as a customer success team. Finally, to support this customer-centric paradigm, the provider shall empower its support teams with the appropriate toolset required to monitor the performance of the SECU in use and identify behavioural patterns that may need to be addressed and/or supported (e.g. specific operating conditions or recurring scenarios).

Finally, the teams, the PU suppliers and the FIA shall also be able to pro-actively provide feedback via a dedicated support ticketing system.

For each of the following sections, providers are requested to fill-in the "GENERAL" sheet in Appendix 1.

11.1 Ticket Management System

An essential part of such support system shall be a web portal dedicated to centralising the requests coming from either the teams, the PU suppliers or the FIA. Those support requests usually fall into three major categories:

- Bugs,
- Missing features,

- Confusing or hidden features.

Altogether, the web portal shall allow all stakeholders to create, update, and follow-up issues. For the sake of transparency and promoting collaboration between the stakeholders:

- Issues reported by the provider internal employees shall be included,
- All issues shall be visible by all customers, with the ability for others to comment and/or supply additional material.

In effect, users evolving in a constrained high-pressure environment such as trackside personnel will raise requests using whatever is easiest for them at the time. This may be emails, chats or in-app feature.

In all cases, the Ticket Management System shall concentrate most, if not all, of the support conversations in one place. For this reason, a robust and versatile integration with third-party modern communication tools would be a strength.

11.2 Knowledge Base

It is well-known that more than two users out of three select self-service support overreaching the support person when they feel that it will meet their needs. Therefore, along with the Ticket Management System, the provider shall put in place a knowledge base in the form of a repository or collection of resources and documentation aimed at helping SECU users. This shall cover all the aspects of the SECU, including but not limited to hardware, embedded software, software tool suite and APIs.

To supplement this self-help knowledge base and make it a success, the provider shall put in place the associated toolset necessary to measure its performance and leverage that knowledge to improve it. Over the duration of the supply contract, the aim should be for the cost of the support to decrease, which is why improving the self-support system is key in order to maintain the level of service.

This will also help reducing the workload of the customer support team and help redirect resources on the customer success aspect. Both share the same purpose but have different approaches when it comes to interacting with clients.

11.3 Customer Support Mission

The customer support team will work as a traditional safety net for the users. For that purpose, on-track support will be required at each competition so that the Teams, the PU suppliers and/or the FIA can reach out for help and solve their challenges with minimal delay and disruption to their normal operations.

During the early stages of the supply contract, the support agents will be key in individually assisting and educating the users, going through the numerous minor issues and questions inherent to the introduction of a new product.

For more complex issues that may require extra time and/or expertise that need to be sourced at base, the knowledge and experience of the whole support and development teams may be leveraged. Yet, as a group, the track support team shall be sufficiently autonomous to help quantifying how bad the problem is and provide alternative suitable solutions.

As first point of contact, the role of the track support agents will always be to capture the issue and all the necessary elements of context necessary to its resolution. They shall then transmit that information back at base in a comprehensive form, popularizing the track specificities when necessary. Finally, they shall pro-actively follow-up the issue and make the bridge between the progress at base and the customer. Throughout the process, they shall also report the progress to the FIA and document the knowledge base accordingly.

11.4 Customer Success Mission

Putting in place a customer success team revolves around admitting that providers should take a hands-on approach with their users. This means contacting the customers and working with them to understand and achieve their goals.

An important aspect is that not all users may find the PRODUCT that will be supplied intuitive. It was notably highlighted that in effect, a very heterogeneous panel of users rely on the provider Data Viewer to access data. Some of them can be distinguished by a smaller frequency of use of the software and hence less familiarity. Success agents can usher those users and introduce them to the features the platform will propose. Users may receive assistance over the phone, via email or through specific knowledge-based article and/or tutorial.

A successful team will make sure that customers are using your products and that all users are getting the most out of them. By monitoring how the PRODUCT performs on the field and identifying what may have been hidden behavioural patterns, the success agents can make sure that everything that is learnt from support is fed back into PRODUCT design, missing or messed-up features and customer success.

12 Value delivery chain

To supplement the technical proposal, providers shall document a detailed analysis of the main activities of their company value delivery chain. Only the aspects related to what may have an impact on the SECU supply over the duration of the contract shall be discussed.

This includes but may not be limited to technological developments, inbound logistics, operations, outbound logistics, sales, service and procurement. In each case, the provider shall make a clear and objective evaluation of the potential risks and how these may interfere with the delivery of the PRODUCT. When applicable, a mitigation plan shall be provided.

For each of the following sections, providers are requested to fill-in the “GENERAL” sheet in Appendix 1.

12.1 Human resources

Providers should detail:

- The allocated resource plan to cover both the development and in service years for the hardware,
- The allocated resource plan to cover both the development and in service years for the support,
- The allocated resource plan to cover both the development and in service years for the embedded and off-car software,
- In each case, how much of those resources is combined with other businesses.

12.2 Quality processes and testing equipment

Offers should include a description of the following items:

- Quality processes,
- Test and calibration equipment including Automated Test Equipment (ATE),
- FMEA (Failure Mode and Effects Analysis) and FTA (Fault Tree Analysis) of the PRODUCT.

12.3 Technological developments

It is assumed that not all the technologies pertaining to the equipment, hardware, software, procedures and technical knowledge contained in the provider proposal may have reached full maturity at the point where the tender will be submitted.

It is important that in such cases, the provider discloses the associated risks and the extent of the alternative solutions that it may have to put in place.

Providers are requested to fill-in the “DEVELOPMENTS” sheet in Appendix 1.

12.4 Obsolescence management

Providers are requested to detail the obsolescence management strategy that they will put in place for all the elements included in the PRODUCT on offer over the duration of the contract.

Additional information should be provided for the five following years after the term of the contract, to safeguard the use of the PRODUCT and its functionalities beyond the term of the contract.

12.5 Service

Offers should include information about any service intervals expected by the provider for each element of the SECU. Such service intervals should not be shorter than a Championship season.

Detail strengths and weaknesses of all the activities put in place to guarantee that the PRODUCT remain in a fully functional state after it has been sold and delivered to the Teams, the PU suppliers and/or the FIA. In the case where this may not be the case, a detailed view of how the issue is managed should be provided.

12.6 Procurement

Any procurement activity related to the acquisition of either services or works from an outside external source shall be clearly highlighted. A contingency plan in case of failure of the associated supply chain shall be provided.

12.7 Inbound logistics

Detailed strengths and weaknesses analysis of the materials and/or parts supply chain required to manufacture or assemble the SECU system and sub-systems. This shall include both internal (e.g. inventory management) and external (e.g. purchase and sale of suppliers company) risks investigation.

12.8 Operations

Detailed strengths and weaknesses analysis of the processes and resources that converts the raw inputs into the final SECU hardware, software and associated services. The breakdown shall include both risks at base (e.g. manufacturing) and risks on remote operations (e.g. track support).

12.9 Outbound logistics

Detailed strengths and weaknesses analysis regarding storage and delivery of the final PRODUCT as well as the associated information flow to the end users.

12.10 Sales

Detail strengths and weaknesses for creating and delivering offerings related to the PRODUCT including but not limited to hardware, software and support.

13 Appendix

Appendix 1: Tables of Requirements

See the attached document "Appendix1_TechnicalSpecificationsTables.xlsx"

Appendix 2: System Configuration Software REST API example

It is encouraged that the System Configuration Software would exclusively rely on the use of such an API and that the supplier provides access to this API along with the associated documentation. In any case, it is fundamental that the REST API cover exhaustively all the functionalities of the System Configuration Software.

The present example is inspired from HTTP-based RESTful API that are defined by the following aspect:

- a base URL such as http://system.configuration.software.local:5001,
- a set of standard HTTP methods such as GET, POST, PUT and DELETE,
- a media type for composing the requests such as a URI (Uniform Ressource Identifier) which a unambiguously identifies a particular resource via a string of characters.

Below are some examples of API endpoints using the base URL:

```
http://system.configuration.software.local:5001/api
```

PROJECT INFORMATION

The REST API may be use to access basic information such as the details of the active applications, for example using the following standard command:

GET /api/Apps

```
GET system.configuration.software.local:5001/api/Apps?active=true
 -H "accept: text/json"
```

The response body below provides a detailed list of all the active applications IDs on the unit along with the corresponding application name.


```
[
 {
   "id": 12801,
   "name": "FIA"
 },
 {
   "id": 12802,
   "name": "Chassis"
 },
 {
   "id": 12803,
   "name": "Arbitrator"
 },
 {
   "id": 12804,
   "name": "Coordinator"
 },
 {
   "id": 12805,
   "name": "Controller"
 },
 {
   "id": 12806,
   "name": "Dash"
 },
 {
   "id": 12807,
```

```
    "name": "BrakeControl"
  },
  {
    "id": 12808,
    "name": "Math"
  },
  {
    "id": 12800,
    "name": "TAG320BIOS"
  }
]
```


On may also use the REST API to obtain the current value of all the parameters within an application using the following standard request:

GET /api/Apps/{appId}/ScalarParameters/Values

```
GET
system.configuration.software.local:5001/api/Apps/12801/ScalarParameters/Values
-H "accept: text/json"
```

The response body below provides a detailed list of all the scalar parameters from the application with the ID 12801, along with their current value:


```
[
  {
    "symbolKey": {
      "appId": 12801,
      "id": "BAnalogModeFIA005",
      "name": "BAnalogModeFIA005",
      "symbolType": 1,
      "paramType": 0
    },
    "value": 0
  },
  {
    "symbolKey": {
      "appId": 12801,
      "id": "BAnalogModeFIA006",
      "name": "BAnalogModeFIA006",
      "symbolType": 1,
      "paramType": 0
    },
    "value": 1
  },
  {
    "symbolKey": {
      "appId": 12801,
      "id": "BAnalogModeFIA007",
      "name": "BAnalogModeFIA007",
      "symbolType": 1,
      "paramType": 0
    },
    "value": 1
  },
  {
    "symbolKey": {
      "appId": 12801,
      "id": "BAnalogModeFIA008",
      "name": "BAnalogModeFIA008",
      "symbolType": 1,
      "paramType": 0
    },
```

```
    "value": 1
  },
  {
    "symbolKey": {
      "appId": 12801,
      "id": "BAnalogModeFIA009",
      "name": "BAnalogModeFIA009",
      "symbolType": 1,
      "paramType": 0
    },
    "value": 0
  },
  {
    "symbolKey": {
      "appId": 12801,
      "id": "BAnalogModeFIA010",
      "name": "BAnalogModeFIA010",
      "symbolType": 1,
      "paramType": 0
    },
    "value": 0
  },
...
  {
    "symbolKey": {
      "appId": 12801,
      "id": "xOilScaledMax",
      "name": "xOilScaledMax",
      "symbolType": 1,
      "paramType": 0
    },
    "value": 300
  },
  {
    "symbolKey": {
      "appId": 12801,
      "id": "xOilScaledMin",
      "name": "xOilScaledMin",
      "symbolType": 1,
      "paramType": 0
    },
    "value": -10
  }
]
```


LOGGING CONFIGURATION

Another example of use case that a REST API can cover is the case of changing the configuration of the unit. Here one may use a standard POST method on the following hypothetical URI:

```
system.configuration.software.local:5001/api/logging-configuration
```

Below an example where the SECU-Auth connection request includes a token, that was returned in during the connection to the unit, to verify the accreditation of the user. BAnalogModeFIA006 and BAnalogModeFIA007 are added to the FIA1 logging channel at a frequency of respectively 500Hz and 250Hz.

```
POST system.configuration.software.local:5001/api/logging-configuration/
SECU-Auth: 12ab34cd56ef78ab90cd12ef34ab56cd
Content-Type: application/json
[{
  "parameter”: {
```

```
    "channelID": FIA1,
    "appId": 12801,
    "parameterName": "BAnalogModeFIA006",
    "frequencyHz":  500
  }
},
{
  "parameter”: {
    "channelID": FIA1,
    "appId": 12801,
    "parameterName": " BAnalogModeFIA007",
    "frequencyHz":  250
  }
}]”
```


The response may provide feedback by returning the full list of parameters downloaded to the unit with their associated updated configuration.

Whether the POST request is blocking, waiting for the response or whether the response is returned in some other way is a detail of implementation. However, in all cases, the process should be simple for the user who should also receive clear feedback regarding the success or the failure of his operation.

Finally, the same base URI used to modify the configuration via a POST command may also be used for accessing general information about the logging configuration using a set of standard GET commands:

GET system.configuration.software.local:5001/api/logging-configuration

The URI could equally be used to access information for a specific channel:

GET system.configuration.software.local:5001/api/logging-configuration/{channel}

Appendix 3: System Monitoring Software programming interface example

It is encouraged that the provider Data Viewer would exclusively rely on the use of such an API and that the provider supplies to this API along with the associated documentation. This means that the API shall cover all the functionalities of the System Monitoring Software.

The present example presents what the programming interface might look like yet it is left to the provider to make an exhaustive proposal that satisfies the core requirements presented in the rest of the document.

Let's assume the following base class definition:

```
static class StreamHandlerFactory
{
        static IStreamHandler Create(ICallbackObserver callbackObserver,/* other
params to initialize the library */);
}
```

The client uses the IStreamHandler to populate the API with the raw packets coming from the network layer or, alternatively, from some other data source, for example a file:

```
interface IStreamHandler
{
        void HandleData(array<Byte> data);
}
```

Note that an equally suitable approach may be to have the StreamHandler directly handling the data. This would avoid exposing HandleData:

```
static IDisposable Create(ICallbackObserver callbackObserver, IStream stream, /*
params */);
```

In such paradigm, the StreamHandlerFactory will pass the Stream to the StreamHandler. The latter is then used to read the incoming raw packets coming from the network layer or, alternatively, from some other data source, for example a file.

The StreamHandler implementation will decode the incoming raw packets and call the appropriate callbacks on the user-provided observer:

```
interface class IStreamHandlerObserver
{
        void LoggingStarted();
        void LoggingStopped();
        void OnChannelData(UInt64 sTimeStamp, UInt64 sInterval, UInt32
sampleCount, ChannelIdentifierType channelId, array<Byte> rawData);
        void OnLap(UInt64 sTimeStamp, UInt16 lapNumber);
        void OnEvent(UInt64 sTimeStamp, EventIdentifierType eventId,
EventValueType eventValue);
        void OnConfigStatus(ConfigIdentifierType configId);
}
```

ChannelIdentifierType, EventIdentifierType and ConfigIdentifierType can be used to respectively identify a given SECU data channel, event or configuration. These types and the value they may take will of course be identical to what is used by the System Monitoring PC software itself.