Qt Group

Qt for MOSA

Portable HMIs for FACE Conformant Defense Systems

2025

Executive Summary 3

Qt Framework 4

Qt in Aerospace and Defense 5

Qt for MOSA and the FACE® Technical Standard 6

Qt for MOSA - Specifications 7

- Conformance Details 8
- Architecture 9
- Benefits 10
- Components 11
- Framework Details 12

Executive Summary

Modern defense systems require high-quality software that is modular, portable, and interoperable. Traditional systems development permitted the use of closed, proprietary architectures and interfaces, which led to vendor lock-in and continues to make integration difficult and costly. As the complexity of defense systems increases, legacy systems lack the flexibility to be extended, increasing the cost of technology insertion.

Qt for MOSA provides the **power and efficiency of Qt**—advanced graphics, intuitive controls, rapid development, and more—within the Modular Open System Architecture (MOSA) established under the FACE® Technical Standard, making development there as simple as creating a Qt Application in traditional environments.

With this adaptation to the FACE Technical Standard, Qt Framework can now be used to build the affordable and adaptable systems targeted by the MOSA approach.

Qt Framework

With its comprehensive cross-platform libraries, Qt Framework (or Qt, for short) produces high-performance, easily maintainable, and reusable software for all types of devices—embedded, desktop, mobile, and more.

With 30 years of history, a presence in over 70 industries, and billions of devices *Built with Qt* in the market, Qt Framework has shaped the notion of **digital experience** and helped world-leading OEMs to **efficiently deliver intuitive, reliable, and effective software** across verticals.

Supplemented by dedicated tooling for each stage of the product lifecycle (design-to-code conversion, logic implementation, debugging, profiling, deployment, and beyond), Qt offers the creative environment for developers to efficiently realize their product vision to the full extent.

Qt in Aerospace and Defense

Built with quality and safety at its core, Qt Framework is used extensively across regulated industries, such as Medical and Automotive. In the Aerospace and Defense sector, Qt is known for its ease of development, reliability, and cross-platform capabilities.

Qt is widely used in the development of Command, Control, Communications, Computers (C4), Intelligence, Surveillance, and Reconnaissance (ISR) systems—**C4ISR**, for short—due to its comprehensive libraries of ready-made functionality and efficient tooling, which helps reduce the cost and schedule impact of software development.

A primary advantage of using Qt Framework is its architecture, which provides **modularity, platform independence**, and **openness** at its core.

By abstracting over the complexity of the underlying system and graphics interfaces, Qt libraries make it easy for developers to port advanced functionality to varied devices selectively (only providing what's necessary) and consistently (behaving the same across devices), and to easily integrate third-party solutions.

In a sector seeking to reduce costs through a Modular Open Systems Approach (MOSA), such features are a strong competitive advantage.

Qt Applications in Aerospace and Defense

C4ISR: Advanced HMIs/UIs for demanding control and display requirements

Embedded Systems: Modular cross-platform support for embedded mission and sensor systems

Analysis, Simulation, and Training: Capable 2D and 3D data visualization for operational analysis, simulation, and training

The Modular Open System Approach

MOSA is a Department of Defense (DoD)-mandated strategy emphasizing the use of open standards and modular design principles in military systems. The goal of MOSA is to create more adaptable, cost-effective, and **interoperable** defense systems by reducing reliance on proprietary solutions and vendor lock-in. It is required by U.S. law for all major defense acquisition programs and is a key enabler of innovation and rapid technology insertion.

Qt for MOSA and the FACE® Technical Standard

The Qt Framework was developed with the principles of modularity, portability, and interoperability, even before those principles were adopted by MOSA in service of reducing the cost of developing defense systems. But FACE Conformance is a recent achievement (August 2025). Qt for MOSA, a FACE Unit-of-Conformance (UoC) supplying an adapted version of Qt Framework alongside a reference application, targets the **General-Purpose Profile of FACE Technical Standard, Edition 3.2**.

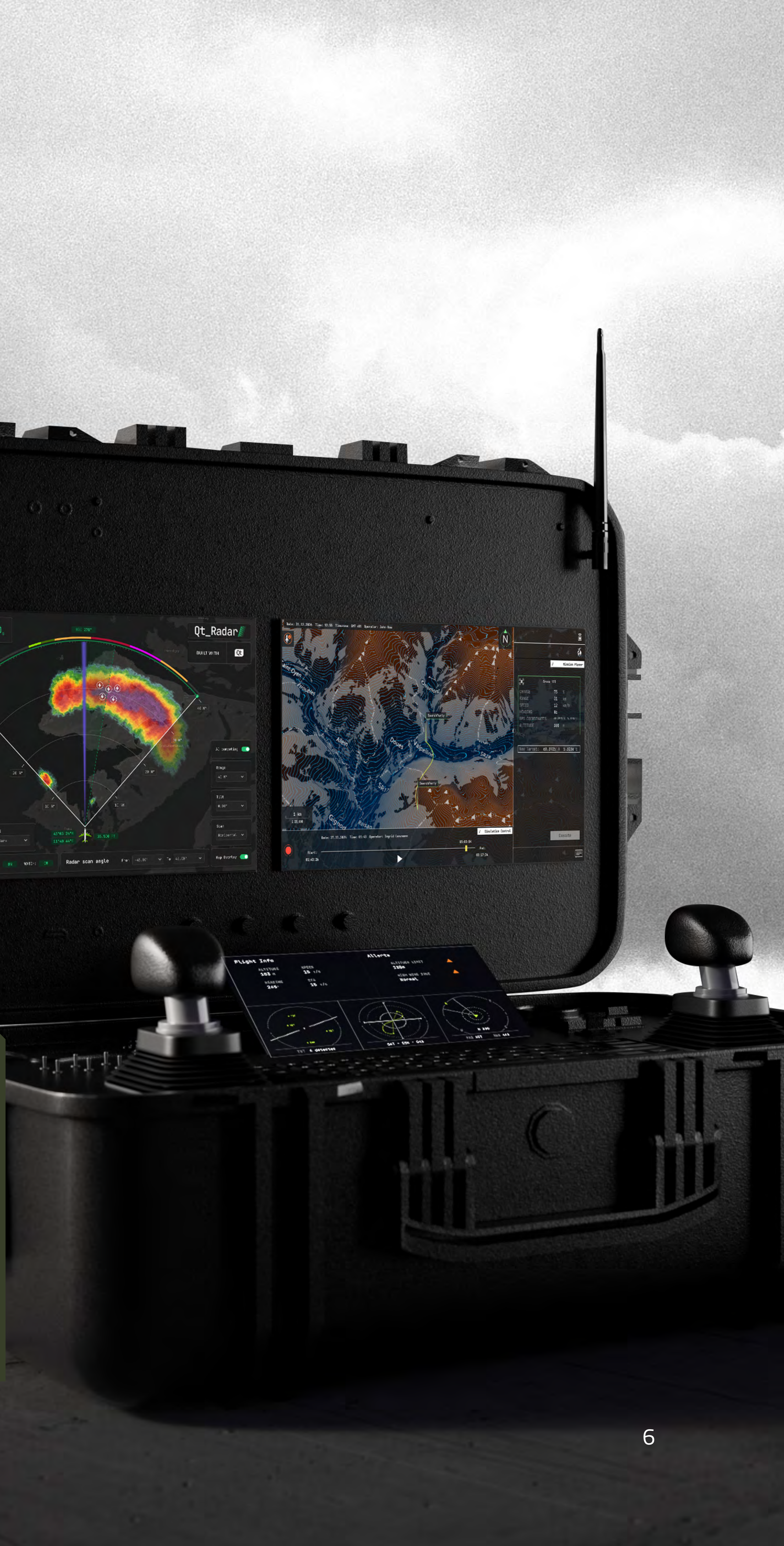
Qt for MOSA is a large subset of Qt Framework—including modules like Qt Quick, Qt Widgets, Qt Charts, and more—that provides an abstraction over the API allowed by the FACE OSS (Operating System Segment) to make rapid development of HMI software for **mission-critical applications** more efficient.

Qt for MOSA is only available to **commercial users** and ships with Maintenance and Support included. It provides much of the power and efficiency that the full Qt Framework offers to traditional development, but it is restricted to capability allowed under the FACE Technical Standard.

Qt for MOSA retains key capabilities of the Qt Framework. It allows development using either traditional Qt Widgets or modern Qt Quick. Event-driven programming is fully supported through the signals-and-slots language construct. Further, the metaobject compiler enables introspection and asynchronous functions even without support from the C++ language runtime.

The FACE® Technical Standard

Managed by The FACE Consortium, the FACE Technical Standard is a set of software specifications designed to codify the principles of MOSA. By promoting open architecture standards, interoperability, and rapid integration, the FACE Technical Standard fosters agility, innovation, and cost efficiency. Where MOSA serves as a high-level directive promoting modularity and openness in defense acquisitions, the FACE Technical Standard provides detailed **software requirements** that enforce these principles.



Qt For MOSA – Specifications

Definitions from FACE Technical Standard

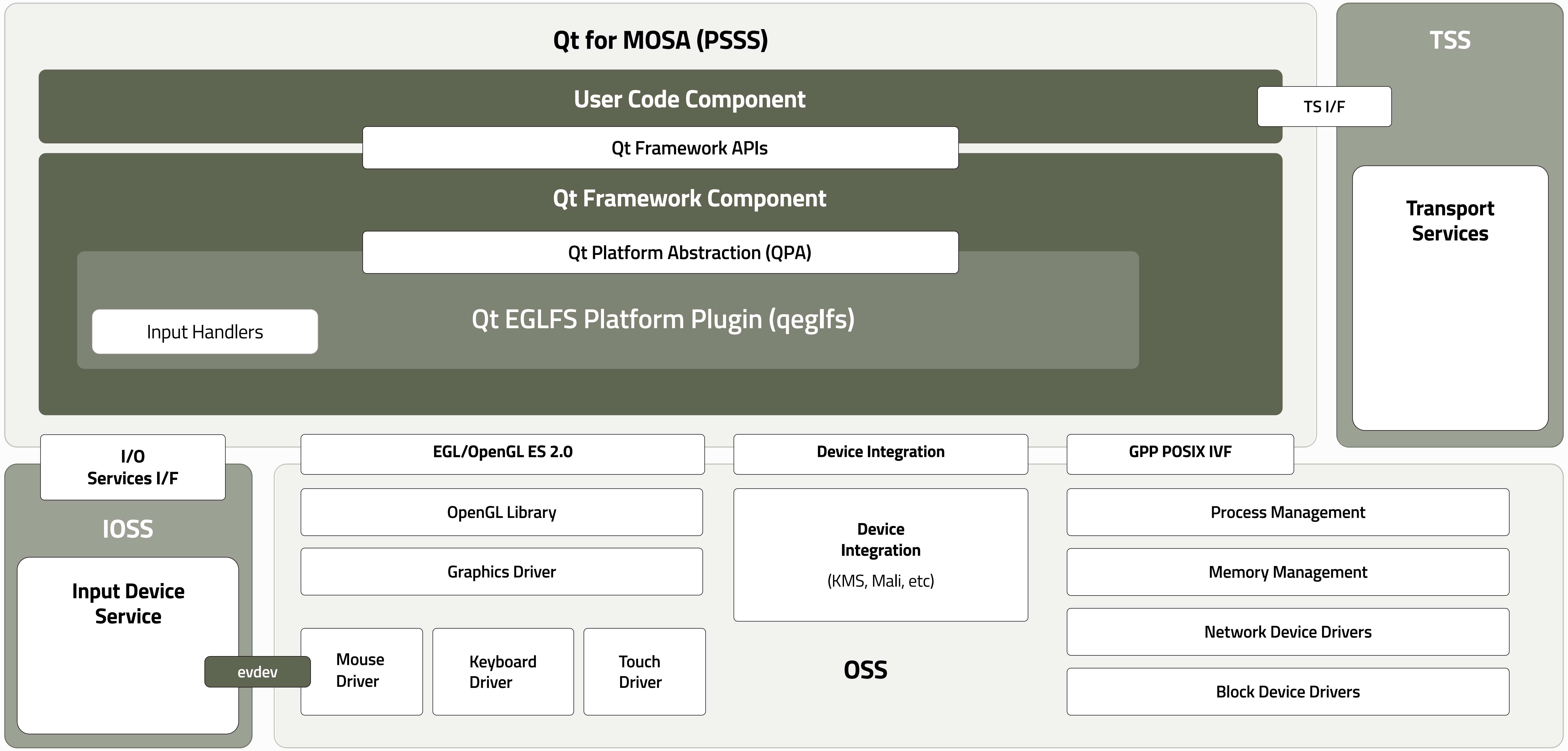
ACRONYM	FULL FORM	DEFINITION
TSS	Transport Services Segment	Segment that abstracts transport mechanisms and data access from software components, facilitating integration into disparate architectures and platforms using different transports
OSS	Operating System Segment	Segment where foundational system services used by all other segments and vendor-supplied code reside
IOSS	I/O Services Segment	Segment where normalization of vendor-supplied interface hardware device drivers occurs
PSSS	Platform-Specific Services Segment	Segment comprised of sub-segments including Platform-Specific Common Services, Platform-Specific Device Services, and Platform-Specific Graphics Services
PSCS	Platform-Specific Common Services	Sub-segment comprised of higher-level services, including Logging Services, Centralized Configuration Services, DPM Services, Streaming Media, and System-Level HMFM
PSDS	Platform-Specific Device Services	Sub-segment where management of data and translation between platform-unique interface control document (ICDs) and the FACE Data Model occurs
PSGS	Platform-Specific Graphics Services	Sub-segment that abstracts the interface specifics of a graphics device driver from software components within the FACE Reference Architecture
HMFM	Health Monitoring and Fault Management	Provide standardized methods for detecting, reporting, and handling faults and failures within the scope of a single system or platform
DPM	Device Protocol Mediation	Mediation of transport protocols supported by OSS interface for platform device
UoC	Unit of Conformance	Software component or domain-specific data model designed to meet the applicable requirements defined in the FACE Technical Standard

Conformance Details

The FACE Technical Standard, initially leveraged for safety-critical airborne systems, is recently being applied more broadly due to the MOSA mandate—including less safety-focused, but still mission-critical, applications like ground vehicles and surveillance systems. Such systems can leverage the less restrictive **General Purpose Profile**, or the Safety Extended Profile, where a general-purpose software framework like Qt is well suited.

- ✓ FACE Technical Standard, Edition 3.2 - General Purpose Profile
- ✓ Platform-Specific Services Segment (PSSS)
- ✓ Platform-Specific Graphics Service (PSGS) Sub-Segment
- ✓ C++11 and C Programming Languages
- ✓ EGL / OpenGL ES 2.0
- ✓ POSIX Operational Environment

Architecture

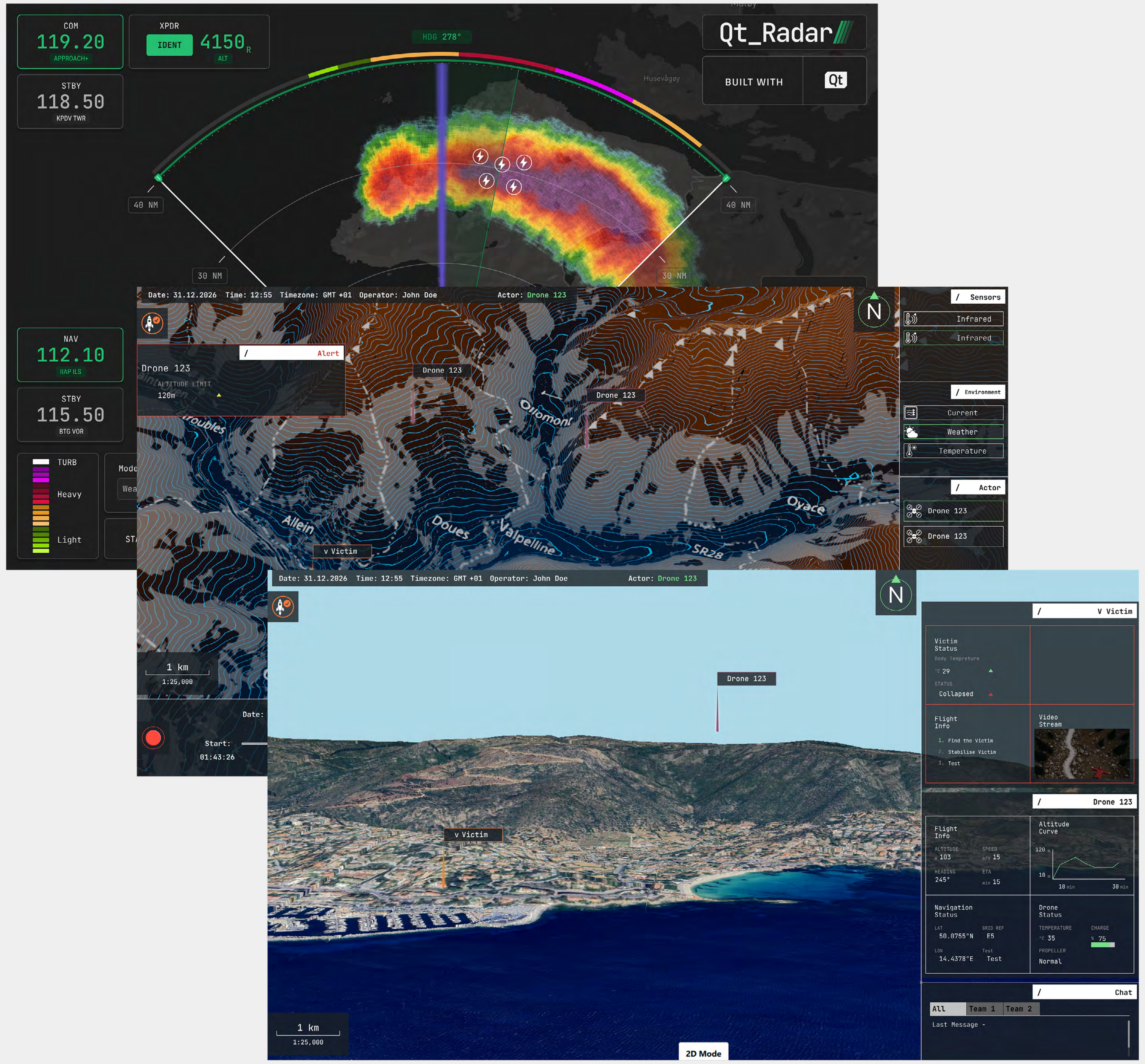


Benefits

Qt for MOSA supports efficient development with core Qt Framework features and libraries. It provides an approachable starting point for new projects seeking FACE Conformance, which can **leverage a proven conformance package**.

Simplified porting of existing Qt applications into FACE Conformant UoCs is also enabled:

- ✓ Develop and test on desktop (Linux, Mac, and Windows)
- ✓ Easily deploy and integrate into target system



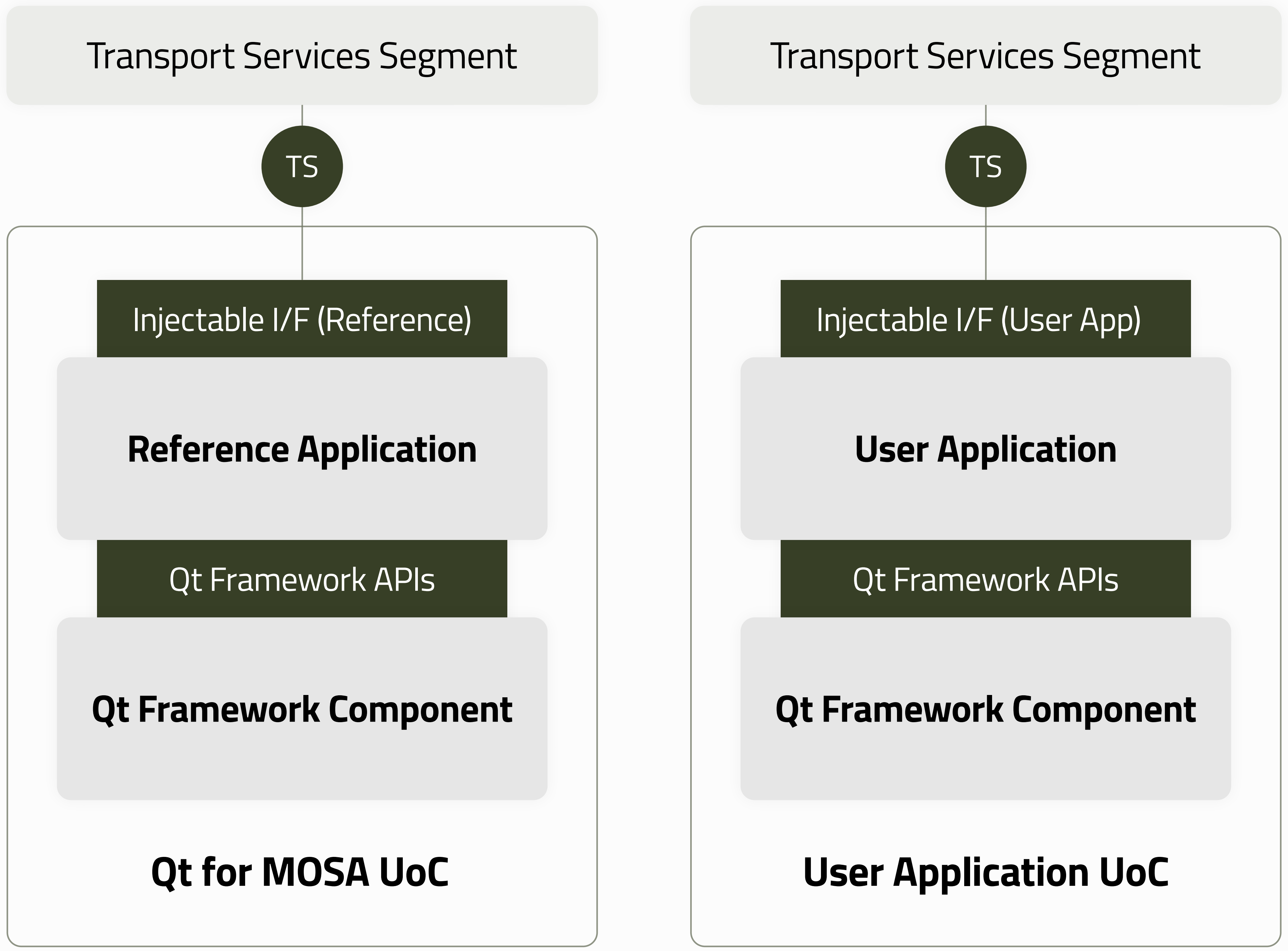
Components

The Qt for MOSA UoC (Unit-of-Conformance) is a FACE Conformant PSSS (Platform-Specific Services Segment) consisting of Qt Framework and a **reference application**. It is bundled with:

- Non-certified IOSS (Input/Output Services Segment) UoC providing example keyboard, mouse, and touch device integrations for Linux
- Graphics Device Integrations that may be used by system integrators to leverage specific graphics hardware

Qt for MOSA also provides the following components:

- Conformance Verification Package
- Documentation



Transport Services Segment

TS

Injectable I/F (Reference)

Reference Application

Qt Framework APIs

Qt Framework Component

Qt for MOSA UoC

Transport Services Segment

TS

Injectable I/F (User App)

User Application

Qt Framework APIs

Qt Framework Component

User Application UoC

Framework Details

Qt for MOSA retains most of the core capabilities of Qt.

- It allows development using either traditional Qt Widgets or modern Qt Quick.
- Event-driven programming is fully supported through the signals and slots language construct.
- The metaobject compiler enables introspection and asynchronous function calls independent of the C++ language runtime.

Supported Modules

Qt Charts	Chart visualization
Qt Core	Core non-GUI classes
Qt GUI	Core GUI classes, includes OpenGL
Qt Image Formats	TIFF, MNG, TGA, WBMP support
Qt QML	QML and JavaScript
Qt Quick	Declarative, dynamic UIs
Qt Quick 3D	High-level 3D content APIs
Qt Quick Controls	Responsive UI components
Qt Quick Dialogs	System dialogs in Qt Quick
Qt Quick Layouts	Layout management
Qt Quick Particles	Particle effects
Qt Quick Test	QML unit testing
Qt Test	C++ unit testing
Qt Virtual Keyboard	On-screen keyboard
Qt Widgets	Classic C++ widgets
Qt XML	SAX and DOM XML parsers

Qt Group

Qt Group offers cross-platform solutions for the entire software development lifecycle.

Qt Group (Nasdaq Helsinki:QTCOM) is a global software company, trusted by industry leaders and over 1.5 million developers worldwide to create applications and smart devices that users love. We help our customers to increase productivity through the entire product development lifecycle - from UI design and software development to quality management and deployment.

Our customers are in more than 70 different industries in over 180 countries. Qt Group is headquartered in Espoo, Finland, and employs over 800 people globally.