MAPUSOFT

INTER-OPERABILITY MADE EASY

MapuSoft is the global leader in software interoperability & reusability solutions that provide freedom, protection and stability to embedded applications

MAPUSOFT

FREEDOM

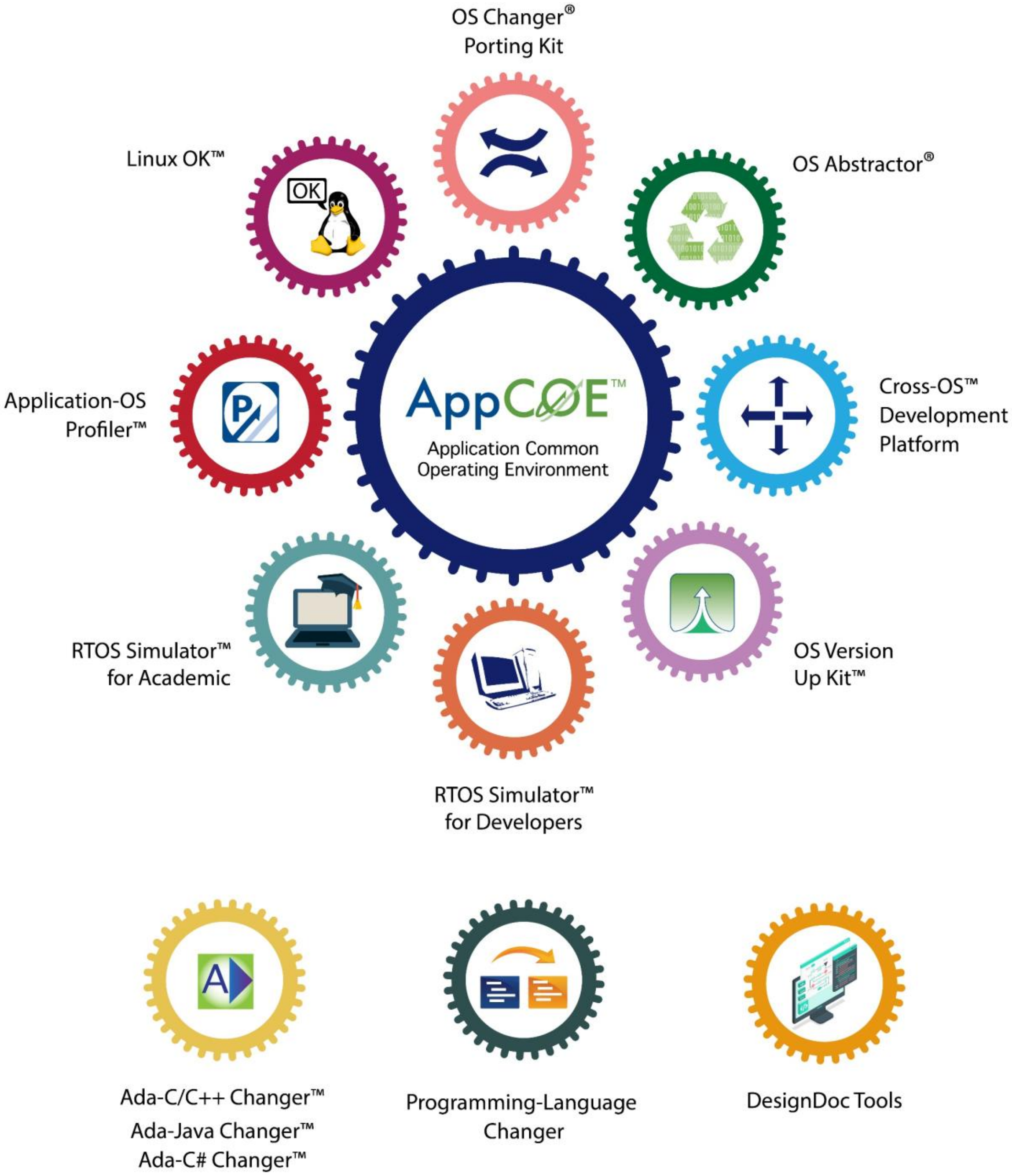
Don't confine your code to one platform

PROTECTION

Protect software investment

STABILITY

Robust and optimized platform



AppCOE™

Application Common Operating Environment

OS Changer®
Porting Kit

OS Abstractor®

Cross-OS™
Development
Platform

OS Version
Up Kit™

RTOS Simulator™
for Developers

RTOS Simulator™
for Academic

Application-OS
Profiler™

Linux OK™

Ada-C/C++ Changer™
Ada-Java Changer™
Ada-C# Changer™

Programming-Language
Changer

DesignDoc Tools

Cross-OS Development Platform

Develop new code that is not locked to the OS for a longer life

Develop C/C++ software using one or multiple OS Interface options and run the same code base on multiple operating systems

OS Abstractor

MapuSoft's OS Abstraction Layer (OSAL)

Is a commercial-grade OS Abstraction Layer (OSAL) designed without a layered implementation in order to provide better performance at lower cost

OS Changer Porting Kit

Rapidly Migrate Software to a new Operating System with Performance Optimization

Re-use your C/C++ software developed for VxWorks®, pSOS®, Linux/POSIX, Windows®, Nucleus®, ThreadX®, µ-ITRON, µC/OS™, FreeRTOS™, VRTX®, QNX® or RTLinux®, on a new Operating System

Application-OS Profiler

Gather Performance Data for Your Application and Platform

Identify performance bottlenecks and compare performance metrics on various target environments

MAPUSOFT

RTOS Simulator for Developers

Develop, Run and Test Applications on a Virtualized Host Platform

Develop, run and test embedded applications on Windows or Linux host environments without the need for the original OS or expensive target hardware

RTOS Simulator for Academics

Develop, Run and Test Applications on Windows or Linux Host and Test on Hardware

Cross-compile, Download and Test Applications on various supported Target Hardware

Linux OK (Linux Optimization Kit)

Increase Performance and stability of Linux Applications

Provides kernel and application specific optimization features aimed at increasing CPU performance, speeding up boot-time and reducing memory footprint for Linux applications

OS Version UpKit

Upgrade Your OS Version without the Manual Porting Effort

Easily upgrade applications to a new version of the OS, without manual porting

Ada-Changer

Automatically convert Ada code to C/C++/Java/C#

Compile Ada code using C/C++/Java/C# tools and support target platforms not supported by Ada

Programming-Language Changer

Automated Software Language Conversion

Converts legacy application code from over 30 programming languages into modern languages.

DesignDoc Tools

Auto Generate Design Docs from Your Code

DesignDoc tools can be used to fully understand your existing software in terms of complexity, similarity, dead code, and external dependencies.

AppCOE

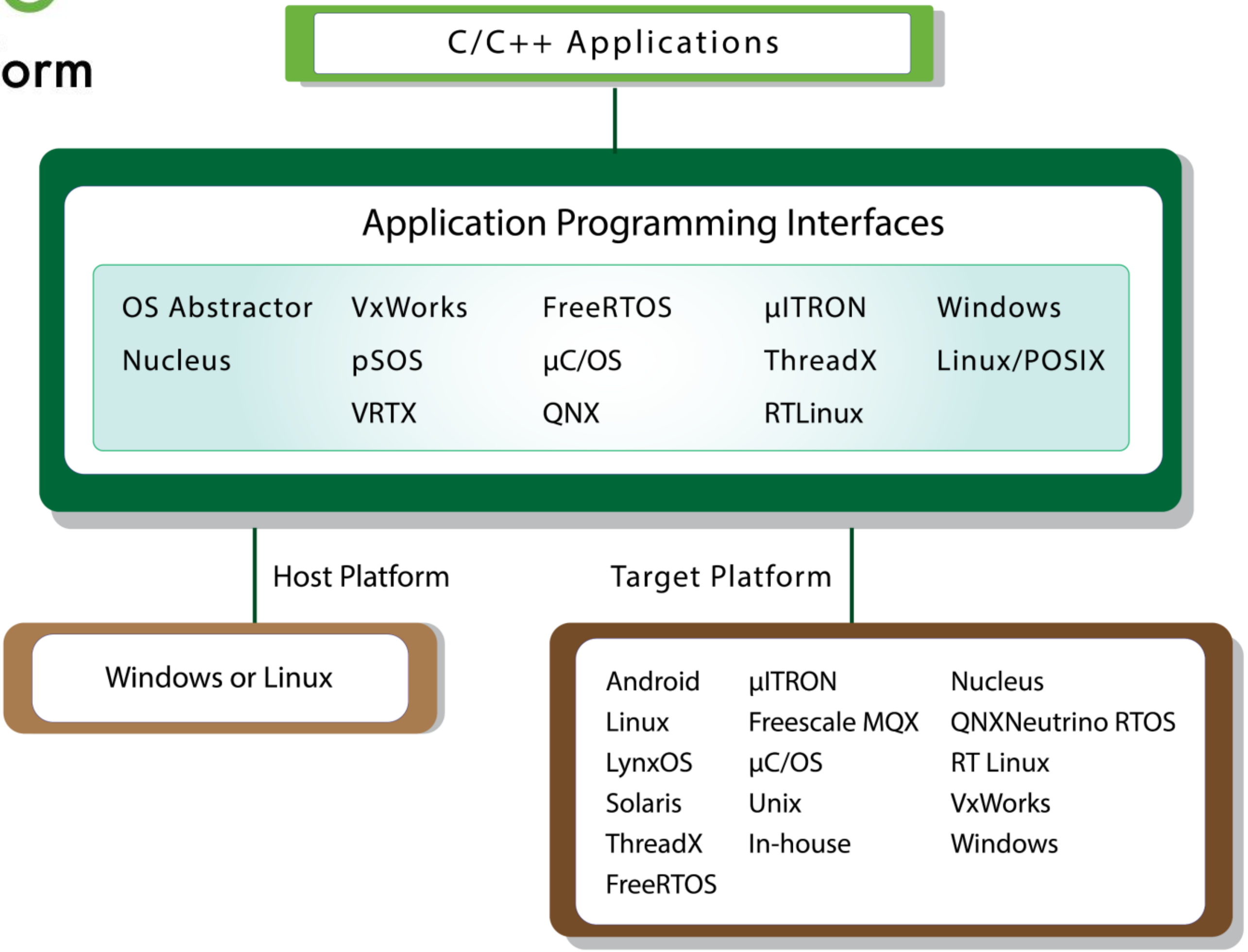
Application Common Operating Environment

AppCOE is a framework of common architecture that promotes code-reuse, software interoperability and Cross-OS platform capabilities among systems and devices.

MAPUSOFT®

Cross-OS® Development Platform

Develop Long-life code



C/C++ Applications

Application Programming Interfaces

- OS Abstractor
- Nucleus
- VxWorks
- pSOS
- VRTX
- FreeRTOS
- µC/OS
- QNX
- µITRON
- ThreadX
- RTLinux
- Windows
- Linux/POSIX

Host Platform

Windows or Linux

Target Platform

- Android
- Linux
- LynxOS
- Solaris
- ThreadX
- FreeRTOS
- µITRON
- Freescale MQX
- µC/OS
- Unix
- In-house
- Nucleus
- QNXNeutrino RTOS
- RT Linux
- VxWorks
- Windows

Cross-OS Development Platform

- Develop new applications that are not locked to the current OS/hardware platform. This greatly increases its lifespan and value while eliminating the risk associated with the OS selection process since the application can be tested on multiple platforms.
- Cross-OS Development Platform Interfaces:
  - OS Abstractor™ API
  - Linux/POSIX API
  - micro-ITRON API
  - RTLinux® API
  - VxWorks® API
  - FreeRTOS™ API
  - µC/Os™ API
  - Windows® API
  - Nucleus® API
  - pSOS® API
  - ThreadX® API
  - VRTX® API
  - QNX® API
- Includes OS Abstractor Target Specific Modules for the target OS or OS’s
- Cross-OS Development Platform is integrated with AppCOE, an Eclipse-based framework, providing a state-of-the-art IDE for development and testing

Cross-OS Development Platform includes:

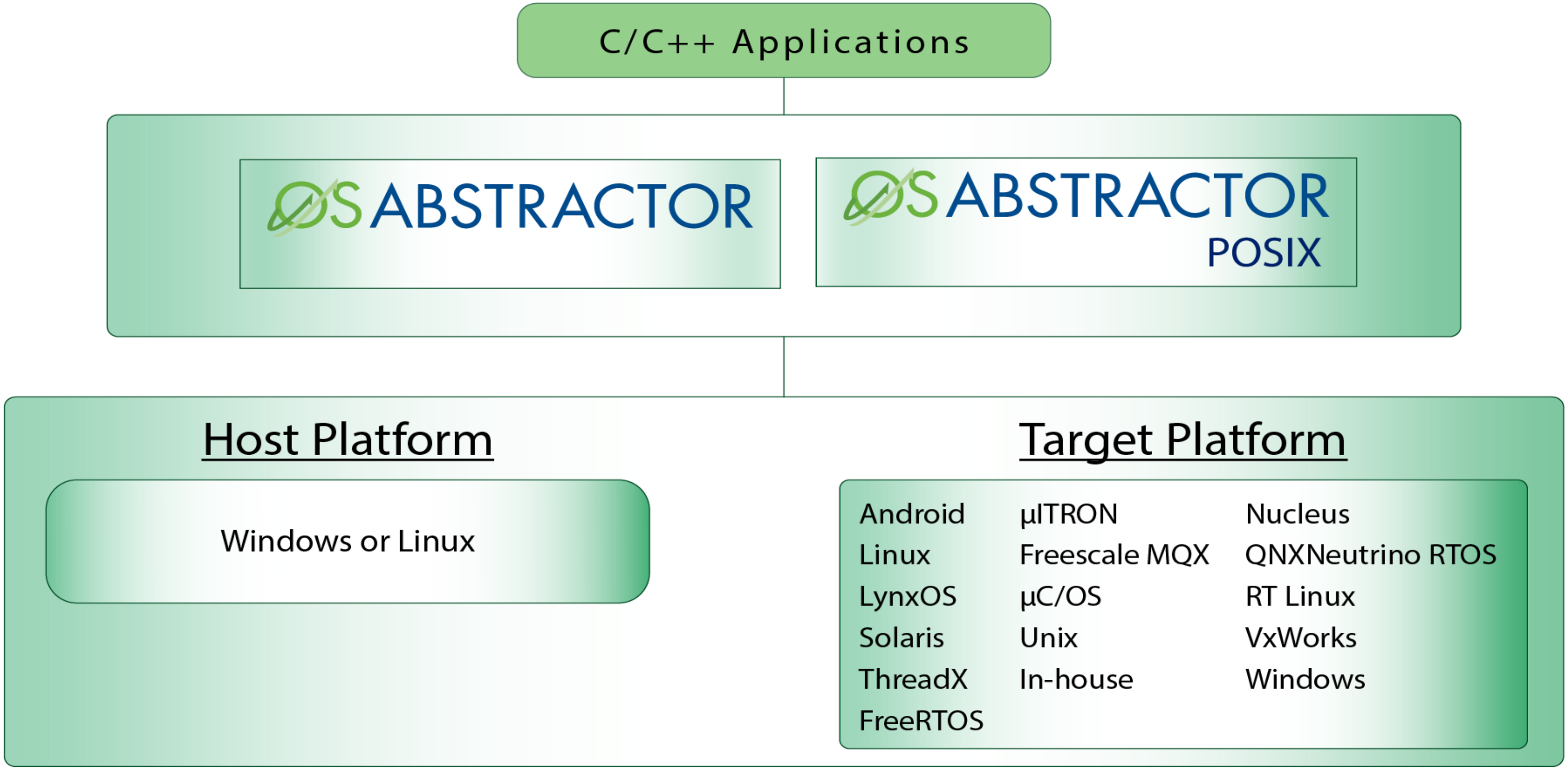
- Library Package Generator
  - Full source code libraries for the Cross-OS Development Platform Interface(s) and OS Abstractor Target Specific Module for your target platform
  - Sample demo applications
  - Project build files for supported tools and IDEs for your target environment
- Optimized Target Code Generator
  - Generates the Cross-OS Development Platform Interface(s) and OS Abstractor Target Specific code
  - Module source code, specifically optimized for your application and target environment
  - Creates project files for your target IDE
  - Includes the system settings you chose in the GUI-based Wizard
- OS Simulator for host application development for your chosen Operating System(s)
- Profiler to view performance data of your application and Cross-OS Development Platform Interface(s) for your target

MAPUSOFT

OS ABSTRACTOR

RTOS Abstraction Layer

Develop code independent of the OS



C/C++ Applications

OS ABSTRACTOR

OS ABSTRACTOR POSIX

Host Platform

Windows or Linux

Target Platform

Android
Linux
LynxOS
Solaris
ThreadX
FreeRTOS

µITRON
Freescale MQX
µC/OS
Unix
In-house

Nucleus
QNXNeutrino RTOS
RT Linux
VxWorks
Windows

MapuSoft's OS Abstraction Layer (OSAL)

- OS Abstractor APIs give users the ability to effectively develop code independent of the underlying OS.
- Protects your software investment and easily expands support to multiple operating systems.
- It allows developers to use a standard OS API interface across multiple OS platforms and greatly reduce the cost associated with code maintenance and learning multiple operating systems.
- It is integrated with AppCOE which provides a multiple OS interface host environment with provisions to generate optimized code for a wide variety of target OS platforms.

OS Abstractor Target Specific Module: Performance Features

- Not your typical wrapper
  - Provides most of the OS features by itself and does not depend on the OS, except for a few features (ex. priority scheduling, change priority, semaphore, messaging, thread suspend/resume)
- Quick support for a new OS
  - MapuSoft can easily add support to a new commercial or in-house OS, typically in two weeks
- Process support to any OS
  - Add software-based process and shared memory functionality to an OS, even if the OS does not have those features
- Advanced process memory allocation scheme
  - Applications can allocate required system heap memory during process creation to ensure that they will always have the required system memory
  - Setting memory limits prevents an application from using up all system memory and impacting others
- Thread pooling
  - Applications can pool threads to increase platform robustness & performance by eliminating the overhead associated with actual task creation & deletion at run-time

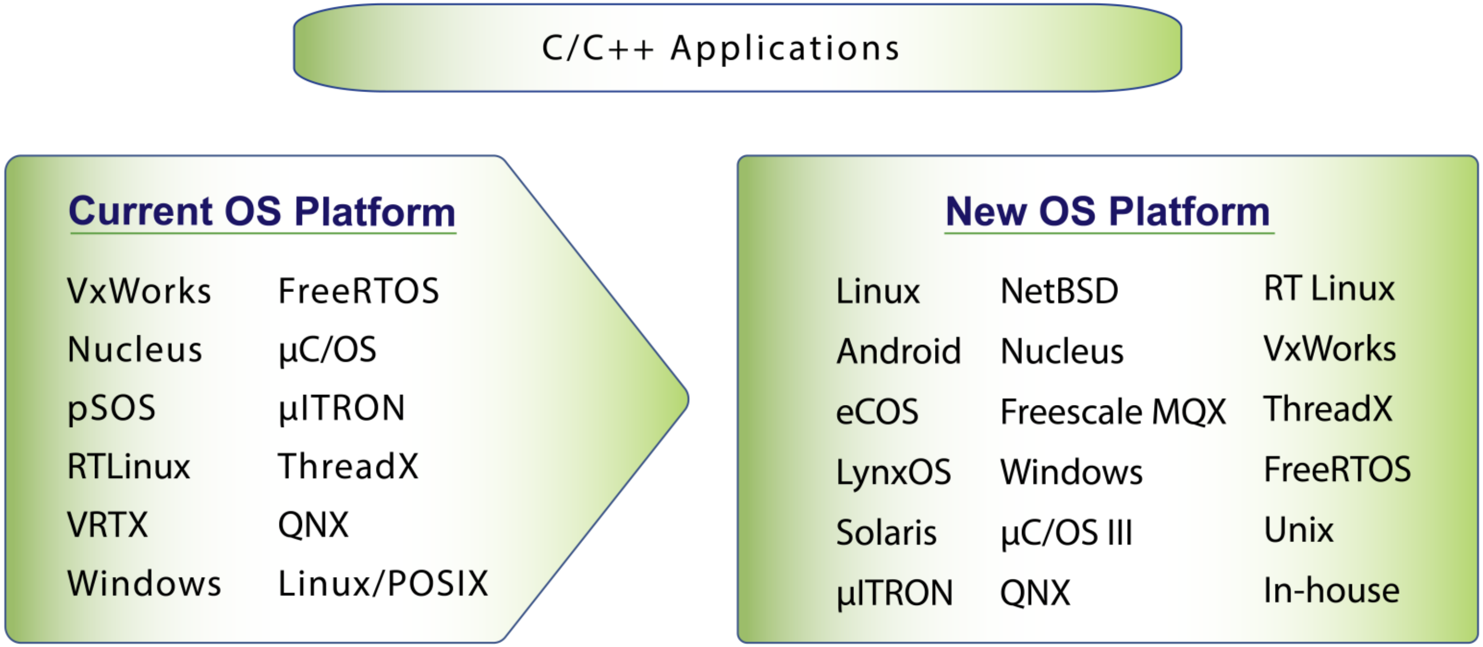
OS Abstractor Target Specific Module: Performance Features

- Mission Critical Features
  - Applications have the ability to recover from software fatal errors through a soft reset by rolling the stack back to the start of the application
- API Flexibility
  - Use one or more of the Cross-OS Development Platform Interface(s)
  - Cross-OS Development Interface(s) can also be used within a single or across multiple applications
  - Combine applications written with different OS APIs and run them on one or many OS
- Zero copy message queues
  - Cross-OS Queue APIs will not introduce data read and copy overhead
- API and application profiling, plus API optimization
  - Profile applications and Cross-OS Interface(s) functions on your target
  - Optimize individual Cross-OS Interface functions based on profiler data
- Scalability specific to your application during code generation
  - AppCOE reads your application to custom generate Interface code that is specific to your application to increase the performance with reduced memory footprint

MAPUSOFT

OS CHANGER®

Easily migrate your code to a new OS



C/C++ Applications

Current OS Platform

- VxWorks
- Nucleus
- pSOS
- RTLinux
- VRTX
- Windows
- FreeRTOS
- µC/OS
- µITRON
- ThreadX
- QNX
- Linux/POSIX

New OS Platform

- Linux
- Android
- eCOS
- LynxOS
- Solaris
- µITRON
- NetBSD
- Nucleus
- Freescale MQX
- Windows
- µC/OS III
- QNX
- RT Linux
- VxWorks
- ThreadX
- FreeRTOS
- Unix
- In-house

ROI: Cost of Porting – Manual vs. OS Changer

Non-Mission Critical Application	Small	Medium	Large
	10K LOC	100K LOC	1M LOC
Manual Porting of Single Application *	$ 95,338	$ 953,380	$ 9,533,800
OS Changer License Fees	$ 12,000 [1]	$ 43,200 [2]	$ 72,000 [3]
Estimated Labor to use OS Changer	$ 8,000	$ 16,000	$ 64,000
Cost of Porting with OS Changer	$ 20,000	$ 59,200	$ 136,000

([1] = Single Seat License, [2] = 10 seat License, [3] = Source Code Site License)

Savings (Manual minus OS Changer)	$ 75,338	$ 918,780	$ 9,397,800
Return on Investment	377 %	2,436 %	6,910 %

* Source: Software Engineering Institute, Carnagie Mellon, 2008 Bureau of Labor Statistics, 2006. Porting estimated at half the cost of new development.

OS Changer Porting Kit

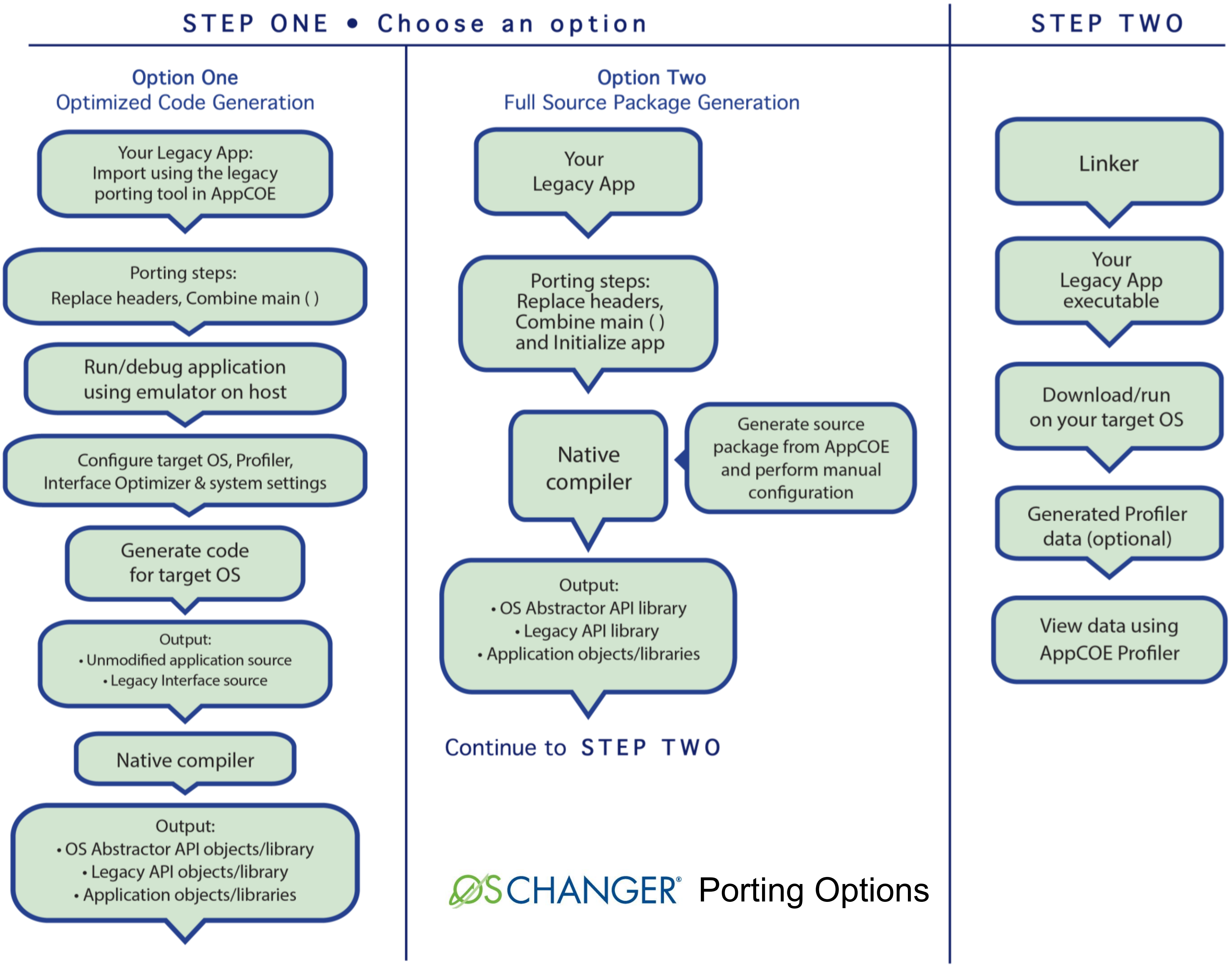
- OS Changer Porting Kit enables automated migration of C/C++ software to a new OS or RTOS. It saves time and money because the C/C++ code remains unchanged and the tedious and error-prone porting work is no longer required.
- It includes one OS Changer Interface for the OS you are moving from and a OS Abstractor Target Specific Module for the target OS you are porting to
- OS Changer Porting Kit includes OS Simulator for host application development for your chosen Operating System
- OS Changer Porting Kit is integrated with AppCOE, an Eclipse-based framework, providing a state-of-the-art IDE for development and testing

OS Changer Porting Kit includes:

- Library Package Generator
  - Full library source code of the OS Changer Interface and OS Abstractor Target Specific Module for your target platform
  - Sample demo applications
  - Project build files for supported tools & IDEs for your target environment

- Optimized Target Code Generator
  - Generates the OS Changer Porting Interface and OS Abstractor Target Specific Module source code, specifically optimized for your application and target environment
  - Creates project files for your target IDE
  - Includes the system settings you chose in the GUI-based Wizard

- Profiler to view performance data of your application and OS Changer Interface for your target



STEP ONE • Choose an option

Option One
Optimized Code Generation

Your Legacy App: Import using the legacy porting tool in AppCOE

Porting steps: Replace headers, Combine main ( )

Run/debug application using emulator on host

Configure target OS, Profiler, Interface Optimizer & system settings

Generate code for target OS

Output:
- Unmodified application source
- Legacy Interface source

Native compiler

Output:
- OS Abstractor API objects/library
- Legacy API objects/library
- Application objects/libraries

Option Two
Full Source Package Generation

Your Legacy App

Porting steps: Replace headers, Combine main ( ) and Initialize app

Generate source package from AppCOE and perform manual configuration

Native compiler

Output:
- OS Abstractor API library
- Legacy API library
- Application objects/libraries

Continue to STEP TWO

STEP TWO

Linker

Your Legacy App executable

Download/run on your target OS

Generated Profiler data (optional)

View data using AppCOE Profiler

OSCHANGER® Porting Options

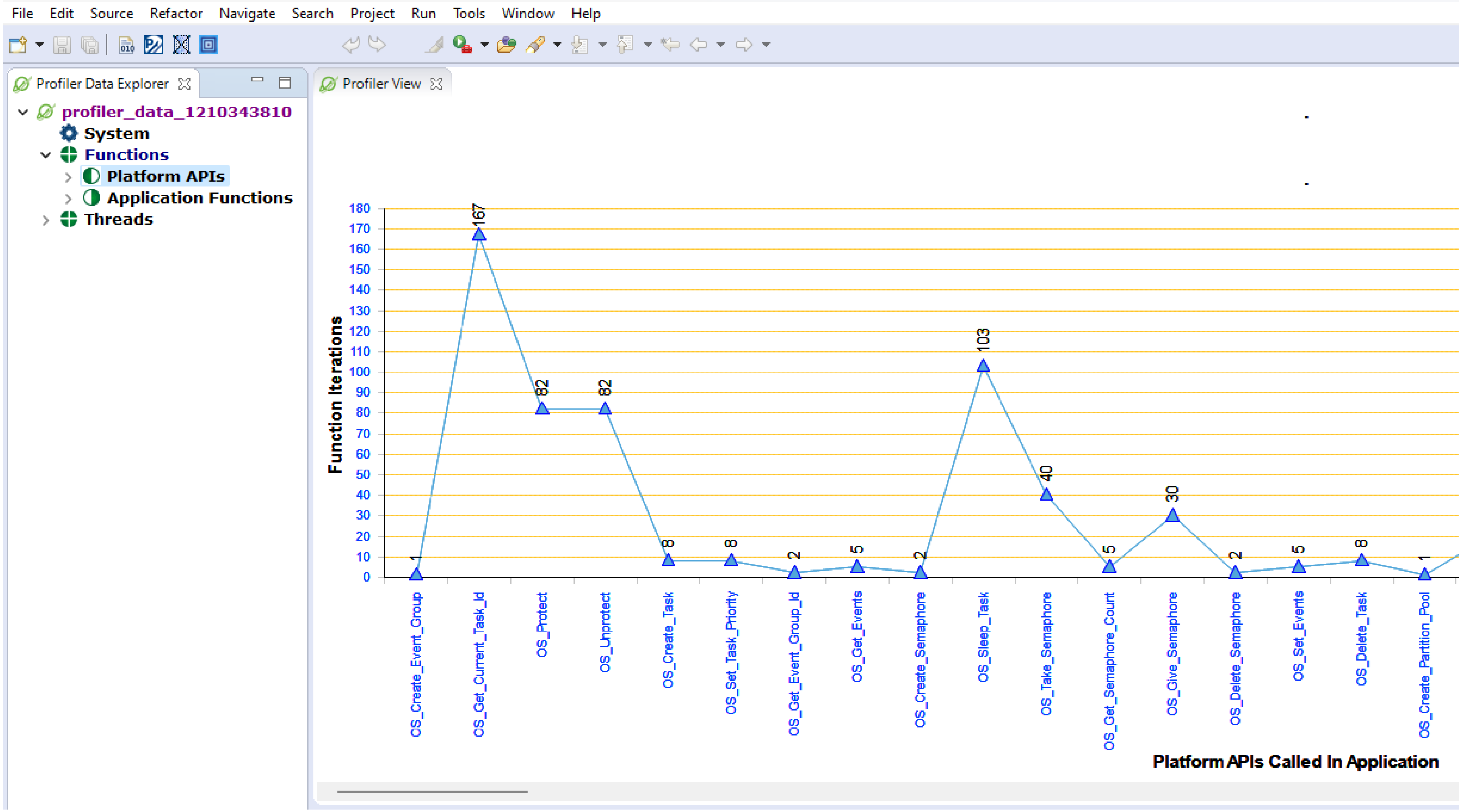
Application-OS Profiler

- When Application runs on target, the profiler starts, and once data collected creates a data file. View data off-line in AppCOE for analyzing Application performance & bottle-necks
- Collect execution timing & execution count Application functions and the calls made by the application to the OS via Cross-OS APIs
- Graphically view data in chart formats system-wide or per task
- Generate report to identify best/worst/average execution times
- Generate comparison report on Application's performance on different environment (like App Configuration, OS, Hardware, etc.)
- Configure profiler task priority, # of records to collect and select functions to profile via AppCOE code generation GUI

MAPUSOFT

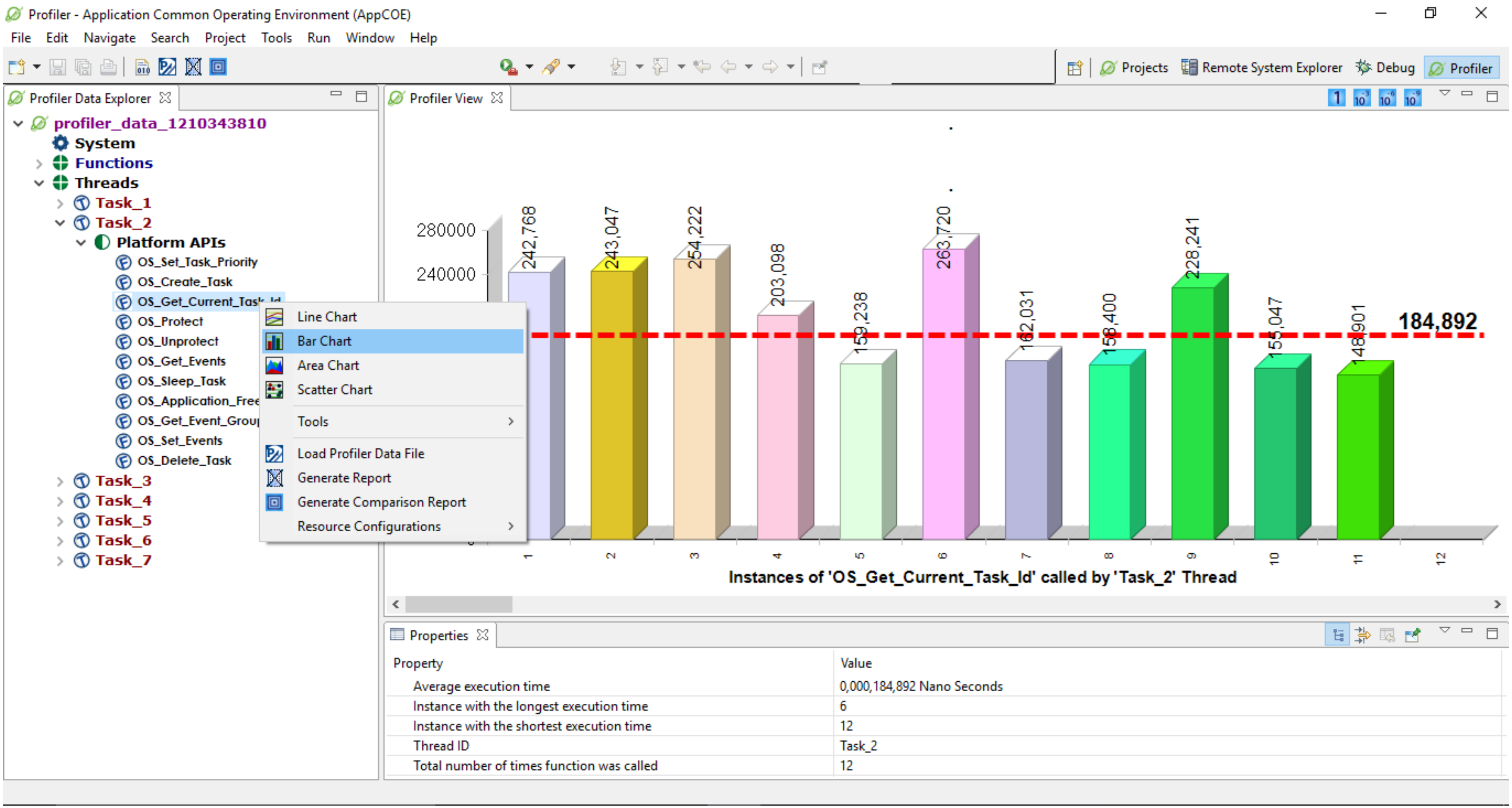
Application-OS Profiler

Execution Count – Line Graph



Application-OS Profiler

Performance Data – Bar Graph

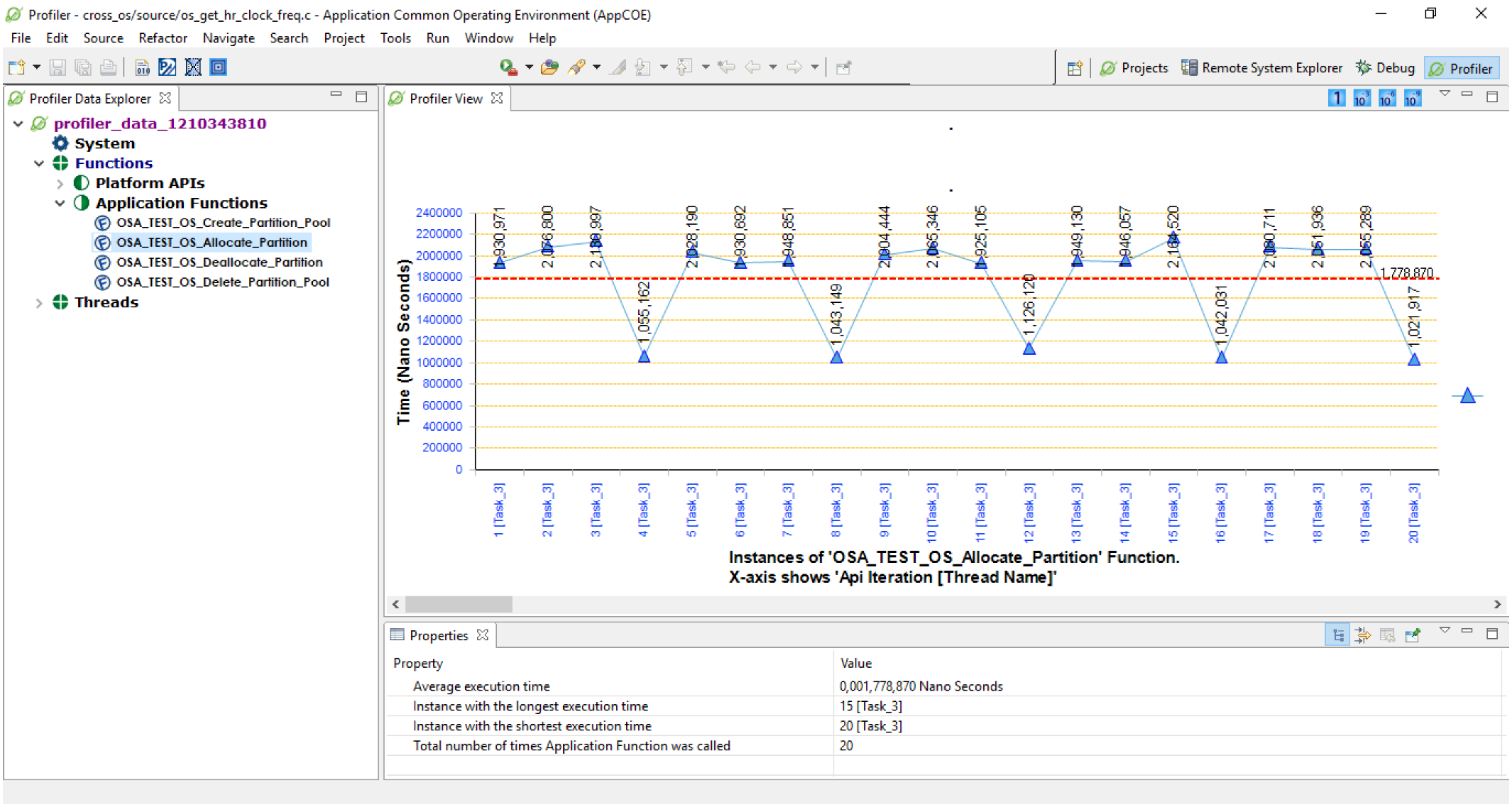


Property	Value
Average execution time	0,000,184,892 Nano Seconds
Instance with the longest execution time	6
Instance with the shortest execution time	12
Thread ID	Task_2
Total number of times function was called	12

MAPUSOFT

Application-OS Profiler

Performance Data – Line Graph



Profiler - cross_os/source/os_get_hr_clock_freq.c - Application Common Operating Environment (AppCOE)

- profiler_data_1210343810
  - System
  - Functions
    - Platform APIs
    - Application Functions
      - OSA_TEST_OS_Create_Partition_Pool
      - OSA_TEST_OS_Allocate_Partition
      - OSA_TEST_OS_Deallocate_Partition
      - OSA_TEST_OS_Delete_Partition_Pool
  - Threads

Instances of 'OSA_TEST_OS_Allocate_Partition' Function.
X-axis shows 'Api Iteration [Thread Name]'

Property	Value
Average execution time	0,001,778,870 Nano Seconds
Instance with the longest execution time	15 [Task_3]
Instance with the shortest execution time	20 [Task_3]
Total number of times Application Function was called	20

MAPUSOFT®

RTOS SIMULATOR™

SIMULATE, DEVELOP AND DEBUG EMBEDDED APPLICATIONS ON HOST AND TARGET HARDWARE

RTOS Simulator eliminates the need for the original OS, as well as the requirement for expensive target hardware during development on a host computer. RTOS Simulator allows you to develop product prototypes on inexpensive target hardware. The result is lower licensing costs, reduced hardware requirements and a shorter time to market. RTOS Simulators are available for eleven popular OS Platforms.

RTOS SIMULATOR for Developers

Develop, Run and Test Applications on a Virtualized Host Platform

RTOS SIMULATOR for Academics

Develop, Run and Test Applications on Windows or Linux Host and Test on Hardware

RTOS Simulator

- Develop and test embedded applications on Windows or Linux host environments

- Eliminates the need for the original OS and expensive target hardware during development
  - Results in lower licensing costs, reduced hardware requirements and a shorter time to market

- Provides a virtualized test platform
  - Simulate a system of applications interacting with each other on one or more CPU cores through simulated devices

- Available Simulators
  - VxWorks®
  - Linux/POSIX
  - micro-ITRON
  - Nucleus ®
  - Windows ®
  - pSOS ®
  - ThreadX ®
  - OS Abstractor ®
  - µC/OS™
  - FreeRTOS™
  - VRTX ®
  - QNX ®
  - RTLinux ®

RTOS Simulator as a Development Platform

- Allows you to simulate real-time applications on Windows or Linux hosts by hardening and optimizing the underlying OS platform
- Includes the Simulated OS Interface and OS Abstractor Interface in non-source, object format
  - Allows for development, simulation, testing and integration of embedded applications on a x86 host environment
  - Optionally, the Simulated OS Interface and OS Abstractor Interface libraries are available in full source code format for use with target native tools/software on Windows or Linux
- RTOS Simulator is integrated with AppCOE, an Eclipse-based framework, providing a state-of-the-art IDE for development and testing

RTOS Simulator as a Virtualized Testing Platform

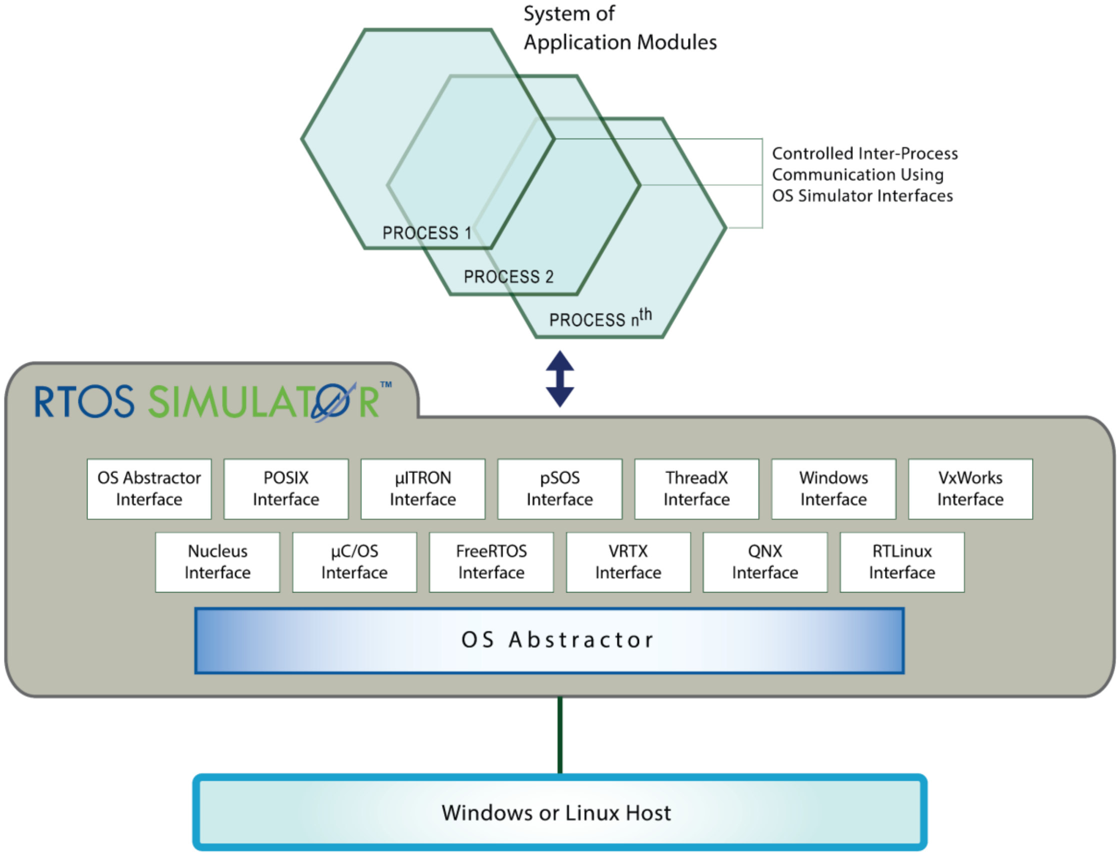
- Host-based testing offers more hardware resources, less complexity and lower costs when compared to testing on a target platform

- Ability to do modular testing
    - Allows applications to be broken down into separate processes with dedicated heap memory and kernel resources so that individual modules can be debugged further without crashing others
    - Can load or re-start other processes dynamically from within an application or statically load them from command prompt
    - Simulate interacting applications with each other on one or more CPU cores using Simulated OS Interface and OS Abstractor Interface for inter-process communication
    - Shared tiered memory pools allow effective application-level data communication with zero copy and without having to pass through physical or virtual devices

- Easy to develop a prototype by creating low-level simulated devices using OS Abstractor APIs

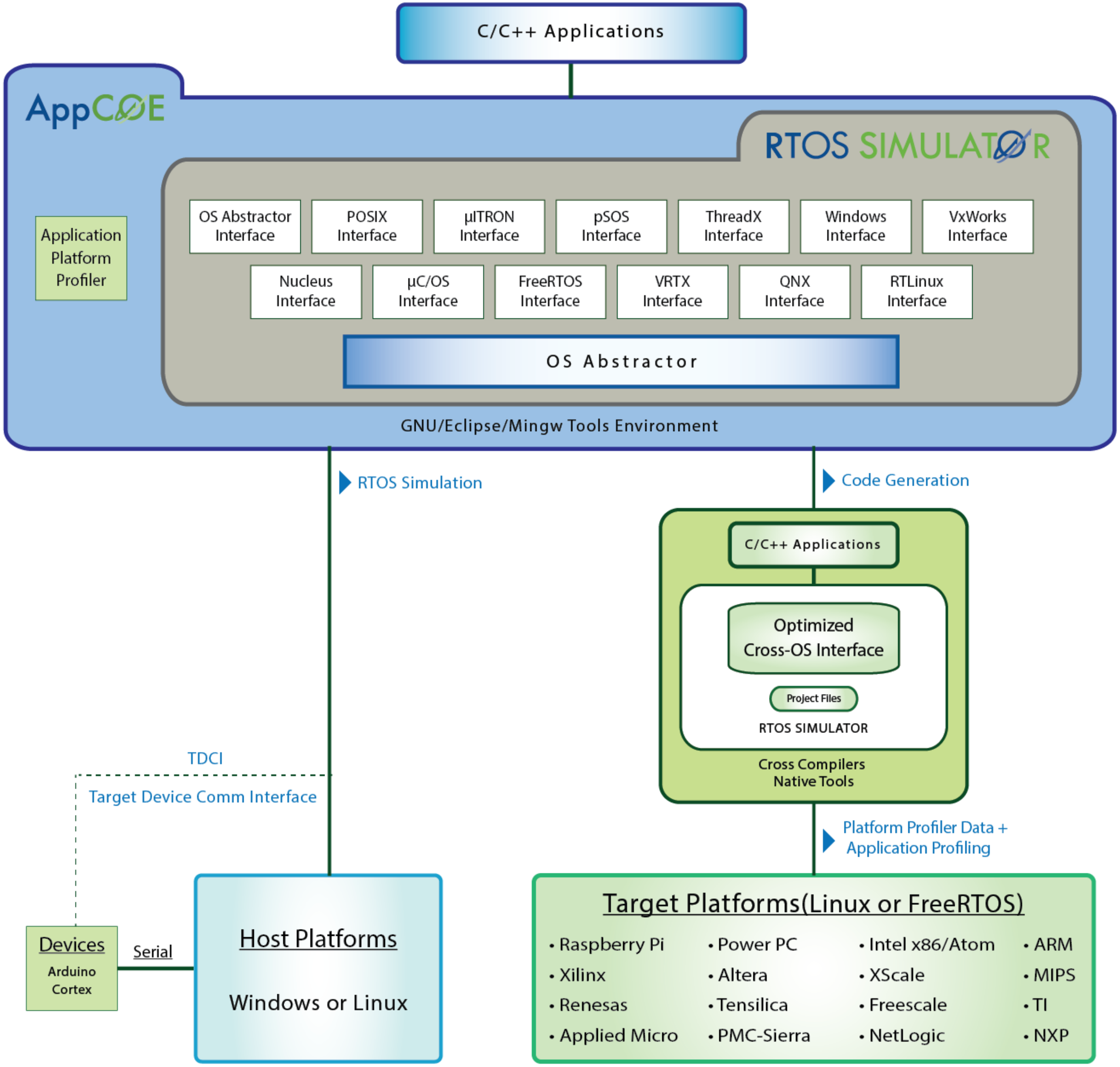
MAPUSOFT

RTOS Simulator as a Virtualized Testing Platform



RTOS Simulator for Academics

- RTOS Simulator helps students develop, integrate, run, debug and test RTOS applications on Windows and Linux host platforms.
- The integrated development environment supports all popular and commonly used RTOS interfaces with code generation and profiling capabilities configured for different hardware such as the ARM, x86, PowerPC etc.
- Developed applications can be deployed and run on target hardware utilizing the included specific target library components.
- Includes Raspberry Pi support package plus one Raspberry Pi board



C/C++ Applications

AppCOE

RTOS SIMULATOR

Application Platform Profiler

OS Abstractor Interface

POSIX Interface

µITRON Interface

pSOS Interface

ThreadX Interface

Windows Interface

VxWorks Interface

Nucleus Interface

µC/OS Interface

FreeRTOS Interface

VRTX Interface

QNX Interface

RTLinux Interface

OS Abstractor

GNU/Eclipse/Mingw Tools Environment

RTOS Simulation

Code Generation

C/C++ Applications

Optimized Cross-OS Interface

Project Files

RTOS SIMULATOR

Cross Compilers
Native Tools

Platform Profiler Data +
Application Profiling

TDCI

Target Device Comm Interface

Devices
Arduino
Cortex

Serial

Host Platforms

Windows or Linux

Target Platforms(Linux or FreeRTOS)

- Raspberry Pi
- Xilinx
- Renesas
- Applied Micro
- Power PC
- Altera
- Tensilica
- PMC-Sierra
- Intel x86/Atom
- XScale
- Freescale
- NetLogic
- ARM
- MIPS
- TI
- NXP

MAPUSOFT

RTOS Simulator for Academics

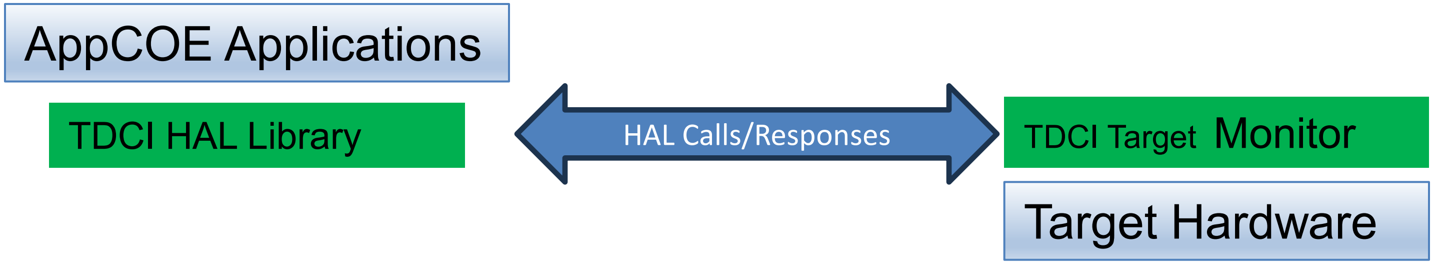
Software Tools for Simulation of Real-Time Systems

- Virtualized low-cost RTOS platforms using AppCOE on Windows/Linux
  - Develop, integrate, run, debug and test RTOS applications on host
  - Target Device Comm Interface: Connect host application to use hardware devices
  - Supports popular RTOS interfaces including VxWorks, Windows, µC/OS, RTLinux, Linux/POSIX, FreeRTOS, micro-ITRON, ThreadX, Nucleus, OS Abstractor and QNX
  - Optimized code generation with profiling capabilities for applications to run on hardware such as the Raspberry Pi over Linux/FreeRTOS
  - Develop applications to run directly over Arduino target
- Advanced embedded lab exercises for RTOS, IOT Devices, Network Socket Interface with lab manuals and associated source code
- Includes Hardware kits, Student Certificates & world-class technical support

TDCI Architecture

- TDCI Host Library - Contains HAL api's for supporting multiple target hardware
- TDCI Target Module – Target monitor program that resides on the target hardware

- Communication between TDCI host-based HAL interface and the target hardware

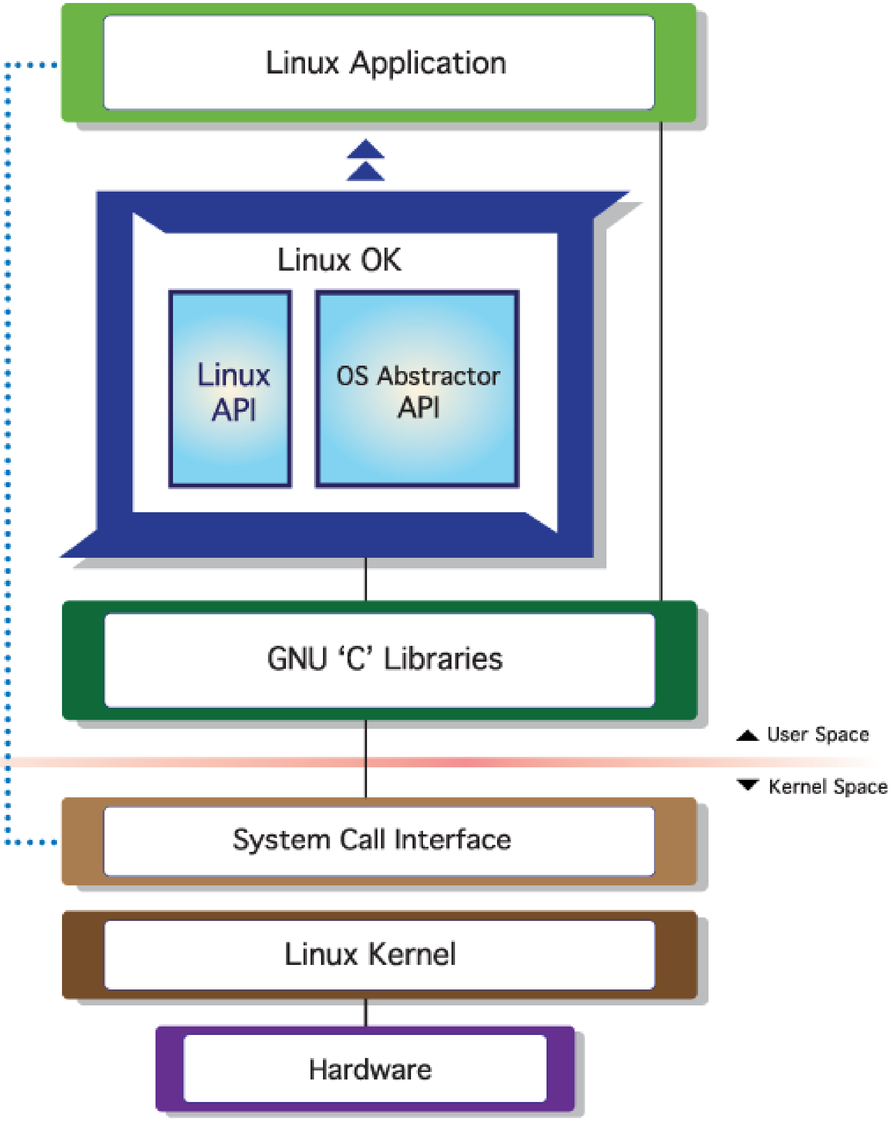


- ○ AppCOE application makes hardware API calls using TDCI host library.
- ○ Target monitor program receive these calls, executes them, and send response back to host

MAPUSOFT

LinuxOK®
Optimization Kit

Increase performance and stability of Linux Applications



Linux Optimization Kit

- Linux Optimization Kit allows users to achieve higher application performance without making changes to the Linux OS kernel.
- Enhanced performance is provided through OS Abstractor target specific module
- Included Linux/POSIX Interface provides better coverage and portability
  - Ability to have one code base for multiple POSIX variants
- Provides header files and template code to re-direct the application to use the Linux OK interfaces

Linux Optimization Kit

- Generates optimized interface code, based on the application's API usage, and according to GUI settings input by the user

- Reduces run-time creation and deletion of OS resources by pooling and reusing them across applications/processes

- Reduces unnecessary task and user/kernel mode switching

- Enhances Linux application performance
  - Statically create and reuse OS resource control blocks across one or more applications
  - Statically create and reuse threads from pool
  - Enhance performance without breaking Linux API compliance or modifying your application
  - GUI to configure target environment and resource requirements for various Linux kernel versions and variants (32/64 bit; SMP/UP systems)

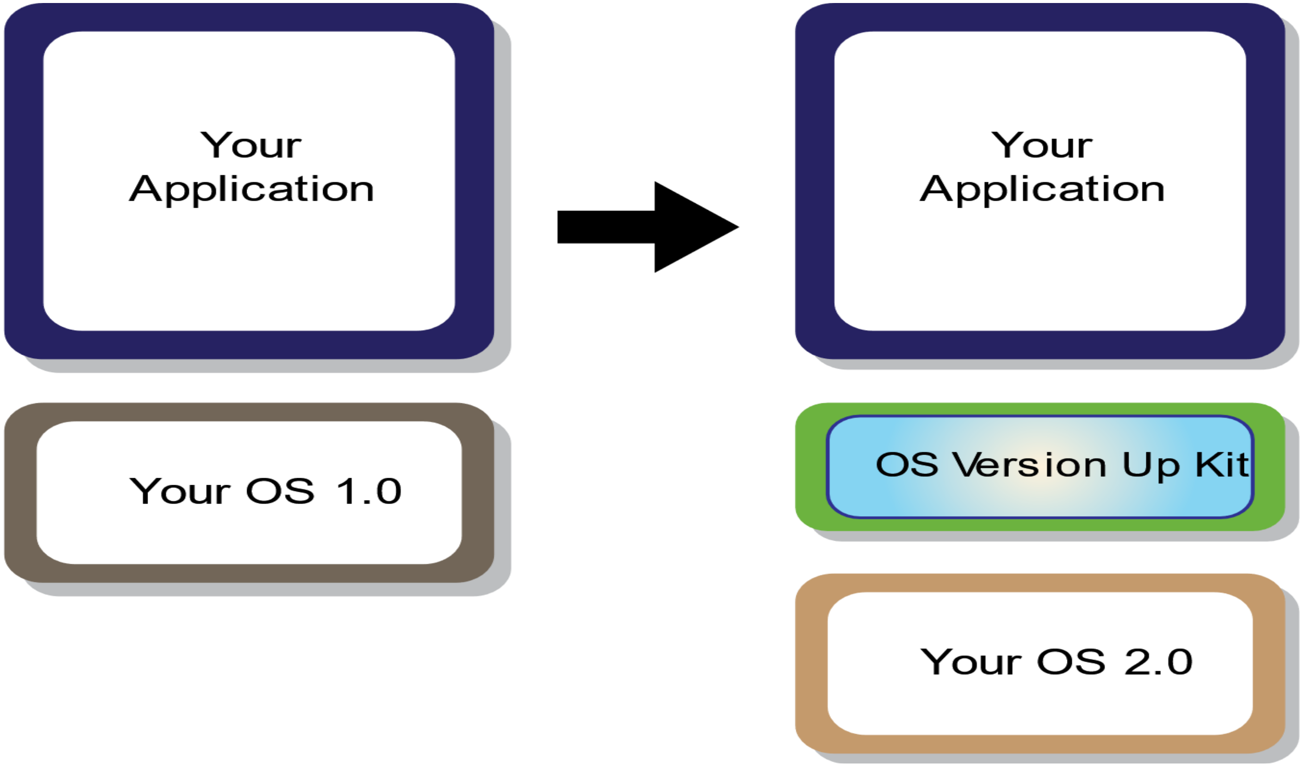
MAPUSOFT

Linux Optimization Kit

- OS Abstractor Interface for advanced features
  - Tiered memory pools to eliminate memory fragmentation
  - Shared memory pool to eliminate data copy across applications
  - Automatically recover from software fatal errors through a soft reset
  - Use the same Interface for inter-process communication within and across individual application processes

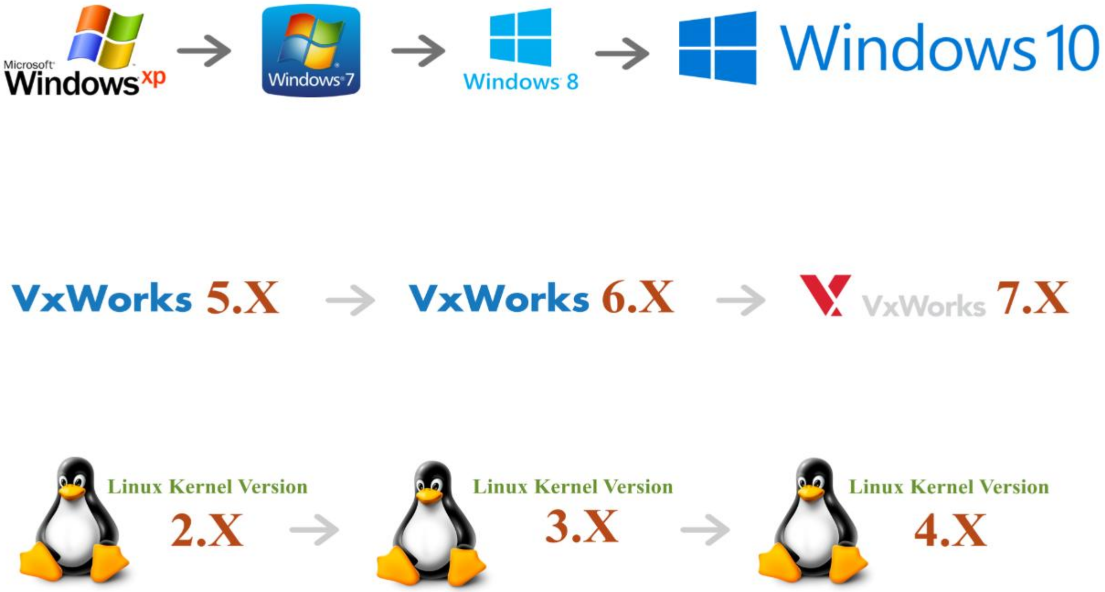
- Scalability and Optimization specific to your application
  - Profiler to identify performance bottlenecks and level of API usage so that specific optimizations can be made
  - Generate project files for your target

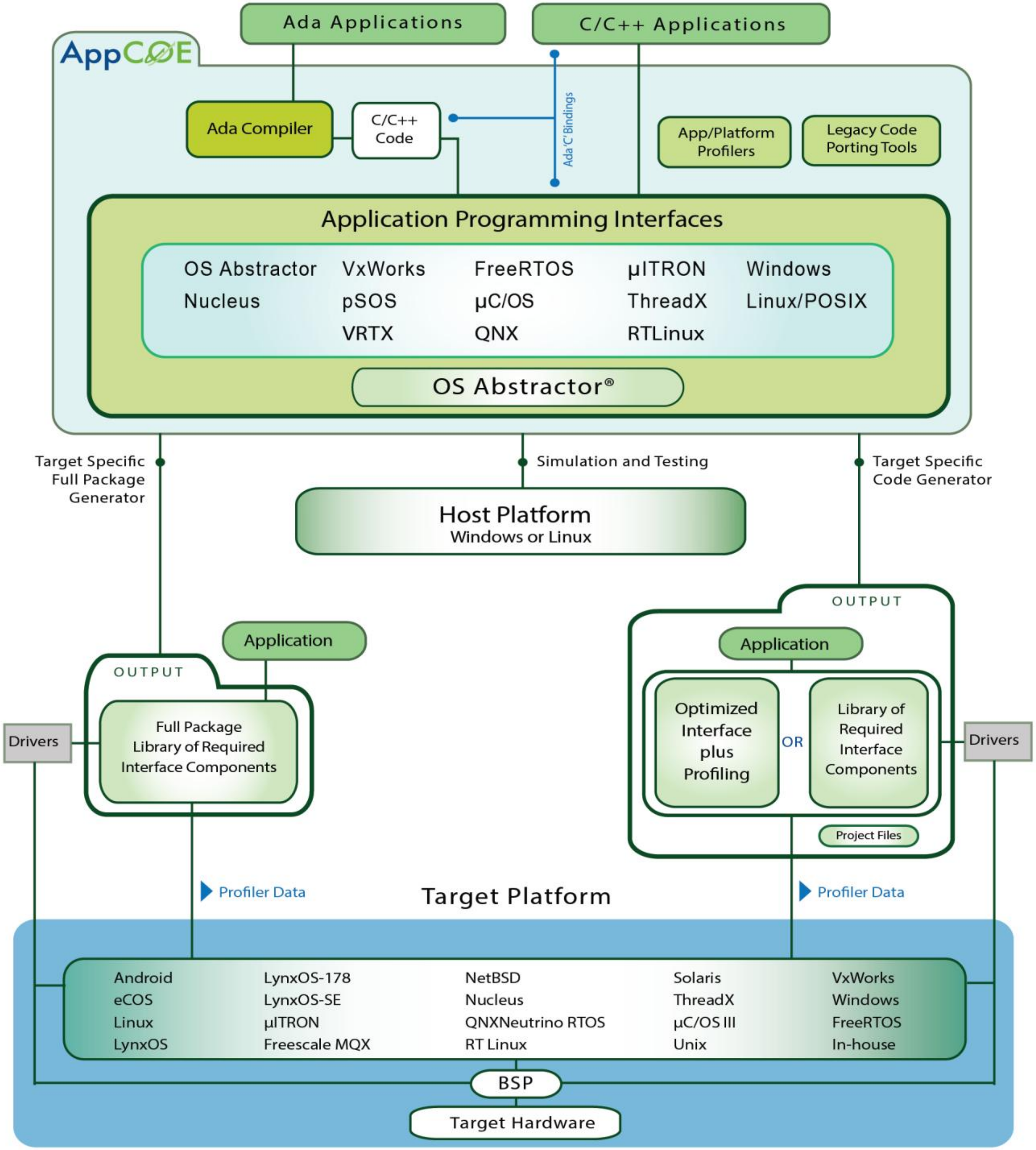
Easily upgrade your application to a new OS Version



OS Version UpKit

- OS Version UpKit allows applications to easily upgrade to newer versions of an OS, without the manual porting effort. It saves a huge amount of time and money!





Ada Applications

C/C++ Applications

AppCOE

Ada Compiler

C/C++ Code

Ada 'C' Bindings

App/Platform Profilers

Legacy Code Porting Tools

Application Programming Interfaces

OS Abstractor, Nucleus, VxWorks, pSOS, VRTX, FreeRTOS, µC/OS, QNX, µITRON, ThreadX, RTLinux, Windows, Linux/POSIX

OS Abstractor®

Target Specific Full Package Generator

Simulation and Testing

Target Specific Code Generator

Host Platform

Windows or Linux

OUTPUT

Application

Full Package Library of Required Interface Components

Drivers

OUTPUT

Application

Optimized Interface plus Profiling

OR

Library of Required Interface Components

Project Files

Drivers

Profiler Data

Target Platform

Profiler Data

Android, eCOS, Linux, LynxOS, LynxOS-178, LynxOS-SE, µITRON, Freescale MQX, NetBSD, Nucleus, QNXNeutrino RTOS, RT Linux, Solaris, ThreadX, µC/OS III, Unix, VxWorks, Windows, FreeRTOS, In-house

BSP

Target Hardware

AppCOE™

Application Common Operating Environment

MAPUSOFT®

Ada-Changer

Automated Conversion of Ada Code to Other Programming Languages

Ada-Changer is the most reliable and powerful Ada language conversion solution on the market. With its Modeling and Re-factoring capability, Ada-Changer not only modernizes the code by converting it to a modern language but also improves the code. Having been developed and tuned by scientists with over 40 years of modernization experience, Ada to C/C++/Java/C# Changer significantly reduces the cost and risk for any Ada modernization project

Ada-C/C++ CHANGER®

Automated Conversion of Ada Code to C/C++

Ada-Java CHANGER®

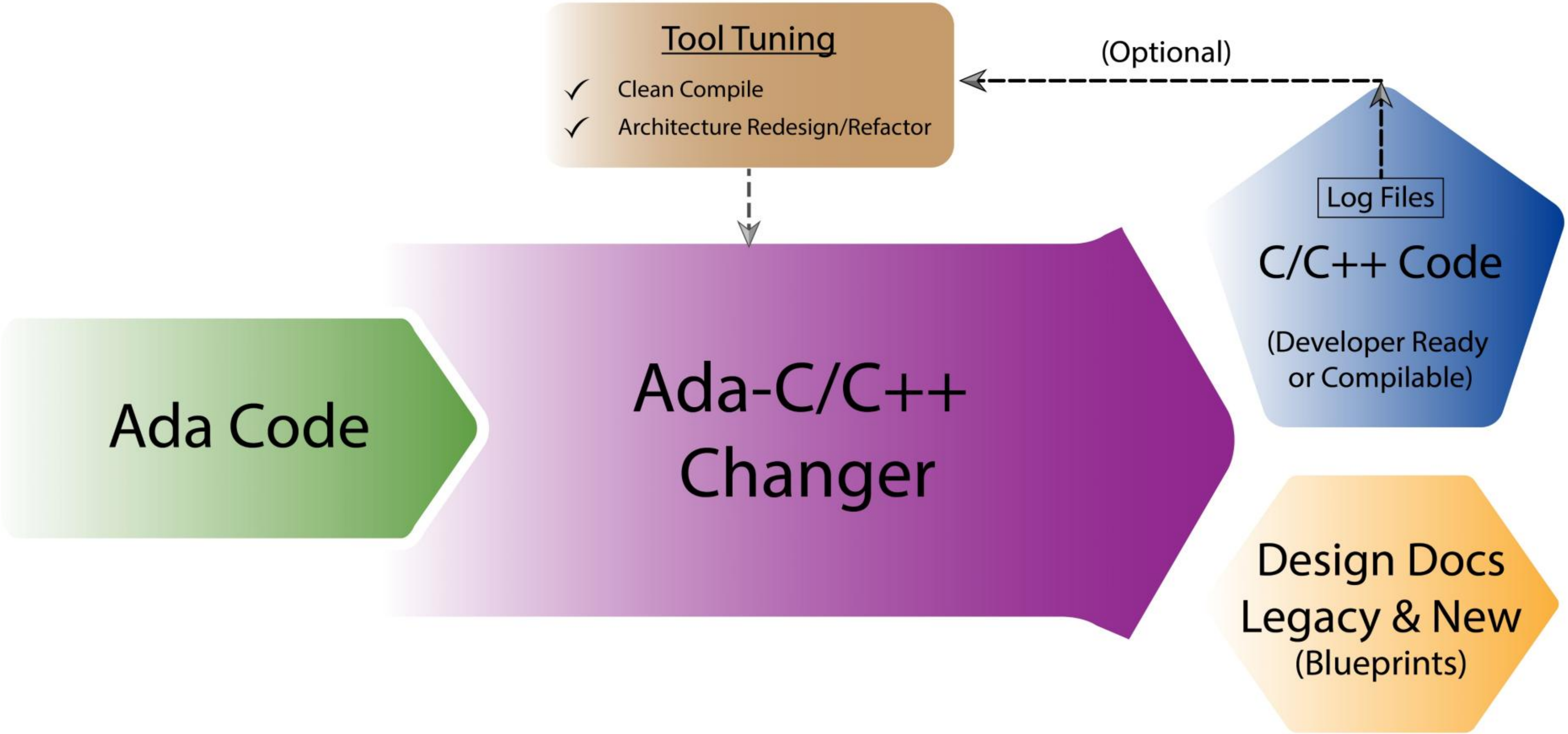
Automated Conversion of Ada Code to Java

Ada-C# CHANGER®

Automated Conversion of Ada Code to C#

MAPUSOFT®

Ada-C/C++ CHANGER®



Tool Tuning

✓ Clean Compile

✓ Architecture Redesign/Refactor

(Optional)

Log Files

Ada Code

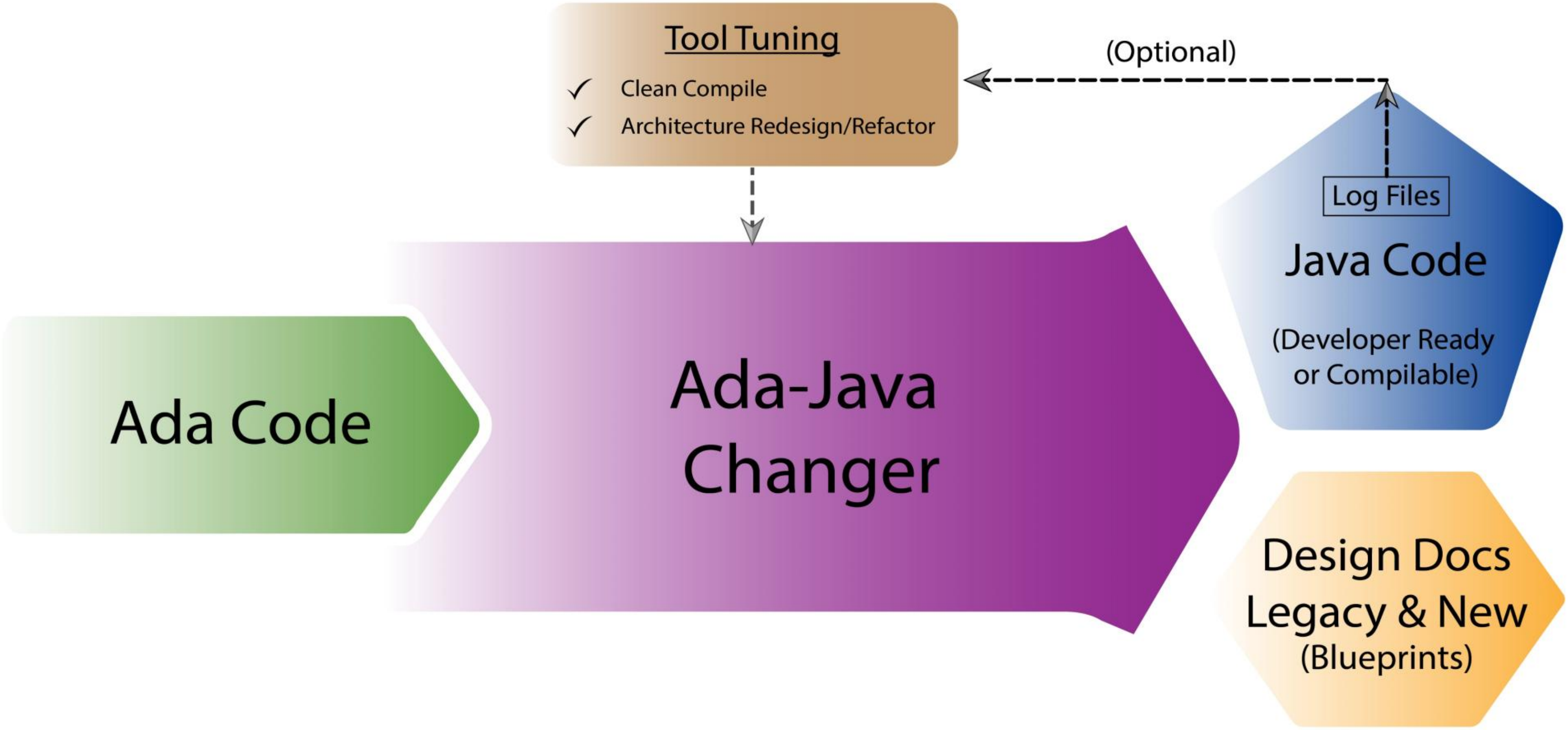
Ada-C/C++ Changer

C/C++ Code

(Developer Ready or Compilable)

Design Docs Legacy & New (Blueprints)

Ada-Java CHANGER®



Tool Tuning

✓ Clean Compile

✓ Architecture Redesign/Refactor

(Optional)

Log Files

Java Code

(Developer Ready or Compilable)

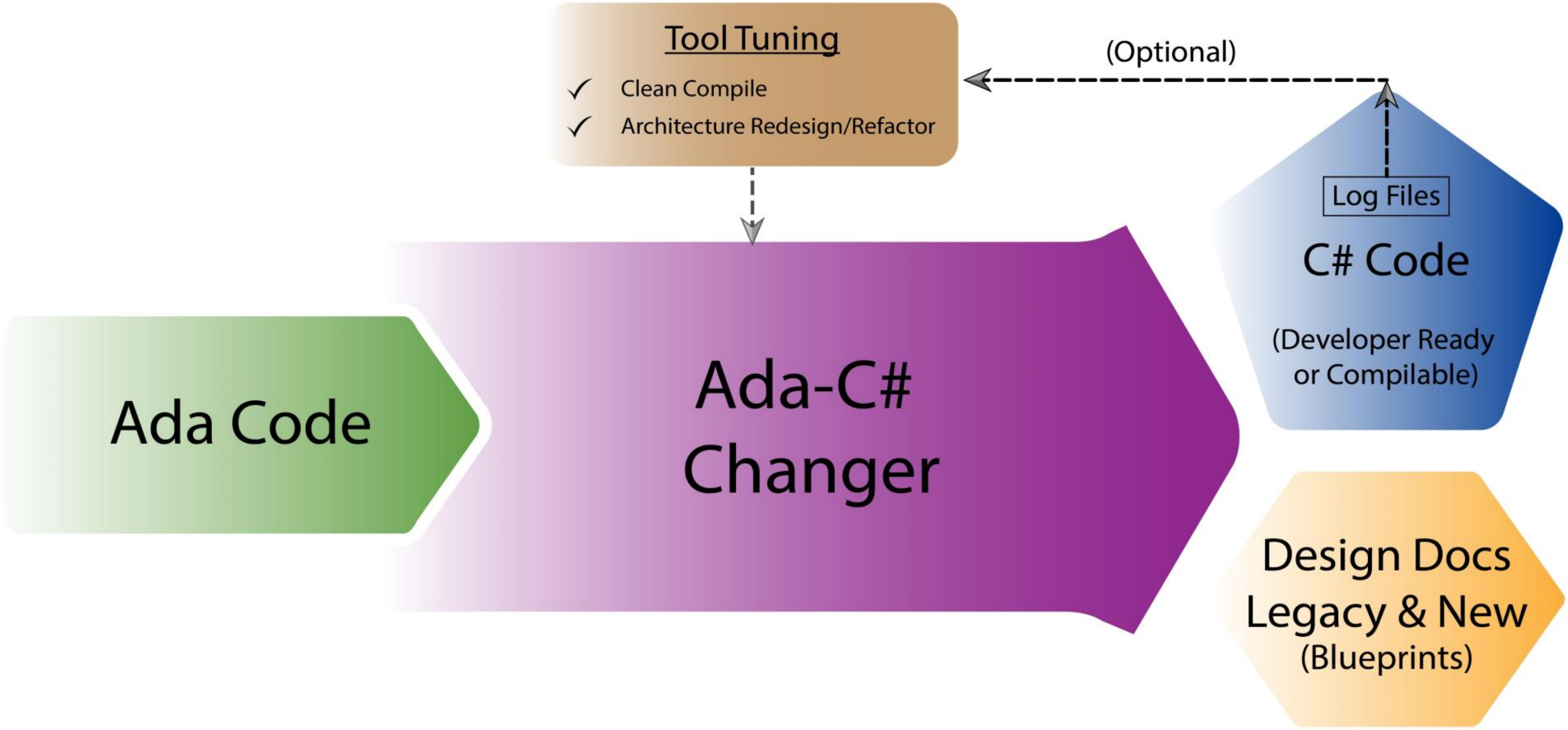
Ada Code

Ada-Java Changer

Design Docs Legacy & New (Blueprints)

MAPUSOFT®

Ada-C# CHANGER®



Tool Tuning

✓ Clean Compile

✓ Architecture Redesign/Refactor

(Optional)

Log Files

C# Code

(Developer Ready or Compilable)

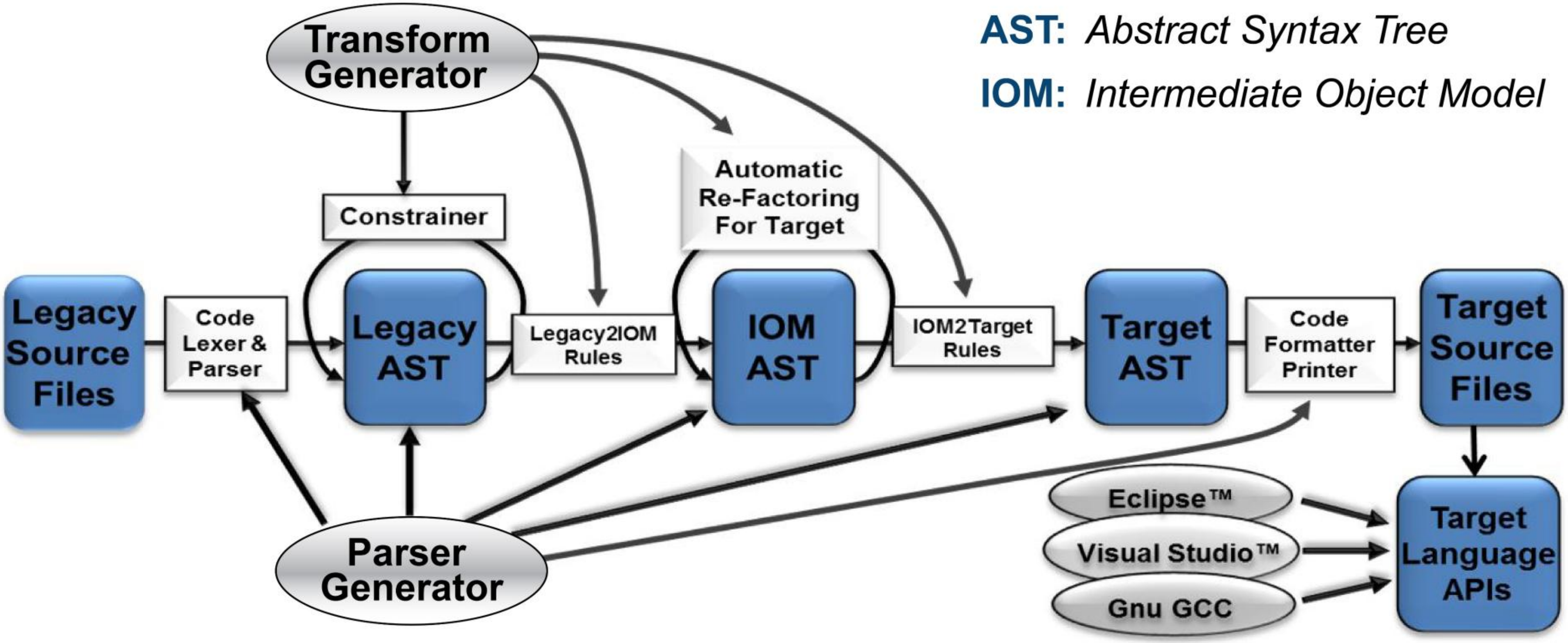
Ada Code

Ada-C# Changer

Design Docs Legacy & New (Blueprints)

MAPUSOFT

Model-Based Rule-Driven Conversion Framework



AST: *Abstract Syntax Tree*

IOM: *Intermediate Object Model*

A Sampling of Past Ada Projects

Over 150+ Automated Modernization Projects since 2000 - 100% Customer Satisfaction

System	Source(s)	Target(s)	SLOC	TTC
TOPSKY – Air Traffic Management – French Version	Ada	Java & C++	495k	12 Months
TOPSKY – Air Traffic Management – Nordic Version	Ada	Java & C++	541k	9 Months
TOPSKY – Air Traffic Management – Australian Version	Ada	Java & C++	638k	9 Months
Variable Message Format Test Tool	Ada	C++	77k	3 Months
Naval Undersea Warfare Center Weapons Control System (WCS)	Ada	C++	800k	7 Months
SAC Strategic Planning System	Ada	C++	40k	2 Months
E-2C Hawkeye Aircraft Mission Fault Isolation Program (MFIP)	Ada	C++	20k	5 Months
Navy Multi-band Terminal (NMT)	Ada	C++	89k	5 Months
Canadian Armed Forces Crypto Material Management System (CMMS)	Ada	C#	20k	4 Months
COBRA DANE Radar Calibration System (SCRS) of Ballistic Missile Early Warning System (BMEWS)	Ada	Java & C++	380k	8 Months
P-3C Orion Aircraft Acoustic Signal Processor System (ASP)	Ada	C++	500k	14 Months
Global Positioning Navigation and Timing Systems (GPNTS)	Ada	C++	221k	8 Months
Aegis ACB-08 Product Line Architecture (Analysis Only)	Ada	C++	1M	1 Month
Advanced Field Artillery Tactical Data System (AFATDS)	Ada	Java	5.5M	9 Months
Hundreds of Other Successful Projects Completed	32+ Source Languages	11+ Targets	1+ Billion	< 12 months

Why use Ada-Changer Instead of Manually Rewriting?

- Conversion with Ada-Changer is completed much faster than manual conversion, meaning the product can be deployed without many years of delay
- Ada-Changer can handle large applications with multi-millions of LOC
- Our technical support team has expertise in Ada, C/C++, target tools and OS platforms
- Eliminate the need for a costly and tedious code re-write for extensive cost and time savings

Ada Conversion - Features

- Supports conversion of Ada 83 and Ada 95 to ANSI C++ 03
- Preserves Ada comments, files structures and variable names
- Converts Ada generics to C++ templates (most cases)
- Supports Ada tasking and rendezvous constructs
- Supports Calendar, System, Unchecked_Conversion, Record Representation Clauses, Variant Records, Declare Block Statements, Ada Tick Functions, Renames (for packages, variables, records, and types), “use” for packages and types, Ada Strings, access (including access all & procedure)
-

Add-on tool features and re-factoring services

- Generate design documents of existing & new code
- Transform Database and User-Interface implementation
- Remove dead or redundant code/data
- Merge and consolidate duplicate code/data
- Reorganize and improve design in new code/data
- Remove flaws in new code that originally existed in old
- Create reusable optimization/packaging/redistribution modules
- Integrate with modern parallel/multi processing and distributed/N-Tier operational environments

Add-on tool features and re-factoring services …

- Modernization of legacy systems to multi-tier architectures
- Modernization of flat file, hierarchical and legacy databases
- Modernization of legacy UI to web or other modern UI
- Re-factor to improve maintainability, security and performance
- Custom pattern-based changes
- Migration to modern cloud architectures
- Testing & integration support solutions
- Extraction of business rules from legacy systems

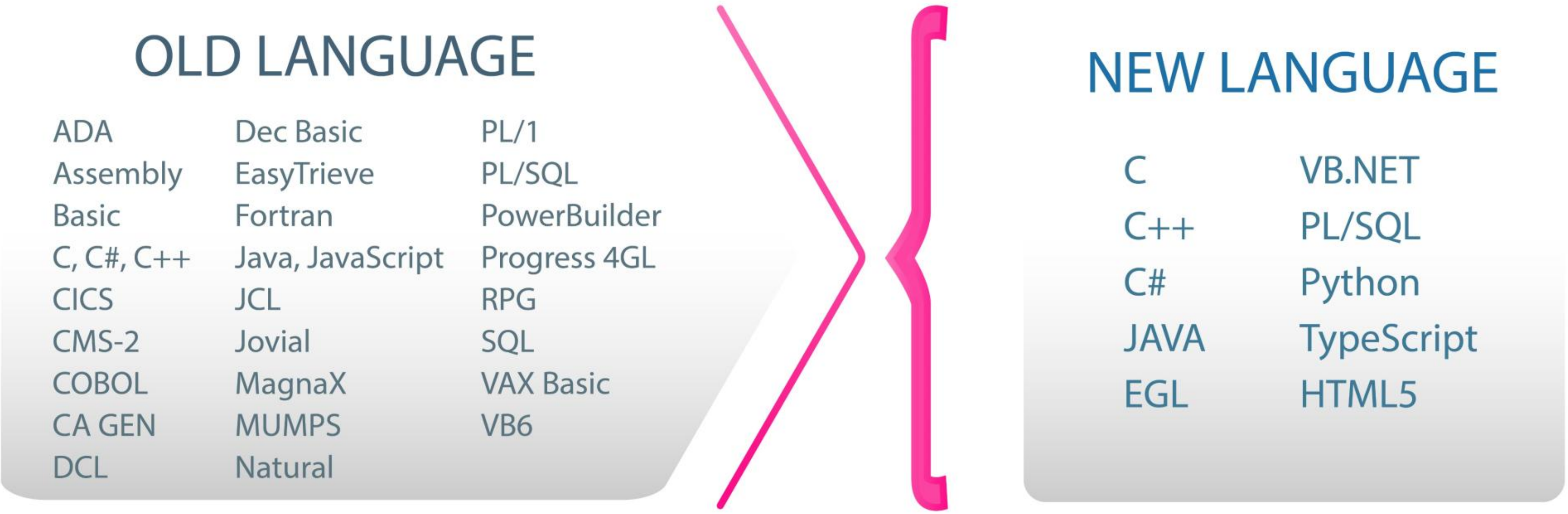
Secured Facilities and Personnel

- Secure Room and Server Facilities
  - Cleared facility and servers for use on classified/secret code
  - Sensitive Compartmented Information Facility (SCIF)
- Cleared Staff
  - Staff maintains active security clearances which are reviewed periodically
  - Over 20 years of modernization experience with highly sensitive code
  - Completed large projects for US and Foreign Governments
- Procedural FSO
  - Employ an FSO with secure code handling procedure

MAPUSOFT

Programming-Language™ Changer

Converts legacy application code from over 30 programming languages into modern languages. The power of model-based automation increases efficiency and accuracy, while significantly reducing cost, errors and risk.



OLD LANGUAGE

- ADA
- Assembly
- Basic
- C, C#, C++
- CICS
- CMS-2
- COBOL
- CA GEN
- DCL
- Dec Basic
- EasyTrieve
- Fortran
- Java, JavaScript
- JCL
- Jovial
- MagnaX
- MUMPS
- Natural
- PL/1
- PL/SQL
- PowerBuilder
- Progress 4GL
- RPG
- SQL
- VAX Basic
- VB6

NEW LANGUAGE

- C
- C++
- C#
- JAVA
- EGL
- VB.NET
- PL/SQL
- Python
- TypeScript
- HTML5

MAPUSOFT

DesignDoc Tools

Auto Generate Design Docs from Your Code

DesignDoc tools can be used to fully understand your existing software in terms of complexity, similarity, dead code, and external dependencies.

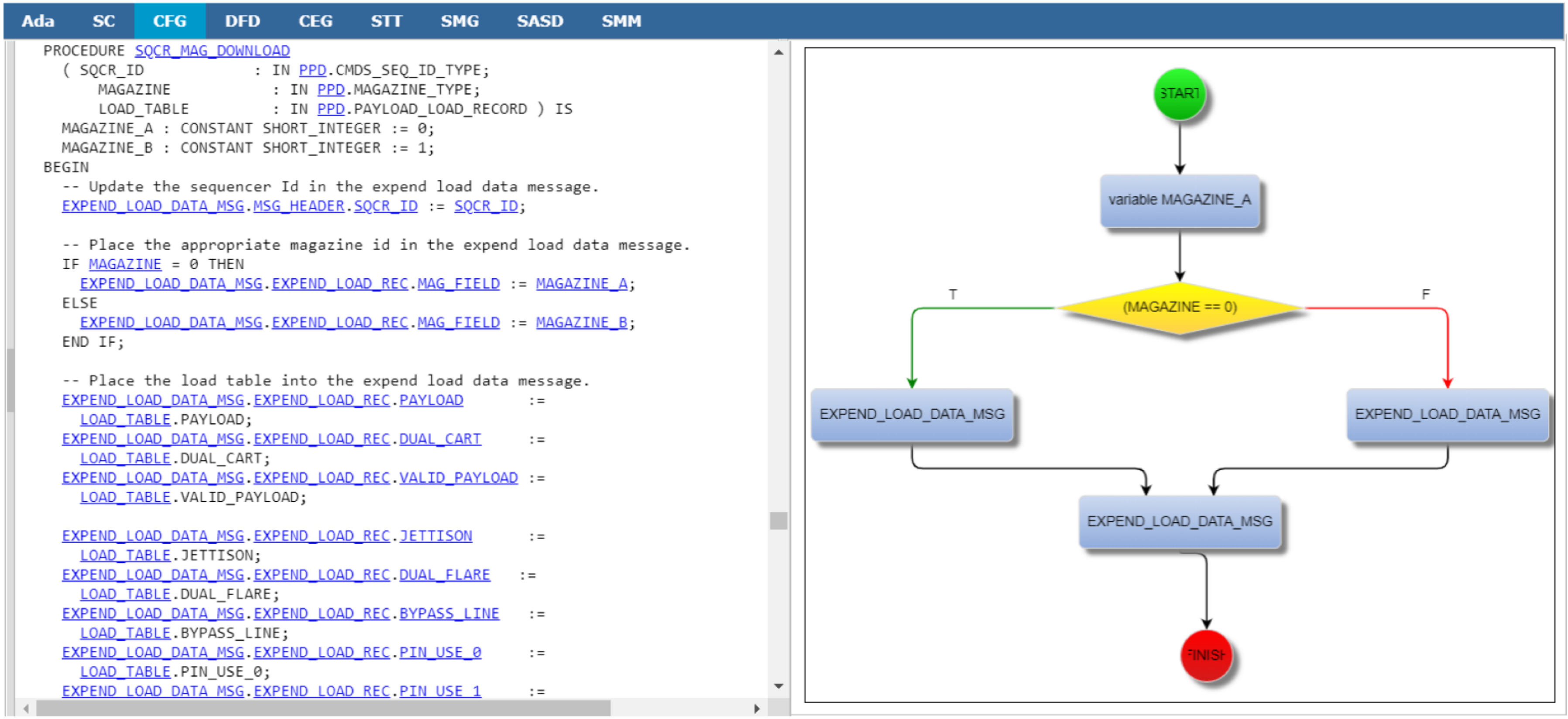
Legacy DesignDoc™ Legacy+New DesignDoc™ DesignDoc Subscription

DesignDoc Supported Programming Languages

C, C#, C++	Java	VB.Net	Python	IBM EGL	PL/SQL
TypeScript	ADA	Assembler	Basic	CICS	CMS-2
COBOL	DCL	Dec Basic	EasyTrieve	Fortran	JCL
Jovial	MagnaX	MUMPS	Natural	PL/I	PowerBuilder
SQL	VAX Basic	Visual Basic 6			

Legacy DesignDoc™

Legacy DesignDoc tool reads your existing application to generate design documentation so that the application code can be understood by engineers maintaining the product.

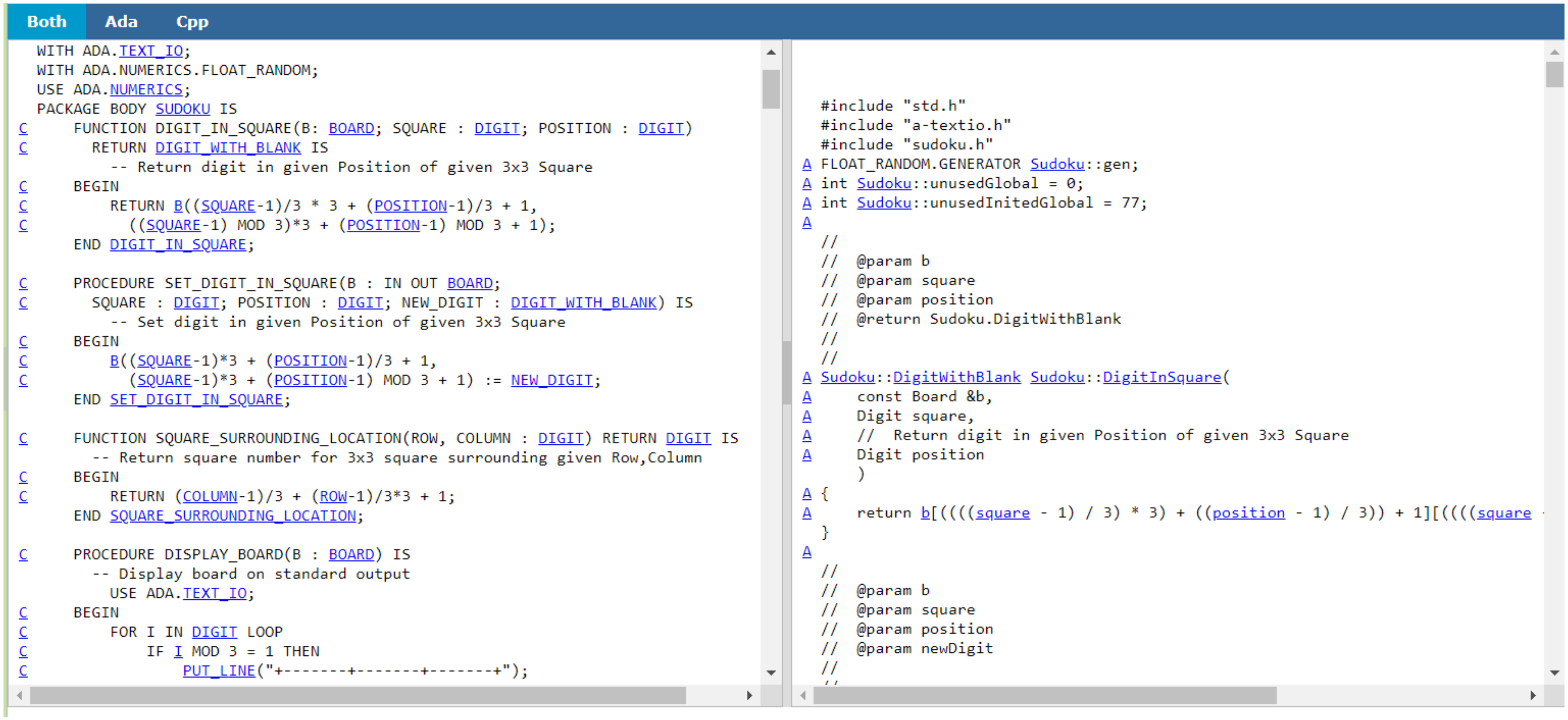


```
PROCEDURE SQCR_MAG_DOWNLOAD
  ( SQCR_ID            : IN PPD.CMDS_SEQ_ID_TYPE;
      MAGAZINE           : IN PPD.MAGAZINE_TYPE;
      LOAD_TABLE         : IN PPD.PAYLOAD_LOAD_RECORD ) IS
  MAGAZINE_A : CONSTANT SHORT_INTEGER := 0;
  MAGAZINE_B : CONSTANT SHORT_INTEGER := 1;
BEGIN
  -- Update the sequencer Id in the expend load data message.
  EXPEND_LOAD_DATA_MSG.MSG_HEADER.SQCR_ID := SQCR_ID;

  -- Place the appropriate magazine id in the expend load data message.
  IF MAGAZINE = 0 THEN
    EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.MAG_FIELD := MAGAZINE_A;
  ELSE
    EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.MAG_FIELD := MAGAZINE_B;
  END IF;

  -- Place the load table into the expend load data message.
  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.PAYLOAD       :=
    LOAD_TABLE.PAYLOAD;
  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.DUAL_CART     :=
    LOAD_TABLE.DUAL_CART;
  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.VALID_PAYLOAD :=
    LOAD_TABLE.VALID_PAYLOAD;

  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.JETTISON      :=
    LOAD_TABLE.JETTISON;
  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.DUAL_FLARE   :=
    LOAD_TABLE.DUAL_FLARE;
  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.BYPASS_LINE   :=
    LOAD_TABLE.BYPASS_LINE;
  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.PIN_USE_0     :=
    LOAD_TABLE.PIN_USE_0;
  EXPEND_LOAD_DATA_MSG.EXPEND_LOAD_REC.PIN_USE_1     :=
```

MAPUSOFT

Legacy+New DesignDoc™

Legacy+New DesignDoc tool reads the legacy application and the newly ***converted code**** to generate side-by-side design documentation comparison.



```
   WITH ADA.TEXT_IO;
   WITH ADA.NUMERICS.FLOAT_RANDOM;
   USE ADA.NUMERICS;
   PACKAGE BODY SUDOKU IS
C      FUNCTION DIGIT_IN_SQUARE(B: BOARD; SQUARE : DIGIT; POSITION : DIGIT)
C        RETURN DIGIT_WITH_BLANK IS
           -- Return digit in given Position of given 3x3 Square
C      BEGIN
C          RETURN B((SQUARE-1)/3 * 3 + (POSITION-1)/3 + 1,
C            ((SQUARE-1) MOD 3)*3 + (POSITION-1) MOD 3 + 1);
       END DIGIT_IN_SQUARE;

C      PROCEDURE SET_DIGIT_IN_SQUARE(B : IN OUT BOARD;
C        SQUARE : DIGIT; POSITION : DIGIT; NEW_DIGIT : DIGIT_WITH_BLANK) IS
           -- Set digit in given Position of given 3x3 Square
C      BEGIN
C          B((SQUARE-1)*3 + (POSITION-1)/3 + 1,
C            (SQUARE-1)*3 + (POSITION-1) MOD 3 + 1) := NEW_DIGIT;
       END SET_DIGIT_IN_SQUARE;

C      FUNCTION SQUARE_SURROUNDING_LOCATION(ROW, COLUMN : DIGIT) RETURN DIGIT IS
         -- Return square number for 3x3 square surrounding given Row,Column
C      BEGIN
C          RETURN (COLUMN-1)/3 + (ROW-1)/3*3 + 1;
       END SQUARE_SURROUNDING_LOCATION;

C      PROCEDURE DISPLAY_BOARD(B : BOARD) IS
         -- Display board on standard output
           USE ADA.TEXT_IO;
C      BEGIN
C          FOR I IN DIGIT LOOP
C              IF I MOD 3 = 1 THEN
C                  PUT_LINE("+-------+-------+-------+");
```

```
  #include "std.h"
  #include "a-textio.h"
  #include "sudoku.h"
A FLOAT_RANDOM.GENERATOR Sudoku::gen;
A int Sudoku::unusedGlobal = 0;
A int Sudoku::unusedInitedGlobal = 77;
A
  //
  //  @param b
  //  @param square
  //  @param position
  //  @return Sudoku.DigitWithBlank
  //
  //
A Sudoku::DigitWithBlank Sudoku::DigitInSquare(
A     const Board &b,
A     Digit square,
A     //  Return digit in given Position of given 3x3 Square
A     Digit position
      )
A {
A     return b[((((square - 1) / 3) * 3) + ((position - 1) / 3)) + 1][((((square
  }
A
  //
  //  @param b
  //  @param square
  //  @param position
  //  @param newDigit
  //
```

DesignDoc Subscription

- Offered as a monthly subscription service so that the latest documentation can be re-generated whenever needed as development continues

- Generated documentation allows users with varying experience levels to quickly familiarize themselves with the structure and flow of the application through easily navigable diagrams, hyperlinking code, and other UML artifacts and graphics

- Code is processed and hosted in the cloud using Amazon Web Services, which assures reliability, security and availability.

Providing Software Re-Use Solutions to All Embedded Markets

- Military/Aero
- Telecom
- Semiconductors
- Medical
- RTOS
- Industrial
- Consumer Electronics
- And more...

For More Information

- To download MapuSoft's free software evaluation visit: https://mapusoft.com/downloads/
- For any additional information please contact MapuSoft at: https://mapusoft.com/contact/
- Toll Free: 1-877-MAPUSOFT
- Email: sales@mapusoft.com