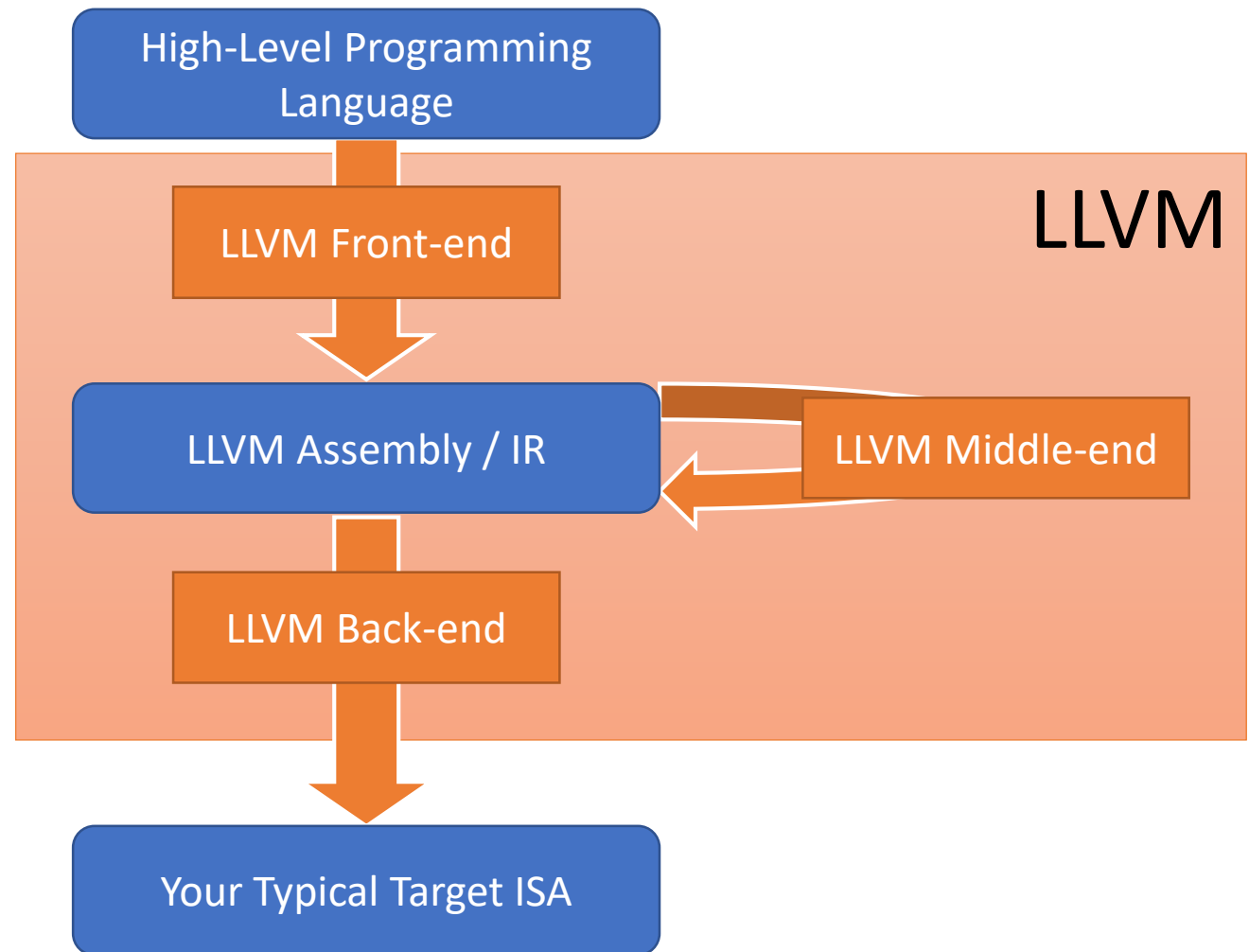




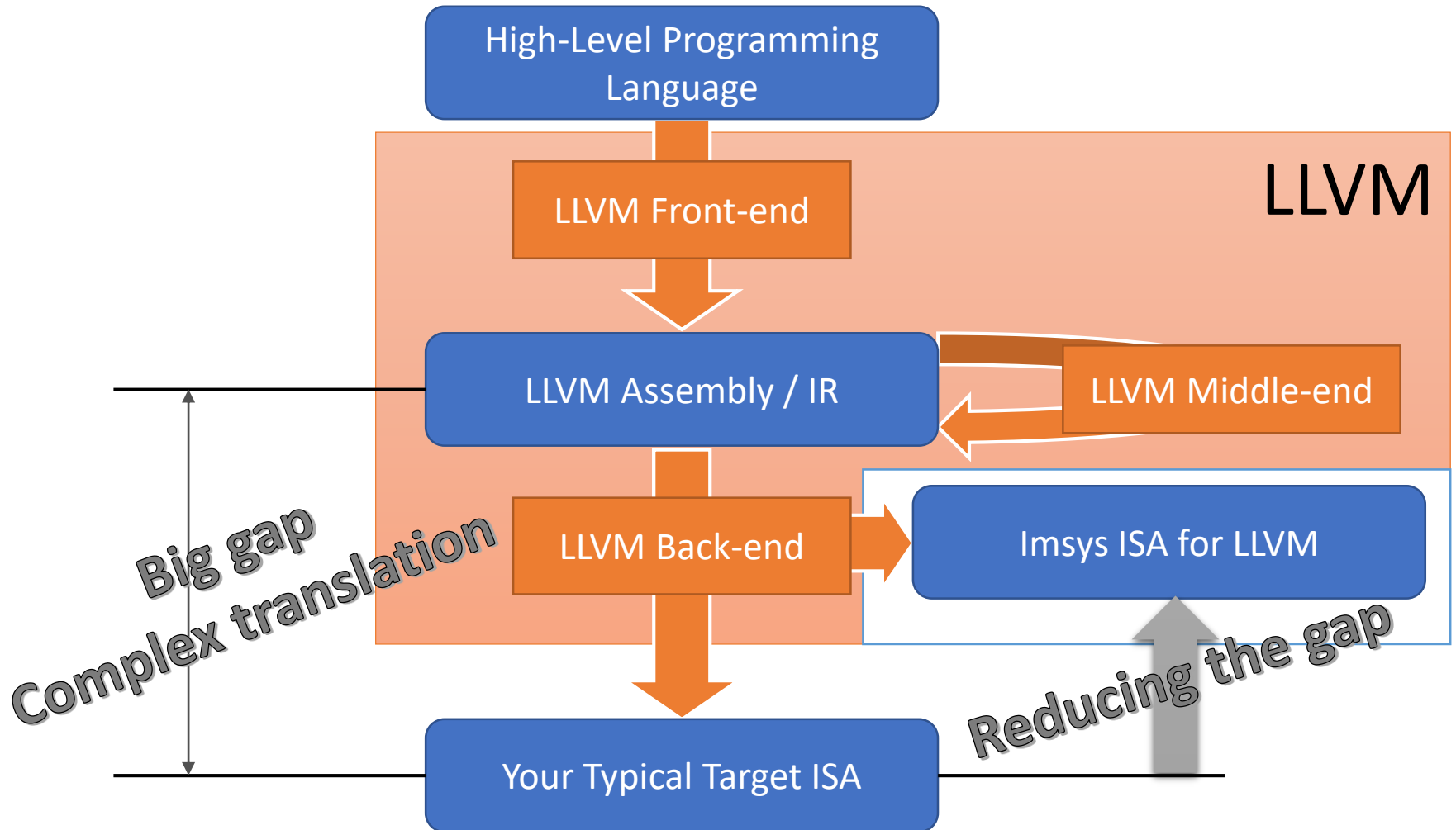
A unique processor architecture meeting LLVM IR and the IoT

Dávid Juhász
david.juhasz@imsys.tech.com

Compiling with LLVM



Improving Efficiency by Lifting Target



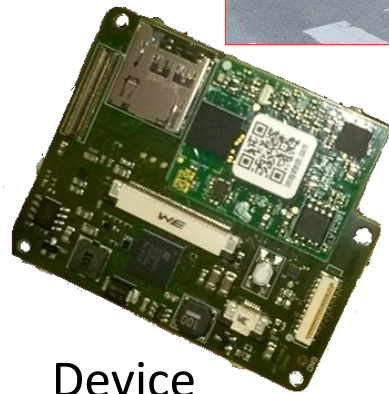


Agenda

1. Imsys Company and the IoT
2. Imsys Lean Processing Technology
3. Imsys ISA for LLVM



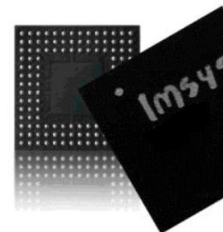
Imsys AB



Device



Module



IC



IP



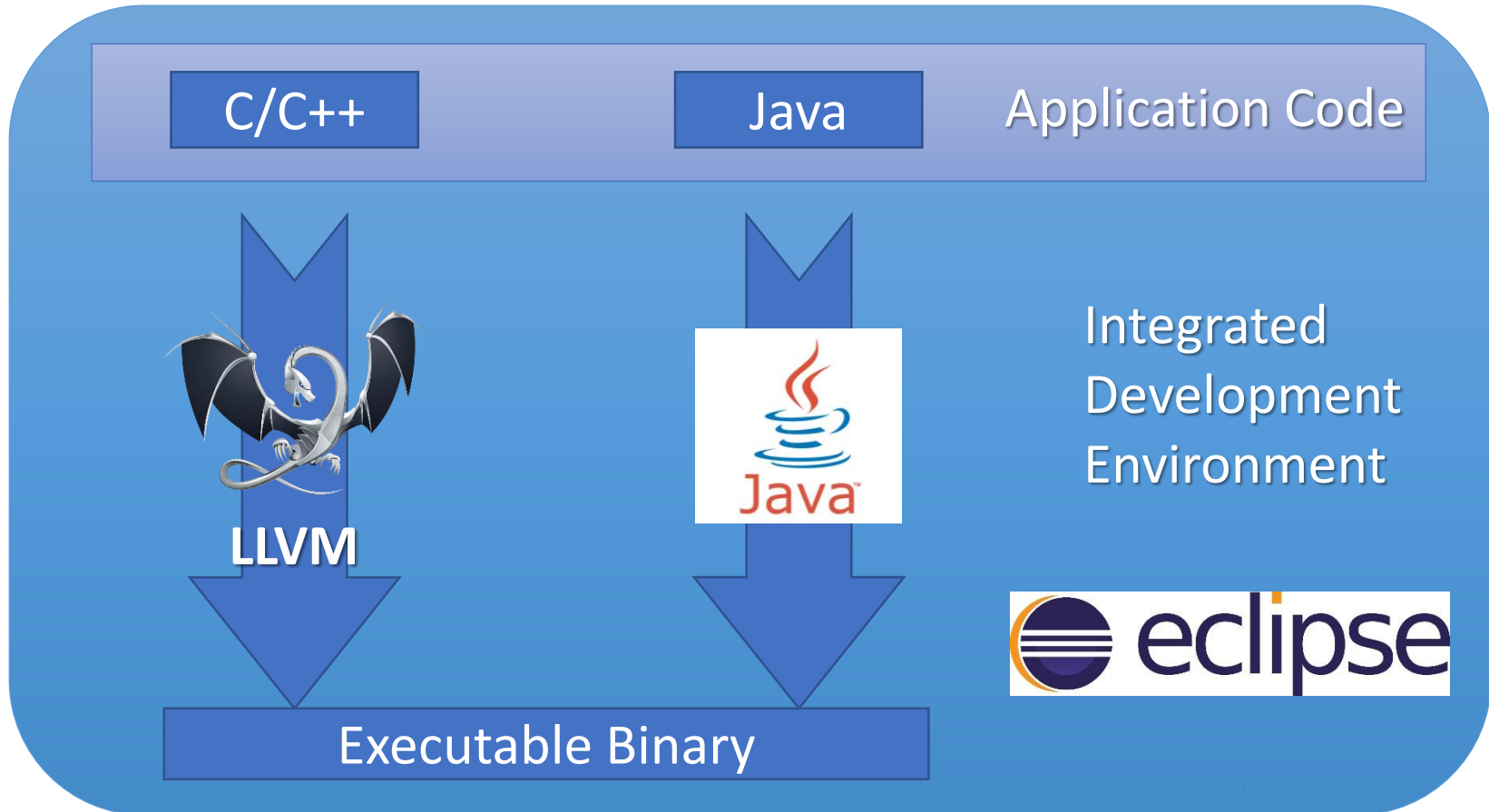


Imsys EMBLA

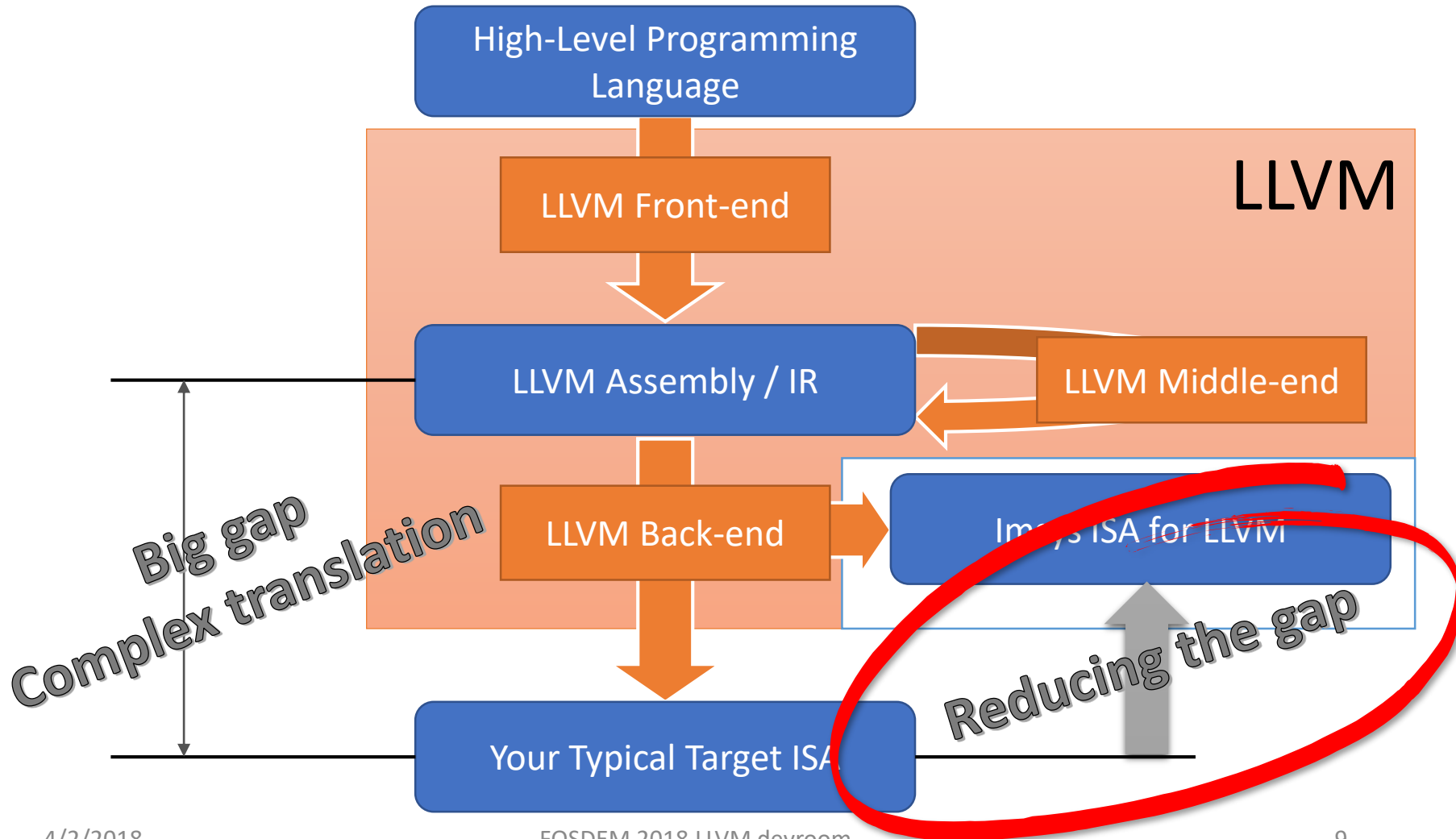
Single Controller Solution for the IoT



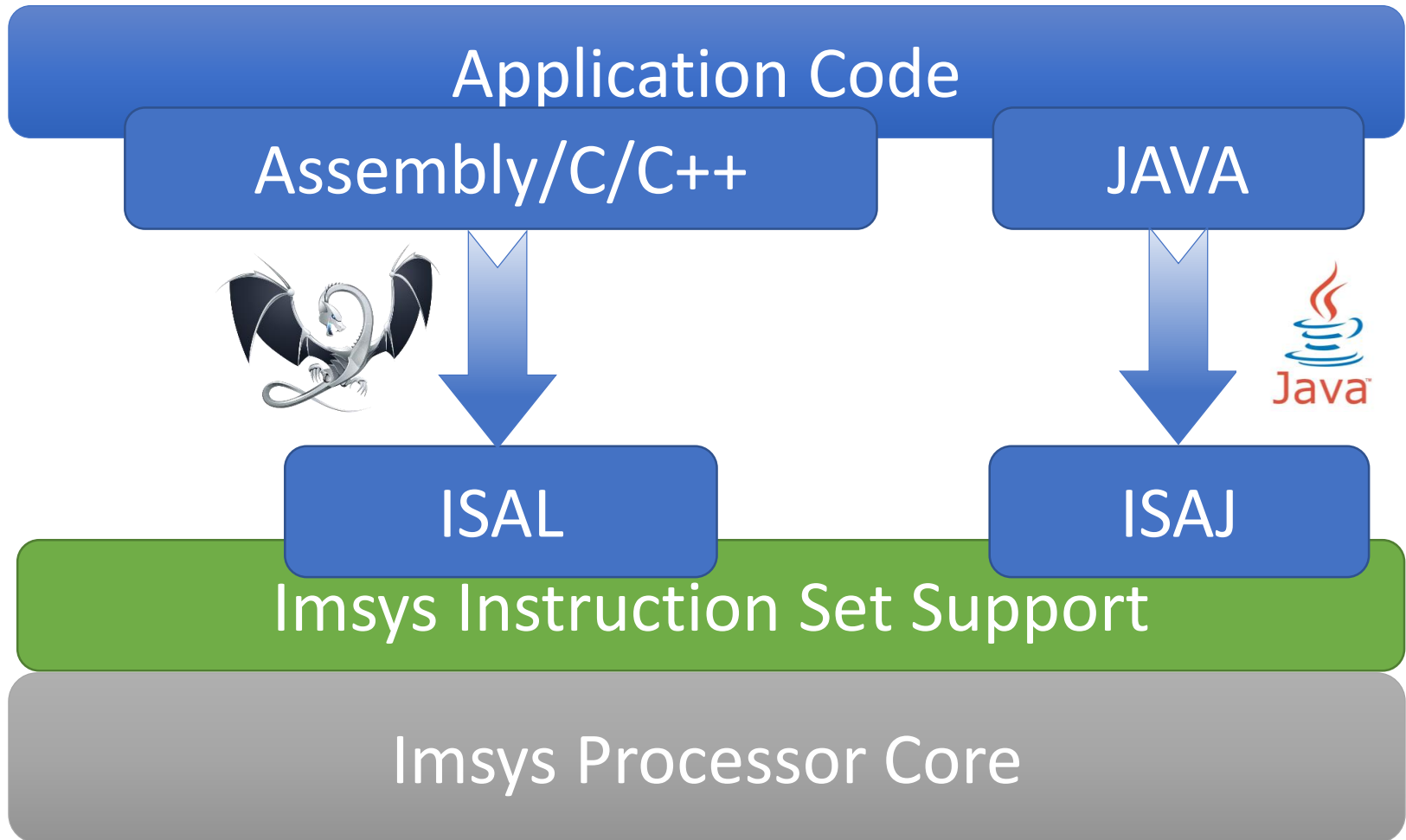
High-Level Software Platform



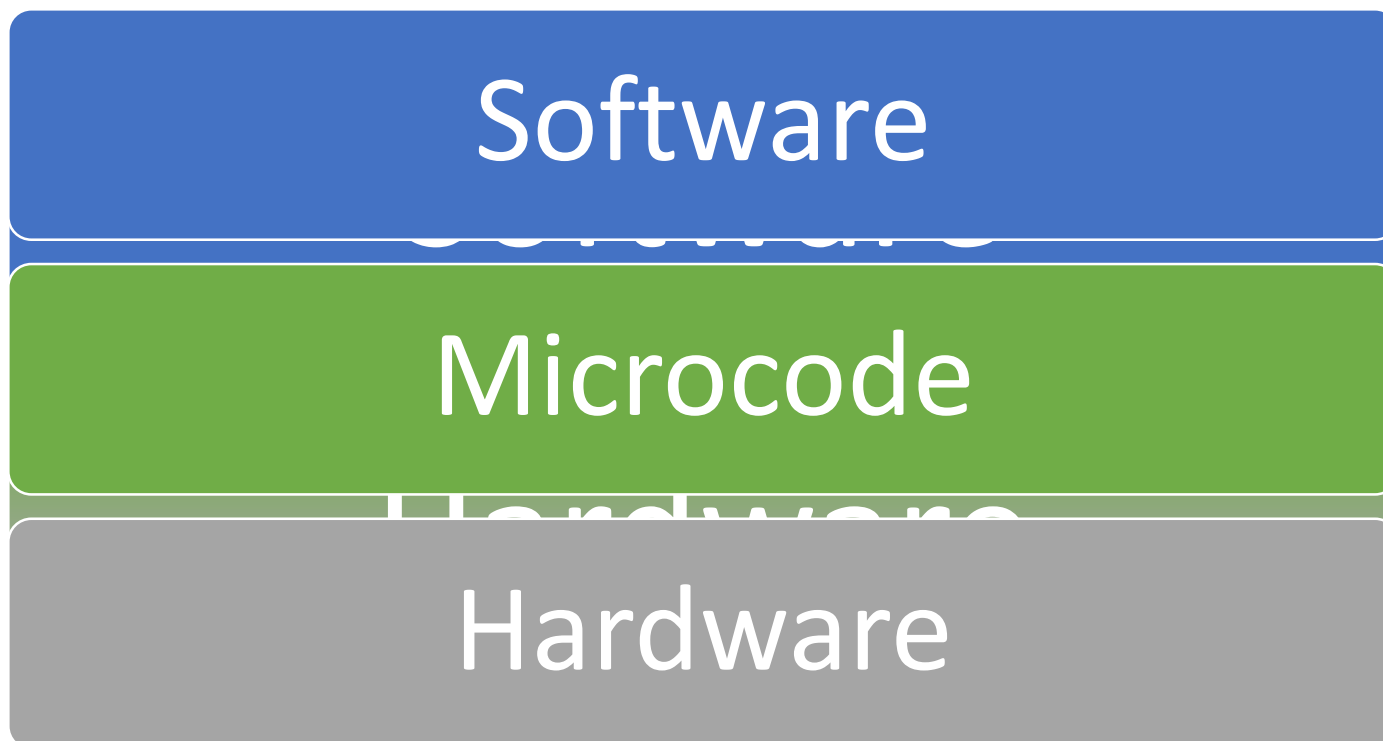
Improving Efficiency by Reducing the ISA Gap



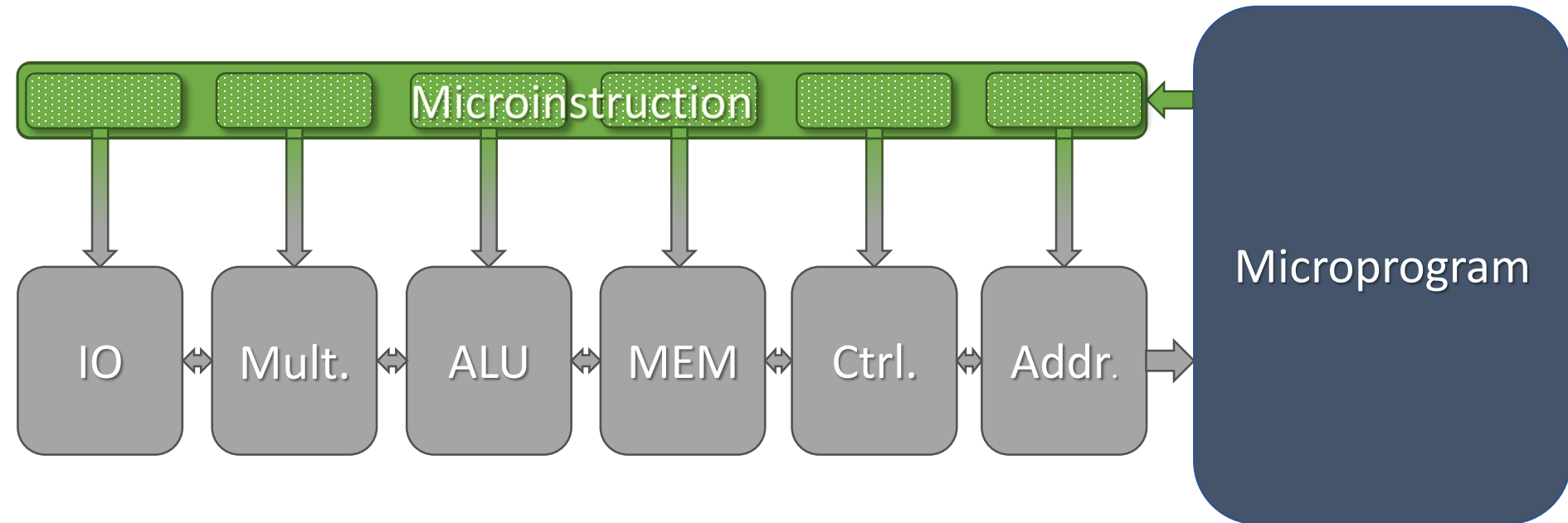
Abstraction Layers



A Forgotten Layer of Abstraction



The Processor Architect's Best Friend



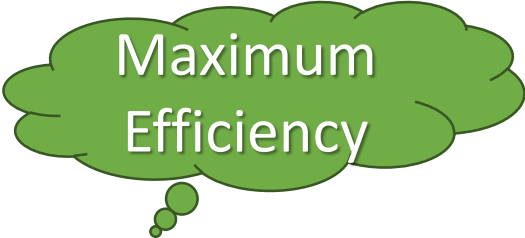
Operation-Oriented Hardware Architecture

A green thought bubble with a small tail pointing downwards and to the right.

Complete
Deterministic
Control

A green thought bubble with a small tail pointing downwards and to the left.

Minimal
Hardware

A green thought bubble with a small tail pointing downwards and to the left.

Maximum
Efficiency

Microcode, what is it good for?

A green thought bubble with a small tail pointing upwards and to the left.

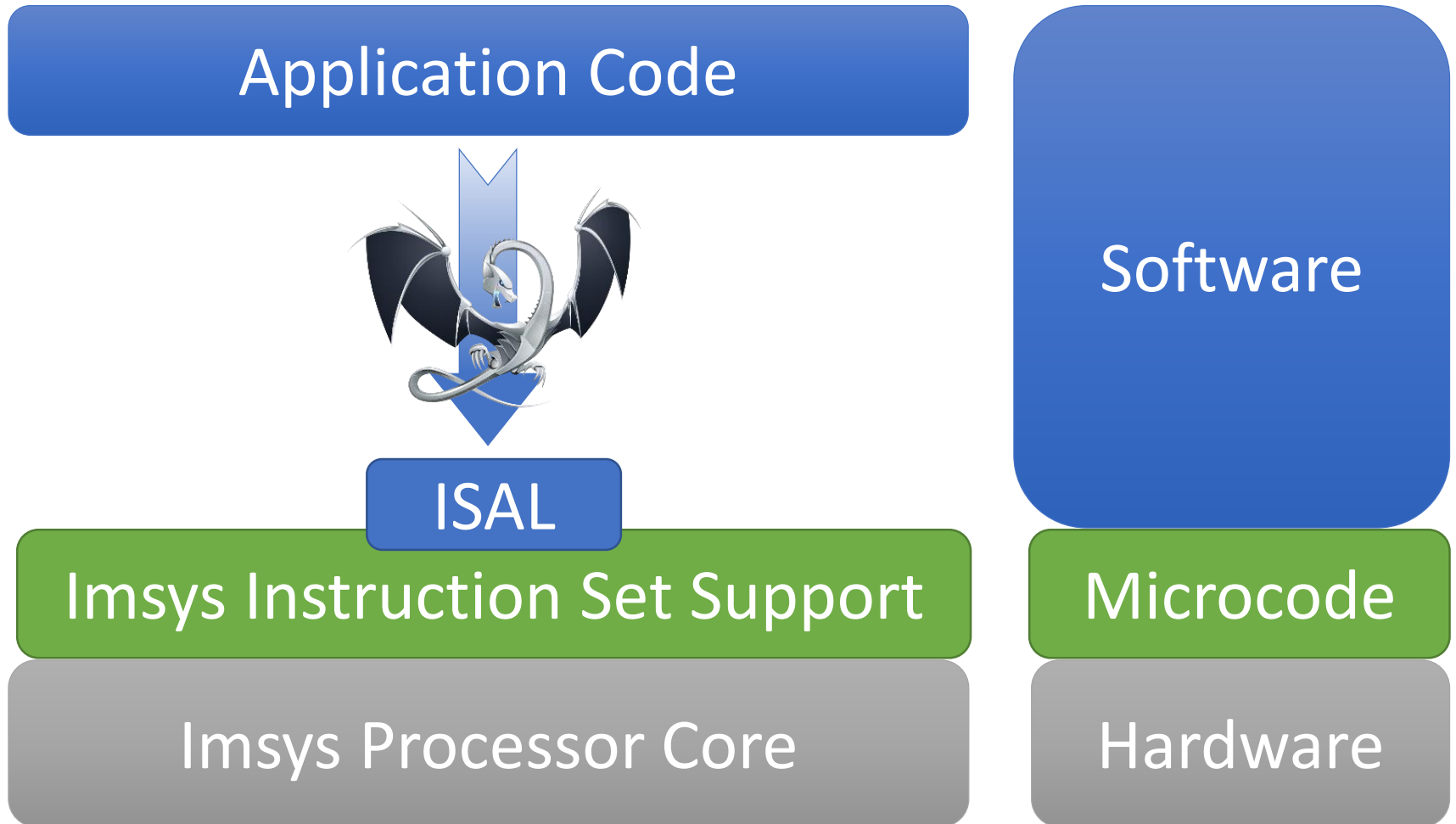
Flexibility

A green thought bubble with a small tail pointing upwards and to the left.

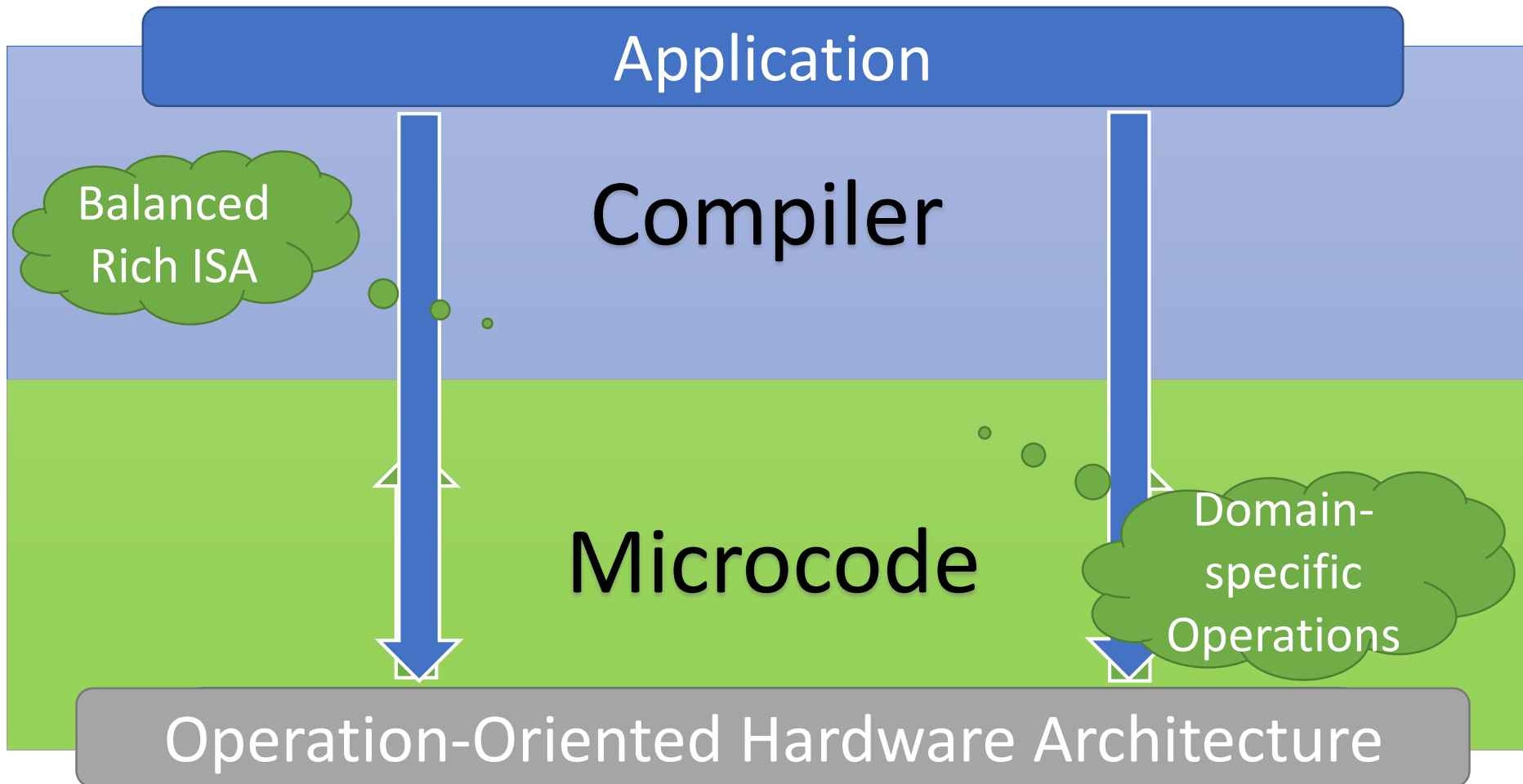
Soft-
Reconfigurability



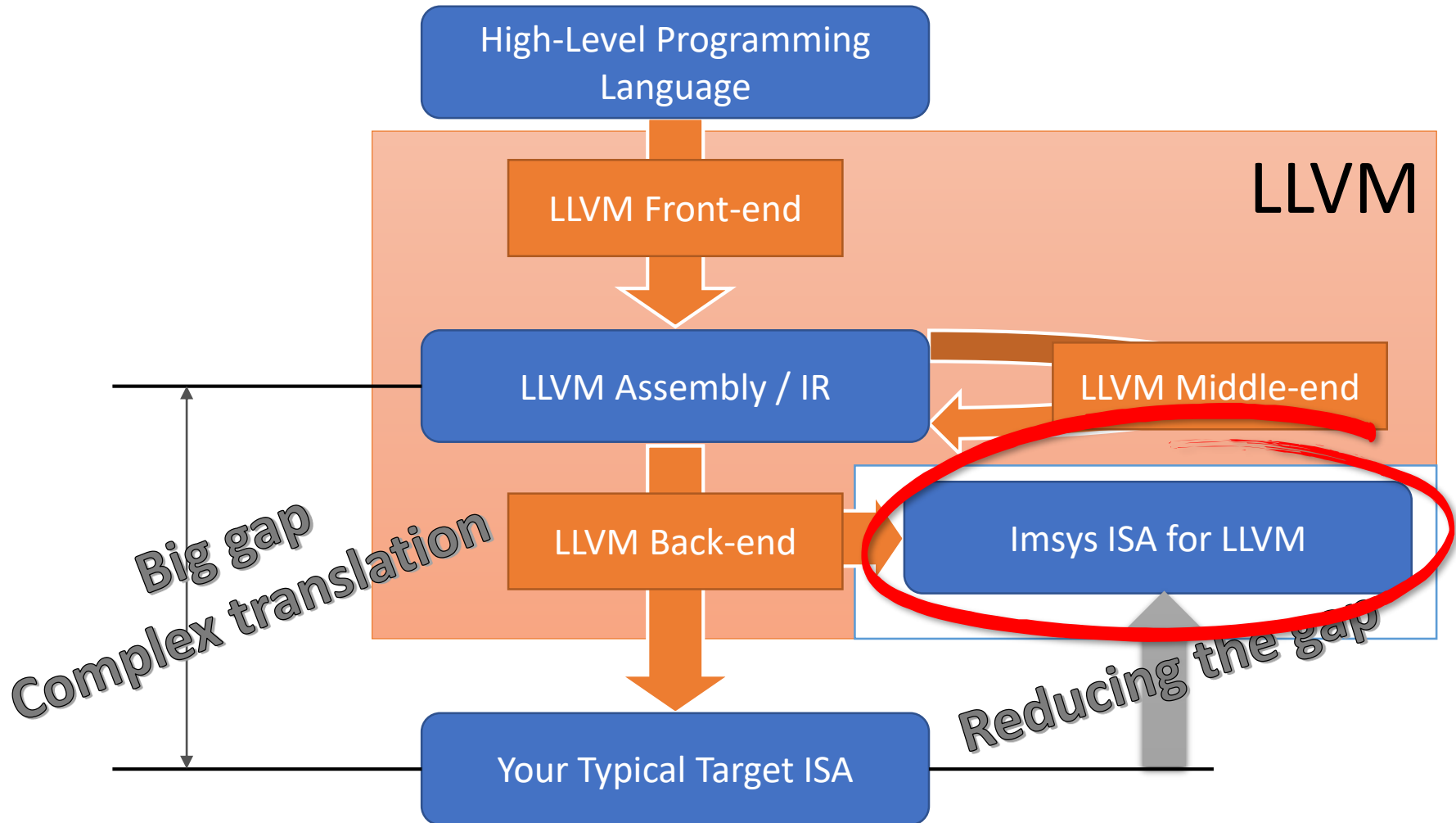
Abstraction Layers Revisited



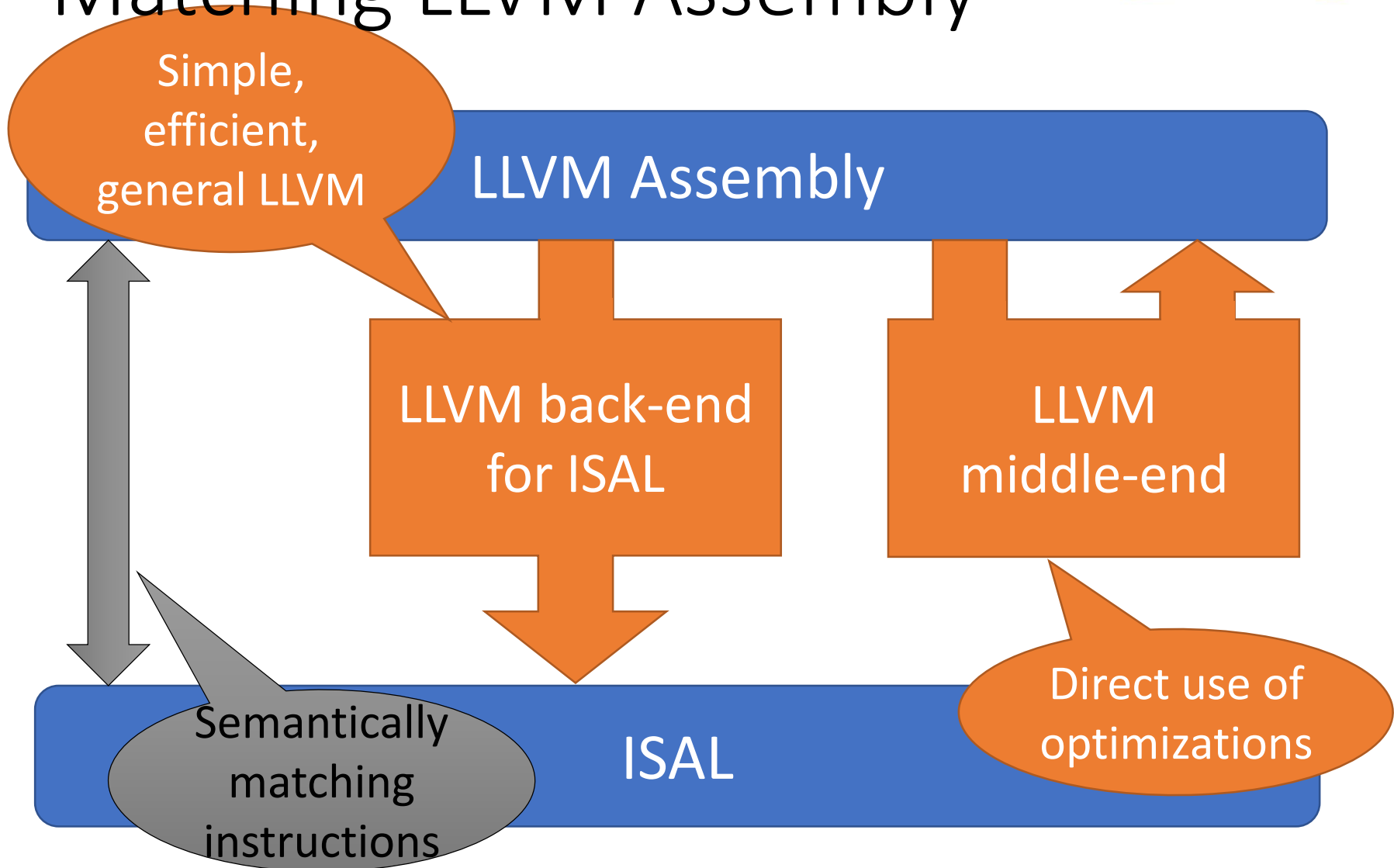
A Unique Balance between Hardware and Software



Improving Efficiency by Matching LLVM Assembly



Matching LLVM Assembly



Meeting a Constrained Reality

The logo for lmsys, featuring the text "lmsys" in a white, stylized font on a purple, brush-stroke-like background.

	LLVM Assembly	ISAL
Operations	Instructions & intrinsic functions	Additional system operations
Single Value Types	Virtually unlimited	Set of integer, floating point, pointer, and vector types
Registers	Unlimited	Register windows
Arguments	Source and destination registers	Registers with support of accumulating in source registers, immediate values
Binary Representation	Bitcode	Custom dense binary coding

Optimized Operation Sequences

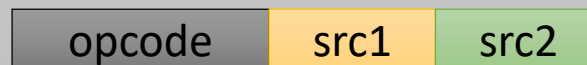
$a := a + b$

Accumulating in source register

add a a b



add.upd a b



$a := a + 42$

Immediate values

move b 42

add a a b

add.imm a a 42



add.upd.imm a 42

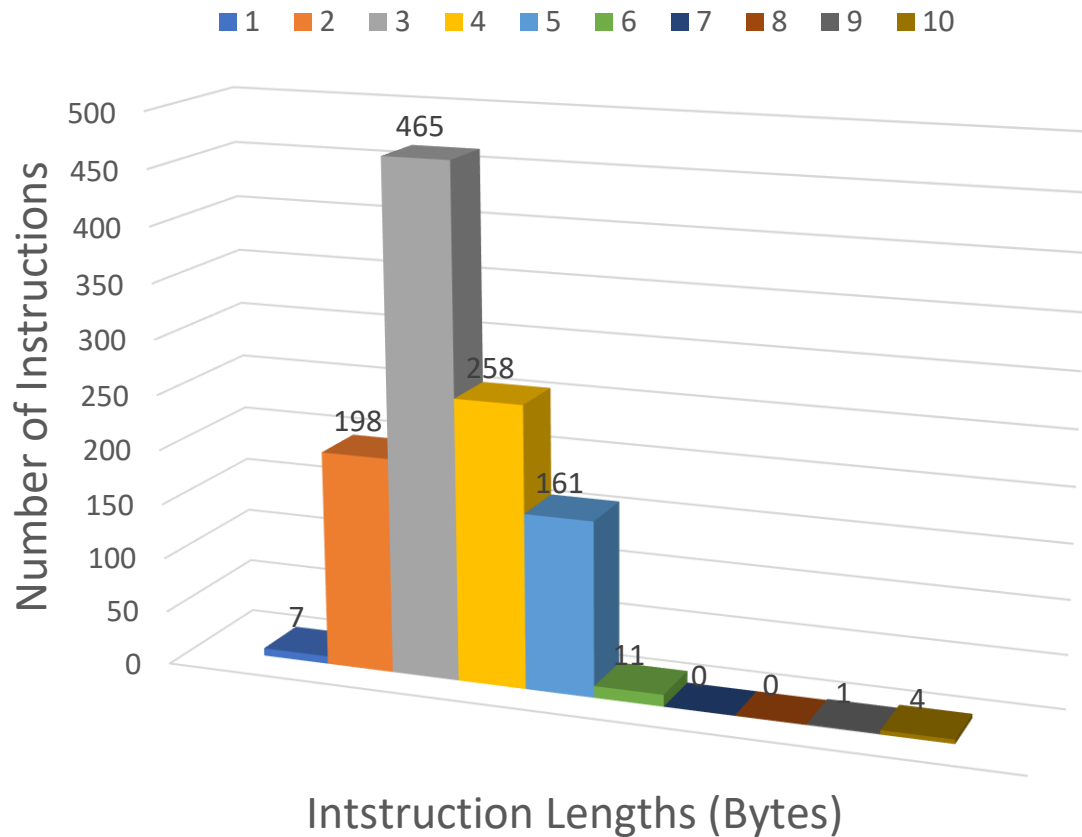


Optimized Binary Representation

Maximize
Code Density

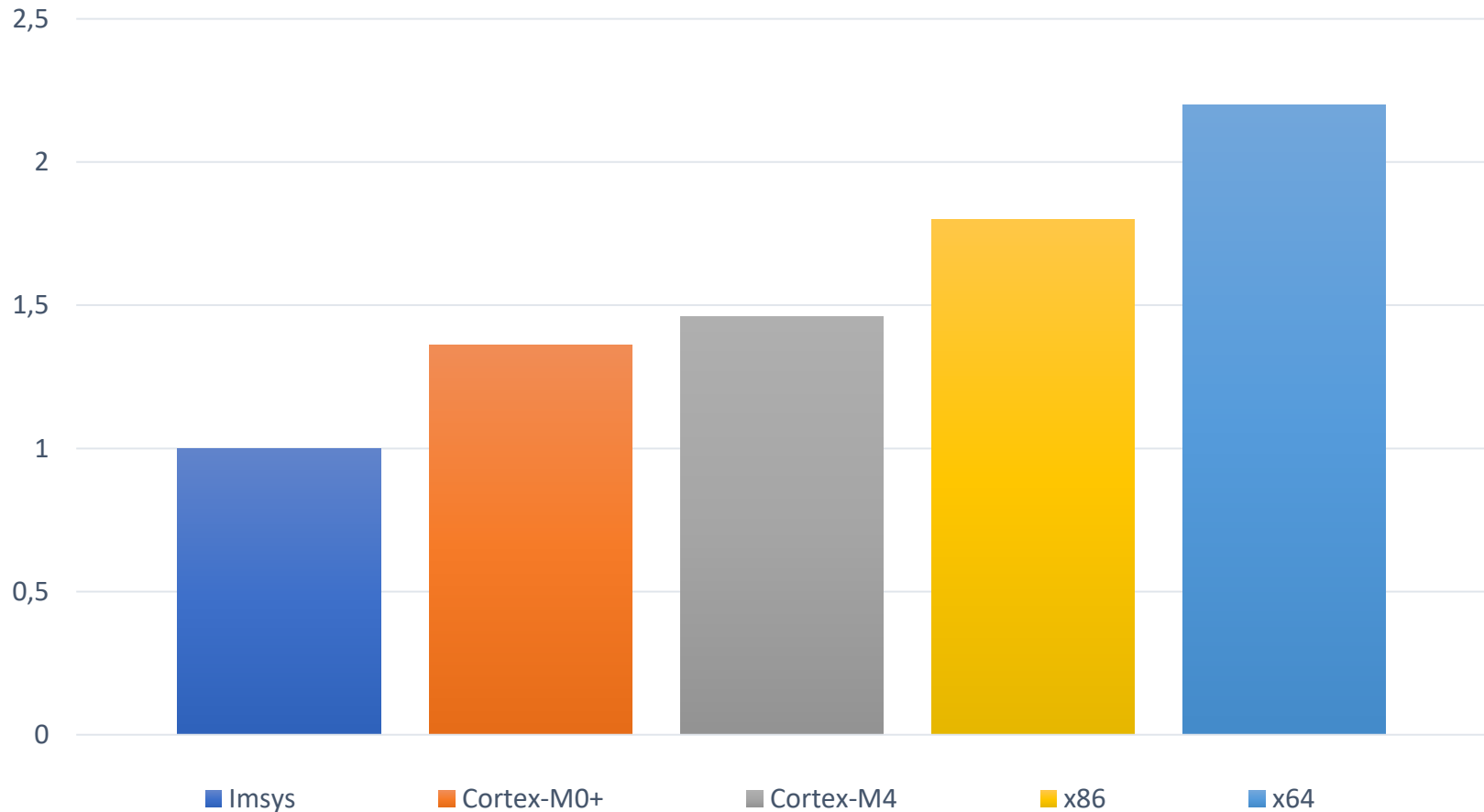
Optimize
Microcode Size

Minimize
Execution Time



Variable-length Instructions

Average Binary Size of TI Suite



ARM Cortex requires 35%+ more, Intel 80%+ more program memory.

LLVM Assembly

LLVM back-end
for ISAL

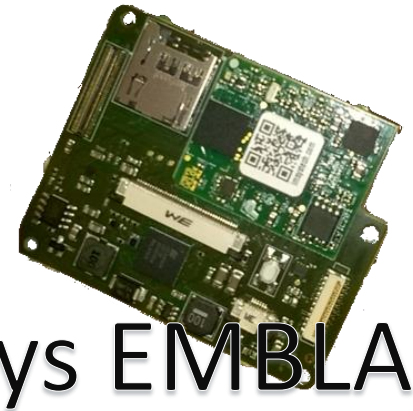
ISAL

Imsys Firmware

Imsys Processor Core

Imsys Lean Processing
Technology

Imsys ISA
for LLVM



Imsys EMBLA
with ISAL



Leaner is Meaner

4/2/2018

Dávid Juhász
david.juhasz@imsys tech.com

FOSDEM 2018 LLVM devroom

23