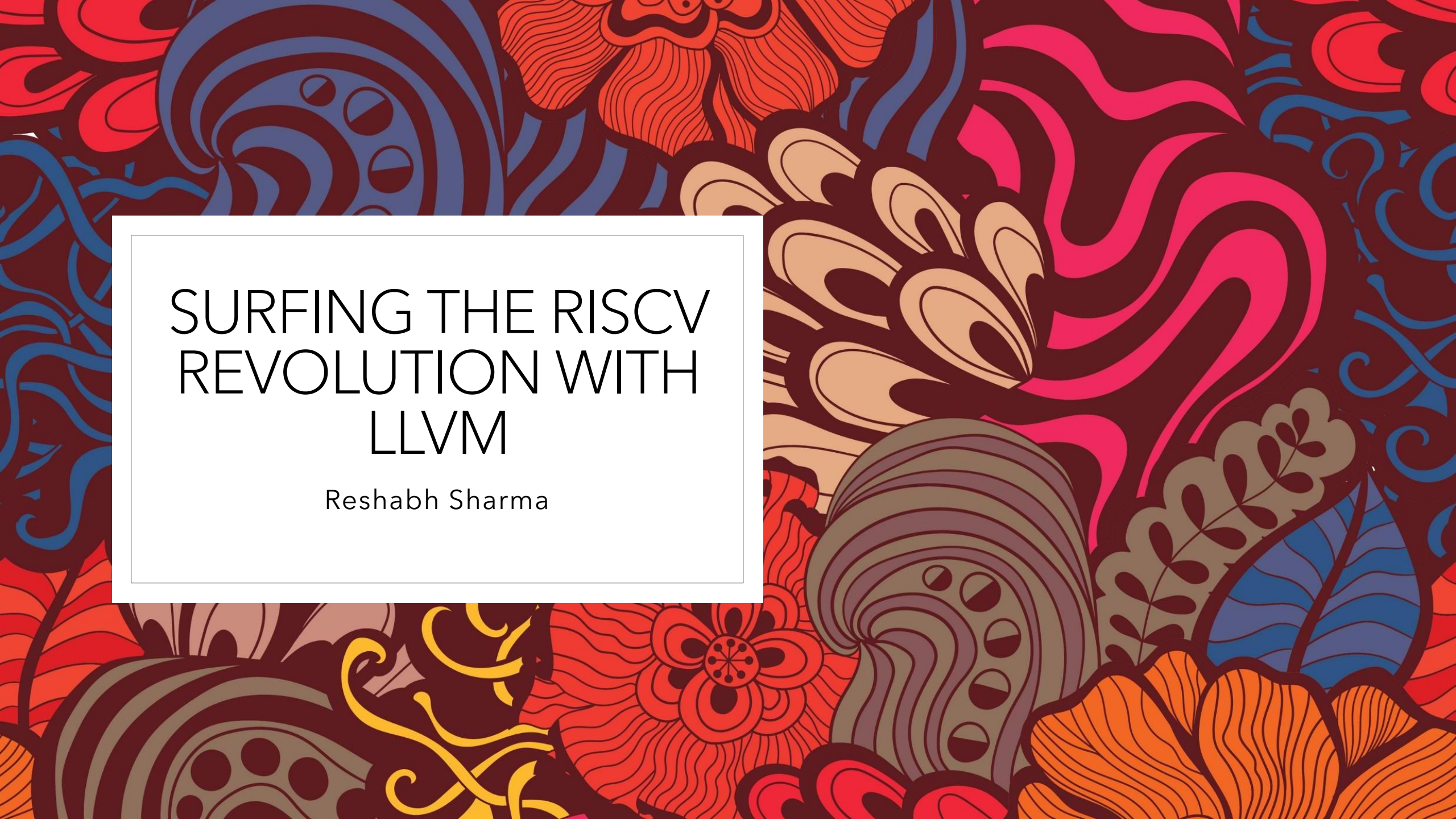
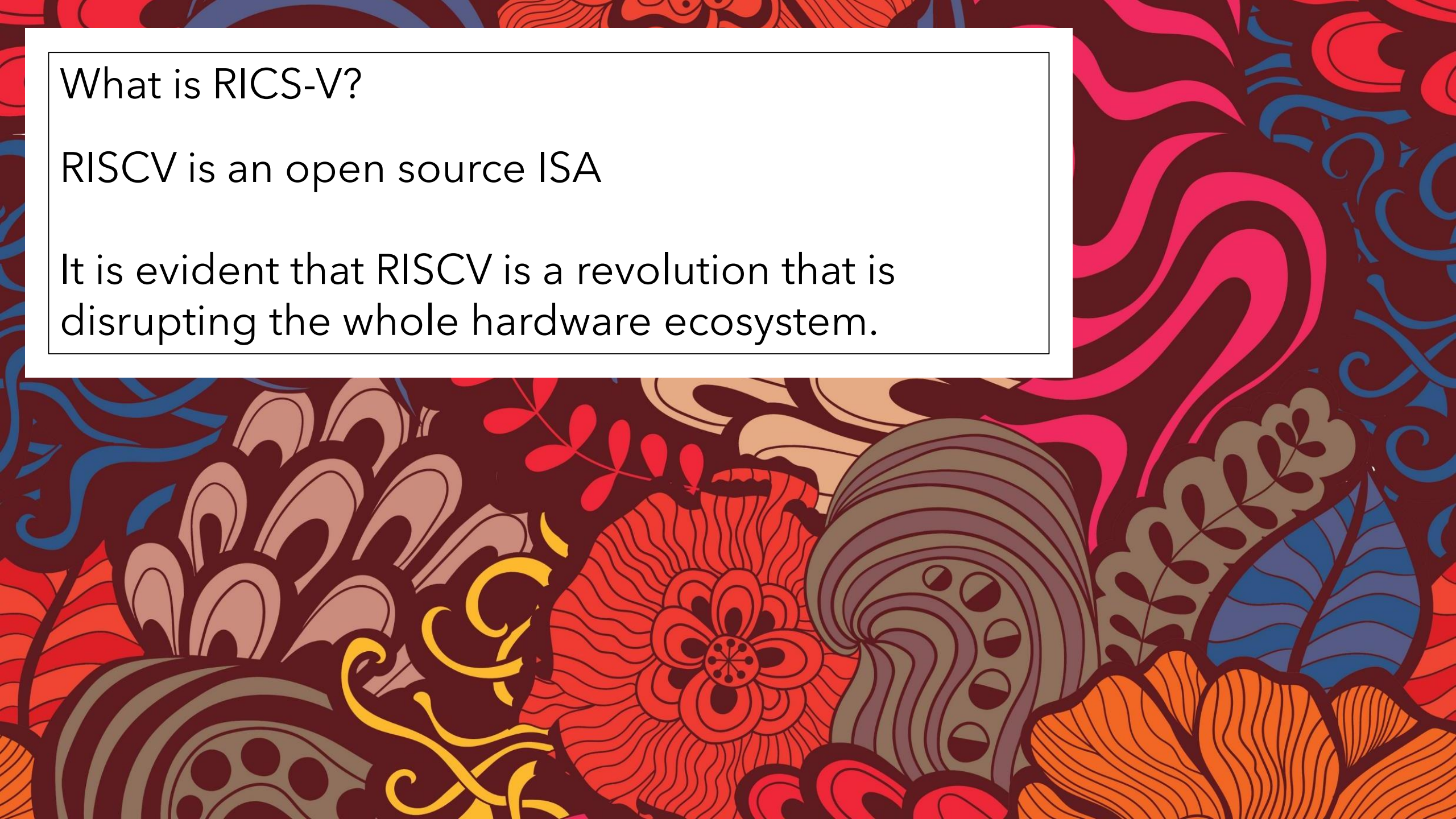

SURFING THE RISCV REVOLUTION WITH LLVM
Reshabh Sharma


What is RICS-V?
RISCV is an open source ISA
It is evident that RISCV is a revolution that is disrupting the whole hardware ecosystem.

What is RICS-V?

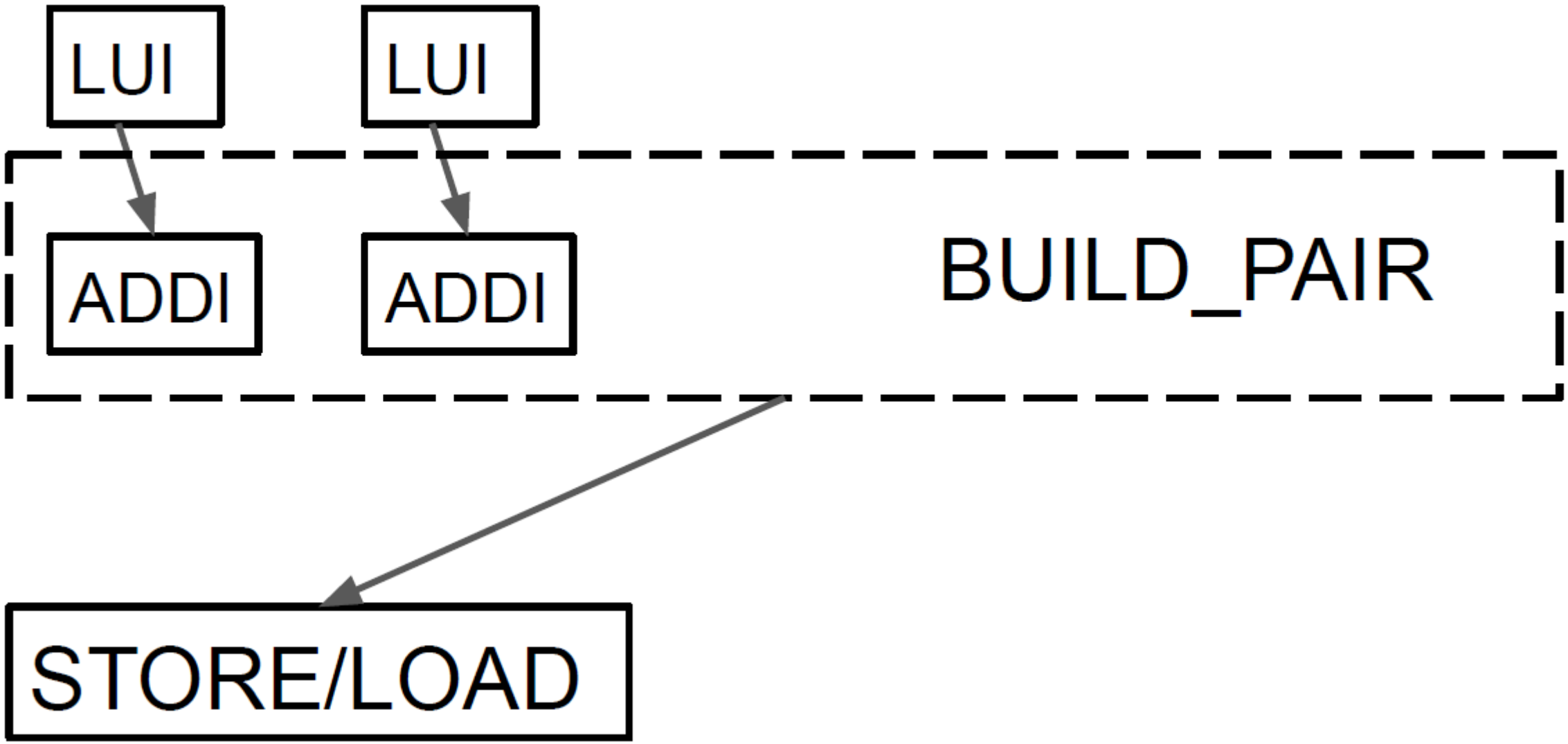
RISCV is an open source ISA

It is evident that RISCV is a revolution that is disrupting the whole hardware ecosystem.

Why should we care as compiler engineers?

It opens a can of OPPORTUNITY for us and the RISCV LLVM backend is the “gold standard” for writing LLVM backends.

Supporting 64 bit pointers in RV32



Supporting 64 bit pointers in RV32

Adding a new instruction in the RISCV LLVM backend is fairly simple

What to do when you have stepped on the tail of a sleeping dragon? (The legalizer)

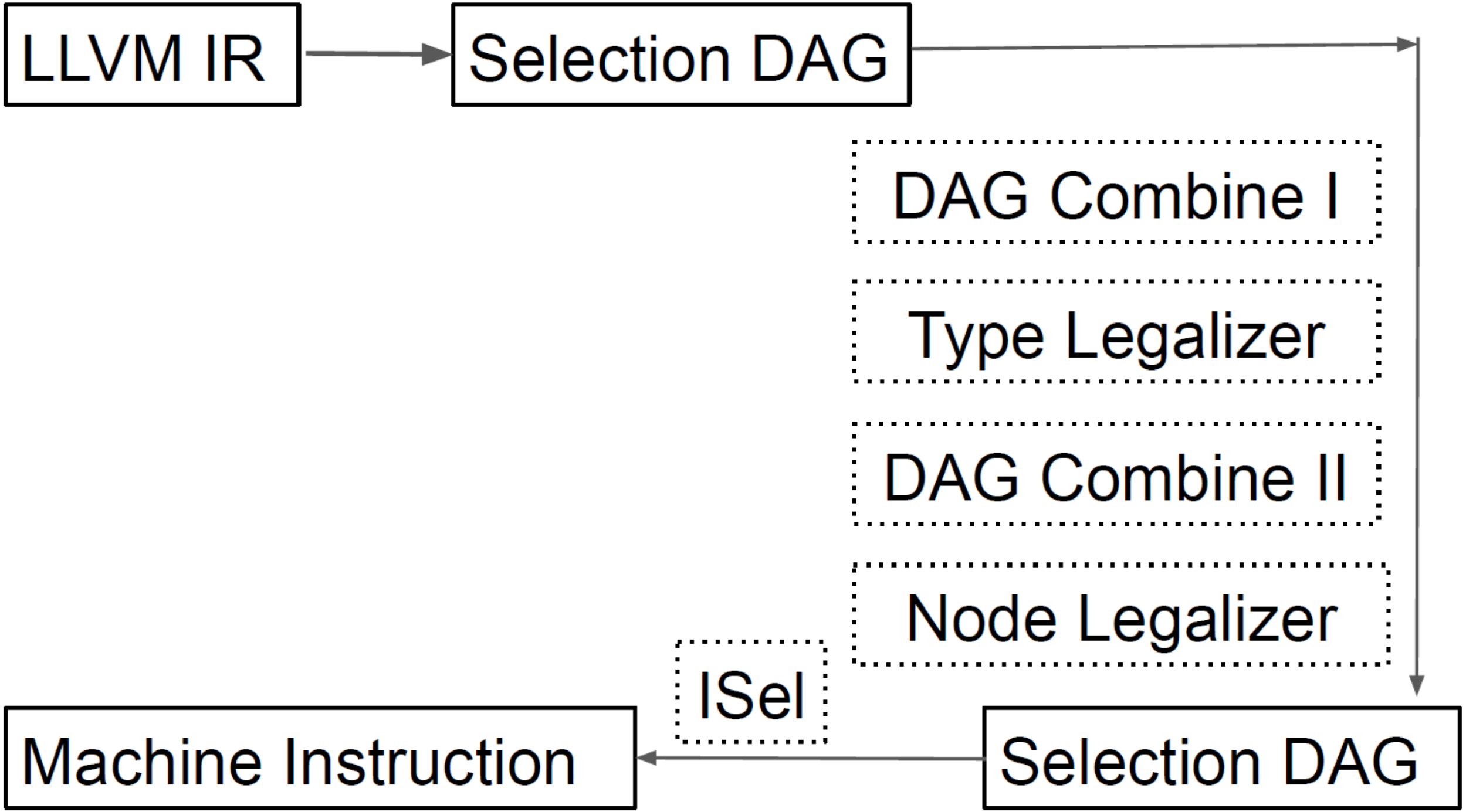
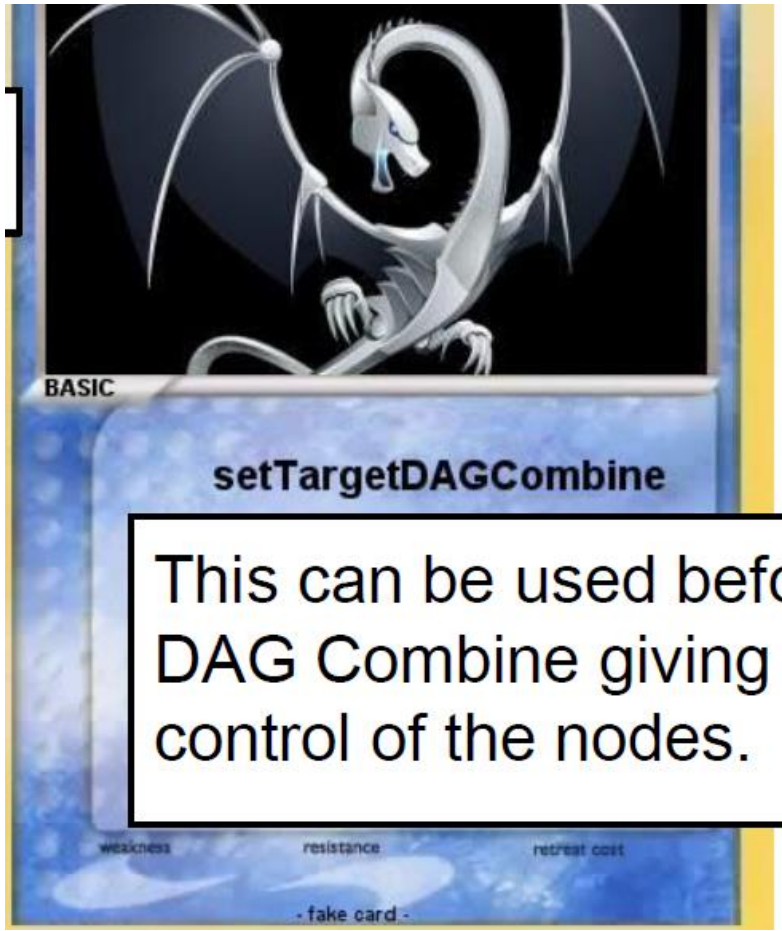

LLVM IR
Selection DAG
DAG Combine I
Type Legalizer
DAG Combine II
Node Legalizer
ISel
Machine Instruction
Selection DAG

Image credits:
Dragon illustration: Vintage vector created by stockgiu - www.freepik.com
Smiley image: The logo belong to the awesome band nirvana

LLVM IR → Selection DAG

PerformDAGCombine



DAG Combine I

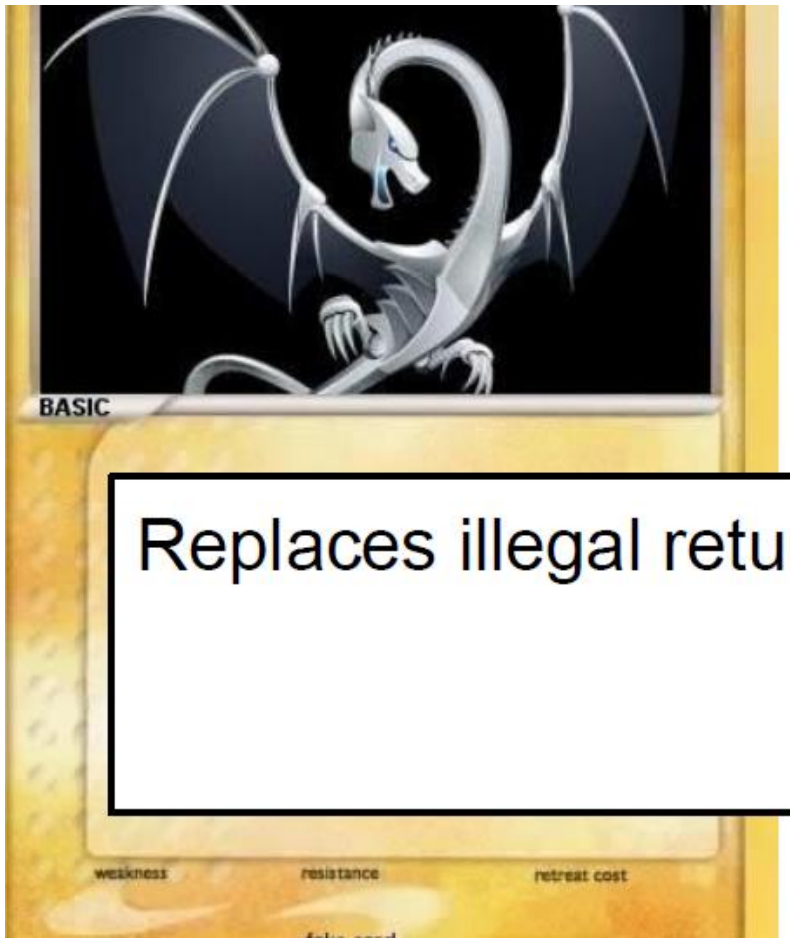
Type Legalizer

DAG Combine II

Node Legalizer

LLVM IR → Selection DAG

ReplaceNodeResults



Replaces illegal return type.

DAG Combine I

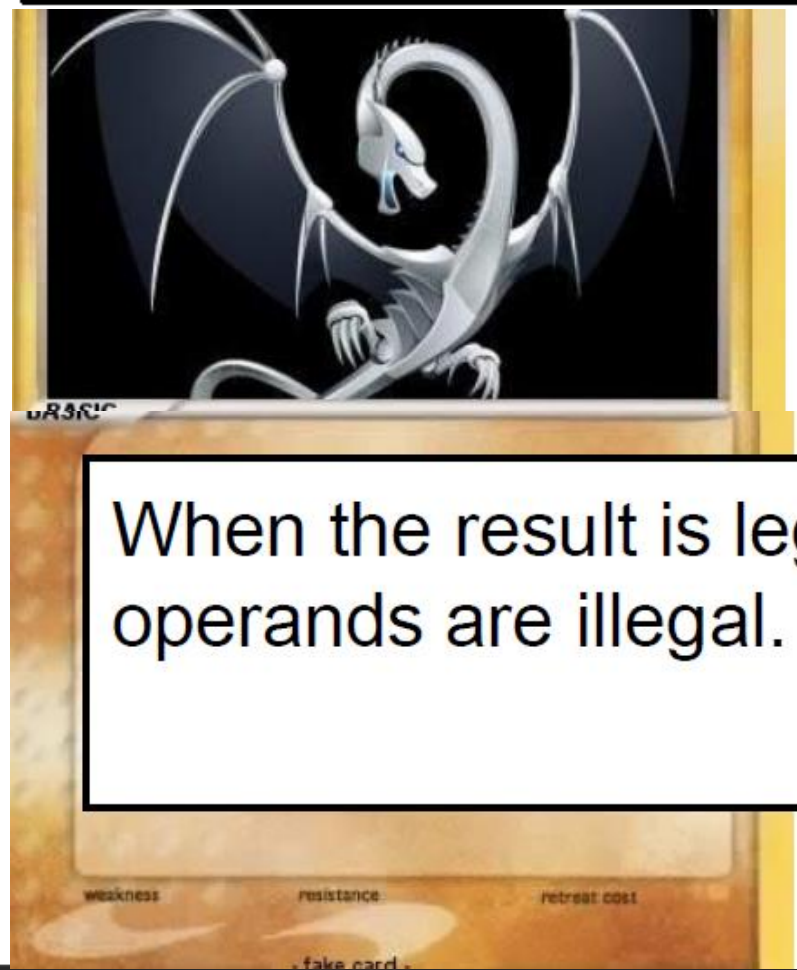
Type Legalizer

DAG Combine II

Node Legalizer

LLVM IR → Selection DAG →

LowerOperationWrapper



When the result is legal and operands are illegal.

DAG Combine I

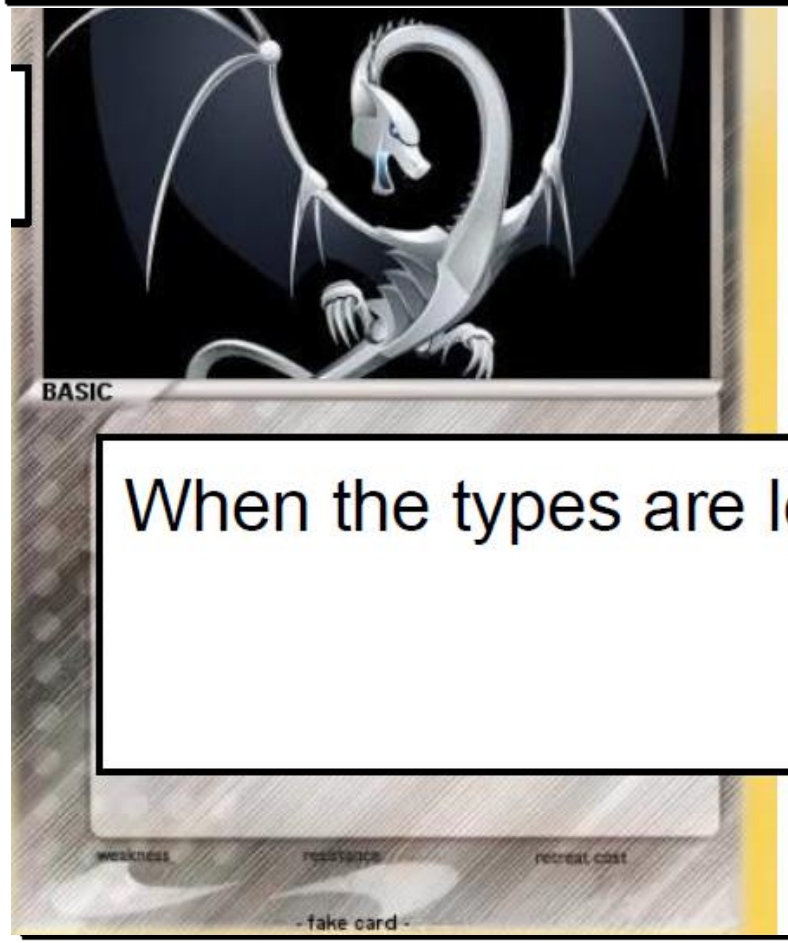
Type Legalizer

DAG Combine II

Node Legalizer

LLVM IR → Selection DAG

LowerOperations



When the types are legal

DAG Combine I

Type Legalizer

DAG Combine II

Node Legalizer

THANK YOU!