



# **SPIR 2.0 Provisional**

**SIGGRAPH, Vancouver  
August 2014**



**S**tandard  
**P**ortable  
**I**ntermediate  
**R**epresentation

## Goals

1. Portable interchange format for partially compiled OpenCL C
2. Target format for other languages

*Enables compiler ecosystem for  
portable parallel programs*

# OpenCL as Parallel Language Backend



JavaScript binding for initiation of OpenCL C kernels



Language for image processing and computational photography



MulticoreWare open source project on Bitbucket



Embedded array language for Haskell



Java language extensions for parallelism



River Trail Language extensions to JavaScript



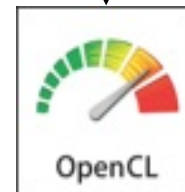
Compiler directives for Fortran, C and C++



PyOpenCL Python wrapper around OpenCL



Harlan High level language for GPU programming

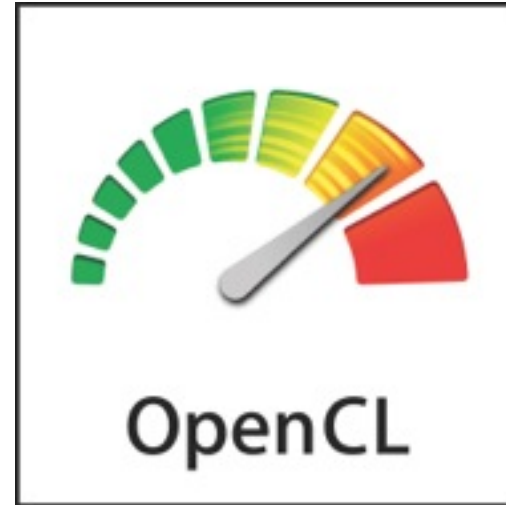


OpenCL provides vendor optimized, cross-platform, cross-vendor access to heterogeneous compute resources

# Builds on LLVM and OpenCL



- Optimizing compiler toolkit
- Portable, flexible, well understood
- Open source platform for innovation



- Proven platform for heterogeneous parallel programming
- Multi-vendor: CPU, GPU, FPGA etc.

# Why use SPIR?

- Without SPIR:
  - Vendors shipping source
    - Risk IP leakage
  - Vendors shipping multiple binaries
    - Complexity
    - Miss optimizations in new compilers
    - Forward compatibility issues
- With SPIR:
  - Ship a single binary per platform
    - E.g. SPIR file can support Intel & AMD
  - Many vendors support SPIR consumption
  - Shipped application can retarget new devices and new vendors

*Opportunity to unleash innovation:  
Domain Specific Languages, C++ Compilers, Halide, ....*

# What's new in SPIR 2.0?

- Full support of OpenCL 2.0 “C” kernel language
  - Generic address space
  - Device side kernel enqueue
  - C++11 atomics
  - Pipes
  - More...
- LLVM 3.4 with restrictions and conventions

*If you can do it in OpenCL C  
You can do it in SPIR*

# SPIR ecosystem is...

- **IR definition**
  - Portable non-source encoding for OpenCL 1.2 or 2.0 device programs
  - SPIR 1.2 is based on LLVM 3.2
  - SPIR 2.0 is based on LLVM 3.4
- **Consumption API for target hardware**
  - cl\_khr\_spir extension to OpenCL runtime API
- **Example generator**
  - Open source patch to Clang translates OpenCL C to SPIR IR
  - Available in github: <https://github.com/KhronosGroup/SPIR>
- **Ease of use tools**
  - SPIR Verifier, SPIR built-ins name mangler
  - Available in github: <https://github.com/KhronosGroup/SPIR-Tools>

# Longevity and Versioning

- SPIR to track both LLVM and OpenCL versions
  - SPIR 1.2 ← LLVM 3.2 + OpenCL 1.2
  - SPIR 2.0 ← LLVM 3.4 + OpenCL 2.0
- SPIR consumer tells you what versions can be loaded
- Khronos members contributing to mainline LLVM+Clang
  - Backward compatibility fixes and tests
  - Full SPIR support in Clang
  - Ease of use tools



# Call to Action

- **Seeking feedback on SPIR 2.0 provisional**
  - A Provisional specification
  - <http://www.khronos.org/registry/spir/>
  - [https://www.khronos.org/opencl/spir2\\_0\\_feedback\\_forum](https://www.khronos.org/opencl/spir2_0_feedback_forum)
- **Innovate on the Front end**
  - New languages, abstractions
  - Target production quality backends
- **Innovate on the Back end**
  - New target platforms: Multi core, Vector, VLIW...
  - Reuse production quality frontends
- **Innovate on Tooling**
  - Program analysis, optimization



# Getting Started

- IR Specification

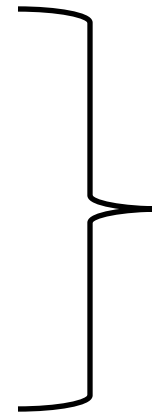
- Khronos SPIR registry - <http://www.khronos.org/registry/spir/>

- Front end

- Khronos-patched Clang from Github

- Verifier

- LLVM pass checks SPIR validity
- Khronos Github



Same open source license  
as mainline  
LLVM and Clang

- Backend

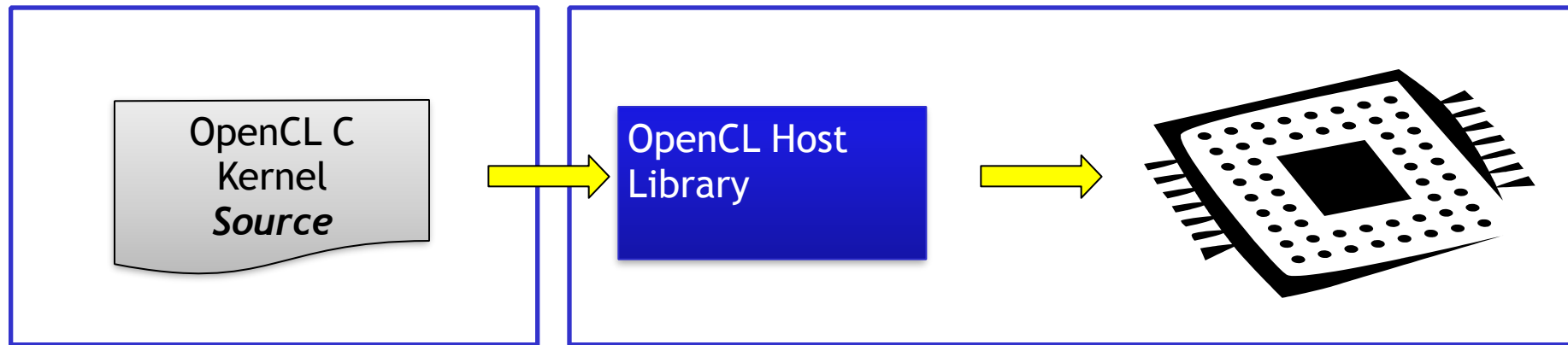
- Check your favorite OpenCL implementation for cl\_khr\_spir



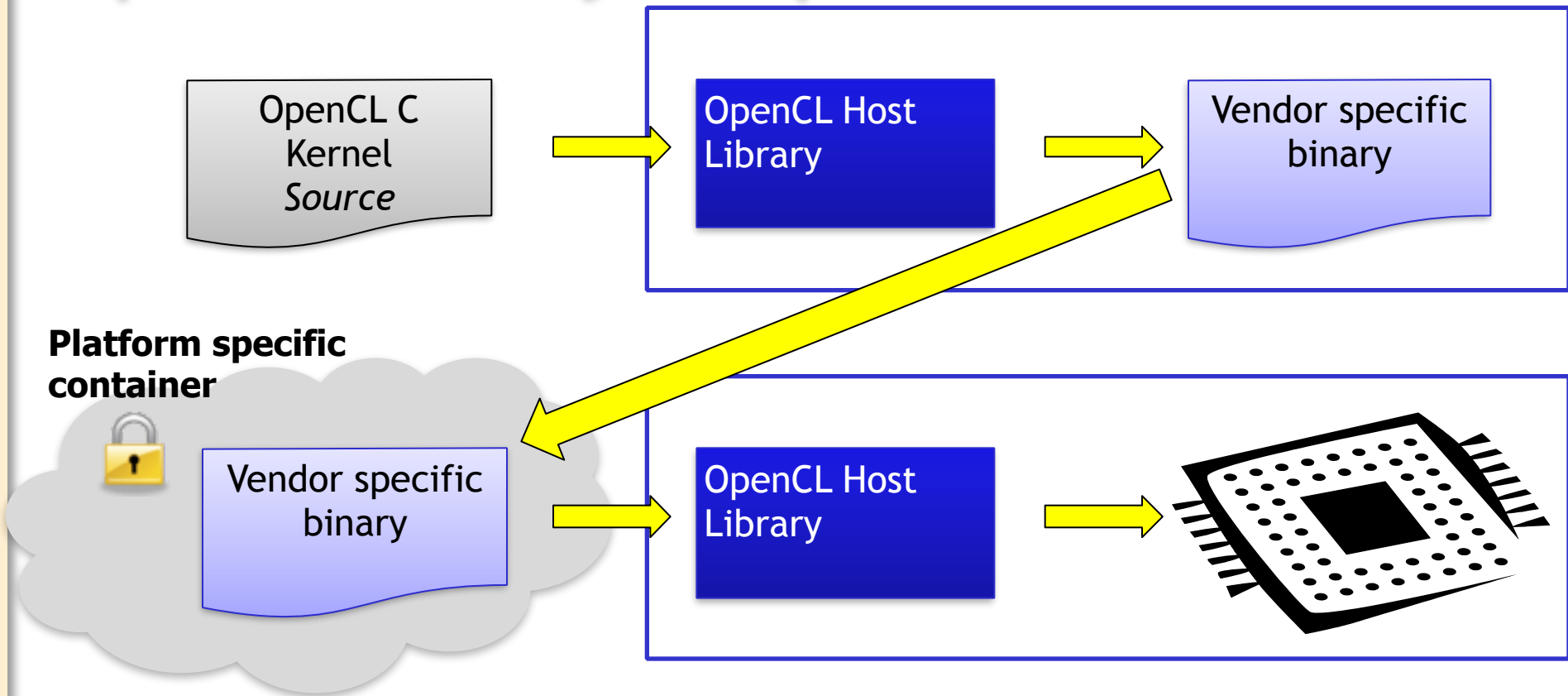
# More About Flows

# OpenCL: Source Compilation Flow

- ISV ships their kernel source
  - Exposes their IP
- Supports only OpenCL C



# OpenCL: Binary compilation flow

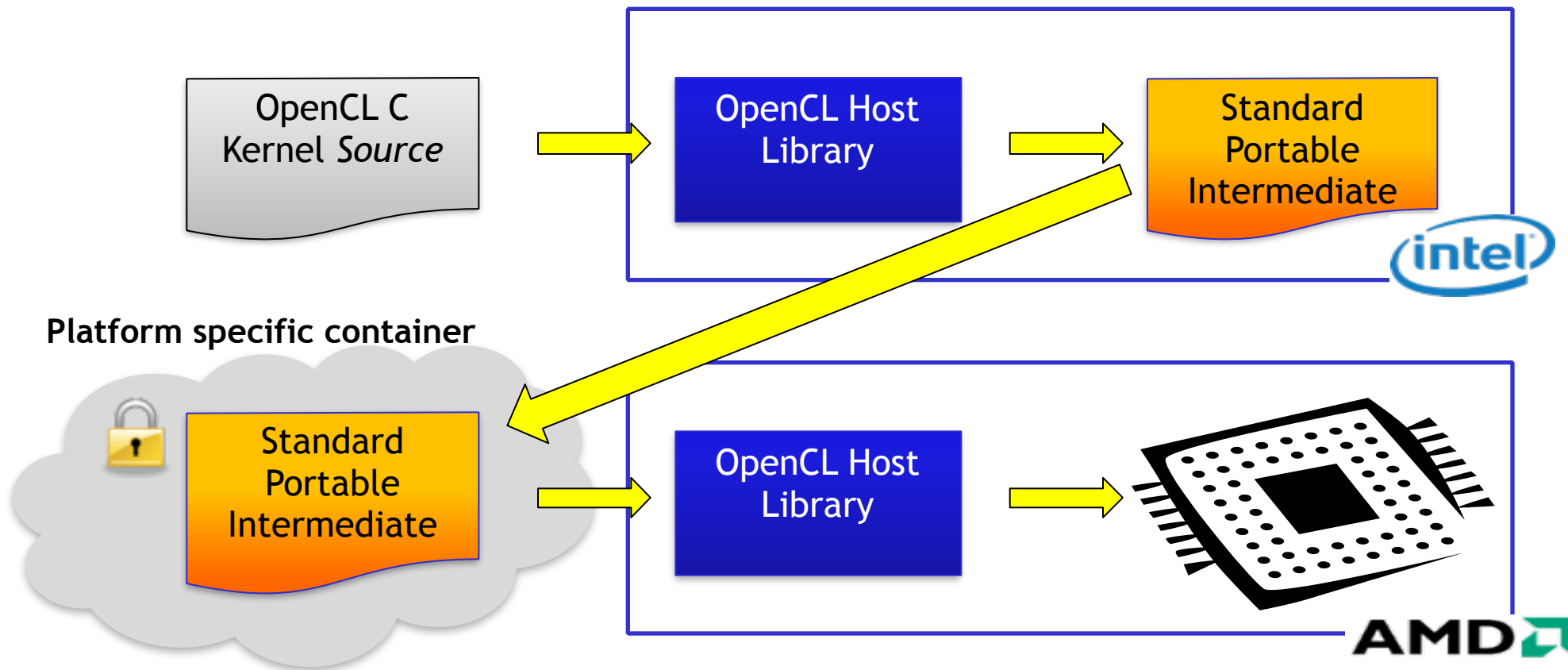


- ISV ships vendor-specific binary
  - Proliferation: devices, driver revisions, vendors
  - Market-lagging: target *shipped* products

# OpenCL: SPIR flow

ISV ships kernels in SPIR form

- User runs application on platform of their choice



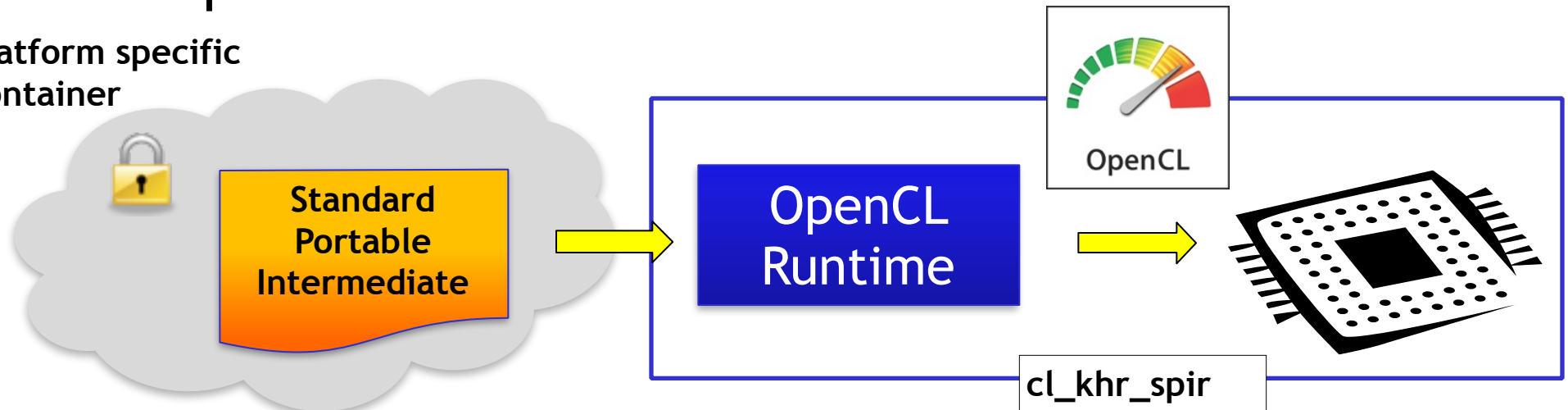
# SPIR Reference Flow

## Generation



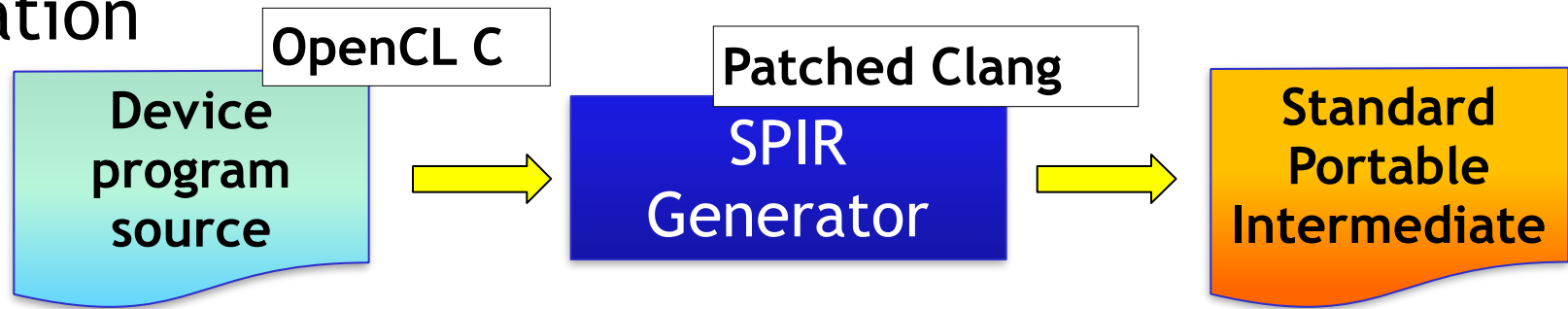
## Consumption

Platform specific container

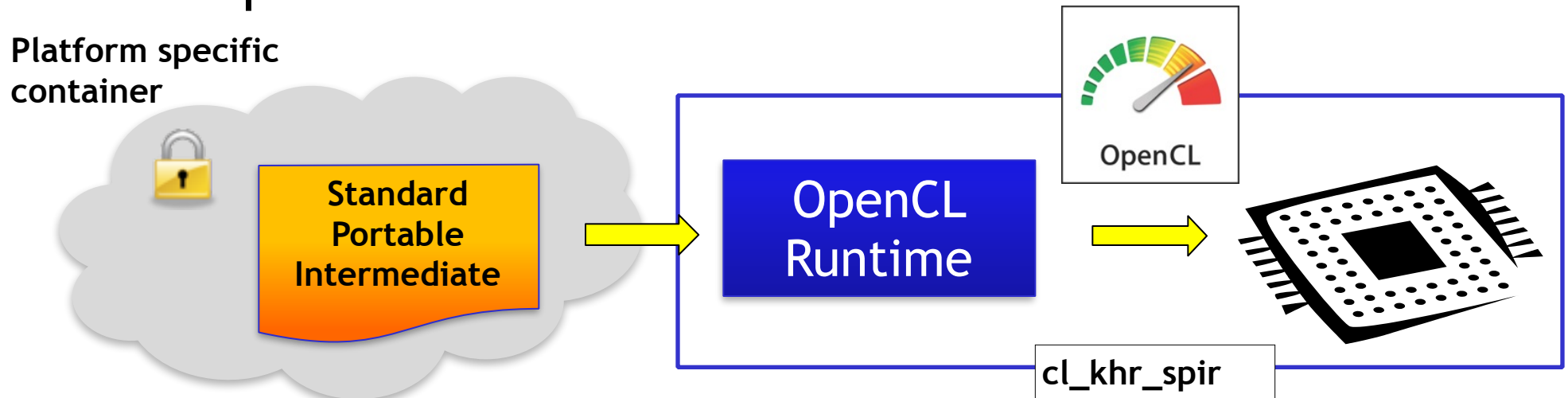


# SPIR Today

## Generation

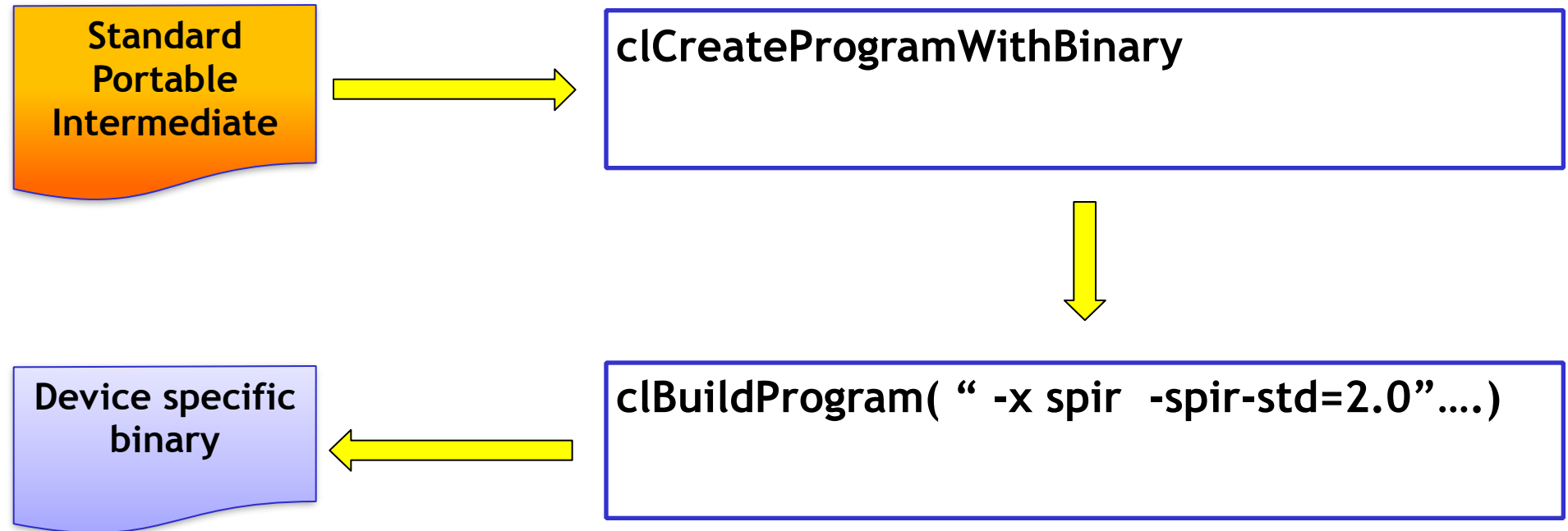


## Consumption

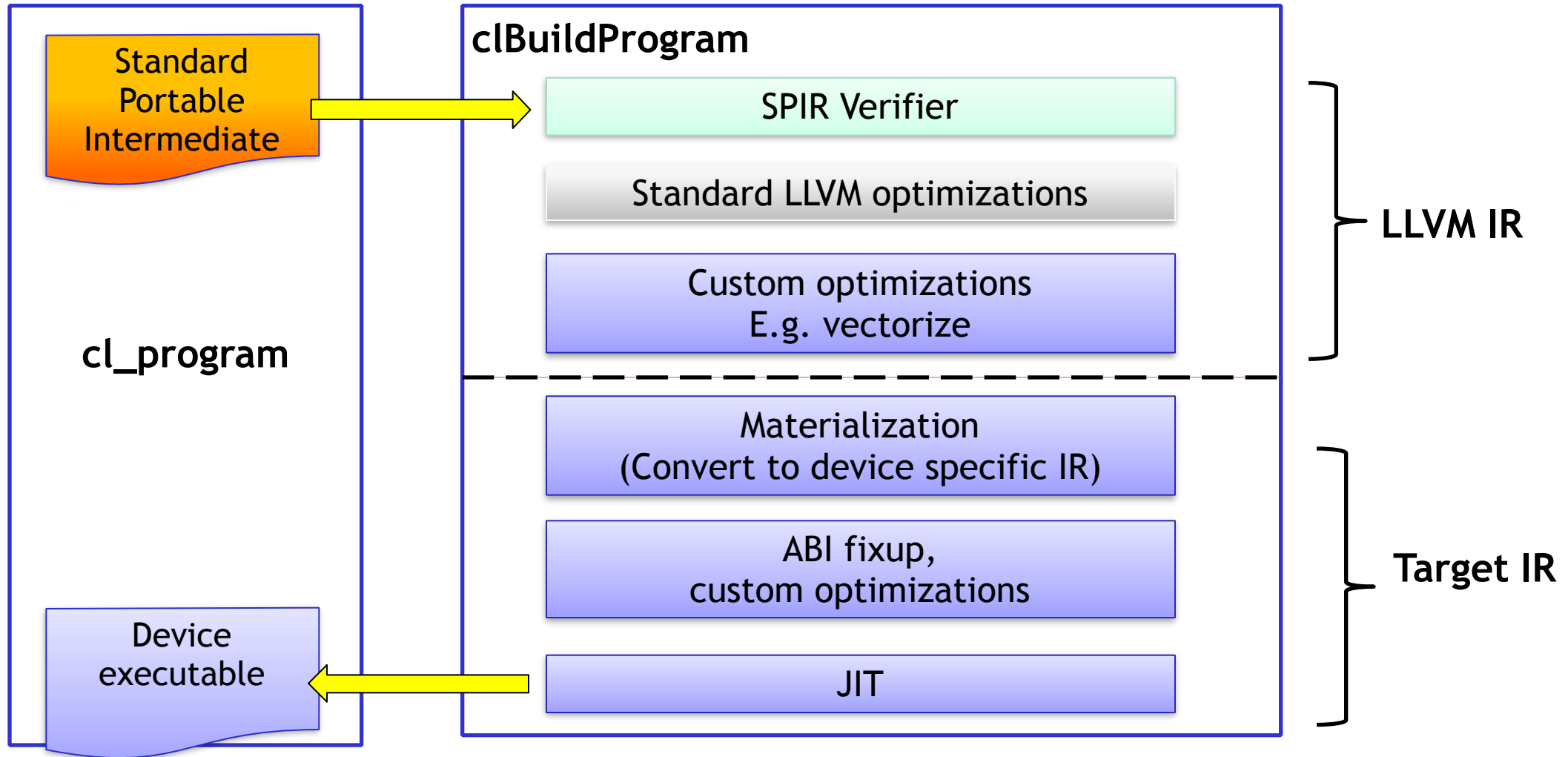




# Sample SPIR Consumption Flow



# Sample SPIR Flow: Room for Optimizations



# Resources

- IR Specification
  - Khronos SPIR registry - <http://www.khronos.org/registry/spir/>
- Feedback Forum Thread
  - [https://www.khronos.org/opencl/spir2\\_0\\_feedback\\_forum](https://www.khronos.org/opencl/spir2_0_feedback_forum)
- Khronos-patch Clang and Tools
  - <https://github.com/KhronosGroup/SPIR>
  - <https://github.com/KhronosGroup/SPIR-Tools>
- Backend
  - Check your favorite OpenCL implementation for cl\_khr\_spir



**Questions?**