

K H R O N O S
G R O U P[™]

SPIR[™] **1.2**

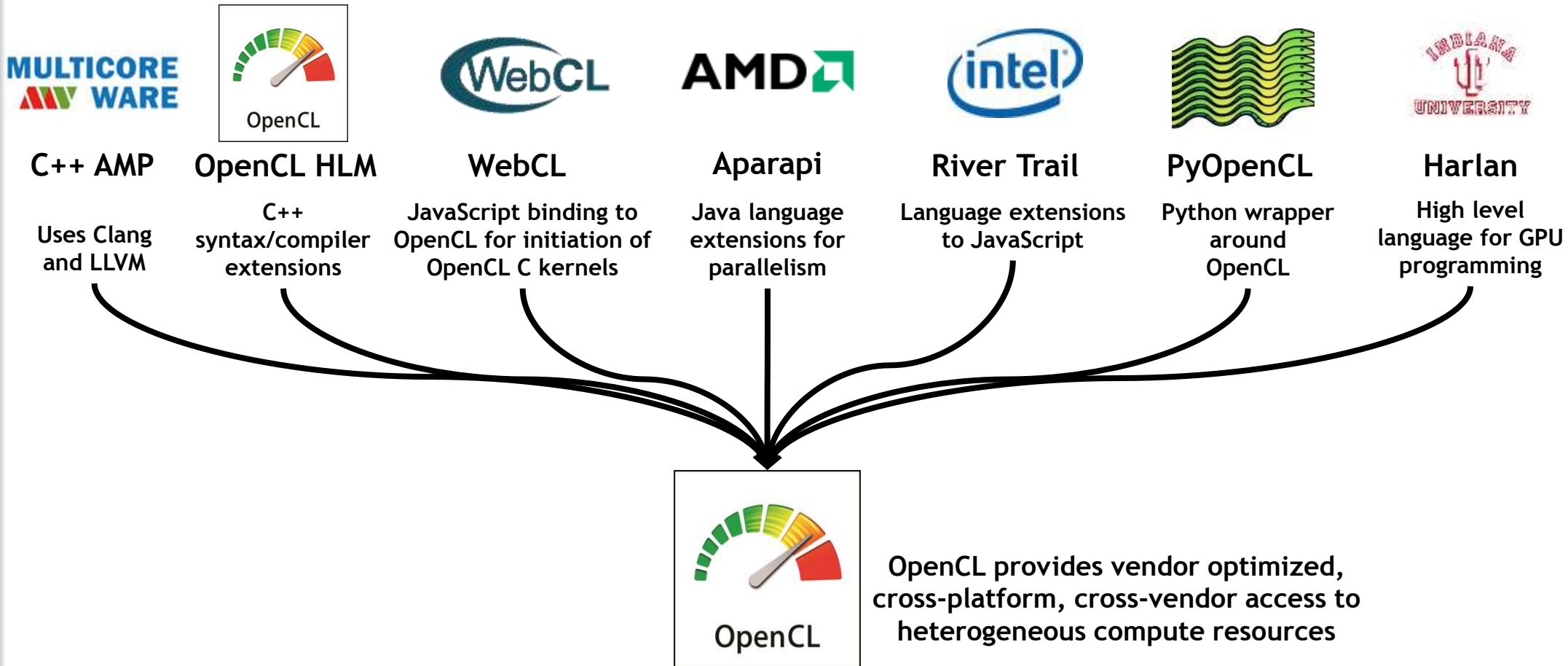
HiPEAC 2014
January 2014



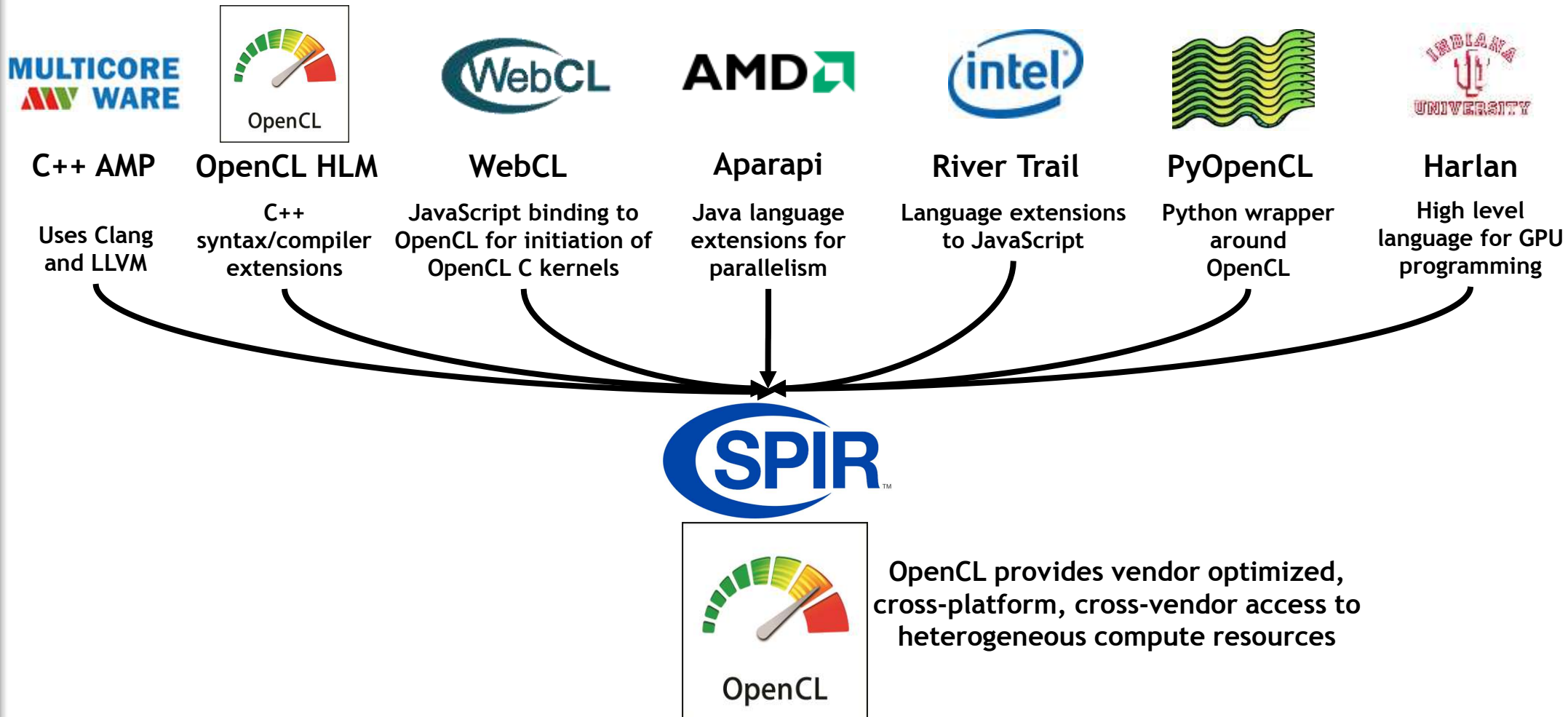
Standard
Portable
Intermediate
Representation

*Enables compiler ecosystem for
portable parallel programs*

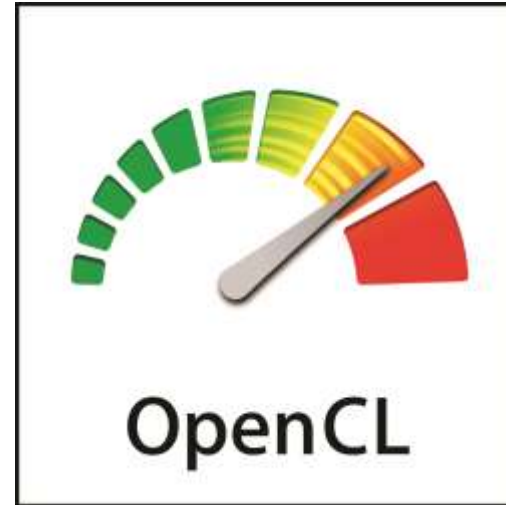
OpenCL as Parallel Compute Foundation



SPIR as compiler-oriented middleware



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

SPIR ecosystem is...

- **IR definition**

- Portable non-source encoding for OpenCL 1.2 device programs
- Based on LLVM 3.2
- Khronos ratified specification
 - Thorough open consultation with LLVM community

- **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

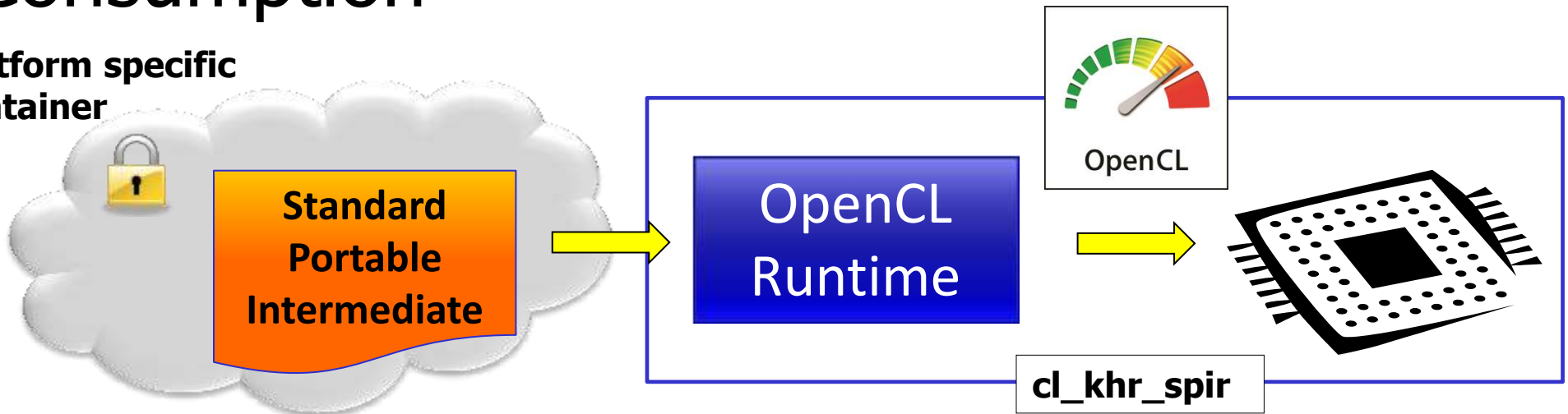
SPIR reference flow

Generation



Consumption

Platform specific container



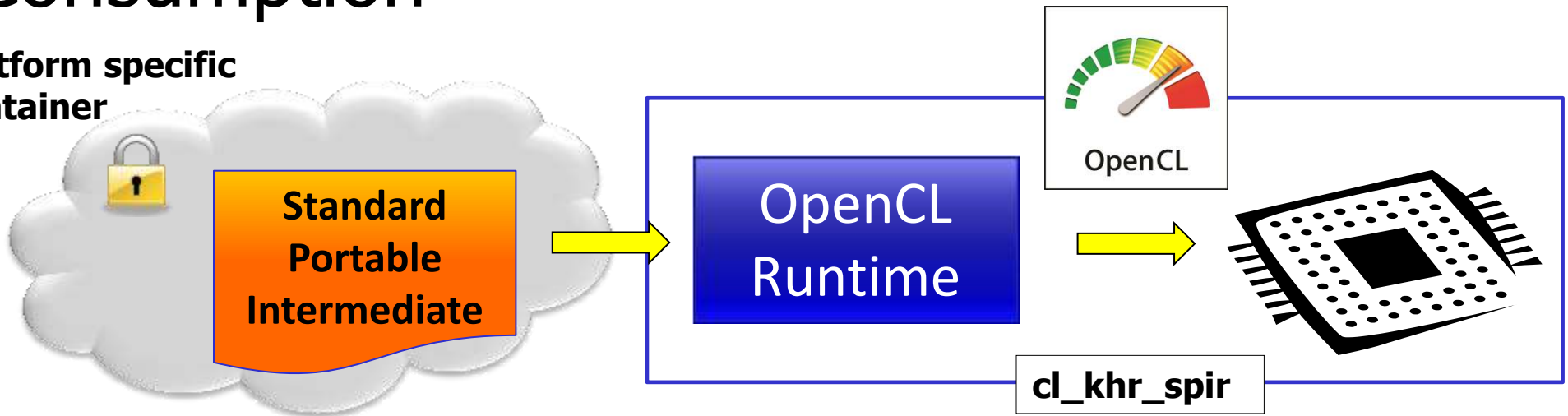
SPIR today

Generation



Consumption

Platform specific container



Call to Action

- **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



Why use SPIR?

- **Without SPIR:**

- Some vendors shipping source
 - Risk IP leakage
- Some vendors shipping multiple binaries
 - Complexity
 - Miss optimizations in new compilers
 - Forward compatibility issues

- **With SPIR:**

- Ship a single binary per platform
 - (Example: one SPIR file can be supported on both Intel and AMD)
- Vendor must support SPIR consumption extension
- Shipped application can retarget new devices and new vendors

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

Longevity and versioning

- **SPIR to track both LLVM and OpenCL versions**

- Today: SPIR 1.2 ← LLVM 3.2 + OpenCL 1.2
- Tomorrow: SPIR 2.0 ← LLVM ? + OpenCL 2.0



- **SPIR consumer tells you what versions can be loaded**

- OpenCL 2.0 cl_khr_spir platform could run both SPIR 1.2 and SPIR 2.0

- **Khronos members contributing to mainline LLVM+Clang**

- Backward compatibility fixes and tests
- Full SPIR support in Clang

A taste for SPIR IR

- **LLVM 3.2 with restrictions and conventions**
- **Fixed set of primitive data types matching OpenCL C**
 - Bool, standard integers, half, float, double
 - Vectors of length 2, 3, 4, 8, 16
- **Supports images, samplers, events**
- **Captures metadata about OpenCL programs**
 - E.g. kernel attributes, argument info
- **Mangling for overloaded builtin functions in OpenCL C**
- **OpenCL NDRange parallel implicit loop for kernel functions**

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

Getting Started

- **IR Specification**

- Khronos OpenCL registry www.khronos.org/registry/cl/

- **Front end**

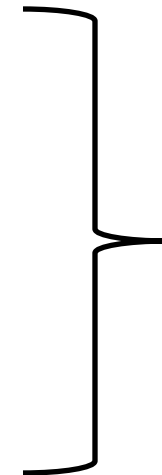
- Khronos-patched Clang 3.2 from Github

- **Verifier**

- LLVM pass checks SPIR validity
- Khronos Github

- **Backend**

- Check your favorite OpenCL implementation for cl_khr_spir



Same open source license as mainline LLVM and Clang



Thank you