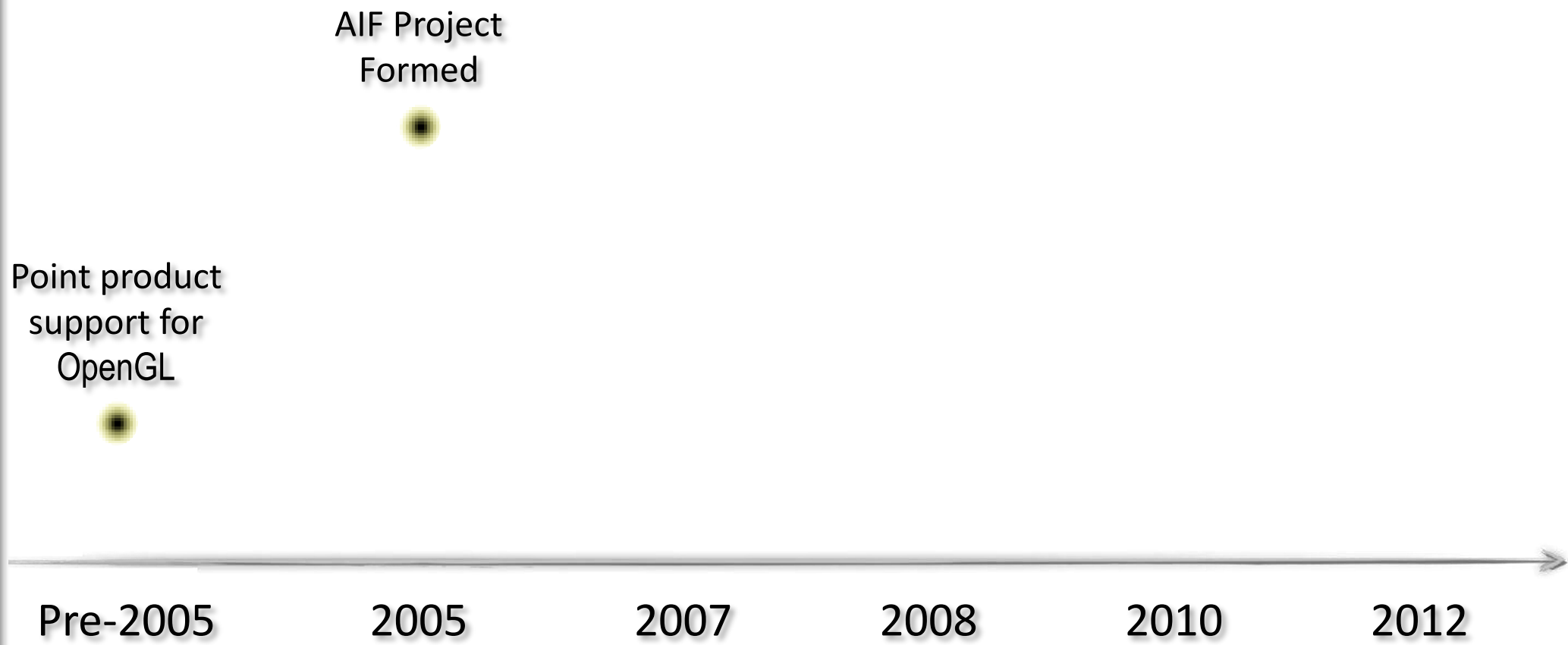# OpenCL @ Adobe

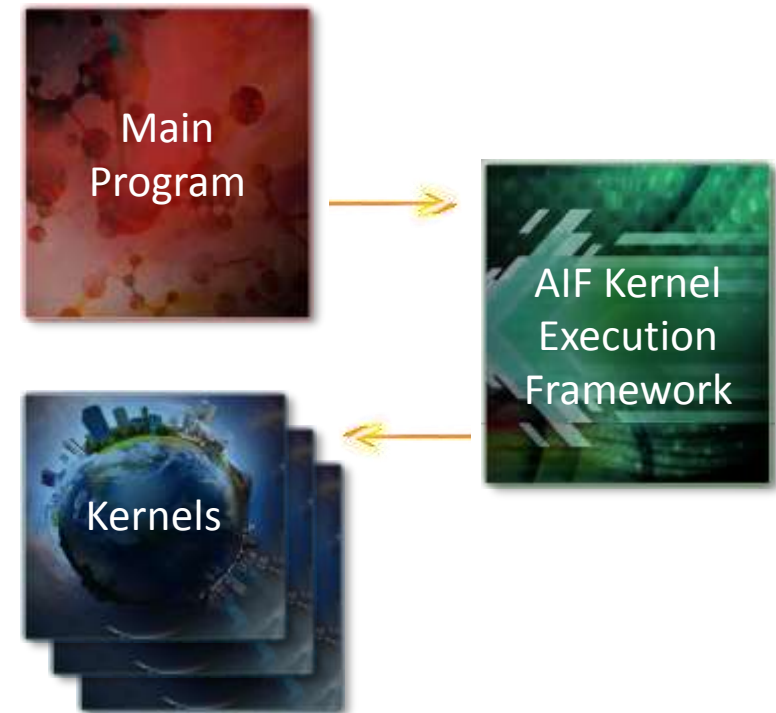**Eric Berdahl**
Adobe Imaging Foundation Engineering Manager
Adobe Systems Inc.

History of Heterogeneous Computing at Adobe

AIF Project
Formed

Point product
support for
OpenGL

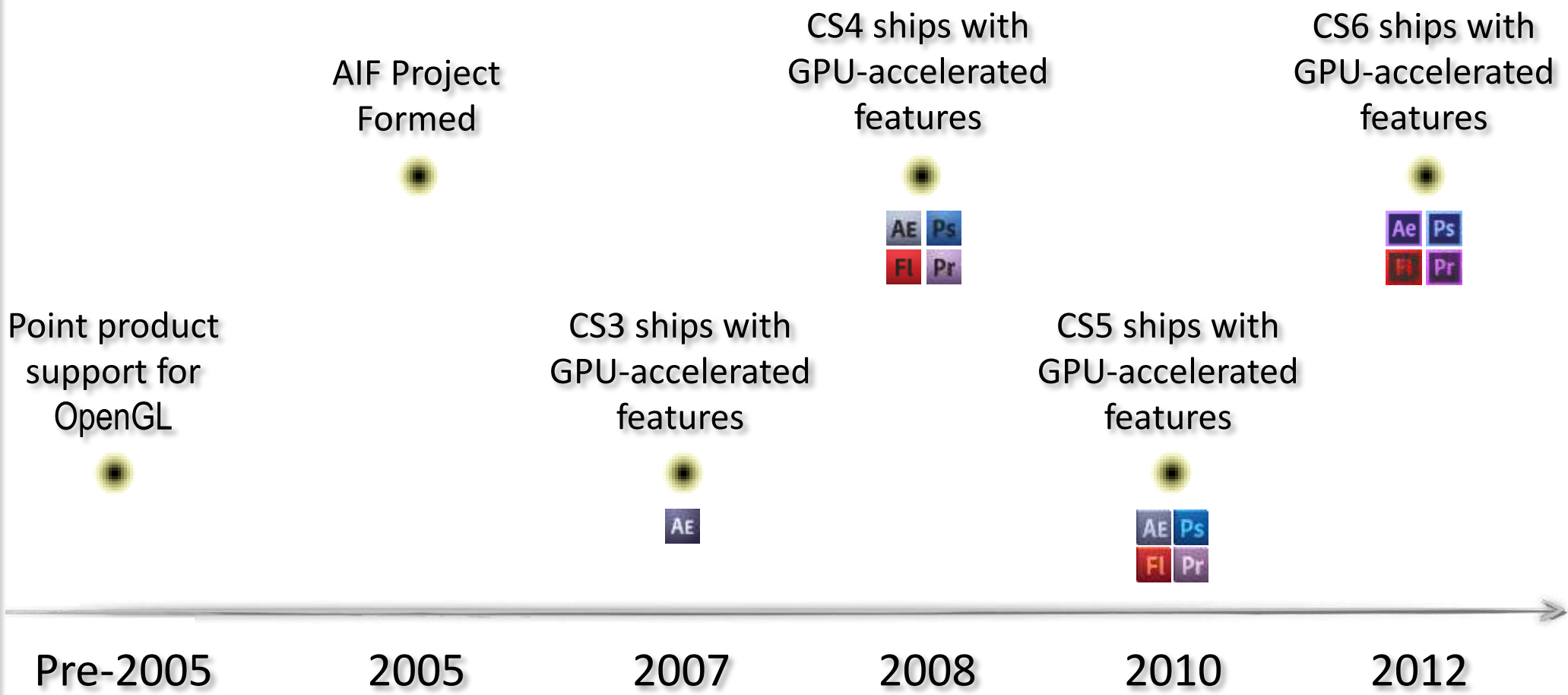Pre-2005   2005   2007   2008   2010   2012
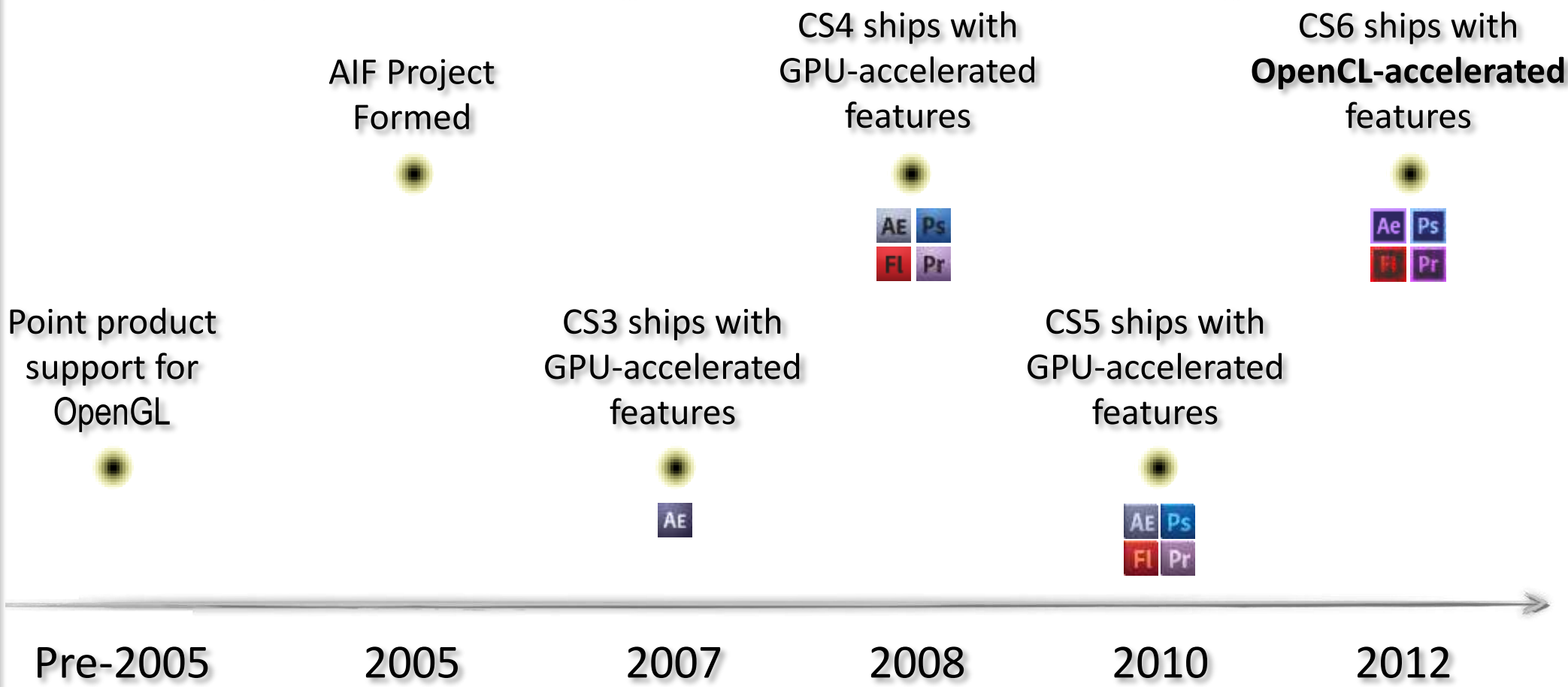
Adobe Imaging Foundation

- **Image processing system**
  - Apply filters to images
  - Do for pixels what OpenGL does for triangles

- **Support for heterogeneous hardware**
  - GPU, (multi-core) CPU

- **Framework**

- **Programmable language (Pixel Bender) for kernels**

Main Program

AIF Kernel Execution Framework

Kernels

History of Heterogeneous Computing at Adobe

CS4 ships with
GPU-accelerated
features

CS6 ships with
GPU-accelerated
features

AIF Project
Formed

Point product
support for
OpenGL

CS3 ships with
GPU-accelerated
features

CS5 ships with
GPU-accelerated
features

Pre-2005   2005   2007   2008   2010   2012

History of Heterogeneous Computing at Adobe

CS4 ships with
GPU-accelerated
features

CS6 ships with
OpenCL-accelerated
features

AIF Project
Formed

Point product
support for
OpenGL

CS3 ships with
GPU-accelerated
features

CS5 ships with
GPU-accelerated
features

Pre-2005   2005   2007   2008   2010   2012

Adobe ♥ OpenCL

- **Compute API supported across vendors**

- **Programming model familiar to C programmers**

- **Demonstrated performance**

- **Same compute kernels on CPU and GPU!**

- **Adobe is now active member of OpenCL working group**

  - Contributing Adobe's experience and minds to continue OpenCL evolution

OpenCL in Photoshop CS6

Sarah Kong
Photoshop Engineering Manager
Adobe Systems Inc.

Blur Gallery Demo

SIGGRAPH 2012

KHRONOS GROUP

Why OpenCL

- **Only cross-platform GPGPU solution**

- **Advantages over OpenGL**
    - Learning curves; data formats; debugging

- **Increasing maturity and ubiquity**

How did we do it?

- **OpenCL kernel was naïve port of SSE2 function (scheduled with TBB on CPU)**
- **Broken into 2K x 2K blocks for GPU**

Challenges

- **Need good candidate algorithms**
  - Bandwidth, compute, parallel

Challenges

- **Need good candidate algorithms**
    - Bandwidth, compute, parallel
- **Need debugged C algorithm first**

Challenges

- **Need good candidate algorithms**
    - Bandwidth, compute, parallel
- **Need debugged C algorithm first**
- **GPU win over CPU variable and unpredictable**

Challenges

- **Need good candidate algorithms**
  - Bandwidth, compute, parallel

- **Need debugged C algorithm first**

- **GPU win over CPU variable and unpredictable**

- **Resource limits**
  - Win/Mac Timeout issues on low end cards.

Challenges

- **Need good candidate algorithms**
  - Bandwidth, compute, parallel

- **Need debugged C algorithm first**

- **GPU win over CPU variable and unpredictable**

- **Resource limits**
  - Win/Mac Timeout issues on low end cards.
  - Out of memory issues on low end cards.

Challenges

- **Need good candidate algorithms**
  - Bandwidth, compute, parallel

- **Need debugged C algorithm first**

- **GPU win over CPU variable and unpredictable**

- **Resource limits**
  - Win/Mac Timeout issues on low end cards.
  - Out of memory issues on low end cards.
  - Used bandwidth test to enable OpenCL acceleration
    - Affected by power-gating on mobile GPU

Challenges

- **Need good candidate algorithms**
  - Bandwidth, compute, parallel

- **Need debugged C algorithm first**

- **GPU win over CPU variable and unpredictable**

- **Resource limits**
  - Win/Mac Timeout issues on low end cards.
  - Out of memory issues on low end cards.
  - Used bandwidth test to enable OpenCL acceleration
    - Affected by power-gating on mobile GPU

- **Platform variation**
  - Driver Issues
  - Various compiler issues

Performance Comparison

- **Systems of standard configuration show 4-8x gain for typical use-cases**
- **Gains improve with Blur radii (results from MacBookPro running 10.7.4 listed below)**
  - General application processing accounts for majority of time in smaller workloads

Radius in Pixels (13 Megapixel image)	Radeon 6750m	Core i7 (2.3 GHz)
50	4.7s	16.2s
100	7.0s	31.9s
150	9.4s	47.9s

Future Plans

- **More OpenCL in future versions**
- **Investigate OpenCL for CPU**

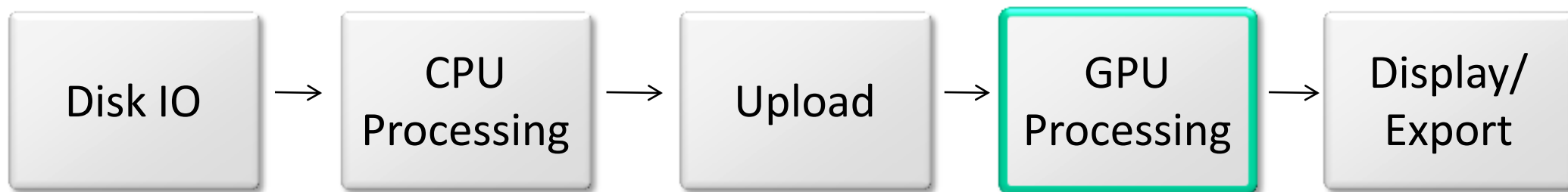# OpenCL in Premiere Pro CS

**David McGavran
Premiere Pro Engineering Manager
Adobe Systems Inc.**

Demo

Pipeline

Disk IO → CPU Processing → Upload → **GPU Processing** → Display/Export

Accelerated Effects

Intrinsics
- Adjustment layers
- Color space conversion
- Deinterlacing
- Compositing
- Blending modes
- Nested Sequences
- Multicam
- Time remapping

Transitions
- Additive dissolve
- Cross dissolve
- Dip to black
- Dip to white
- Film Dissolve
- Push

Effects
- Alpha adjust
- Basic 3D
- Black & white
- Brightness & contrast
- Color balance
- Color pass
- Color replace
- Crop
- Directional blur
- Drop Shadow
- Extract
- Fast blur
- Fast color corrector
- Feather edges
- Gamma correction
- Garbage matte
- Gaussian blur
- Horizontal flip

- Invert
- Luma corrector
- Luma curve
- Noise
- Proc amp
- RGB color corrector
- RGB curves
- Sharpen
- Three-way color corrector
- Timecode
- Tint
- Track matte
- Ultra keyer
- Vertical flip
- Video limiter
- Warp Stabilizer

Implementation

- **Pipeline entirely floating point**

- **10-bit display supported**

- **Subtree rendering for non-accelerated effects**

- **Draw with OpenGL interop**

Filter Concatenation

- **Entirely manual**
- **Host kernel for successive pointwise effects**
- **Read once, apply multiple filters, write once**
- **Major bandwidth savings**

Future

- **Increase set of supported effects**
- **Supporting third party effects & codecs**
- **GPU encoding & decoding**
- **Multiple GPU support**
- **GPU Scopes**