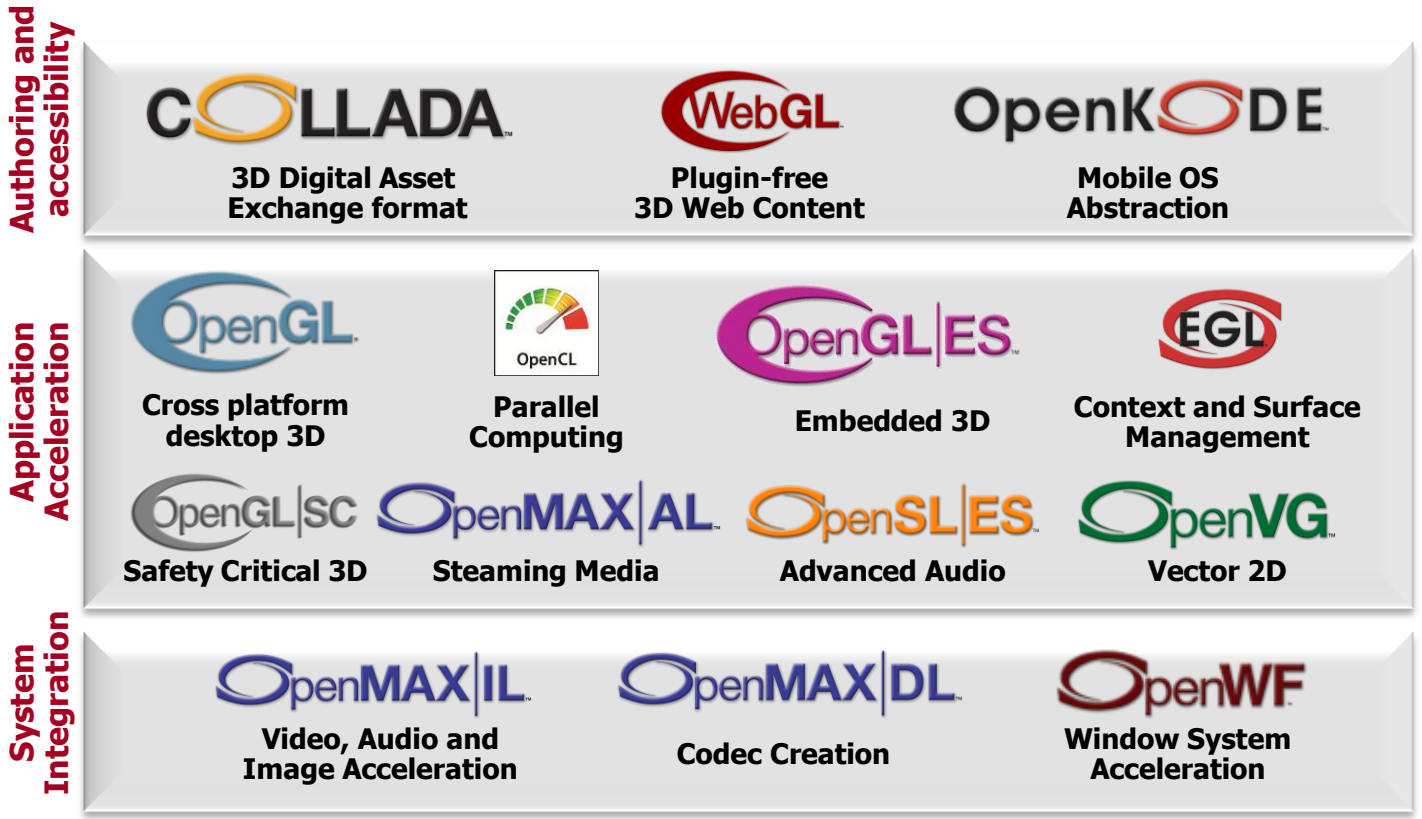
Applications and Implementations

Hwanyong LEE
CTO and Technical Marketing Director
HUONE

Khronos Family of Standards



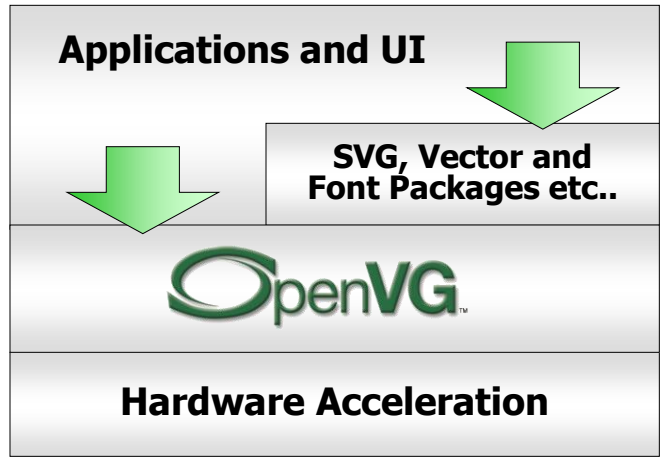
KHRONOS GROUP

A coordinated ecosystem of compute, graphics and media standards and APIs

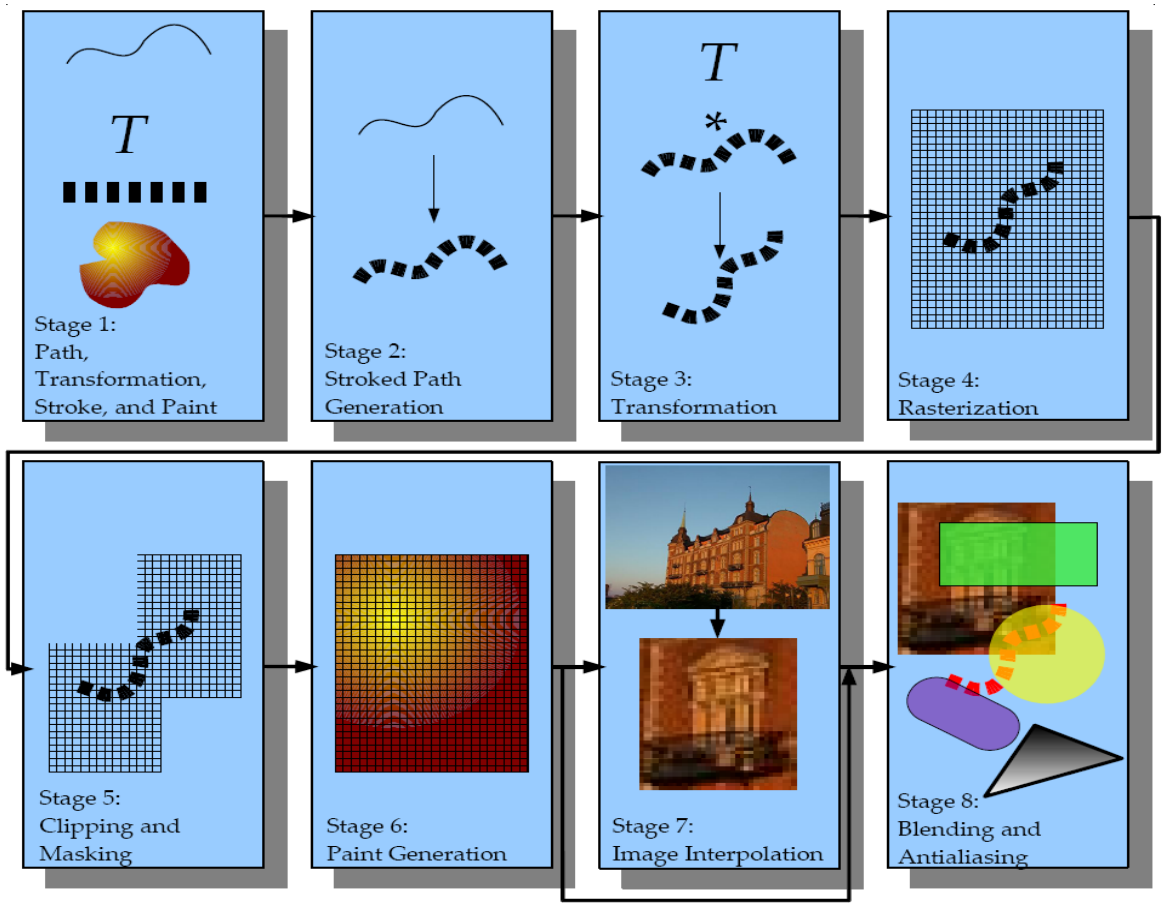
Hundreds of man years invested by industry experts to create a coordinated visual computing ecosystem for accelerated parallel computation, 3D, video, audio and image processing on desktop, embedded and mobile systems

OpenVG

- **Royalty-free open standard API**
- **Low-level 2D vector graphics rendering API**
- **OpenGL-style programming model**
- **Advanced feature set enables**
  - SVG,
  - Flash,
  - PDF, Postscript,
  - Java (JSR 287, 271, 226)
  - etc.
- **Portable content**
- **Map Applications**
- **Hardware Acceleration**

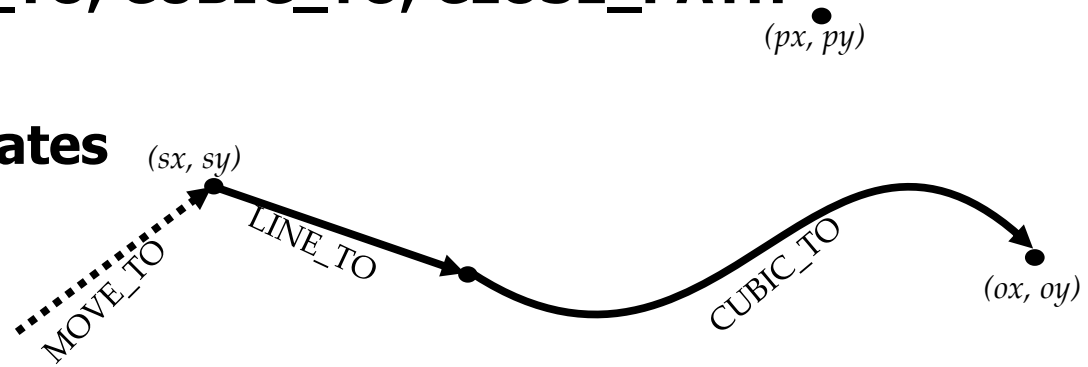


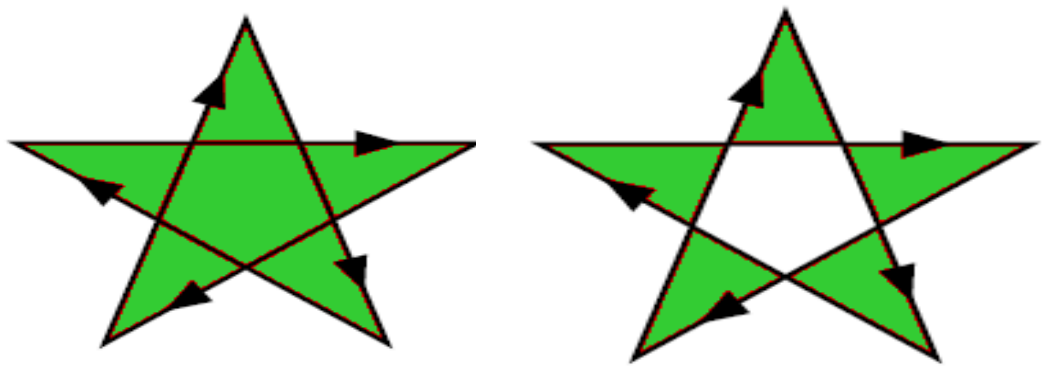
OpenVG Rendering Pipeline



Path Definition

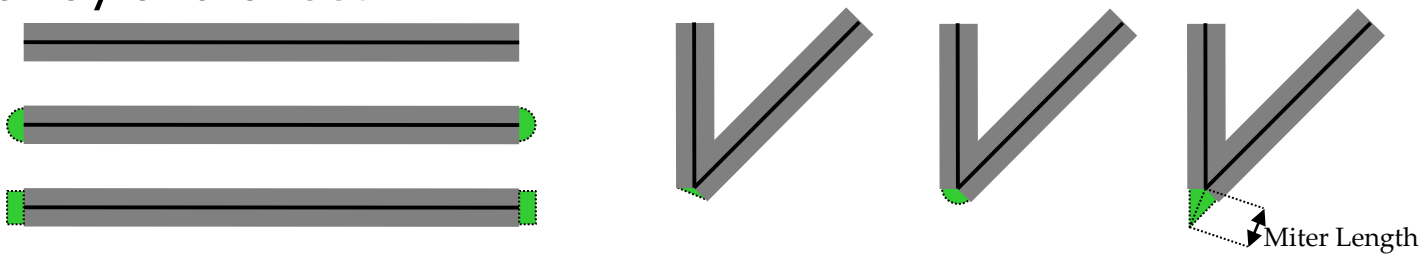
- **MOVE_TO, LINE_TO, QUAD_TO, CUBIC_TO, CLOSE_PATH**
- **Elliptical Arcs**
- **Absolute / Relative Coordinates**
- **Smooth Curves**
- **Path Interpolation**
- **Path Queries:**
  - Bounding Boxes
  - Transformed Bounding Boxes
  - Point along path
  - Tangent along path
- **Non-Zero and Even-Odd fill rules**



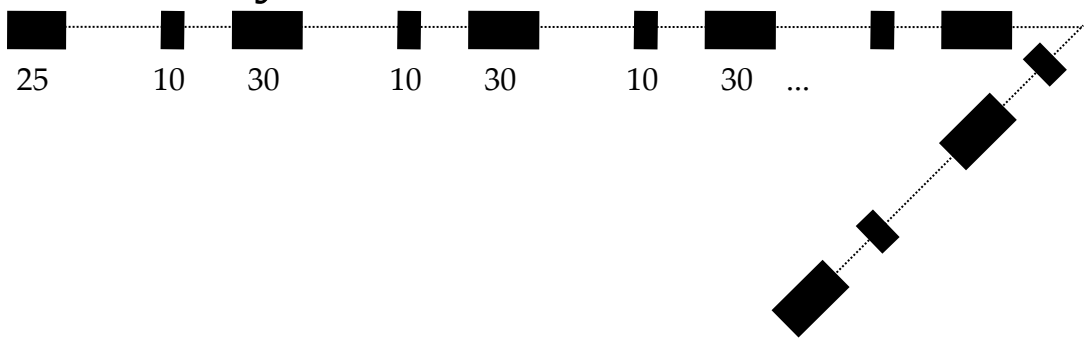


Stroking

- **Stroking takes a path and defines an outline around it:**
  - Line Width
  - End cap style (Butt, Round, or Square)
  - Line join style (Bevel, Round, or Miter)
  - Miter limit (to convert long miters to bevels)
  - Dash array and offset



Dash array = { 10, 20, 30, 40} / Dash Phase = 35



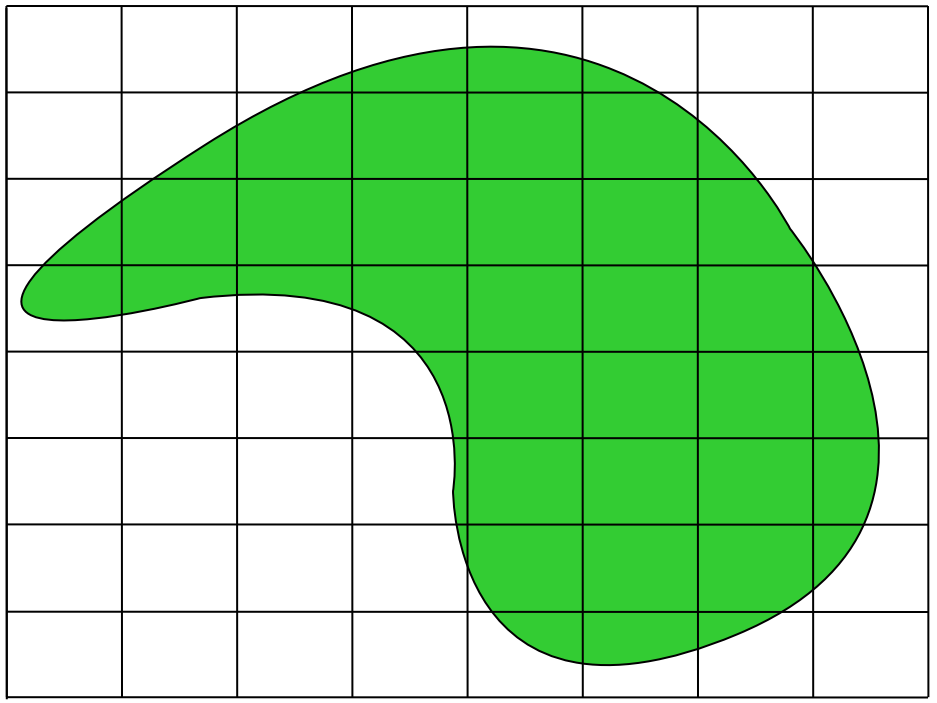
Transformations

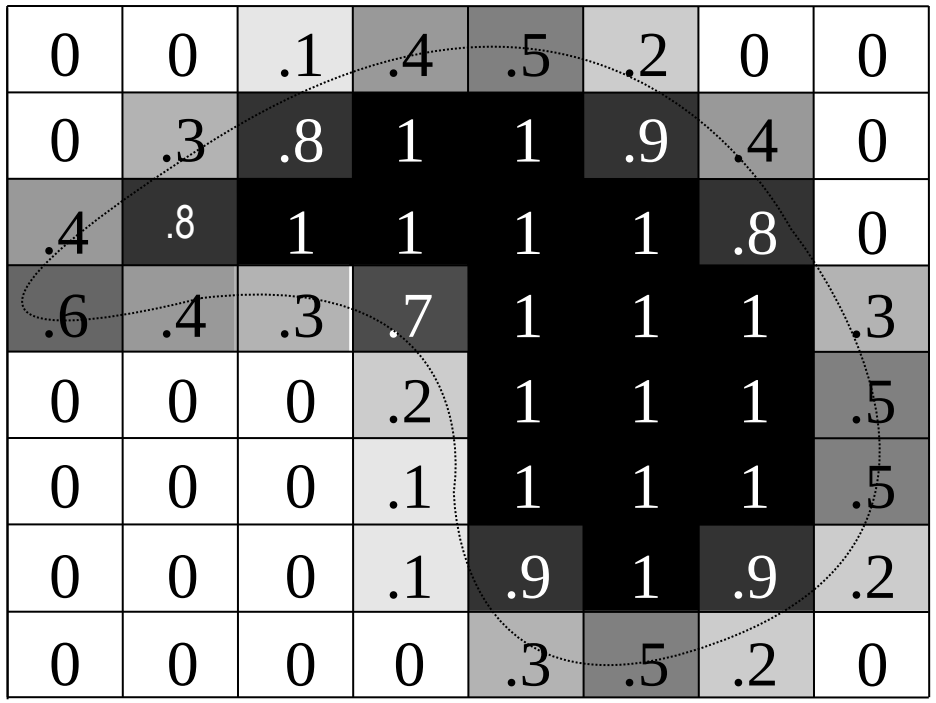
- **Paths use 2x3 affine transformations**
- **Images use 3x3 perspective transformations**
- **Transformation functions are similar to OpenGL:**
  - vgLoadIdentity
  - vgLoadMatrix
  - vgGetMatrix
  - vgMultMatrix
  - vgScale
  - vgRotate
  - vgTranslate
  - vgShear

$$\begin{bmatrix} 1.080 & 0.101 & 0 \\ 0.209 & 0.691 & 0 \\ 1.28\times 10^{-3} & -1.19\times 10^{-3} & 1 \end{bmatrix}$$

Rasterization (continued)

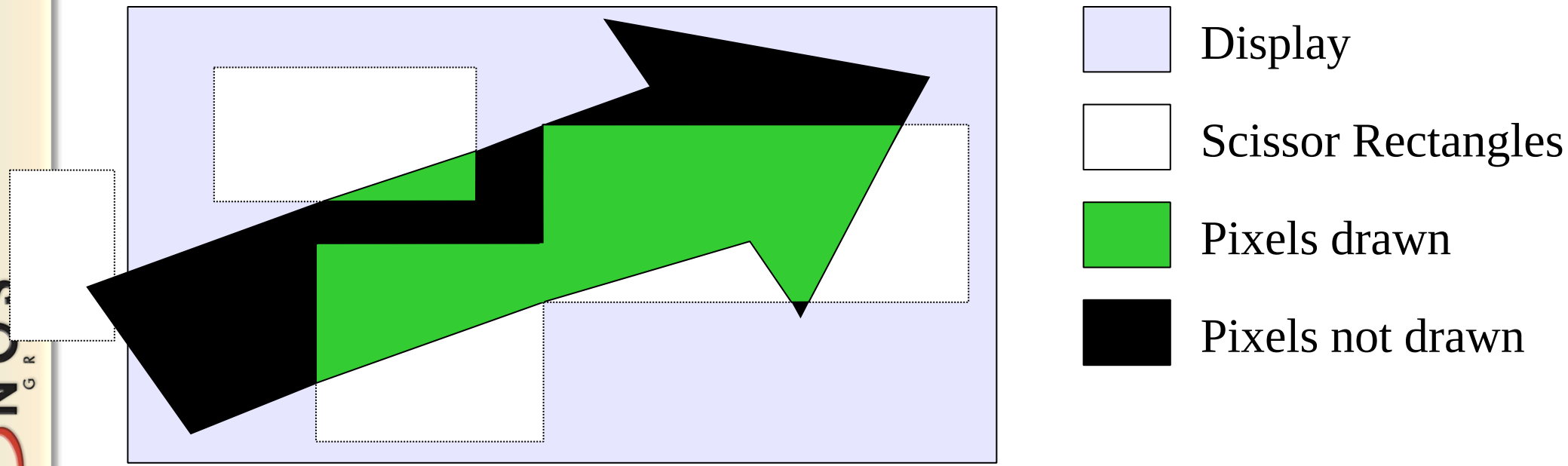
- The goal of rasterization is to determine a filtered alpha value for each pixel, based on the geometry around that pixel
- Filters may be up to 3 pixels in diameter





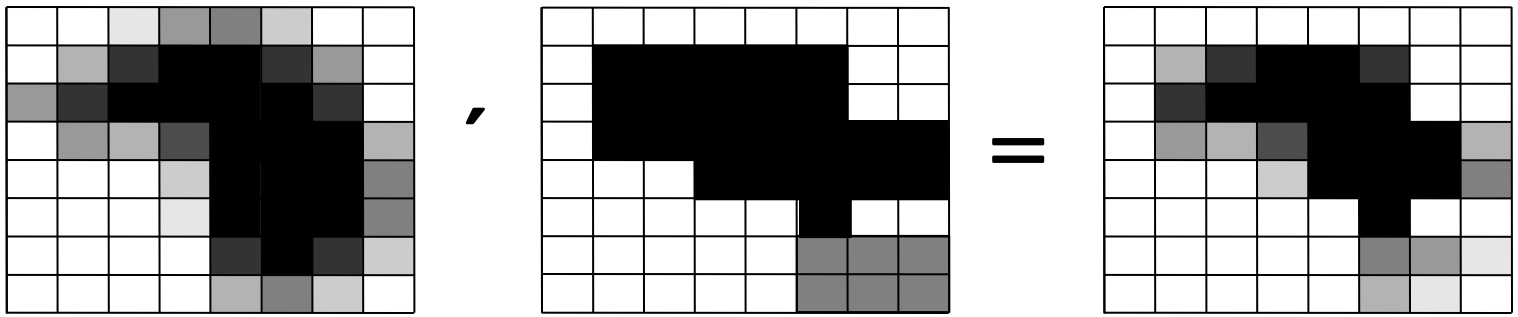
Scissoring

- Only pixels inside a set of scissor rectangles are drawn
- Scissoring is disabled by default



Masking

- In addition to scissoring, a per-pixel mask may be applied
- The mask has an alpha value at each pixel that is multiplied by the alpha from the rendering stage
- May be used to "cut out" an area, create area transitions
- Mask values may be modified using image data
  - Fill, Clear, Set, Add, Subtract, Intersect



Paint Generation

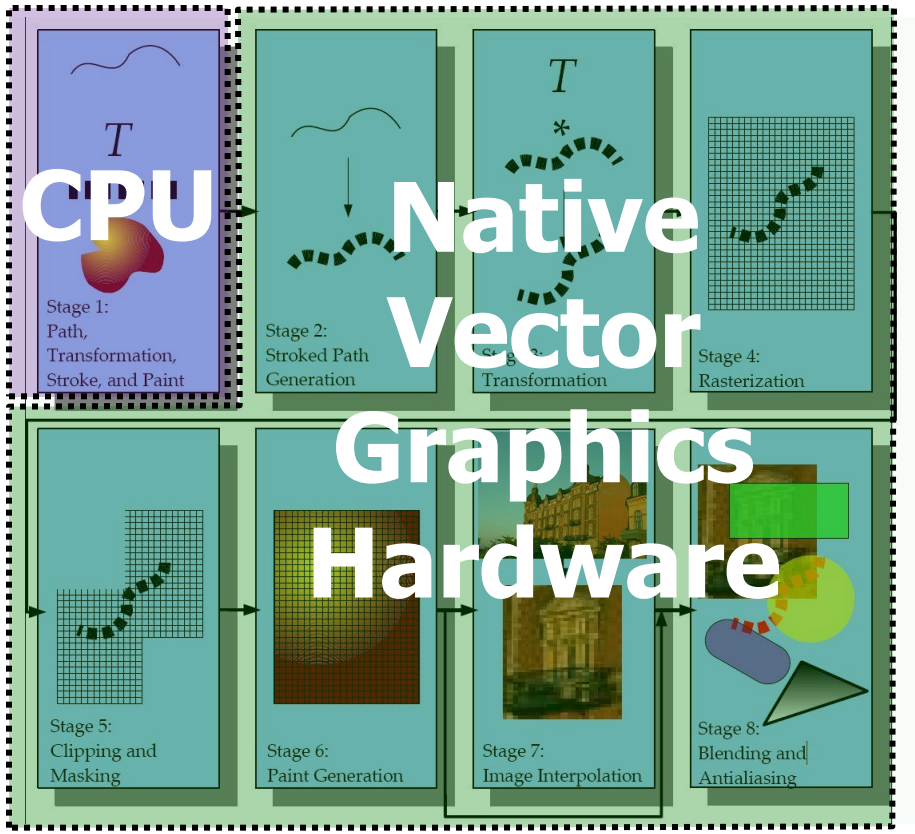
- **Paint is generated pixel-by-pixel and applied to geometry**
- **The alpha from the previous stage (rendering + masking) is used to determine how much paint to apply**
- **Separate paint objects for stroking, filling**
- **Paint is transformed by an affine transform**
- **Four types of paint are supported:**
  - Flat color paint
  - Linear gradient paint: points (x1, y1) and (x2, y2), color ramp
  - Radial gradient paint: center (x, y), focus (x, y), radius, color ramp
  - Pattern paint based on an image, tiling mode

Blending

- **Combine masked alpha from path with paint alpha**
- **Blend the result onto the drawing surface**
- **Blending is a function of:**
  - The paint (R, G, B) color
  - The masked alpha value (path alpha ´ mask alpha ´ paint alpha)
  - The destination (R, G, B) color
  - The destination alpha value (1 if no stored alpha)
- **There are 8 blending functions:**
  - Porter-Duff "source" mode (copy source to destination)
  - Porter-Duff "source **over** destination"/ "destination **over** source"
  - Porter-Duff "source **in** destination"/ "destination **in** source"
  - Lighten (choose lighter of source and destination)
  - Darken (choose darker of source and destination)
  - Multiply (black source pixel forces black, white leaves unchanged)
  - "Screen"(white source pixel forces white, black leaves unchanged)
  - Additive (add pixel values, add alpha up to 1)

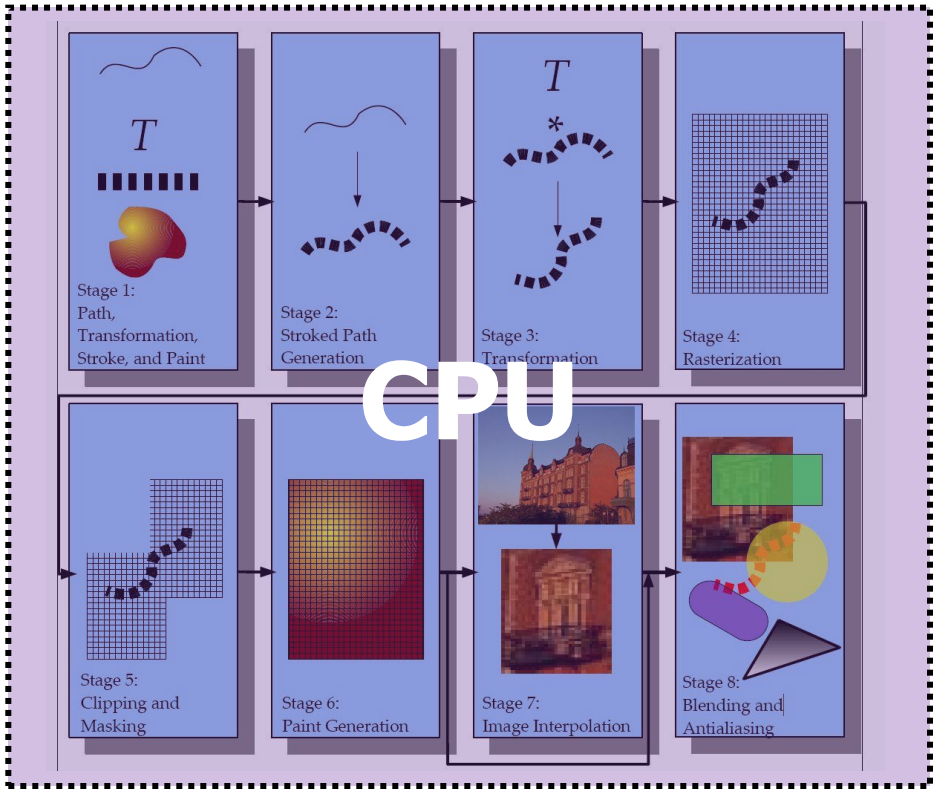
OpenVG with Native Graphics Processor

- CPU sending data and commands to OpenVG hardware
- OpenVG rendering pipeline is in the hardware



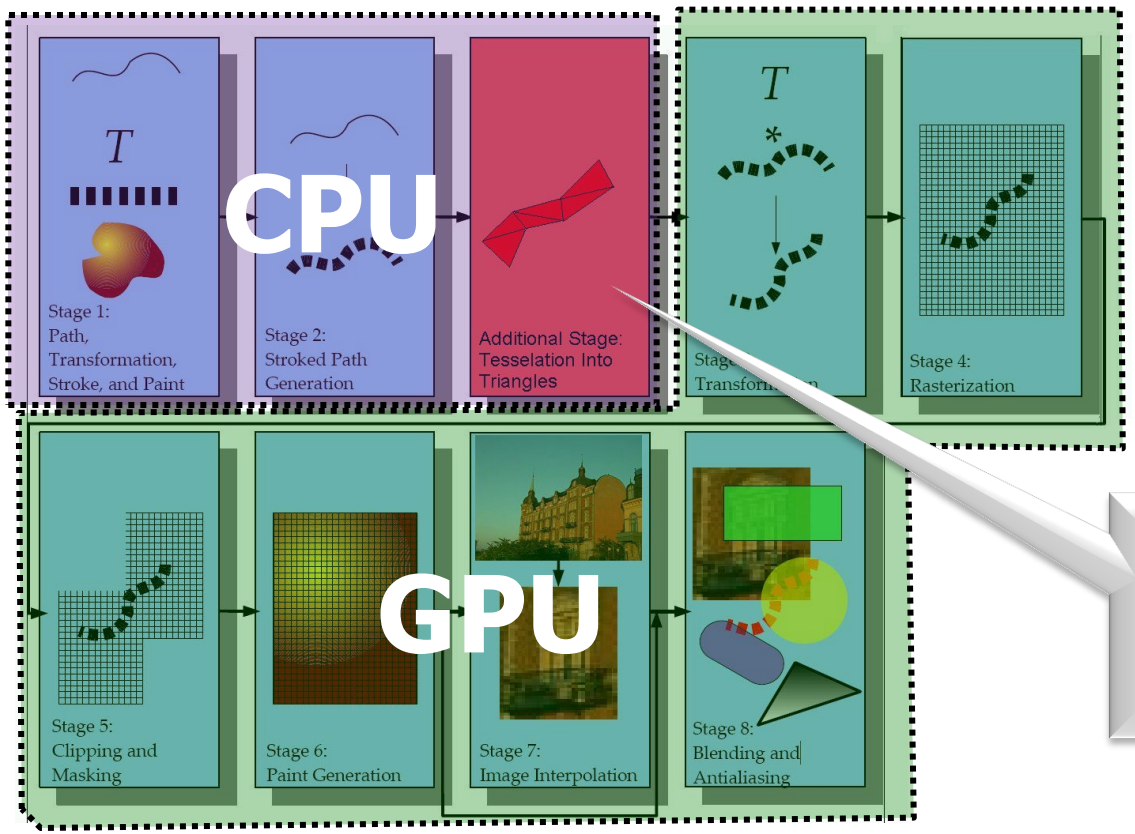
Software Implementation

- **Many implementation alternatives**
- **Everything is processed on the CPU (or +FPU)**
  - Power consumption is often the CPU maximum power



OpenVG with a 3D GPU (tessellation)

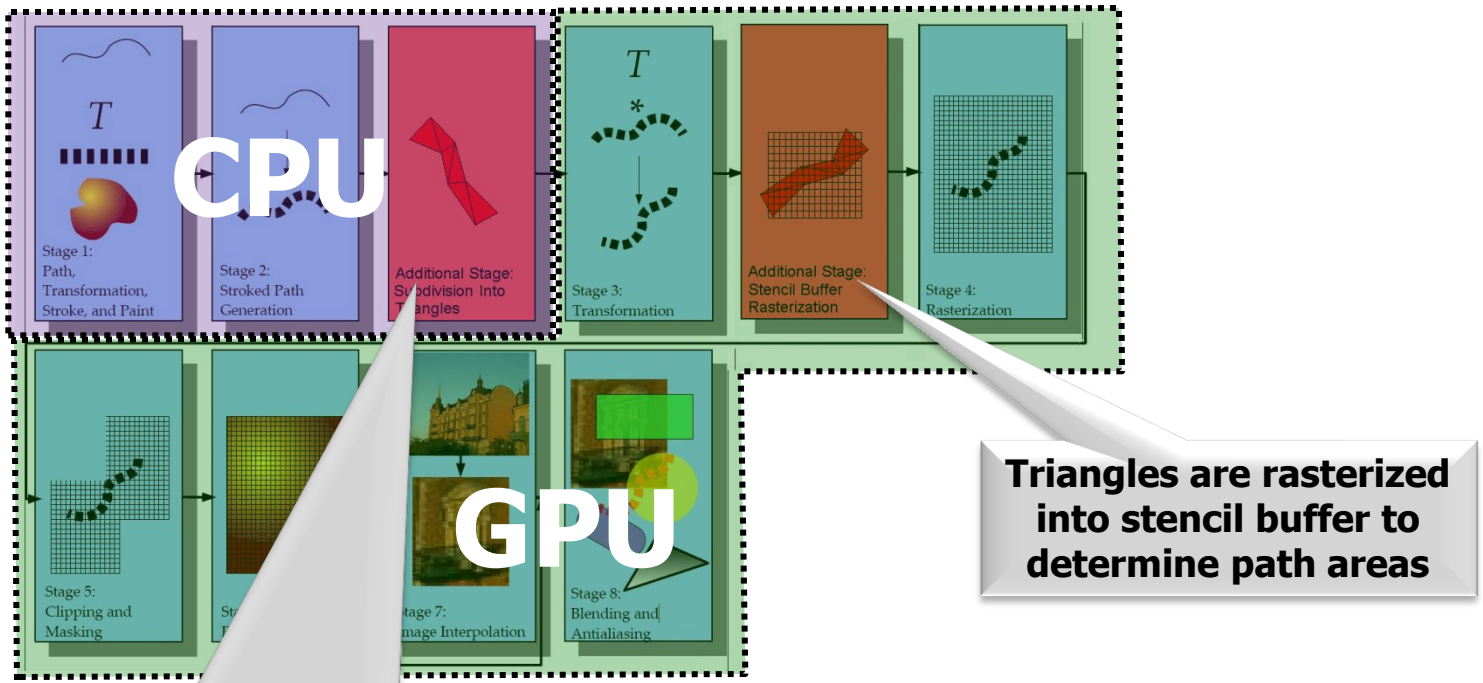
- 3D hardware is not tailored for tessellating arbitrary paths to triangles which causes significant pre-processing to the CPU
- Frame-rates can be discontinuous when tessellation computed infrequently



Tessellation-based implementations require complex polygon tessellation in the software driver

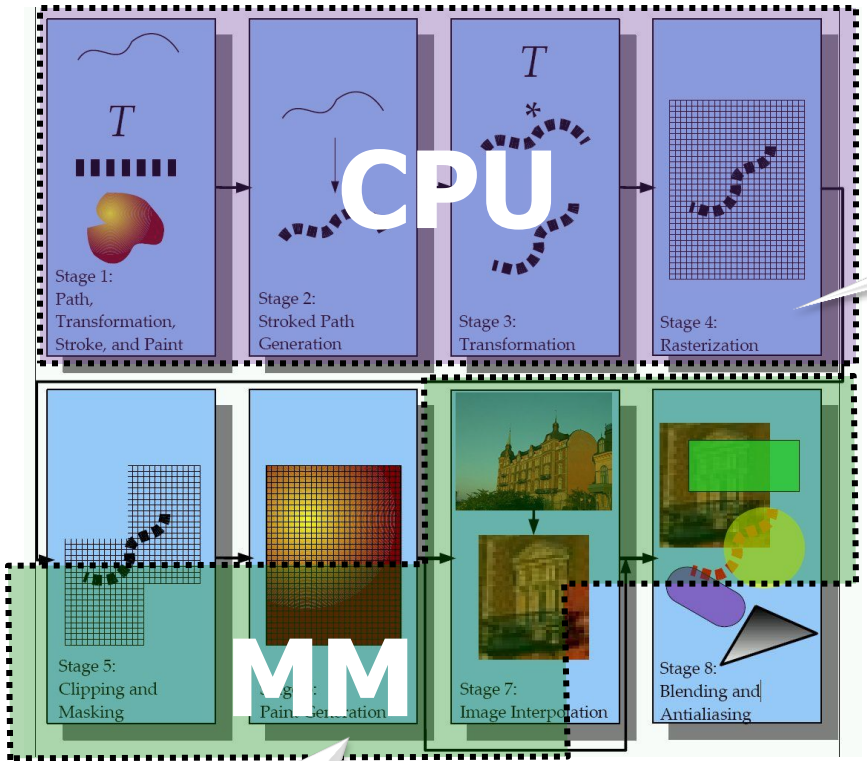
OpenVG with a 3D GPU (stencil)

- Path subdivision is computationally simpler than path tessellation
- Stencil buffer rasterization consumes a lot of memory bandwidth



OpenVG with Multimedia Hardware

- Using double buffer, fast image processing, BitBlt



Most of rasterization (Vector Graphics)

Selective hardware acceleration

Implementations

- **Conformant Products**
  - Takumi GV series
  - DMP dandelion VG
  - Vivante GC series
  - ARM Mali series
  - Inst. for Information Industry DeltaVG
  - Imagination PowerVR SGX, MBX with VGP
  - Broadcom – VideoCore III
  - HUONE – AlexVG engine
  - NVIDIA – NVIDIA AP

Implementations

- **Not conformant but based on OpenVG**
  - ShivaVG – Open Source OpenVG on OpenGL
  - Mazatech – AmanithVG (Software and on OpenGL ES)
  - Hooked Wireless – Hooked OpenVG (on OpenGL ES)
  - HUONE – AlexVG forge (on OpenGL ES)

Applications

- **GUI**
  - Mobile Devices GUI – Samsung
    - Bitmap Graphics, Flash lite, OpenVG, Flash lite on OpenVG
  - Qt Rendering on OpenVG
- **Mapping**
  - GPS, map drawing
- **Flash lite player**
  - Flash 10.1 → Accelerating on OpenGL ES 2.0
- **SVG Mobile Player**
  - MMS, DCD, MPEG4-LASeR, WAP, eBook, WebKit
  - Java Bindings - JSR226, JSR271, JSR287
- **Games**

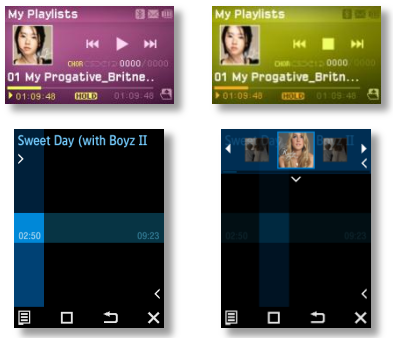
Applications: GUIs

Croix GUI

Magic Touch GUI

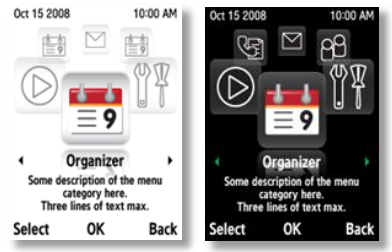
TouchWiz GUI

Multimedia Player



BEAT DJ

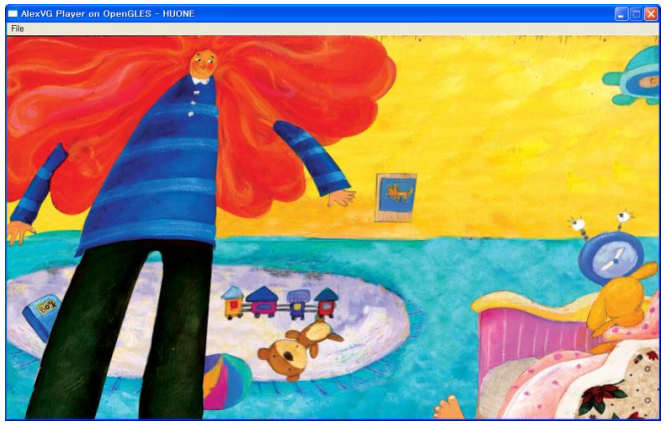
Simplified Main Menu

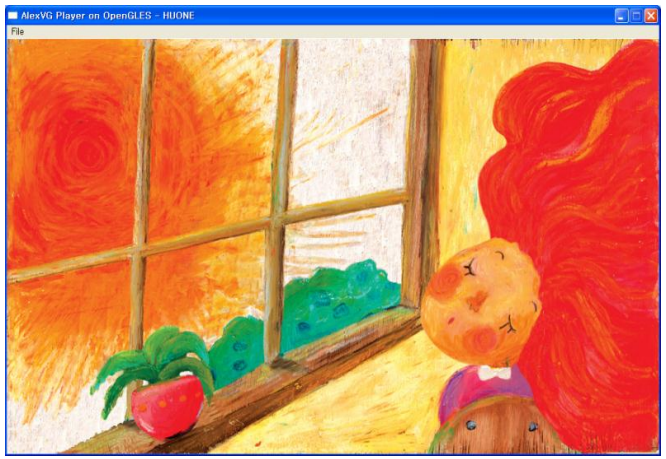


Nokia 500

- OpenVG accelerating Smart Phone
- UI Acceleration

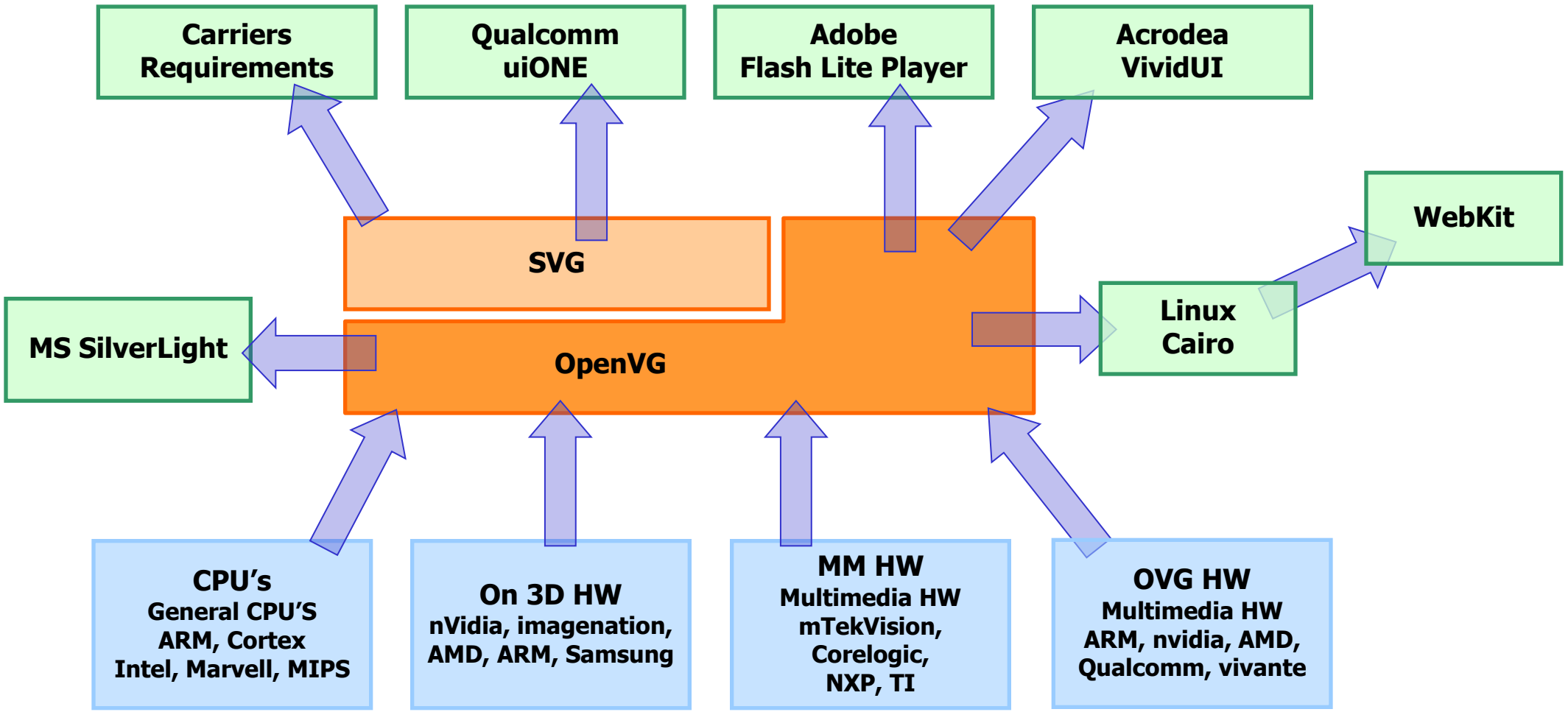
Applications: Web, eBook, Rich Media





Applications: Monitoring, Mapping

Vector Graphics and Other Technologies



OpenVG Roadmap

- **OpenVG Express – Streamlined, refactored**
  - Investigations
    - No Addition of feature
    - No Removal of feature
    - Reduce HW and SW complexity
      by reducing CTS complexity and pass conditional score
      and limited and fixed Color Space
    - Major Target Application – SVGTiny 1.1/1.2 , Flash Lite Player

    - CTS - Making it more effective and more accessible to new vendor

Examples of OpenVG Application

- Demo