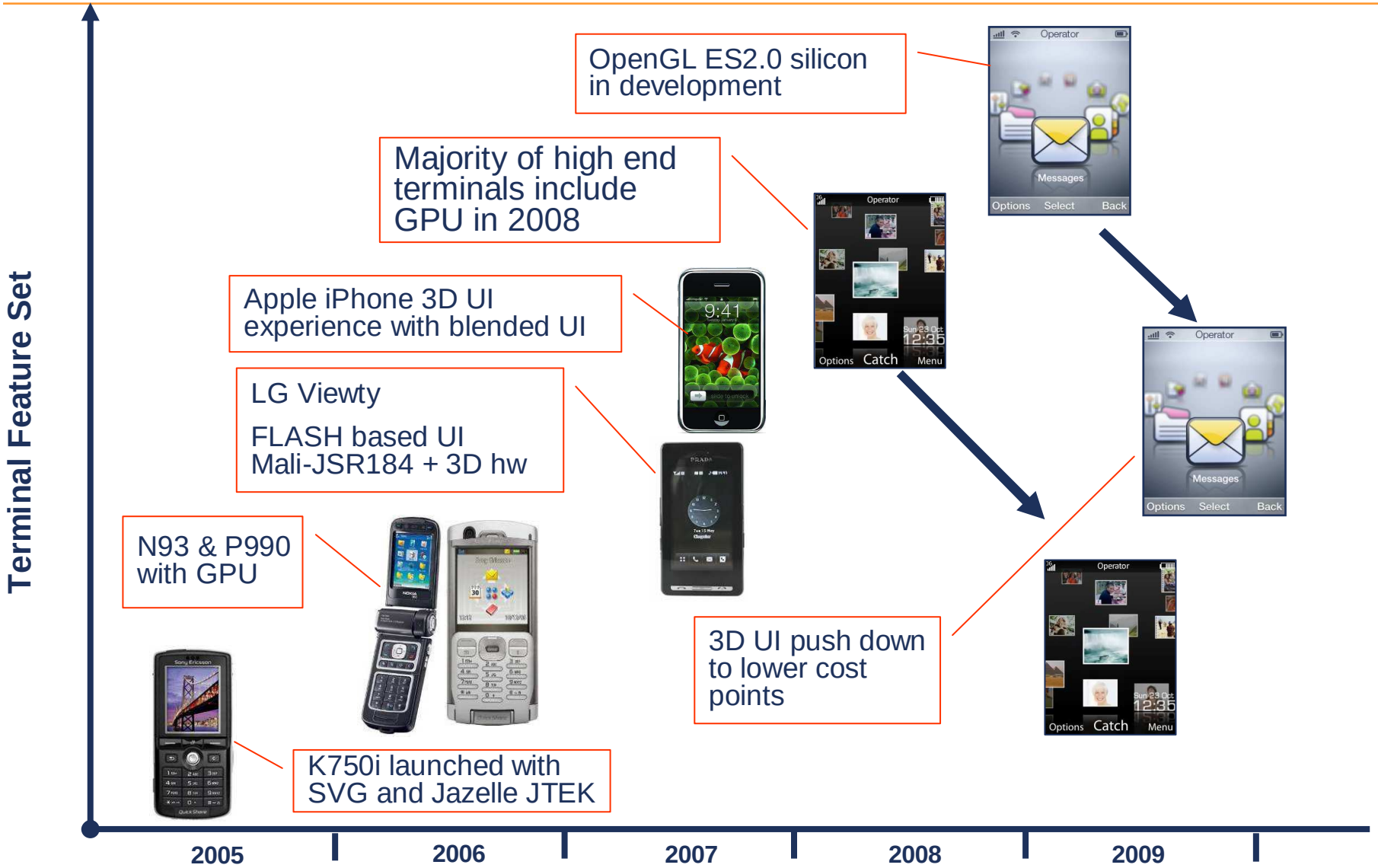
Overcoming The Challenges Of Multimedia System Design

Chris.Porthouse@arm.com

Senior Product Manager

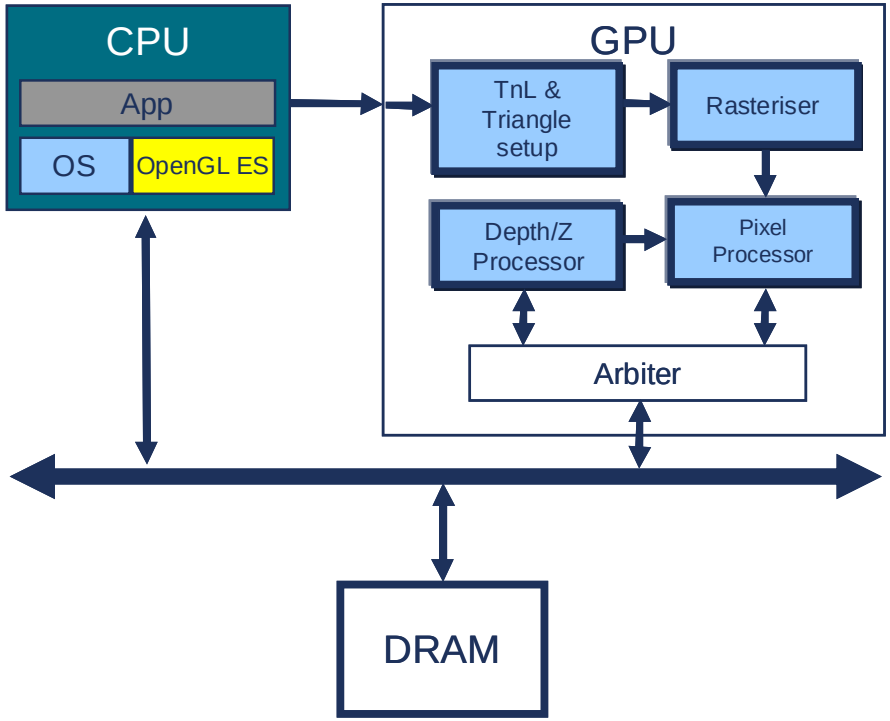
ARM Media Processing Division

Two trajectories for graphics



Legacy Graphics - SoC Configuration

- "Push" style architecture:
  - CPU to GPU via dedicated channel
  - Each rendering step pushed to GPU
    - GPU dependent on CPU for vertex data
    - CPU dependant on memory for app and scene data
    - GPU dependent on memory for texture & frame buffer data
  - Requires all GPU stages at full speed
    - Even if not received data yet



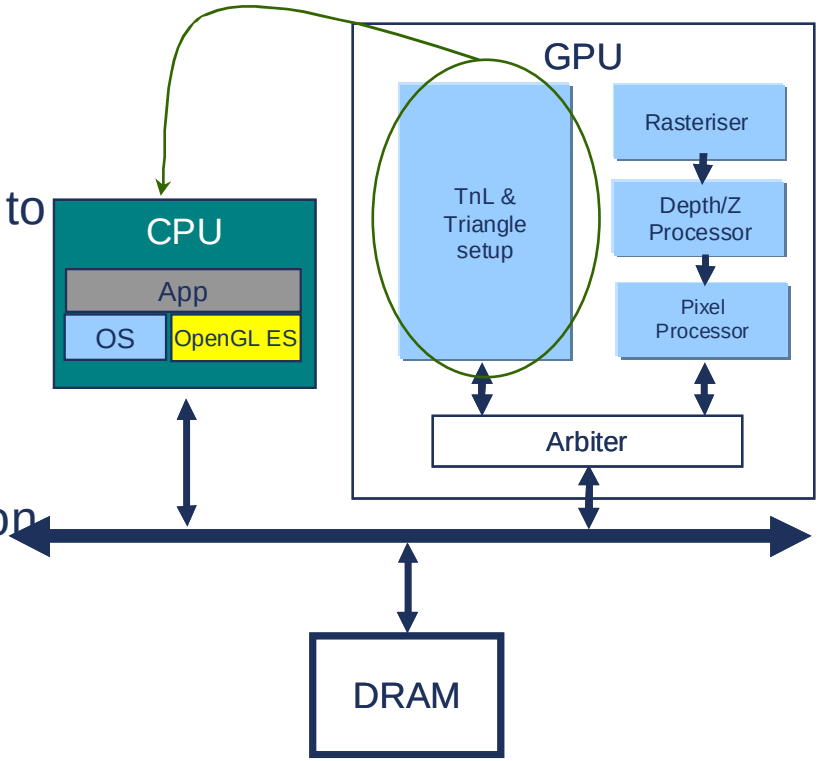
⇨Inefficient processing leads to extra power consumption = lower battery life

⇨Load balancing this architecture is difficult

Improved Graphics - SoC Configuration

- UI requires **limited** geometry processing
  - All about power/performance
- Intermediate processing decoupled
  - Main memory is source of all data
  - While GPU rendering frame, CPU free to generate next frame
  - Decoupling means **no lock ups**
- Power save mode recovery faster
  - Each element in processing chain can free run, idle or shut down depending on load
  - All structures in main memory

CPU de-coupled from GPU…or process the geometry on CPU

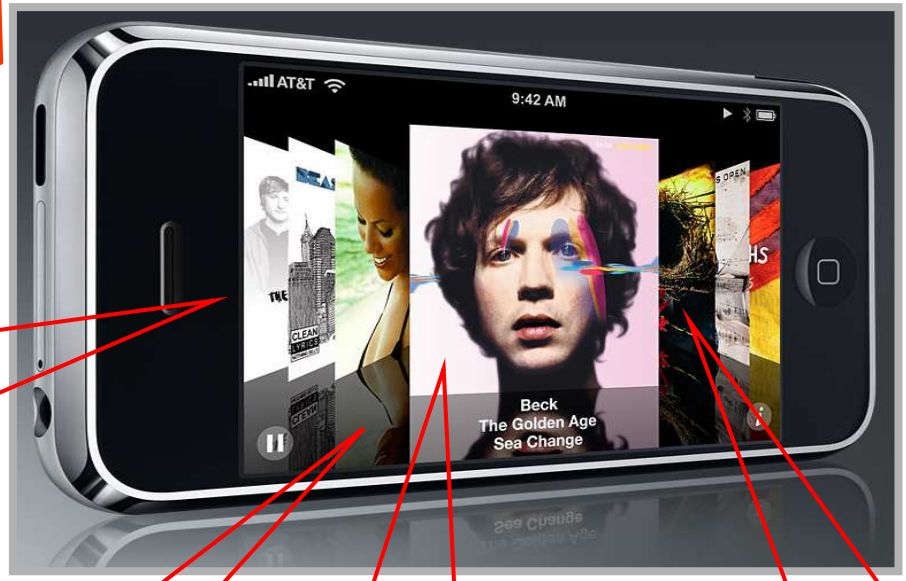


⇨ Benefits – power savings and cost efficient performance

⇨ Scaleable architecture – from low geometry UI to high performance games

New UI's Enabled By Mali55 – in just 1mm²* !

This application has low geometry and high fill rate requirements – ideally suited to the architecture of Mali55



Dynamic User Experience created by:

- Texture Mapping
- Rotation
- Growing and Shrinking
- Flipping

Reflections effects

FSAA enables high quality smooth edge

Tri-linear filtering enables HQ images

Can It Be Done In Software ?

Yes, if you have more than 900MHz free

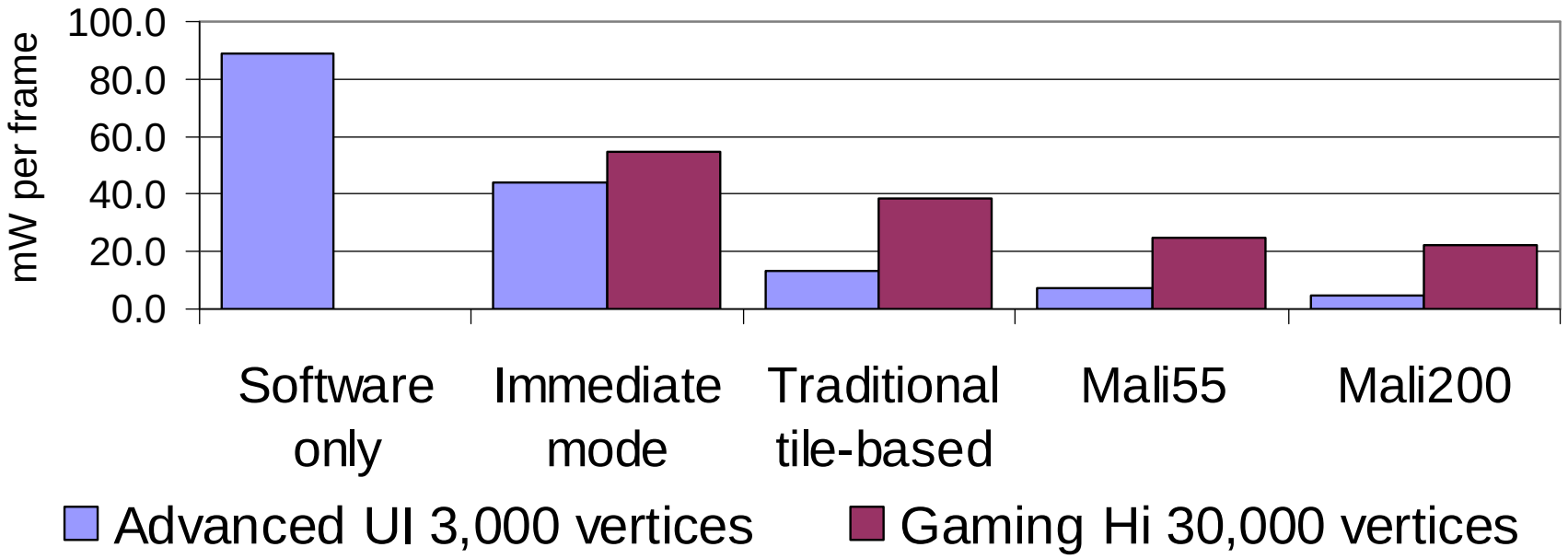
120MHz to transform and animate Album covers

300MHz for reflections effects with FAA

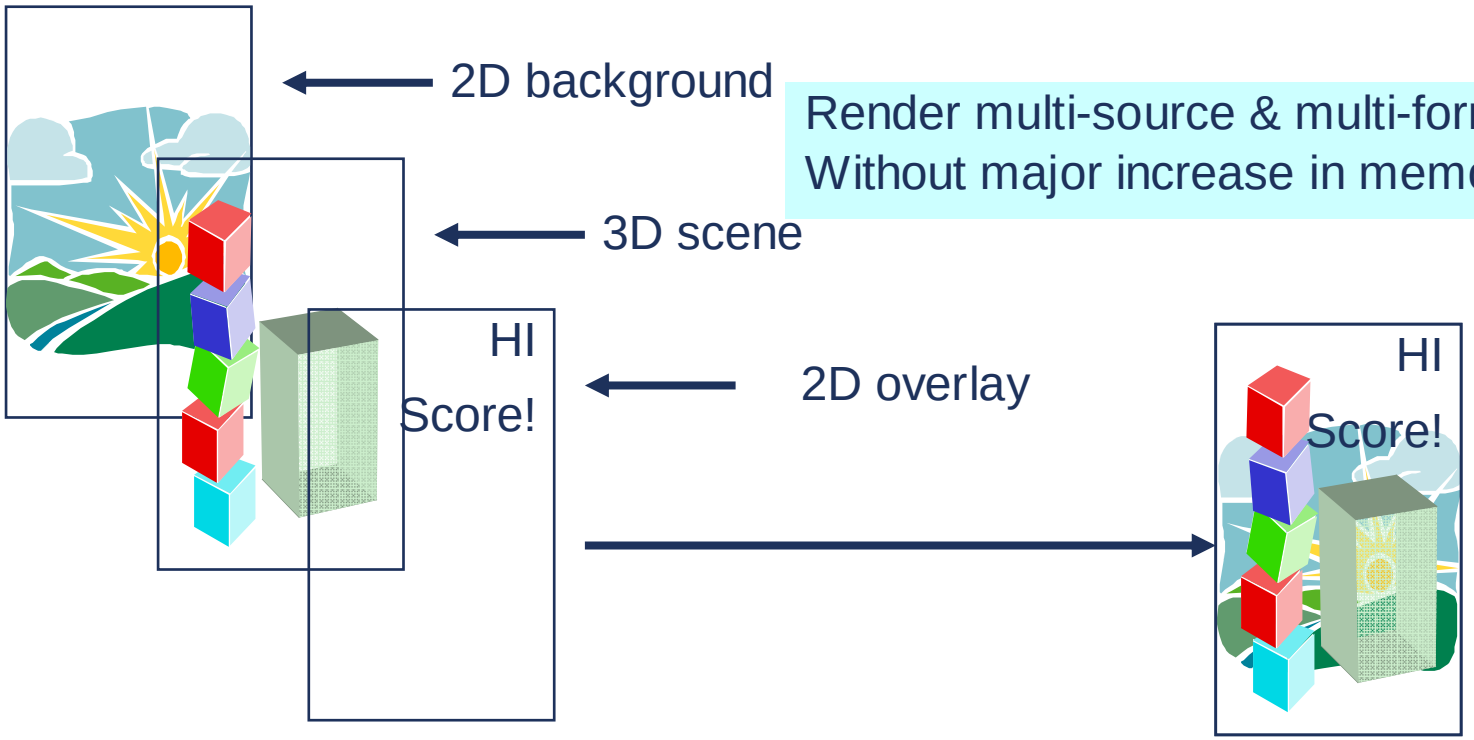
500MHz for Tri-linear filtering with FAA

Power-efficient Graphics

- Memory bandwidth is significant use of power
  - Large proportion is off-chip at 10x the power
- Mali™ architecture significantly reduces memory bandwidth
  - Combines the best of immediate-mode flow and tile-based rendering
- Significant savings for both low and high complexity scenes



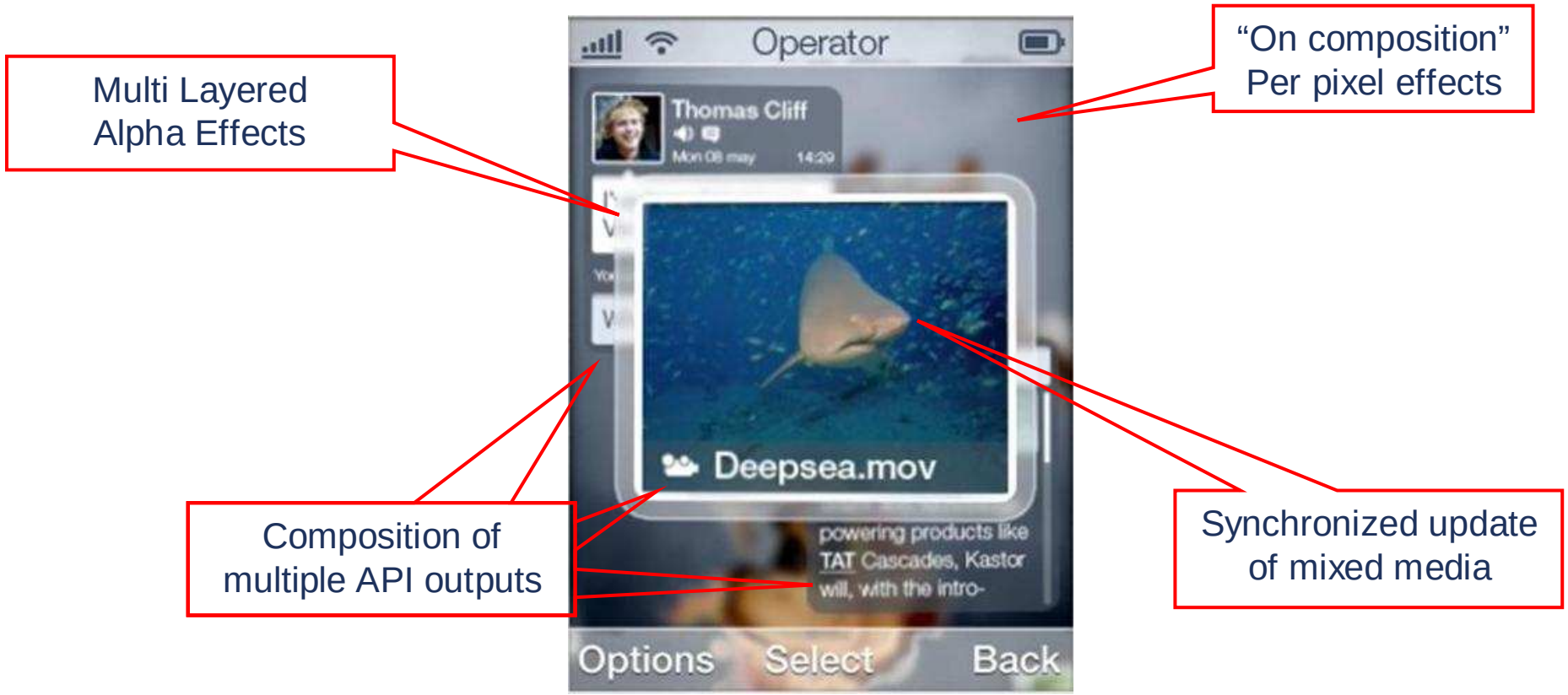
Software challenges: Composition in Java



Render multi-source & multi-format displays
Without major increase in memory & power

- Java allows 2D and 3D to be combined in an image
  - Historically all done in software
  - A lot of GL-ES hardware not designed with Java in mind
- Can be worked around in software…but
  - Reduced performance, excessive copying of screen buffers

Composition complexity increasing



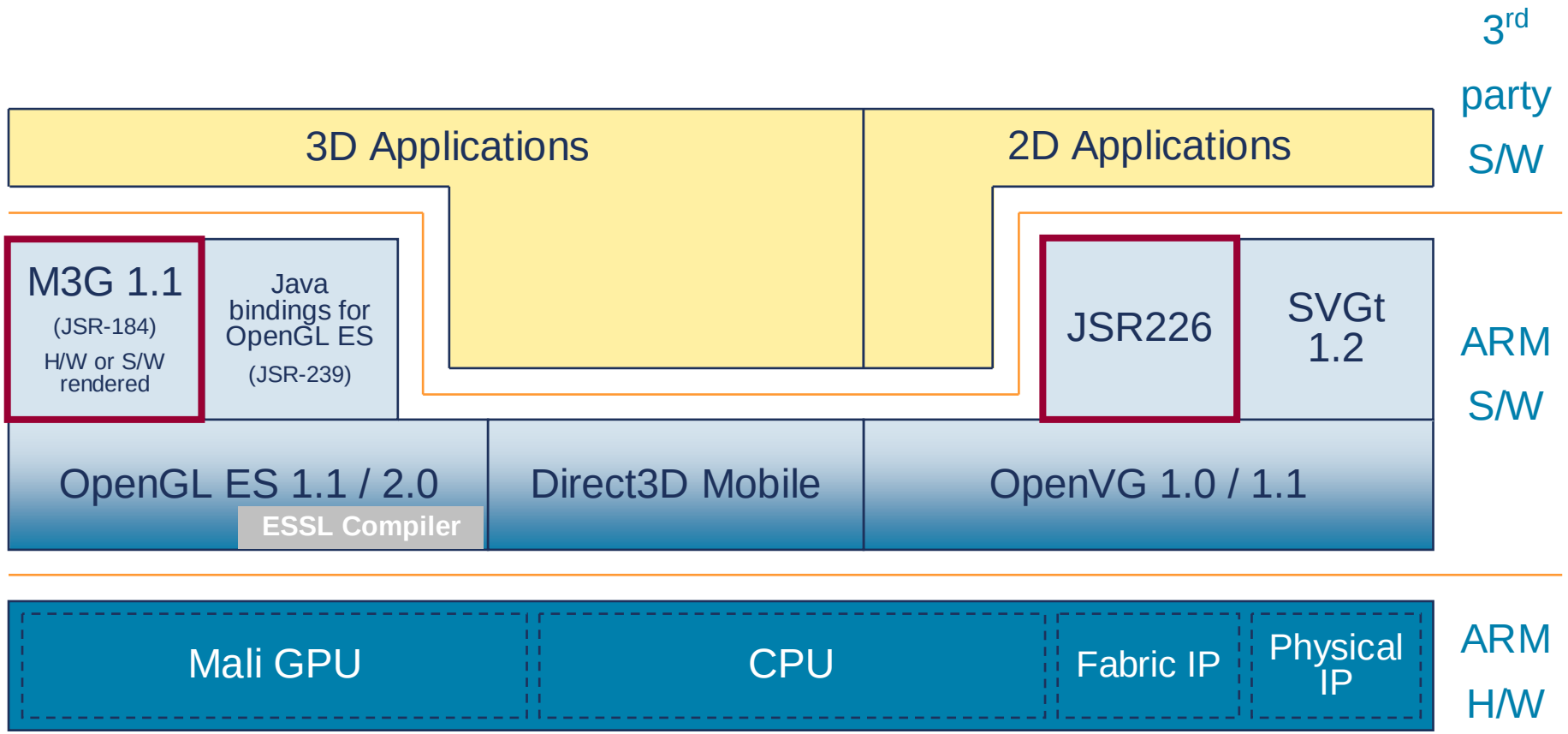
- Becoming too complex to handle with software workarounds…
  - JSR287 allows video as part of 2D vector graphics
  - JSR297 allows dynamic textures (e.g. streaming video) as part of 3D scene

Power and Software Design

- Some components work together better by designing them together
  - For example: avoiding data copying saves energy
- The more parts of the puzzle you are in control of, the easier this is:
  - Content, Java VM, M3G2, OpenGL ES 2.0 drivers...
  - It ***is*** possible to optimise the flow ***and*** remain standards-compliant
- We believe you want a pre-verified, integrated solution

ARM Mali HW & SW Form Full Graphics Stack

- Proven, high quality s/w pre-integrated with Mali - fast time to market
- Code compatibility across h/w-accelerated and s/w-only devices
- 9 software licensees and counting



Cost

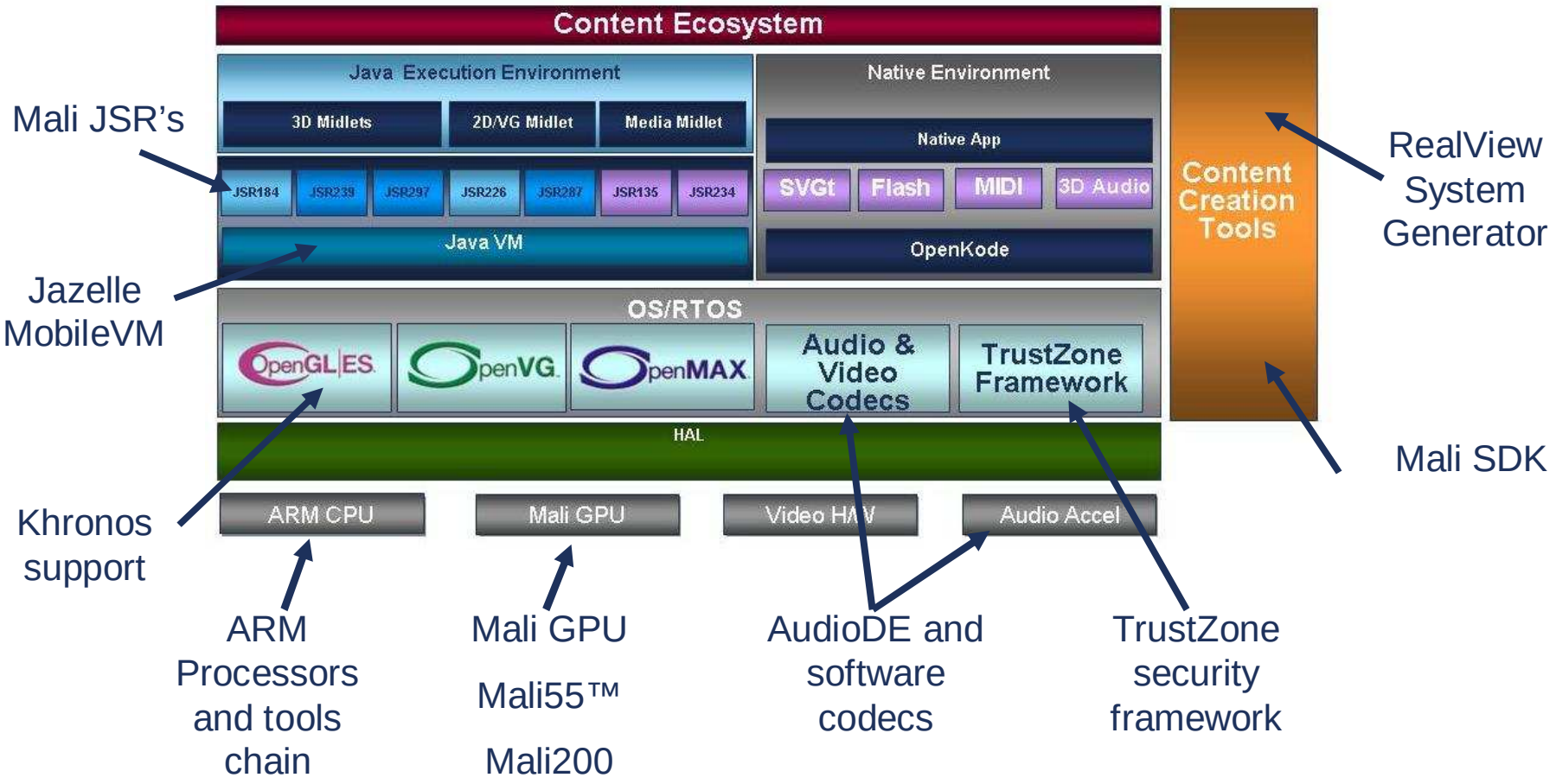
- Meeting demand for user experience while keeping cost of devices for mass market

- What are the cost drivers?
  - IP
  - Silicon
  - Validation
  - Lost market window
  - Etc.
- Obtaining more pre-verified, pre-integrated IP from one supplier will accelerate development
- Will that reduce costs/increase profits overall?

Cost vs. User Experience

- Inevitably, the user exerience has to be tailored to fit the market requirements:
  - Software-rendered graphics
  - CPU-rendered low-resolution video
  - Mali55 OpenGL ES/OpenVG-accelerated user interface
  - Mali200/Sif OpenGL ES 2.0 hardware
  - 1080p video, H.264, VC-1 ...
- What you want are unified stacks
- What you don’t want is to redesign everything between differing platforms

Complete Graphics Stack with ARM Mali h/w & s/w



Proven, high quality software pre-integrated with Mali enables fast time to market

Graphics code compatibility across hardware accelerated and software-only platforms

Summary

- There are no magic bullets
- Sound engineering will still have great value
- Building good systems will still have great value
- Solving “the problem” requires work in a number of disciplines
- Life for suppliers of individual components gets harder
- The world still needs great software, great hardware and great tools

Thank You


mali
ARM® Mali™ Graphics
User Experience Beyond Imagination