Arm Fast Model Technology for Full System Verification with GPU Supported

OSDT 2017

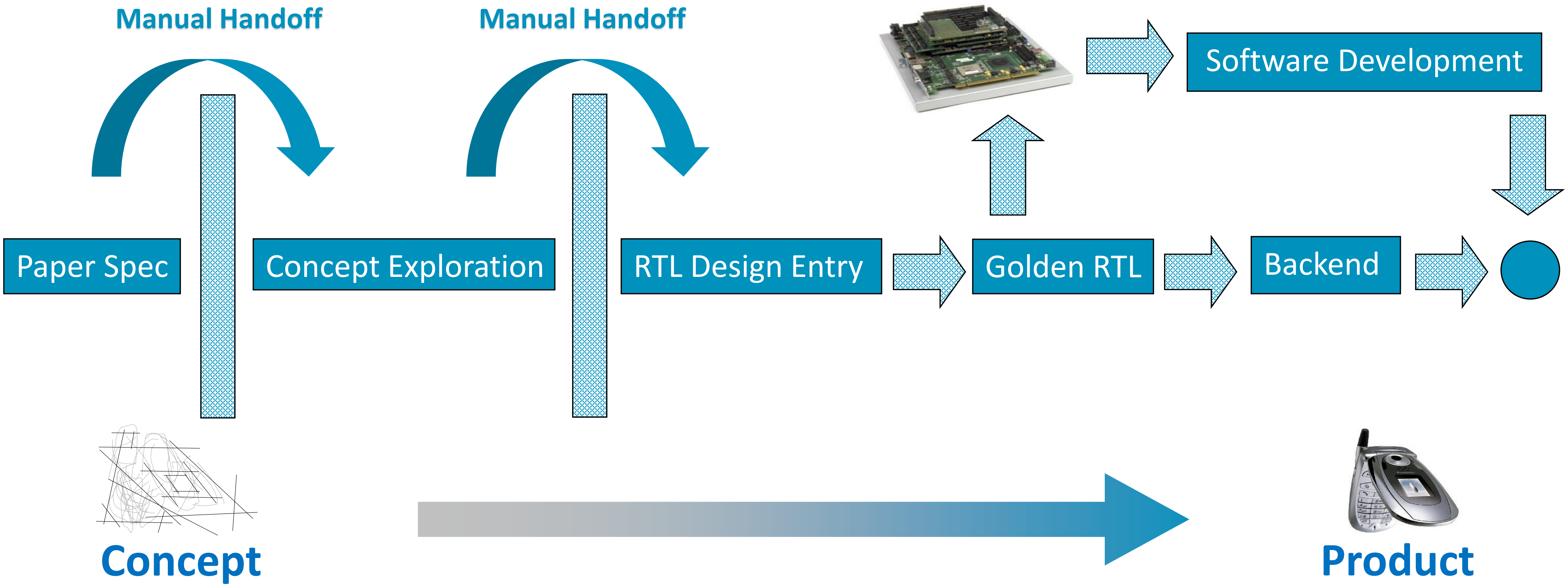
October 23, 2017
Jiangning Liu
Adam McNeeney, Patrick Zhang

Agenda

- Fast Model Introduction
- Graphics Fast Model
- Core Technologies
- Performance Data
- ARMv8 Virtual Platform
- Conclusion

Fast Model Introduction

The Traditional SoC Work-Flow



Manual Handoff

Manual Handoff

Software Development

Paper Spec

Concept Exploration

RTL Design Entry

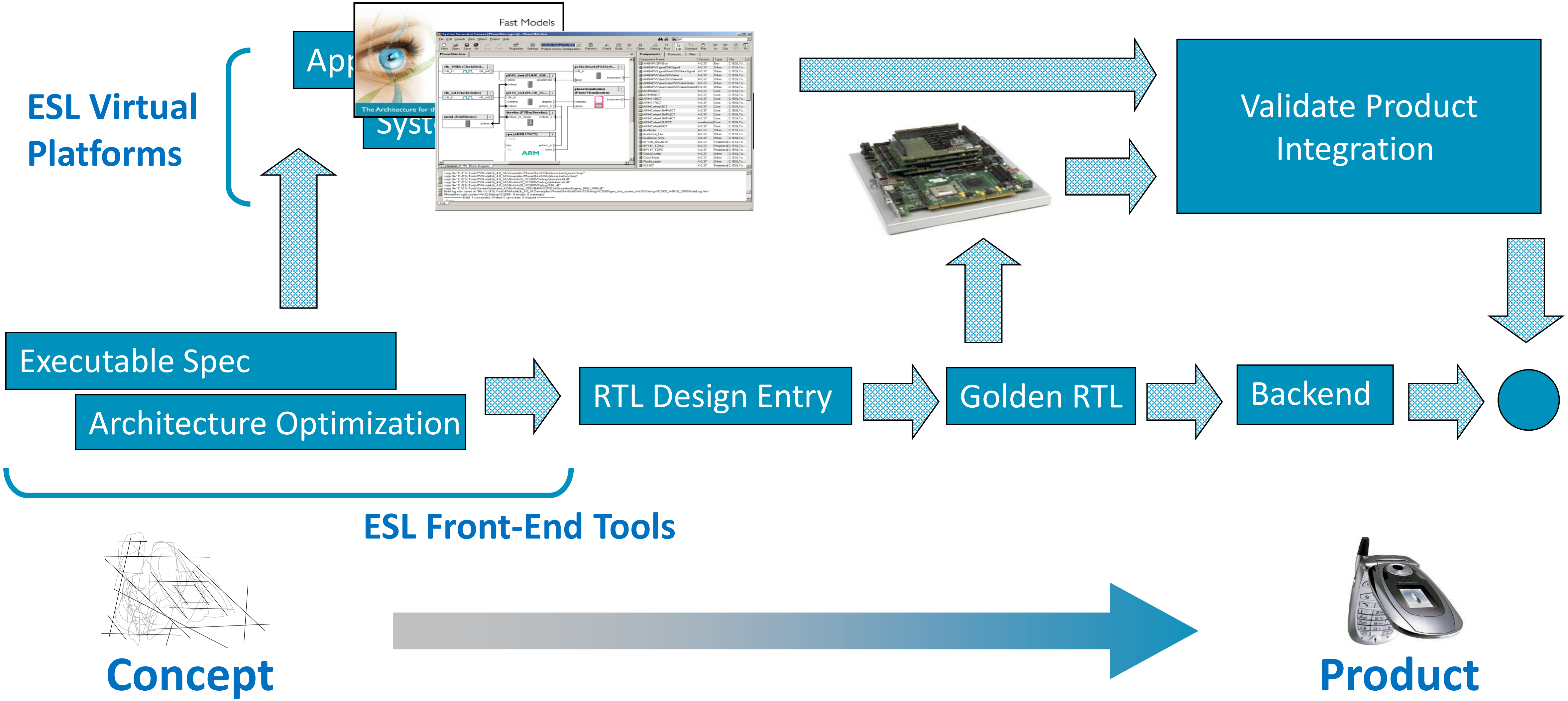
Golden RTL

Backend

Concept

Product

Accelerated with an ESL front-end



ESL Virtual Platforms

App

Syst

Validate Product Integration

Executable Spec

Architecture Optimization

RTL Design Entry

Golden RTL

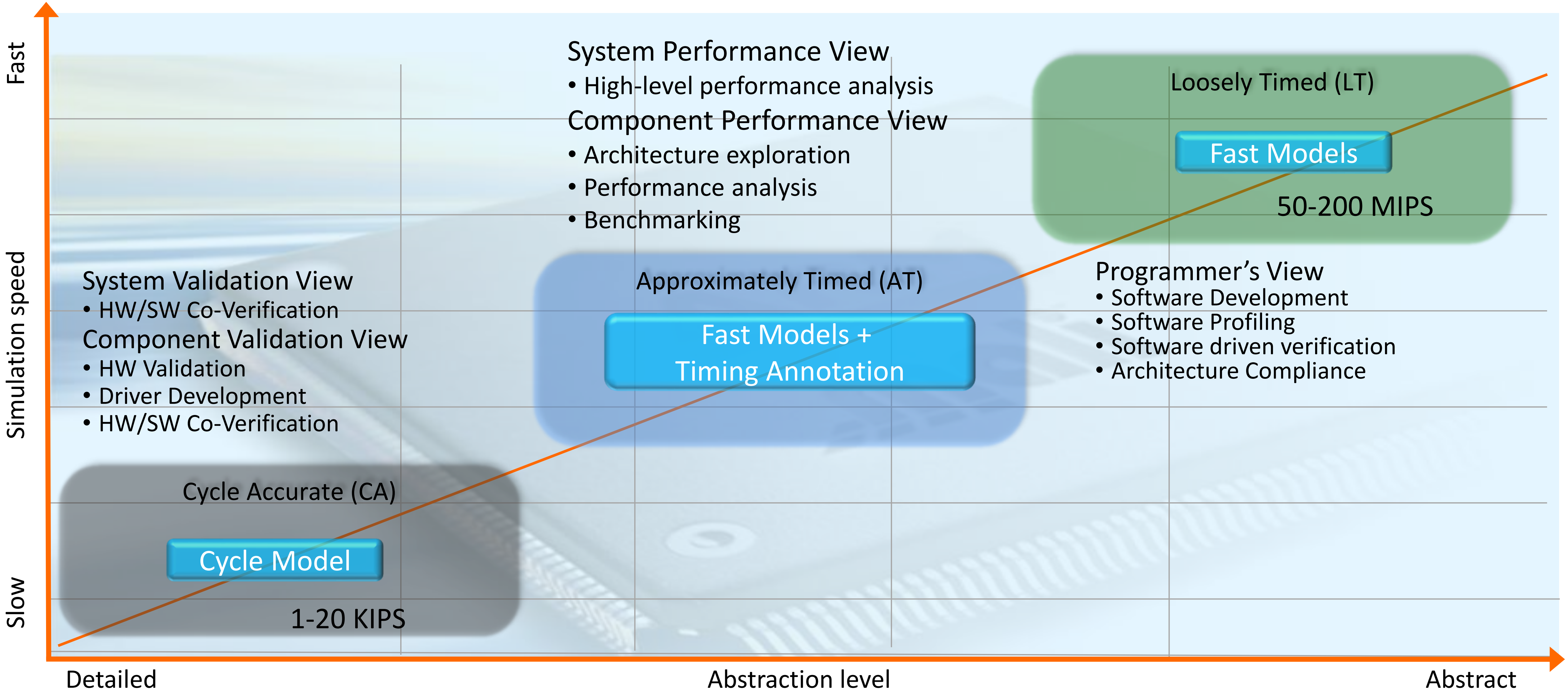
Backend

ESL Front-End Tools

Concept

Product

SoC Simulation Views



Fast

Simulation speed

Slow

Detailed

Abstraction level

Abstract

System Validation View
- HW/SW Co-Verification

Component Validation View
- HW Validation
- Driver Development
- HW/SW Co-Verification

Cycle Accurate (CA)

Cycle Model

1-20 KIPS

System Performance View
- High-level performance analysis

Component Performance View
- Architecture exploration
- Performance analysis
- Benchmarking

Approximately Timed (AT)

Fast Models +
Timing Annotation

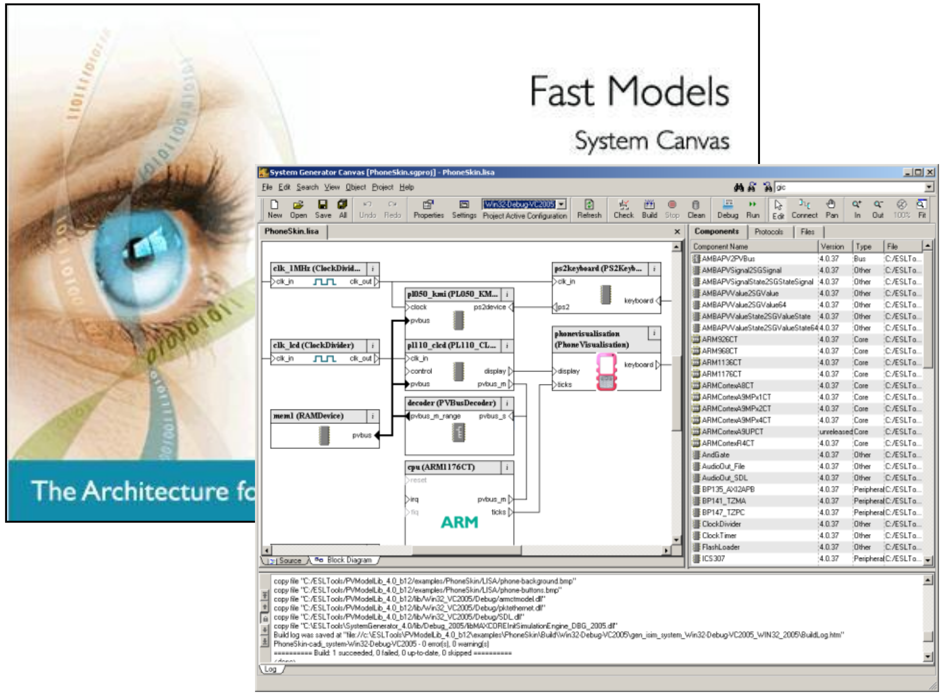
Loosely Timed (LT)

Fast Models

50-200 MIPS

Programmer’s View
- Software Development
- Software Profiling
- Software driven verification
- Architecture Compliance

Fast Models from Arm



Project Active Configuration: Win32-Debug-VC2005

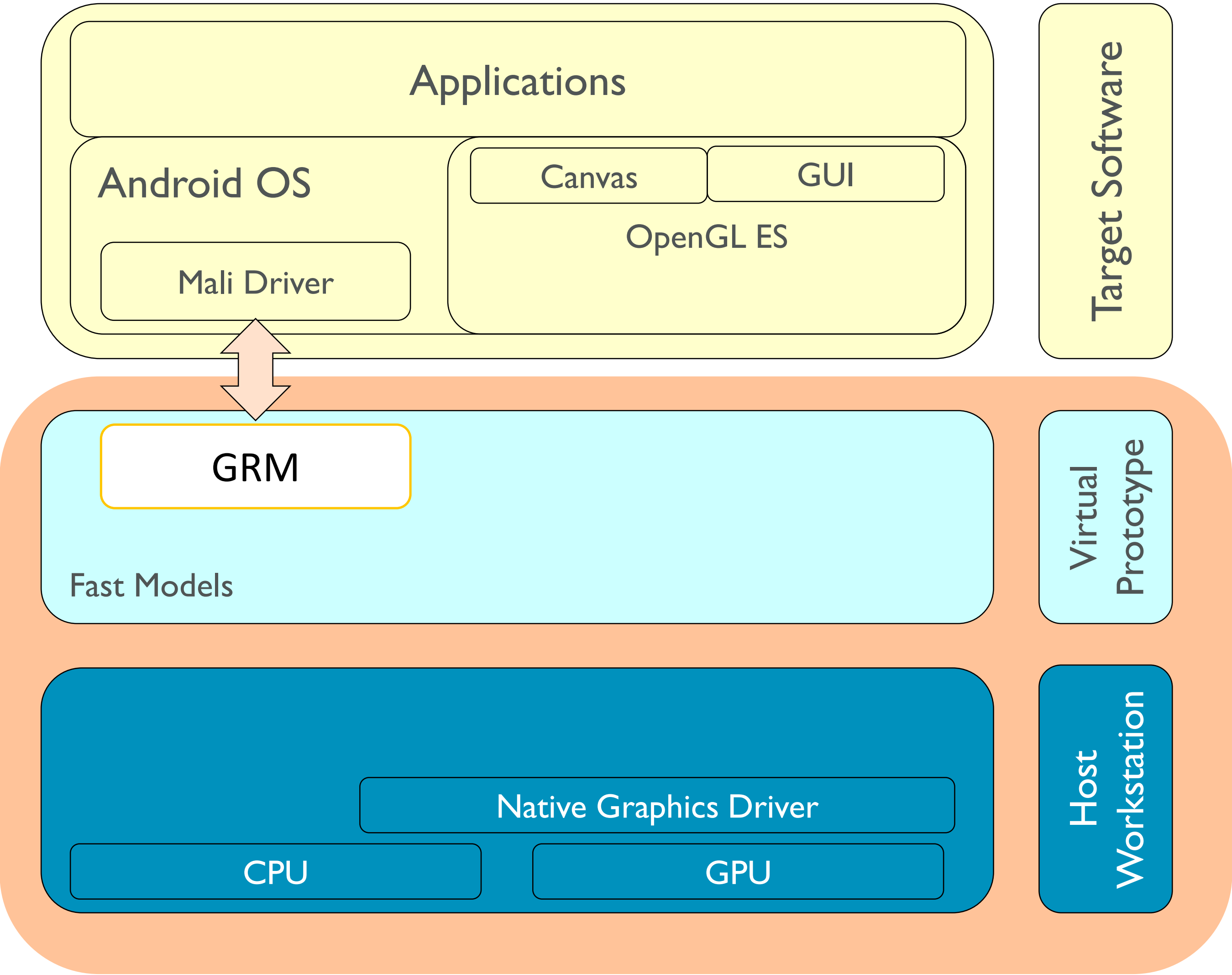
Component Name	Version	Type	File
AMBAPV2PVBus	4.0.37	Bus	C:/ESLTo...
AMBAPVSignal2SGSignal	4.0.37	Other	C:/ESLTo...
AMBAPVSignalState2SGStateSignal	4.0.37	Other	C:/ESLTo...
AMBAPVValue2SGValue	4.0.37	Other	C:/ESLTo...
AMBAPVValue2SGValue64	4.0.37	Other	C:/ESLTo...
AMBAPVValueState2SGValueState	4.0.37	Other	C:/ESLTo...
AMBAPVValueState2SGValueState64	4.0.37	Other	C:/ESLTo...
ARM926CT	4.0.37	Core	C:/ESLTo...
ARM968CT	4.0.37	Core	C:/ESLTo...
ARM1136CT	4.0.37	Core	C:/ESLTo...
ARM1176CT	4.0.37	Core	C:/ESLTo...
ARMCortexA8CT	4.0.37	Core	C:/ESLTo...
ARMCortexA9MPx1CT	4.0.37	Core	C:/ESLTo...
ARMCortexA9MPx2CT	4.0.37	Core	C:/ESLTo...
ARMCortexA9MPx4CT	4.0.37	Core	C:/ESLTo...
ARMCortexA9UPCT	unreleased	Core	C:/ESLTo...
ARMCortexR4CT	4.0.37	Core	C:/ESLTo...
AndGate	4.0.37	Other	C:/ESLTo...
AudioOut_File	4.0.37	Other	C:/ESLTo...
AudioOut_SDL	4.0.37	Other	C:/ESLTo...
BP135_AXI2APB	4.0.37	Peripheral	C:/ESLTo...
BP141_TZMA	4.0.37	Peripheral	C:/ESLTo...
BP147_TZPC	4.0.37	Peripheral	C:/ESLTo...
ClockDivider	4.0.37	Other	C:/ESLTo...
ClockTimer	4.0.37	Other	C:/ESLTo...
FlashLoader	4.0.37	Other	C:/ESLTo...
ICS307	4.0.37	Peripheral	C:/ESLTo...

```
copy file "C:/ESLTools/PVModelLib_4.0_b12/examples/PhoneSkin/LISA/phone-background.bmp"
copy file "C:/ESLTools/PVModelLib_4.0_b12/examples/PhoneSkin/LISA/phone-buttons.bmp"
copy file "C:/ESLTools/PVModelLib_4.0_b12/lib/Win32_VC2005/Debug/armctmodel.dll"
copy file "C:/ESLTools/PVModelLib_4.0_b12/lib/Win32_VC2005/Debug/pktethernet.dll"
copy file "C:/ESLTools/PVModelLib_4.0_b12/lib/Win32_VC2005/Debug/SDL.dll"
copy file "C:\ESLTools\SystemGenerator_4.0/lib/Debug_2005/libMAXCOREInitSimulationEngine_DBG_2005.dll"
Build log was saved at "file://c:\ESLTools\PVModelLib_4.0_b12\examples\PhoneSkin\Build\Win32-Debug-VC2005\gen_isim_system_Win32-Debug-VC2005_WIN32_2005\BuildLog.htm"
PhoneSkin-cadi_system-Win32-Debug-VC2005 - 0 error(s), 0 warning(s)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
<done>
```

- **Fast, functional models of Arm IPs**
  - Cores and peripherals
  - Validated and maintained by Arm
  - APM demo 26 hours 11.5 Trillions (> 100 MIPS) instructions executed
- **Accompanying tools**
  - System Canvas – environment for developing custom models and constructing system
  - System Generator – backend tool which handles system generation
- **Compatible with Arm's other development tools**
  - DS-5
  - MDK (Microcontroller Development Kit)
  - Model Debugger
- **Integrates into SystemC**

Graphics Fast Model

Graphics Register Model (GRM)



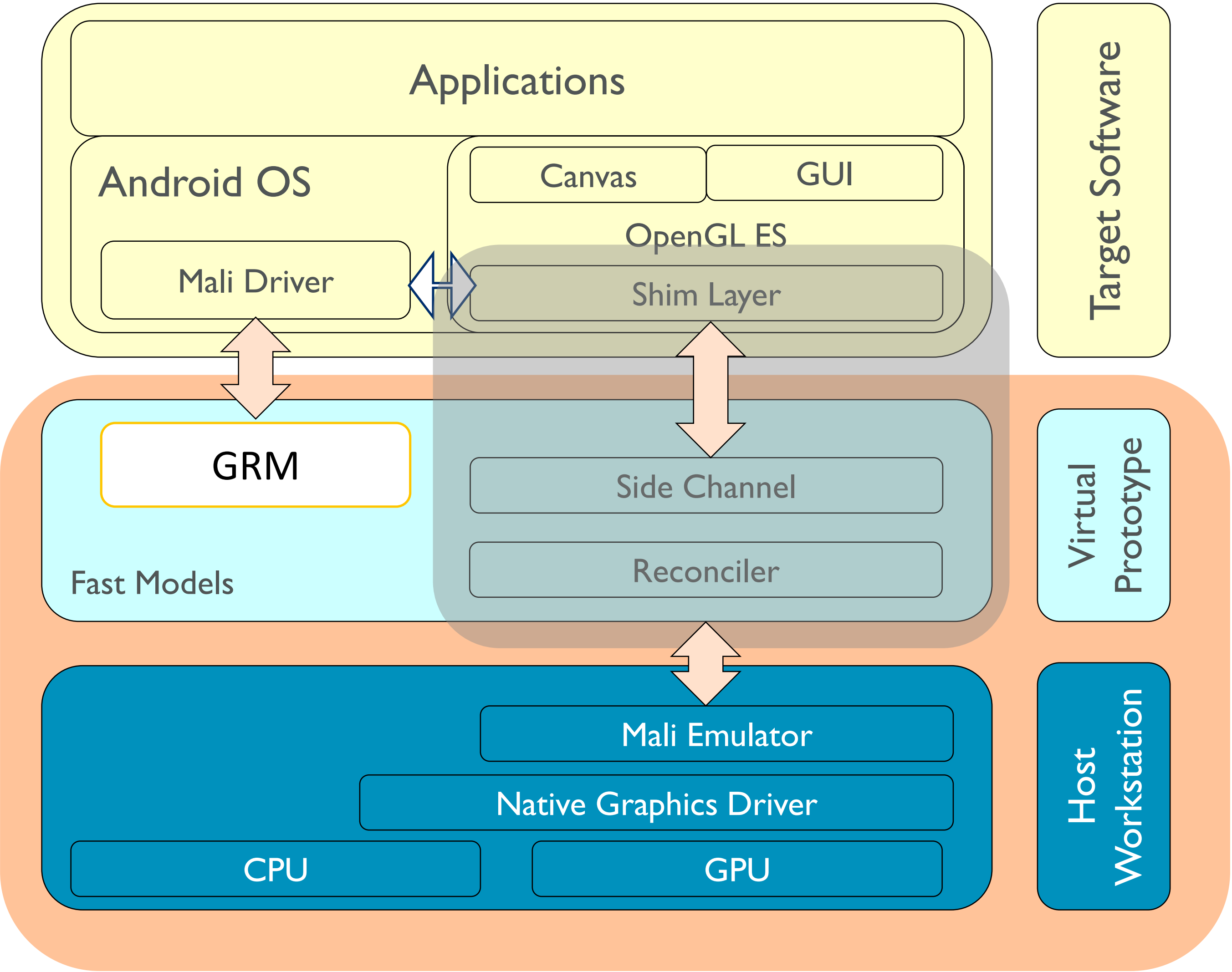
* Graphics Drivers in both kernel and user space can run

GPU Simulation

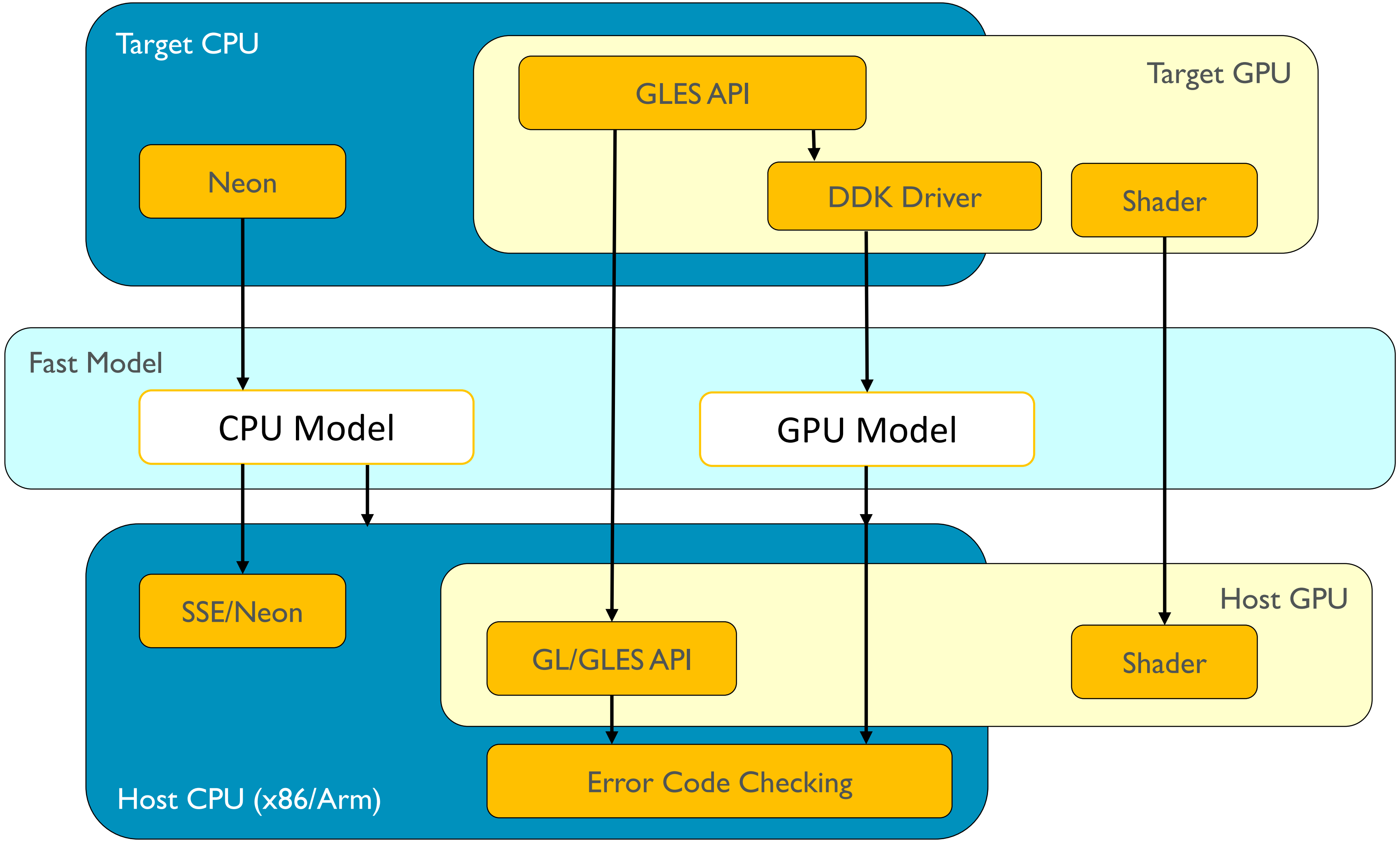
- **Big architecture differences between CPU and GPU, e.g.**
  - GPU has very big number of cores
  - CPU has deeper pipeline and bigger cache size
- **Industry Solutions**
  - Non-GPU model to help CPU centric study. E.g. NoMali model for GEM5
  - Software rendering using CPU through QEMU. E.g. Google Android Emulator
  - Cycle models or C model translated from RTL.
  - Trace replay to help offline GPU analysis
- **Goals of Arm® Fast Model™ Solution**
  - Run commercial graphics benchmarks in subsystem model
  - Verify whole software stack integration as early as possible
  - Zero Android image change at silicon tape-out time

Generic Graphics Accelerator (GGA)

- Leverages host GPU to accelerate OpenGL ES calls in target software
- Side Channel plays a backdoor role of Fast Model
- Mali Emulator can be ignored if graphics APIs can be directly supported by host driver

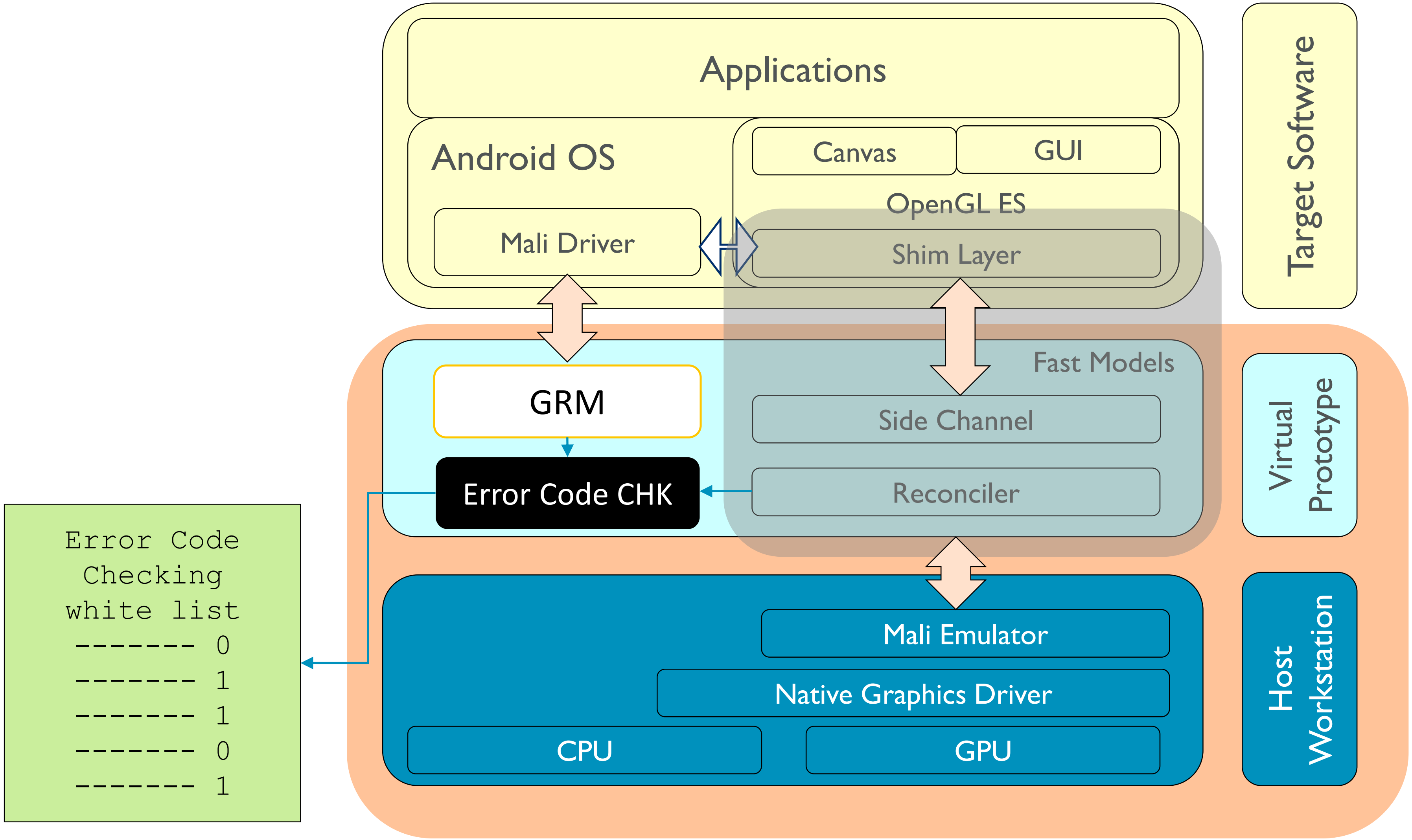


GPU Fast Model Architecture



Graphics Driver Verification

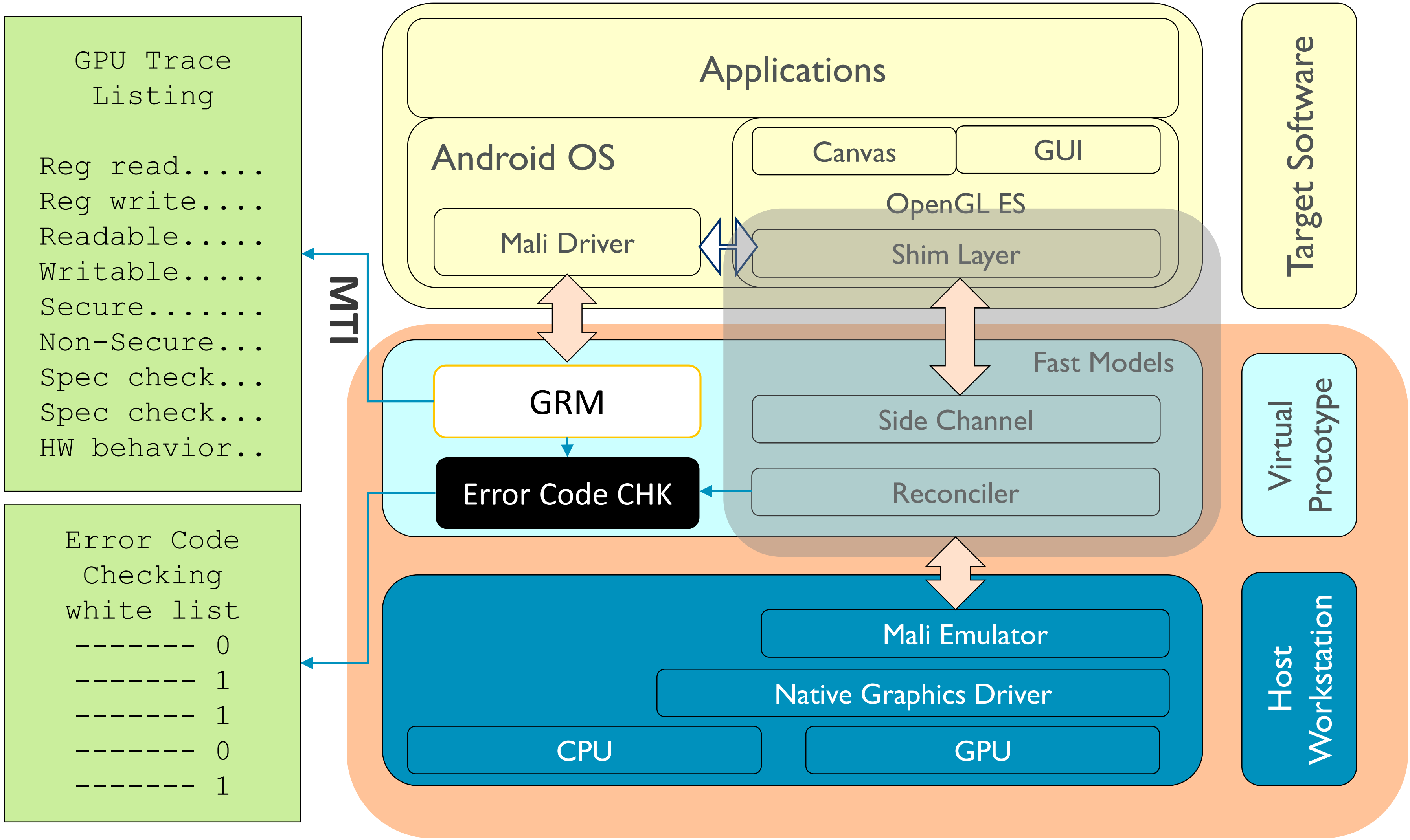
- Error Code Checking



```
Error Code
 Checking
white list
-------- 0
-------- 1
-------- 1
-------- 0
-------- 1
```

Graphics Driver Verification

- Error Code Checking
- GPU Trace Listing
- Register Dump



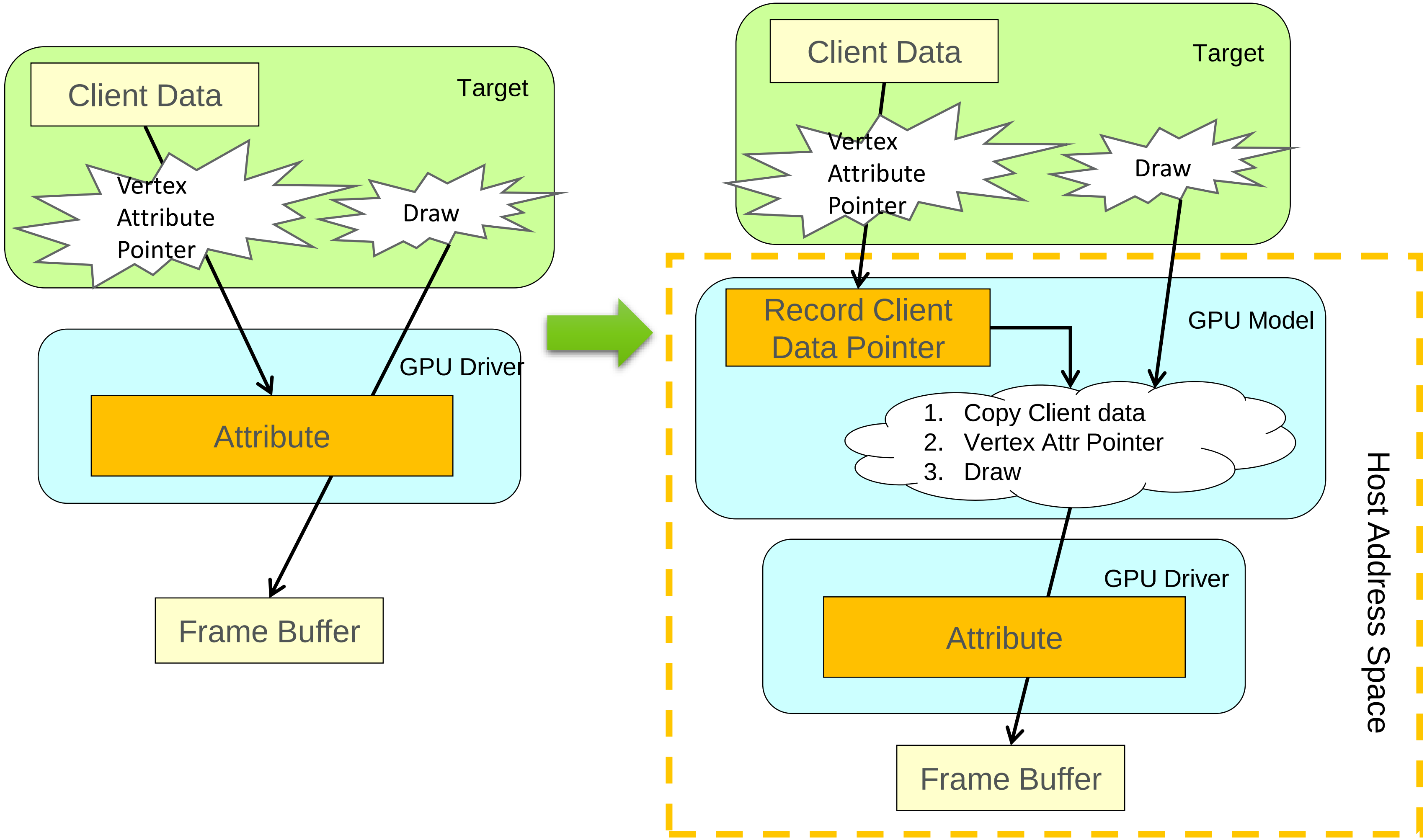
```
GPU Trace
 Listing

Reg read.....
Reg write....
Readable.....
Writable.....
Secure.......
Non-Secure...
Spec check...
Spec check...
HW behavior..
```

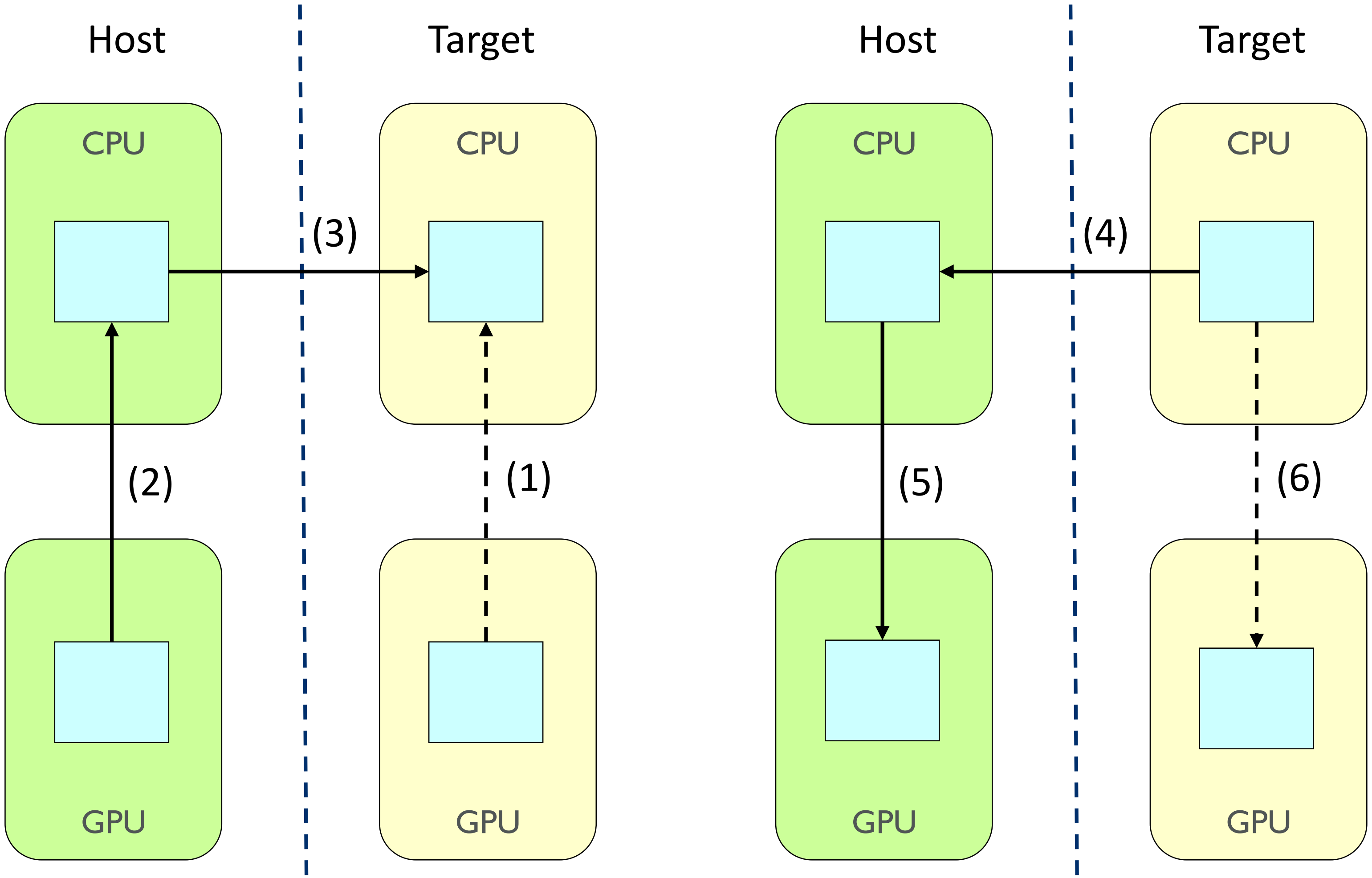
```
Error Code
 Checking
white list
-------- 0
-------- 1
-------- 1
-------- 0
-------- 1
```

Core Technologies

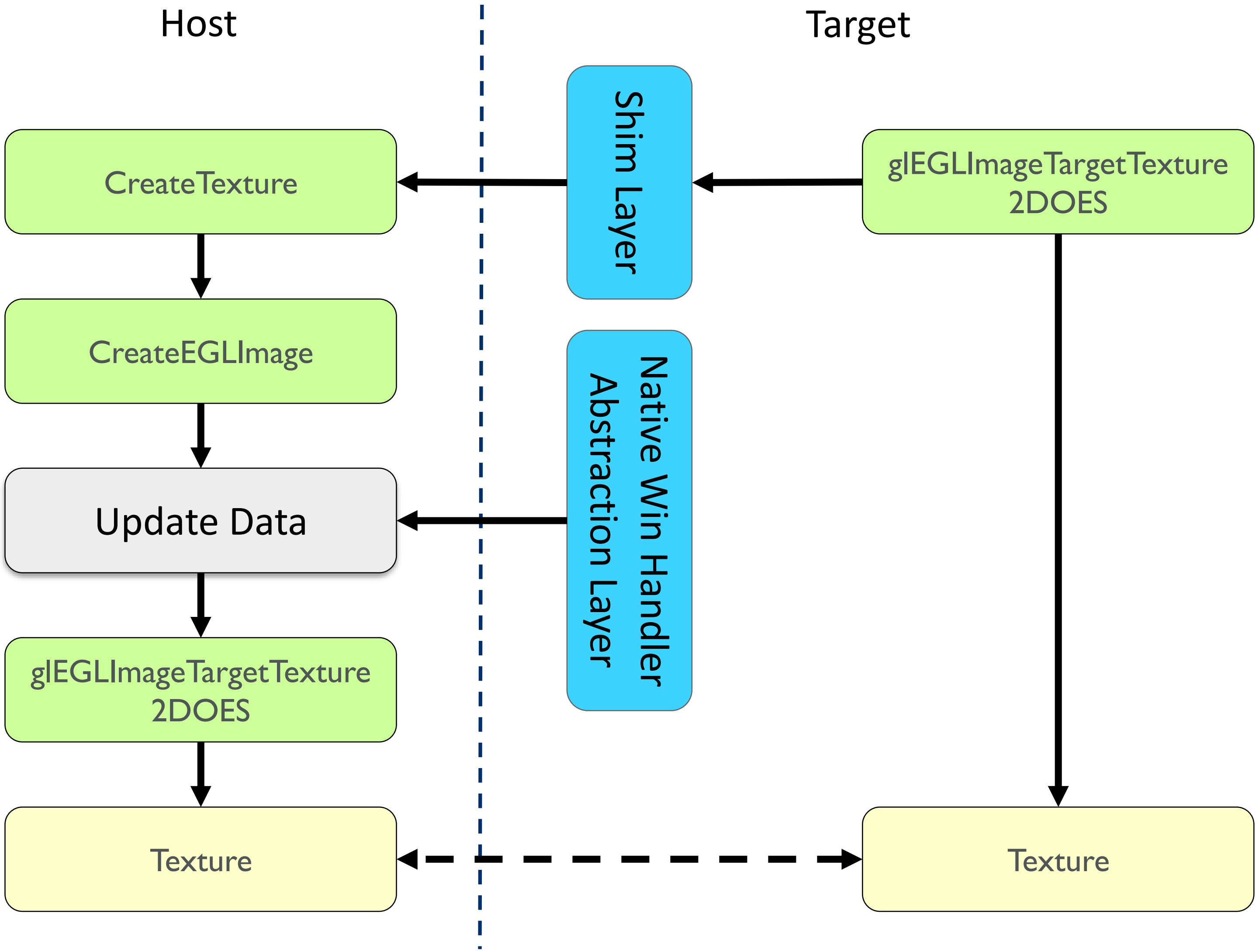
Lazy API Dispatch



Memory Coherence

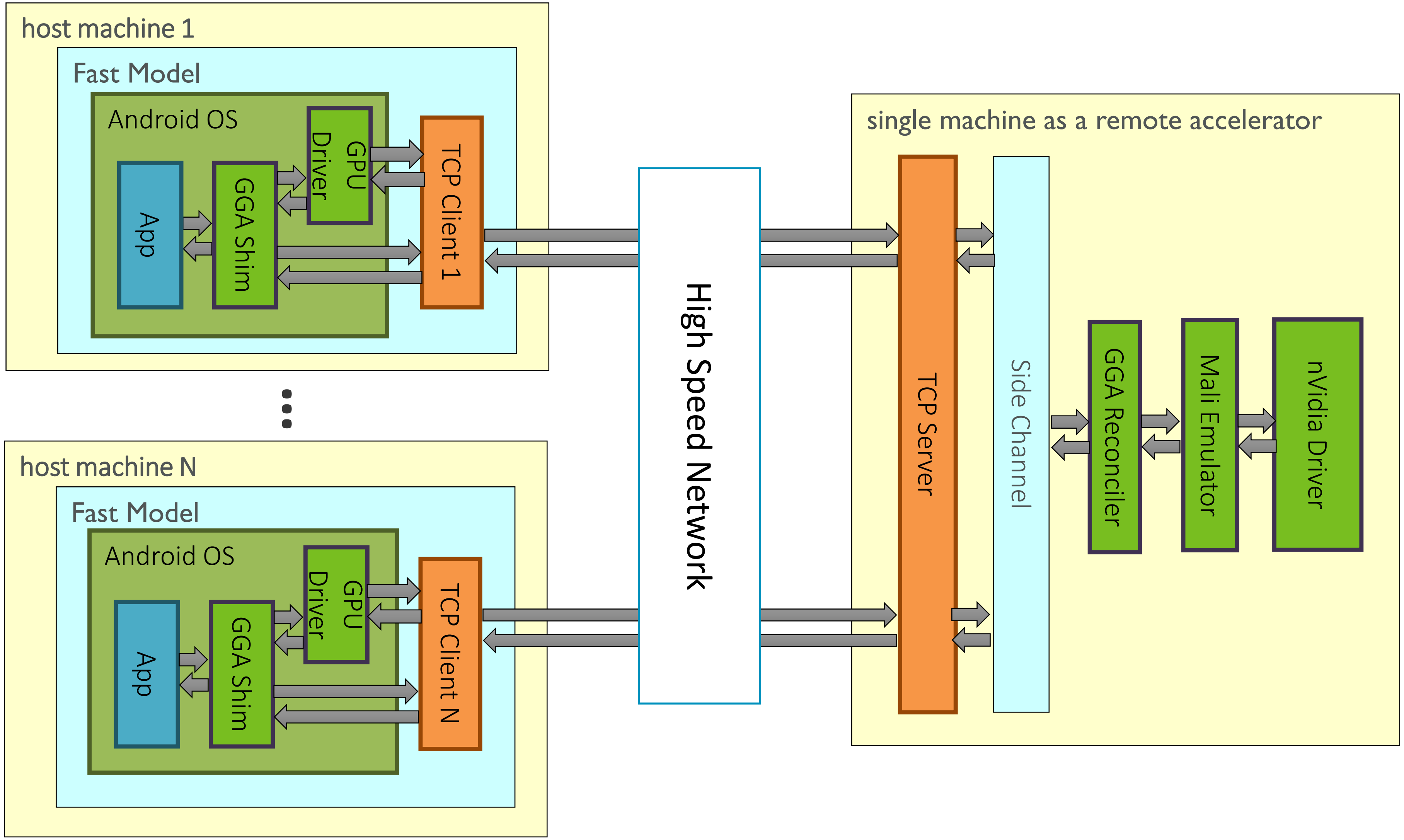


Native Window Handler Abstraction Layer

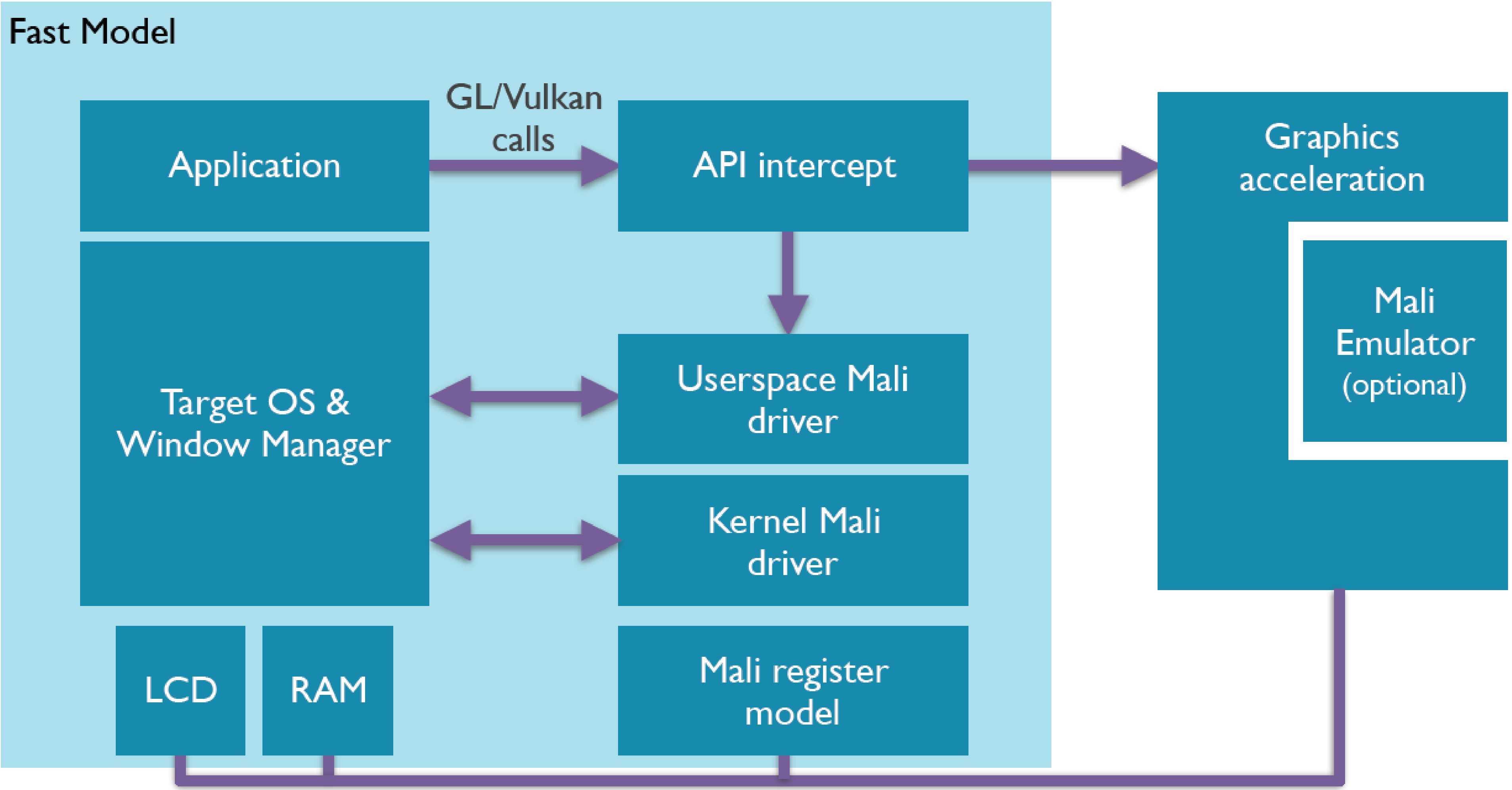


Host
Target
CreateTexture
CreateEGLImage
Update Data
glEGLImageTargetTexture 2DOES
Texture
Shim Layer
Native Win Handler Abstraction Layer
glEGLImageTargetTexture 2DOES
Texture

Remote Acceleration



Subsystem Architecture With Graphics Supported


Fast Model

GL/Vulkan calls

Application

API intercept

Graphics acceleration

Mali Emulator (optional)

Target OS & Window Manager

Userspace Mali driver

Kernel Mali driver

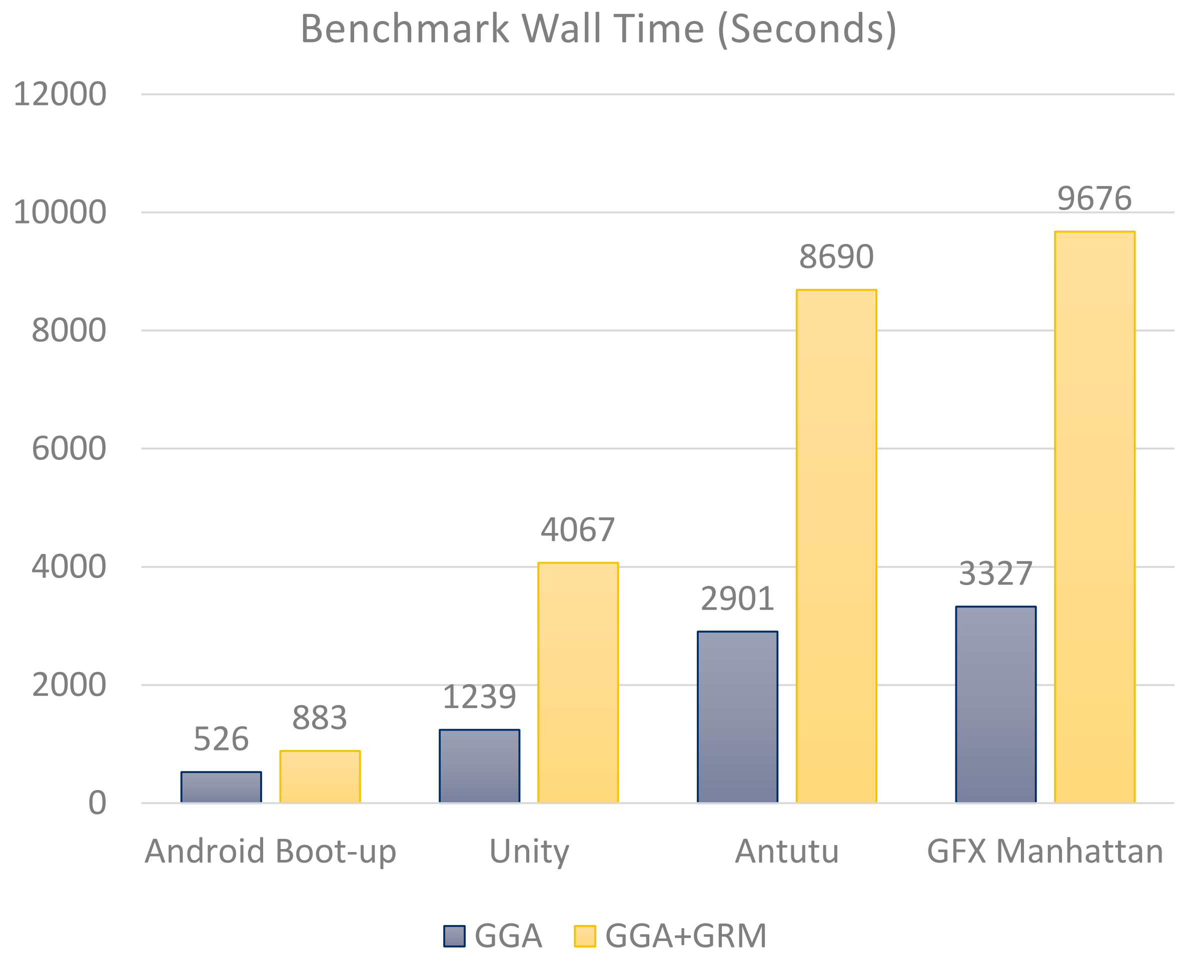
LCD

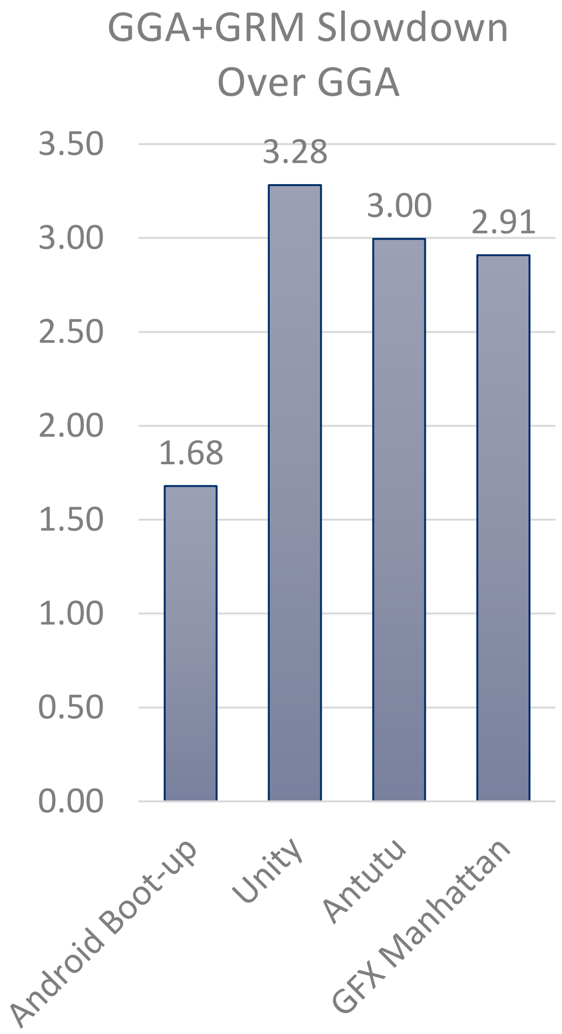
RAM

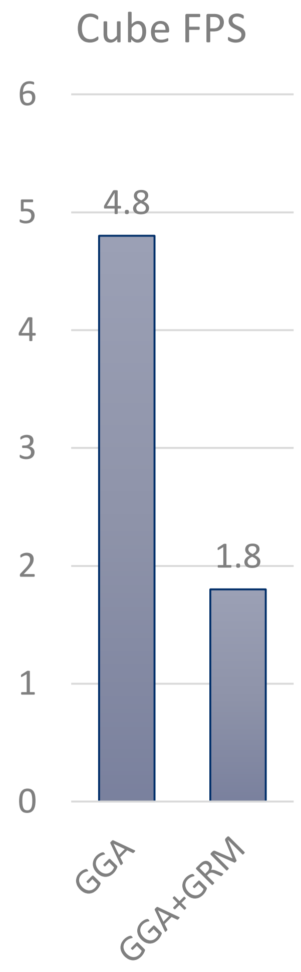
Mali register model

Performance

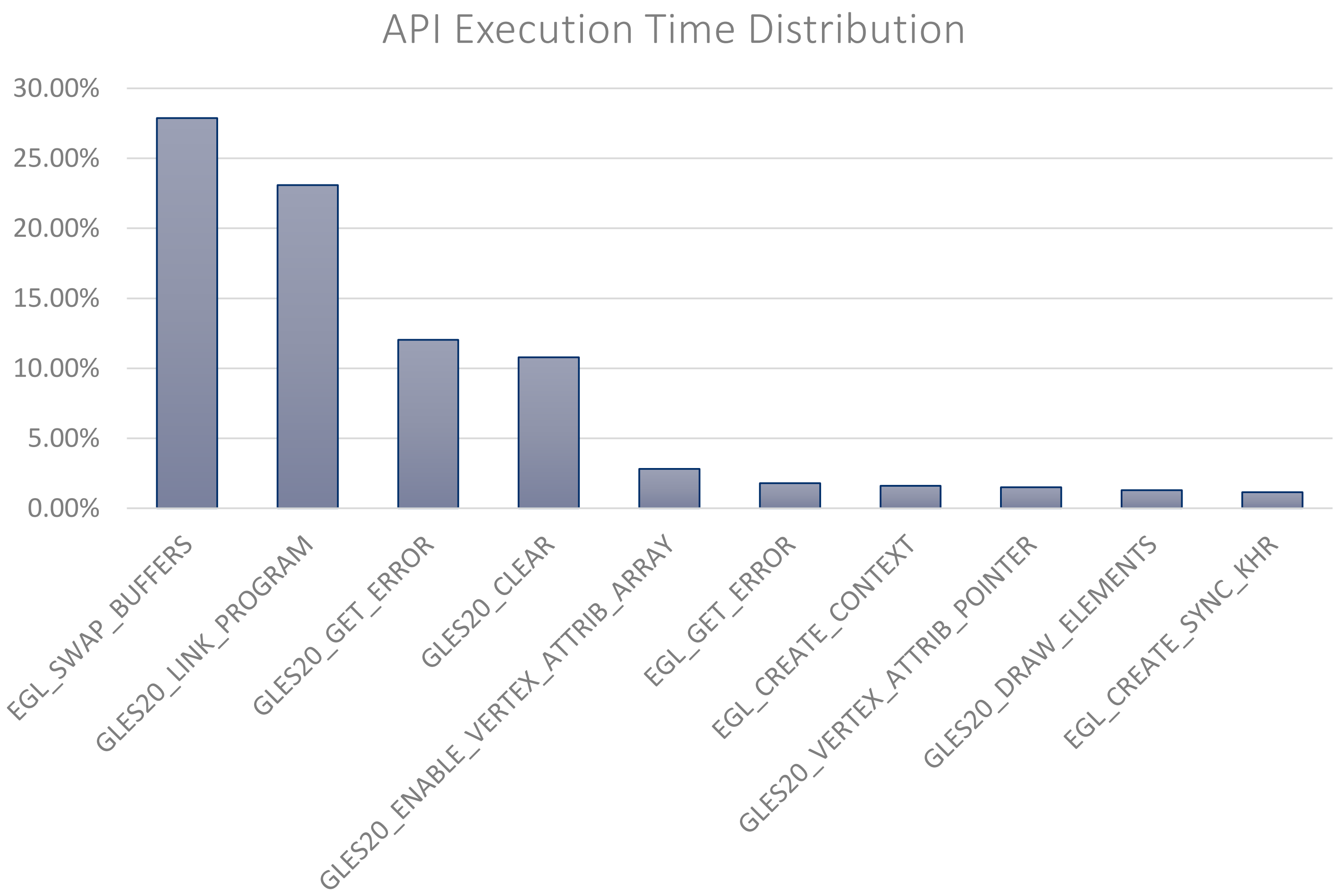
GGA/GRM Performance Data

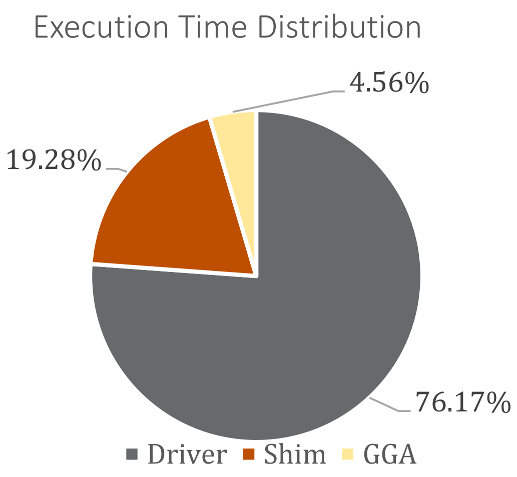


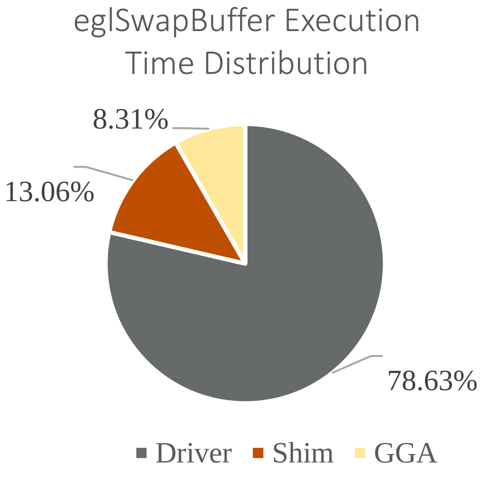




APIs Execution Time Break-down





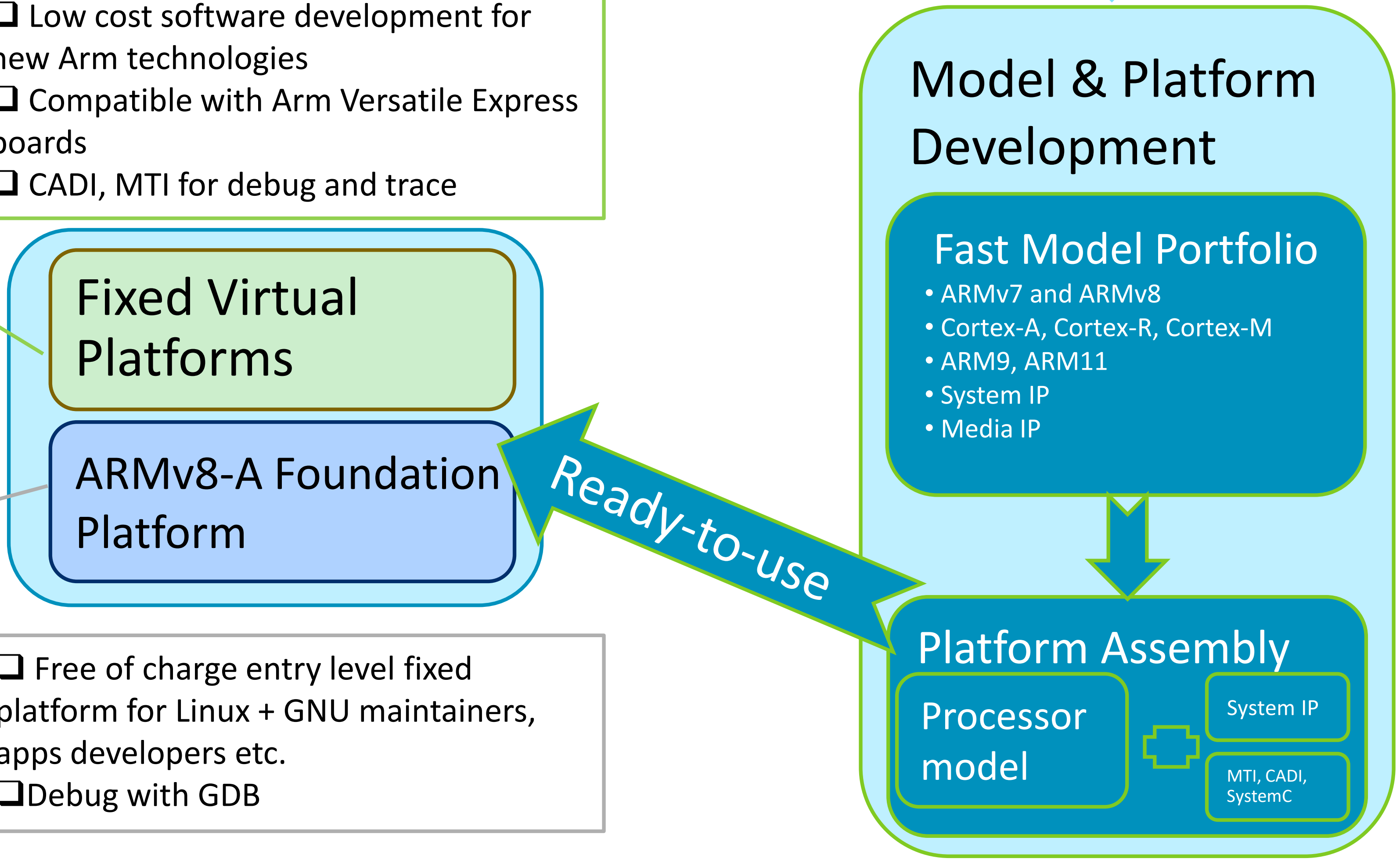


ARMv8 Virtual Prototype

Virtual Prototypes for Software Development

□ Flexible Virtual Platforms tailored to *any* SoC or system
□ Extensible, scalable
□ Integration with Synopsys, Cadence, Mentor, Arm Cycle Model, open source and proprietary TLM simulators

□ Low cost software development for new Arm technologies
□ Compatible with Arm Versatile Express boards
□ CADI, MTI for debug and trace



□ Free of charge entry level fixed platform for Linux + GNU maintainers, apps developers etc.
□Debug with GDB

Subsystem Virtual Prototype

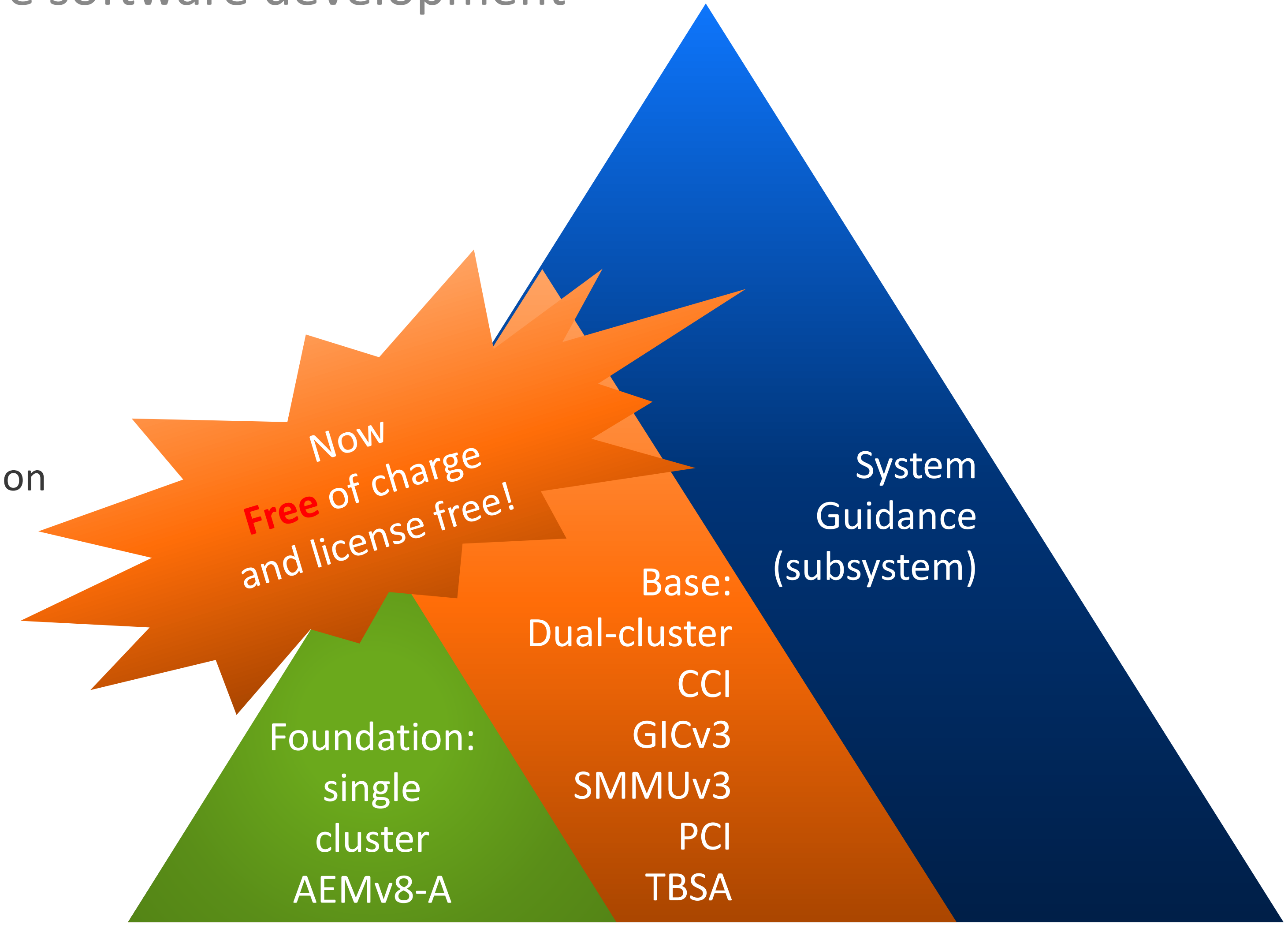
- **Goals**
  - Full support of software development, debugging, and analysis
  - Run whole software stack as early as system IPs’ specification is ready
  - Create virtual platform models that execute with high simulation speeds
- **Simulate multiple systems IPs simultaneously**
  - Hardware solution, E.g. FPGA based emulator
  - Pure software implementation. E.g. Arm® Fast Model™, Synopsys ® VDK
- **Differences from other VMs like VMware, VirtualBox and QEMU**
  - Highly customized for SoC
  - Low level system IP simulation
  - All peripheral devices for a complete subsystem
  - Cross architecture execution

Fixed Virtual Prototypes for Armv8-A

Out of the box solutions for productive software development

- **Foundation: Armv8-A FVP with simple peripheral set for Linux application development**
- **Base: dual-cluster Armv8-A FVP with extended peripheral set including PCIe, GIC, SMMU and CCI models**
  - Supports the latest public Armv8-A specification

  https://developer.arm.com/products/system-design/fixed-virtual-platforms
- **System Guidance: Armv8-A FVPs for Mobile and Infrastructure platforms**



Foundation Platform, Base Platform, Fast Models

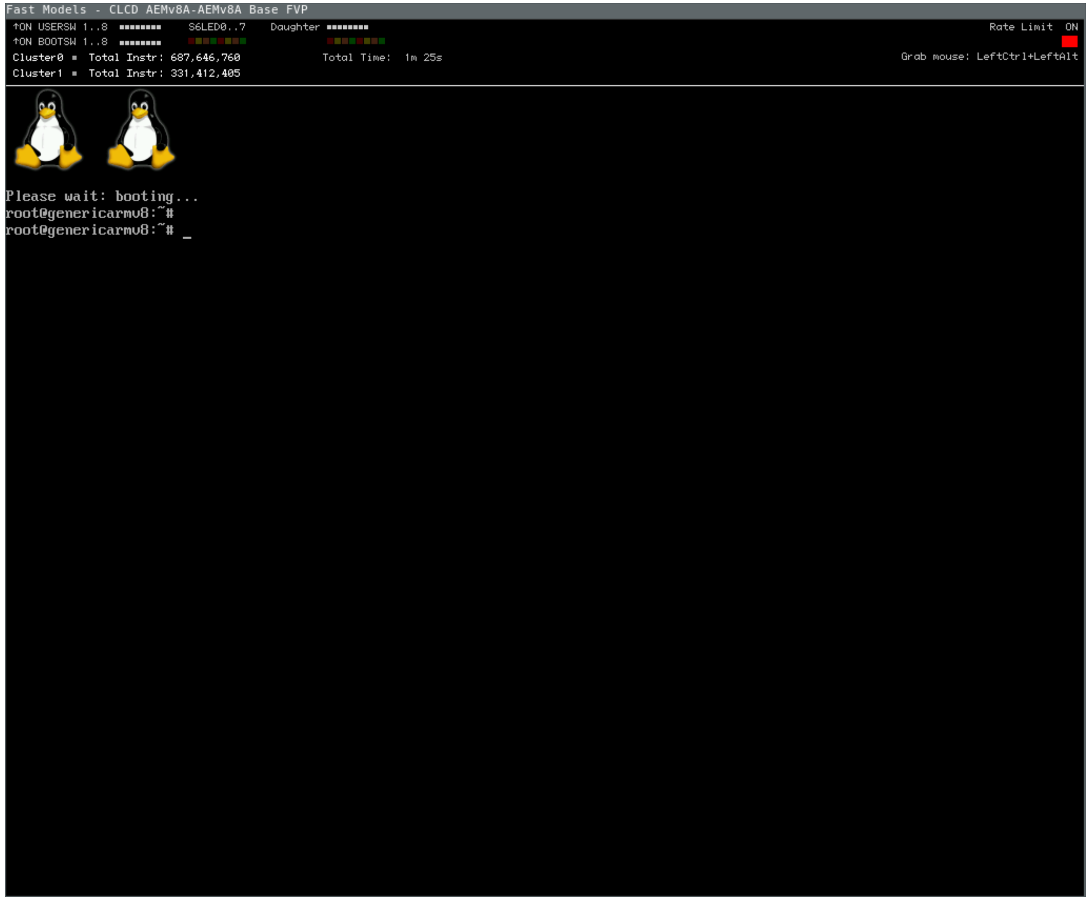
Key Features compared

Foundation Platform	Base Platform (Rev C)	Fast Models
Fixed Virtual Platform (FVP) for AEMv8-A with minimal peripheral set.	Fixed Virtual Platform (FVP) for AEMv8-A with extended peripheral set including PCIe, GIC, SMMU and interconnect models.	Fully flexible package of models and tools to create customised, extensible platforms including GIC, SMMU, GPU, VP, DP and more.
Supports published ARMv8-A specifications	Supports published ARMv8-A specifications	Support for NDA architecture features
Linux application development	Linux/Android support, bare metal software development	Linux/Android support, bare metal software development
Single cluster, 1-4 cores	Dual cluster, 1-4 cores per cluster	Flexible: single/multi cluster, heterogenous platforms, etc
Thumb2EE, crypto not supported	Thumb2EE, crypto supported	Thumb2EE, crypto supported
No SystemC interface	No SystemC interface	SystemC interface
Limited configurability, not extensible.	All platform/model parameters available, not extensible.	Configurable, editable, extensible platforms

Software Stack Supported

The Open Embedded filesystem images are provided with three flavors:

1) minimal — just to get you to a shell prompt
2) SDK — includes developer tools such as a native GNU toolchain
3) LAMP — includes MySQL, Apache, and PHP!


Fast Models - CLCD AEMv8A-AEMv8A Base FVP
↑ON USERSW 1..8
S6LED0..7
Daughter
↑ON BOOTSW 1..8
Rate Limit ON
Grab mouse: LeftCtrl+LeftAlt

```
Cluster0 ▪ Total Instr: 687,646,760        Total Time:  1m 25s
Cluster1 ▪ Total Instr: 331,412,405
```

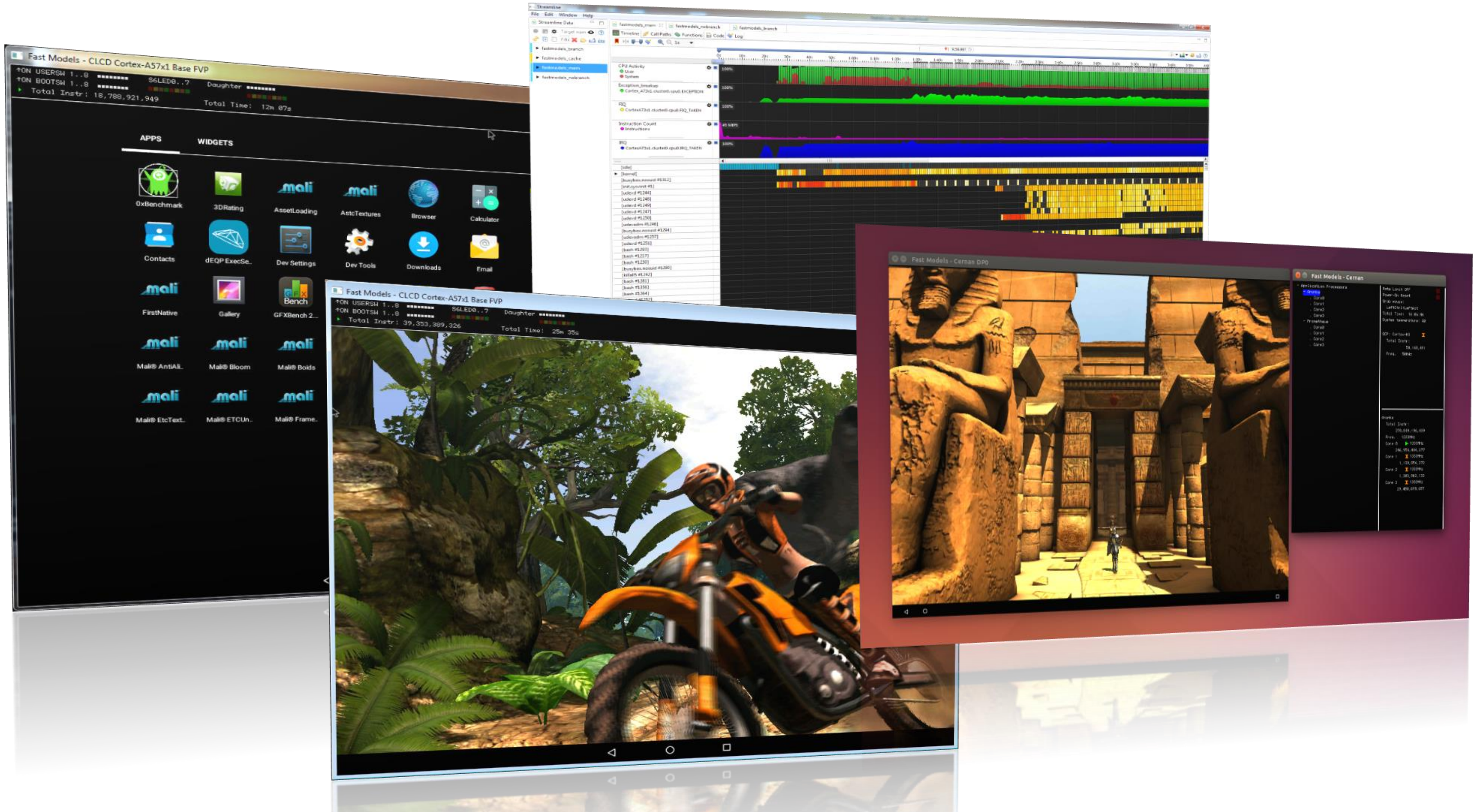
```
Please wait: booting...
root@genericarmv8:~#
root@genericarmv8:~# _
```

Early Software Development

Factor	Emulator	FPGA	Virtual Prototype
Accuracy	Cycle	Cycle	Instruction
Analysis	Medium	Worst	Best
Availability	Later	Latest	Earliest
Capacity	Medium	Lowest	Highest
Cost	Highest	Higher	Lowest
Debug-ability	Medium	Worst	Best
Ease of Use	Hardest	Medium	Simplest
Flexibility	Lowest	Lowest	Highest
Performance	Lowest	Highest	Higher

*** Virtual Prototypes offer the best all-around solution**

Software Development Portfolio



```
Fast Models - CLCD Cortex-A57x1 Base FVP
↑ON USERSW 1..8                S6LED0..7    Daughter
↑ON BOOTSW 1..8
▸  Total Instr: 18,788,921,949              Total Time:  12m 07s
```

```
Fast Models - CLCD Cortex-A57x1 Base FVP
↑ON USERSW 1..8                S6LED0..7    Daughter
↑ON BOOTSW 1..8
▸  Total Instr: 39,353,309,326              Total Time:  25m 35s
```

Conclusion

arm

Conclusion

- Running commercial graphics benchmark in Arm fast model is amazing
- Great support to full system software stack integration and verification
- Significantly shorten SoC development cycle with subsystem virtual prototype

Q
&
A

Thank You!

Danke!

Merci!

谢谢!

ありがとう!

Gracias!

Kiitos!

arm