

# Integrating the driver experience

# Virtio-GPU

2018-10-11 Genivi Bangalor Tech Summit Matti Moell

public

# VISION: Run Guests without modifications



A Virtual Platform allows the development of virtual machine guests that could be moved among different hypervisor systems and/or HW platforms without further modification through establishing an industry standard / de-facto standard.





**Device** refers to the implementation of the virtual/para-virtual device, also known as Backend or Server

Driver refers to the guest driver, also known as Frontend or Client

**Device Host** is the guest that provides the Device to other guests

**Device Guest** is the consumer of a Device

**Guest** is a partition or virtual machine

# Virtualized device Architecture with VIRTIO





Bulk data transport via DMA-like memory model

- Buffer **allocations** handled by "Driver" part (client)
- **Direct** R/W access to allocated buffers in the "Device" part (server)

Metadata transport via virt-queues (ring buffers, asynchronous pipeline)

### VIRTIO-GPU



# Virtio-GPU 2D

- Virtual Display, much like a VGA port
- Accepted for VIRTIO 1.1
- Framebuffers object are allocated in the driver
- Four Basic Commands
  - Get screen information
  - Attach framebuffer via sg\_list
  - Set scan-out by reference
  - Flush scan-out

# Virtio-GPU 3D

- Forward sanitized OpenGL Commands
- Support for OpenGLES 3.1, OpenGL 4.1, Vulkan support in development
- Based on open-source libvirgilrenderer on the host (BSD licensed)
- Based on Mesa driver for guest
- Shaders are transferred in intermediate format (TGSI)

### Virtio GPU Architecture (QEMU)

6 OPENSYNERGY

Apps	Apps	Apps	
MESA			
	Virgl Gallium		
Guest Kernel	DRM/KMS		
	Virtio-GPU driver		Guest
QEMU-KVM			Host
	Virtio-GPU Device libvirgilrenderer		
	LibGL		
Host Kernel	DRM/KMS		
	GPU Driver		
	GPU		

www.opensynergy.com | 6

Virtio GPU

### Virtio GPU Overview



### Virgl Gallium

- Implements Gallium driver for mesa
- Generate virtio-GPU
  Command stream
- Conververts GLSL into TGSI

#### Virtio-GPU driver

- Screen metadata
- Modesetting
- Context Mangement
- 3D resource management
- DMA transfer initiation
- Command stream submission
- Fencing

#### libvirgilrenderer

- Gallium to GL interface conversion
- TGSI->GLSL
- Host side context
  lifecycle management
- OpenGL version
  arbitration
- Issues OpenGL commands to native LibGL





### Headquarter

#### Berlin

OpenSynergy GmbH

Rotherstraße 20 D-10245 Berlin Germany Phone +49 30 / 6098 5400

### **Further Locations**

#### Utah

OpenSynergy, Inc. (USA) 765 East 340 South Suite 106 American Fork, Utah 84003 USA

#### California

OpenSynergy, Inc. (USA)

501 W. Broadway, Suite 832 San Diego, California 92101 USA Phone +1 619 962 1725

#### Munich

OpenSynergy GmbH

Starnberger Str. 22 D-82131 Gauting / Munich Germany Phone: + 49 89 / 8934 1333

E-Mail info@opensynergy.com Web www.opensynergy.com

OpenSynergy, COQOS Hypervisor SDK, Blue SDK, IrDA SDK, Voice SDK, Qonformat, and other OpenSynergy products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of OpenSynergy GmbH in Germany and in other countries around the world. All other product and service names in this document are the trademarks of their respective companies. These materials are subject to change without notice. These materials are provided by OpenSynergy GmbH for informational purposes only, without representation or warranty of any kind and OpenSynergy GmbH shall not be liable for errors or omissions with respect to the materials. OpenSynergy GmbH 2018