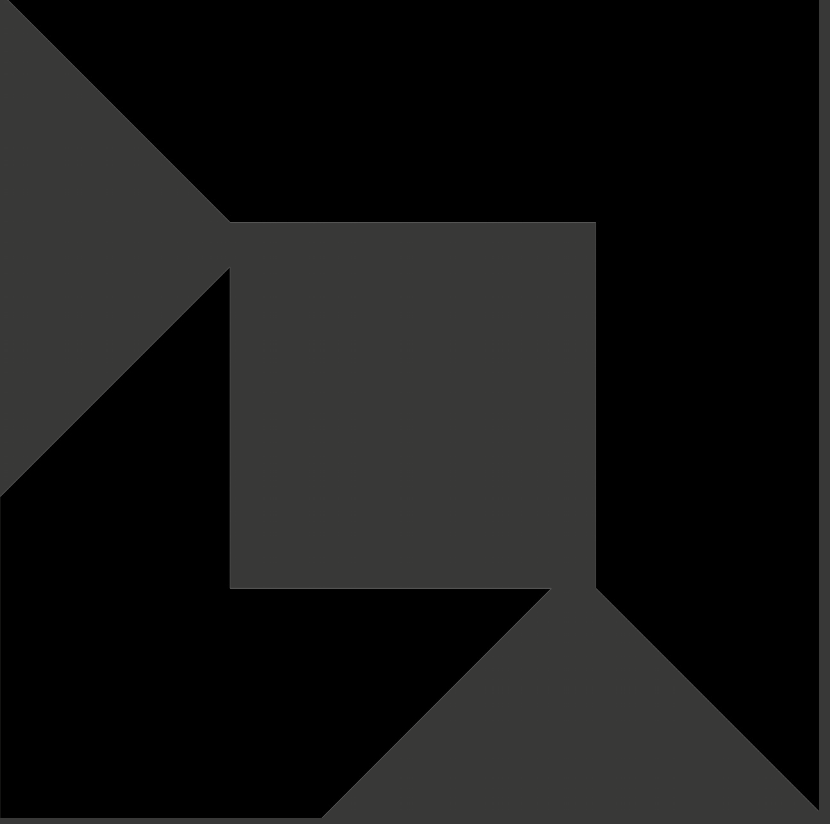


TTM MEMORY MANGER

A GENERAL OVERVIEW AND AN UPDATE ON GRAPHICS MEMORY MANAGEMENT IN THE LINUX KERNEL

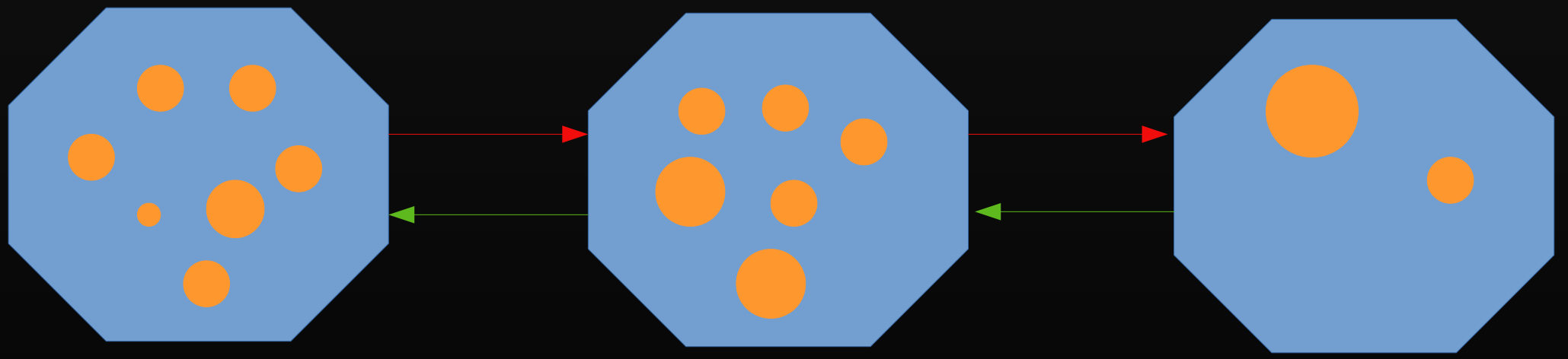
FOSDEM 2020

AMD

WHAT'S TTM AND WHO'S USING IT?

- TTM is the memory manager for graphics devices with dedicated memory.
- Directly used by the following drivers:
  - AMDGPU
  - Nouveau
  - QXL
  - Radeon
  - VMWGFX
- Indirectly through the DRM VRAM helpers by:
  - AST
  - Bochs
  - Hisilicon
  - MGAG200
  - vboxvideo

HOW DOES TTM WORKS?



VRAM

Local memory attached to dedicated GPUs or stolen system memory for integrated GPUs.

System

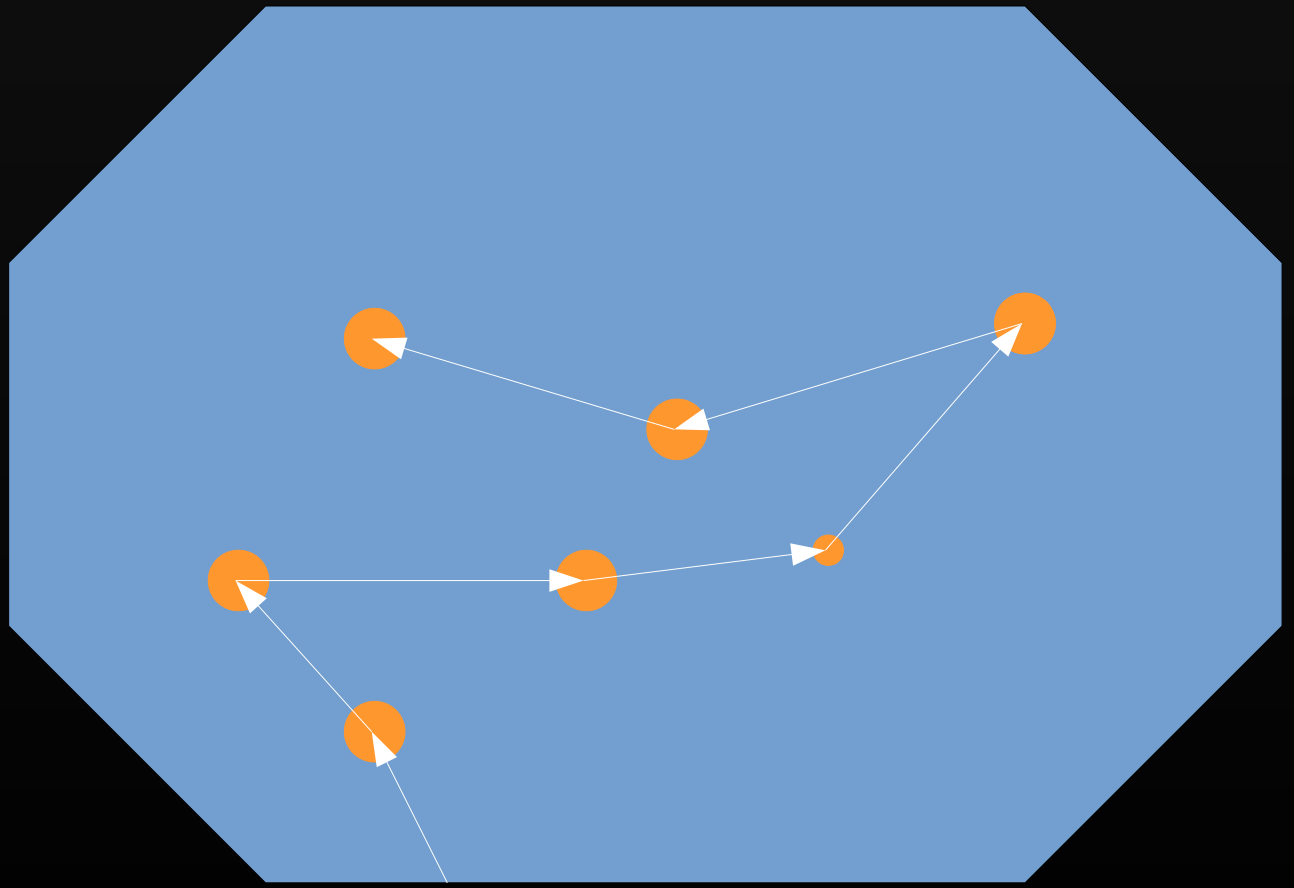
Normal system memory allocated through the DMA API or GFP.

Swap

Either swapped to storage media or otherwise not directly accessible.

HOW EVICTION WORKS

1. Lock BO on LRU
2. Eviction valuable?
3. Evict BO
4. Re-try allocation
5. Repeat



WHAT ELSE DOES TTM PROVIDE?

CPU PAGE FAULT HANDLING

- Correct locking dance for dma_resv objects
- Filling CPU page tables on demand
- Filling in multiple entries at once
- Invalidating all mappings on eviction

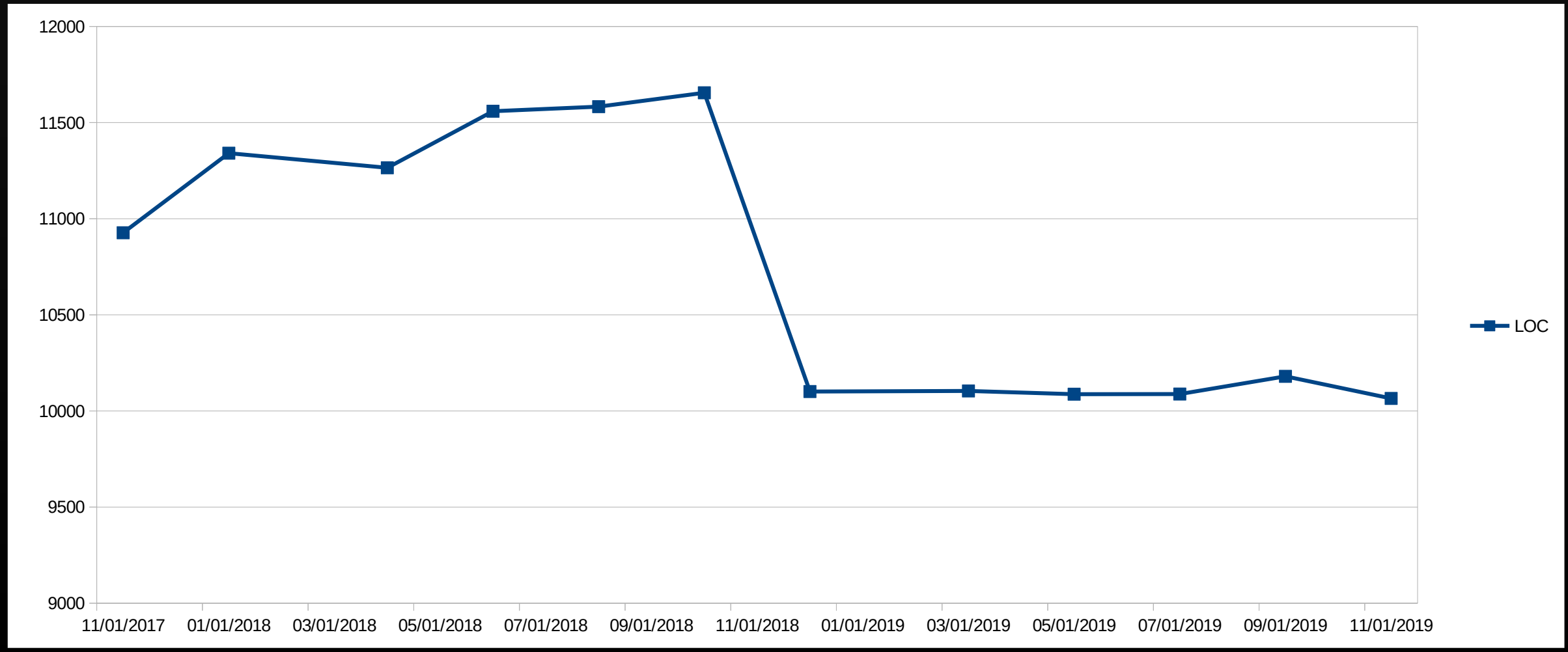
DMA MEMORY PAGE POOLS

- Uncached pool
- Uncached write combined pool
- Decrypted pages pool
- Huge page handling
- DMA32 handling

PROBLEMS?

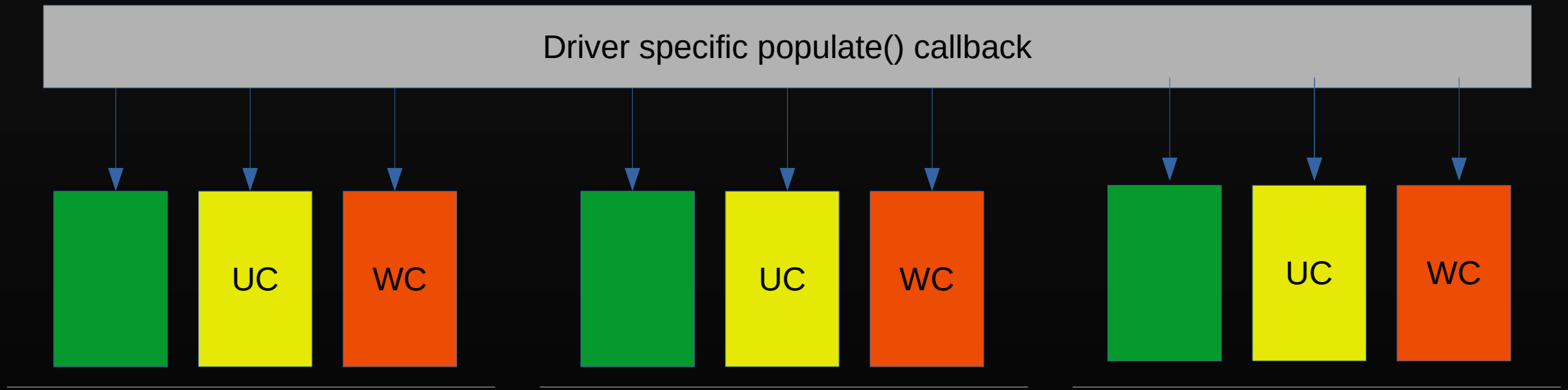
DUMPING GROUND	DMA API ABUSE	DESIGN ISSUES
Driver specific features in common code Features who are entirely unused Hard to move functionality back into drivers	Functionality which belongs into the DMA API Making assumptions how the DMA API works internally	Horrible midlayer design TTM calling back into driver Driver calling back into TTM again Limiting driver memory usage

DUMPING GROUND CLEANUP



AMD

USAGE OF THE DMA API



Normal

Normal system memory without special requirements on the placement.

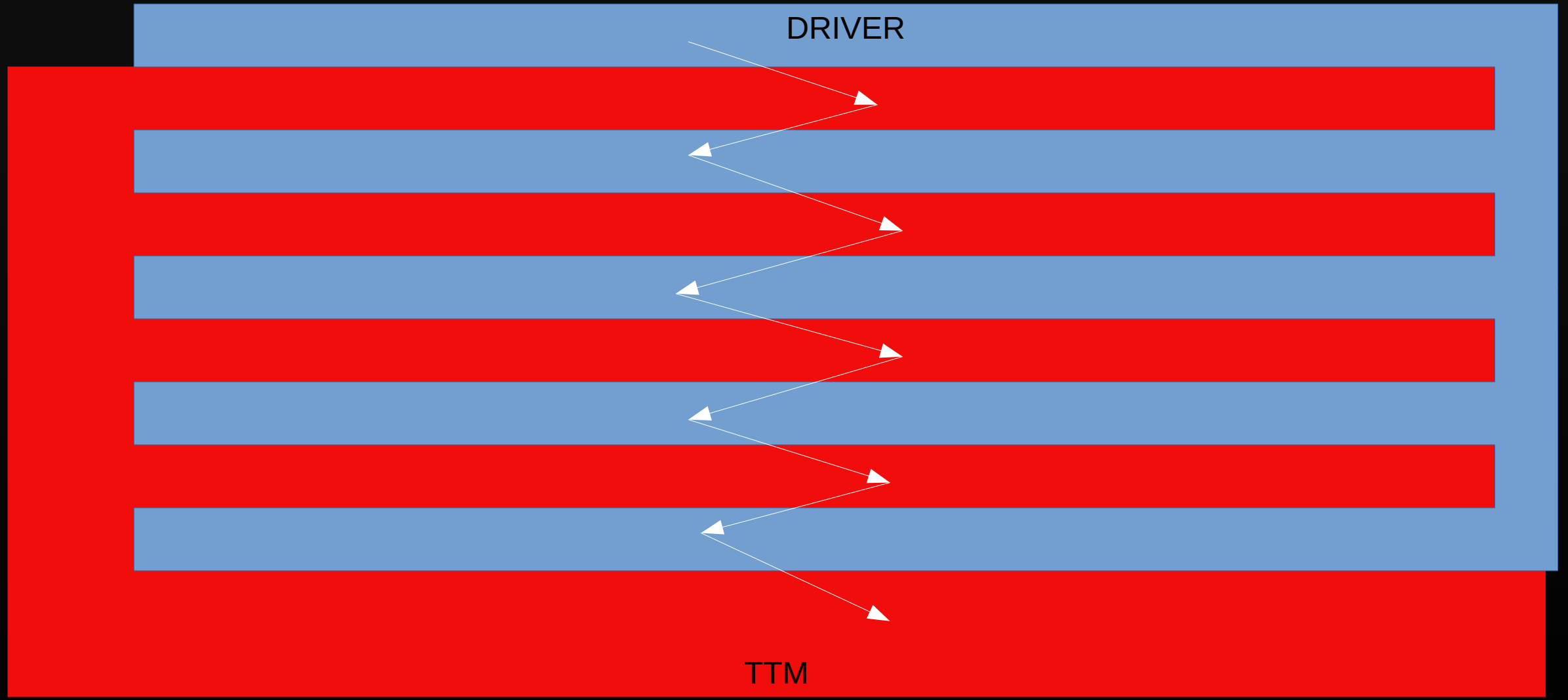
DMA32

Memory for devices which can only work with 32bit addresses.

Huge

Memory allocated continuously for better TLB performance.

DESIGN ISSUES



HOW TO FIX IT?

- Kill it with fire? Rather not!
  - Used by to many drivers.
  - Actually works pretty well.
  - Mostly bug free.
  - Maybe remove AGP support.
- Slow decomposition? Yes, certainly.
  - Move driver specific code/features into driver (mostly done).
  - Remove unnecessary complex handling (partially done).
  - Move code into the DMA API (todo)
- Moving more functionality into new components.
  - DRM/GEM ref count cleanup
  - DMA-buf locking framework
  - LRU cursor

Thanks to:
Alex Deucher <alex.deucher@amd.com>
Huang Rui <ray.huang@amd.com>

DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED ‘AS IS.” AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.