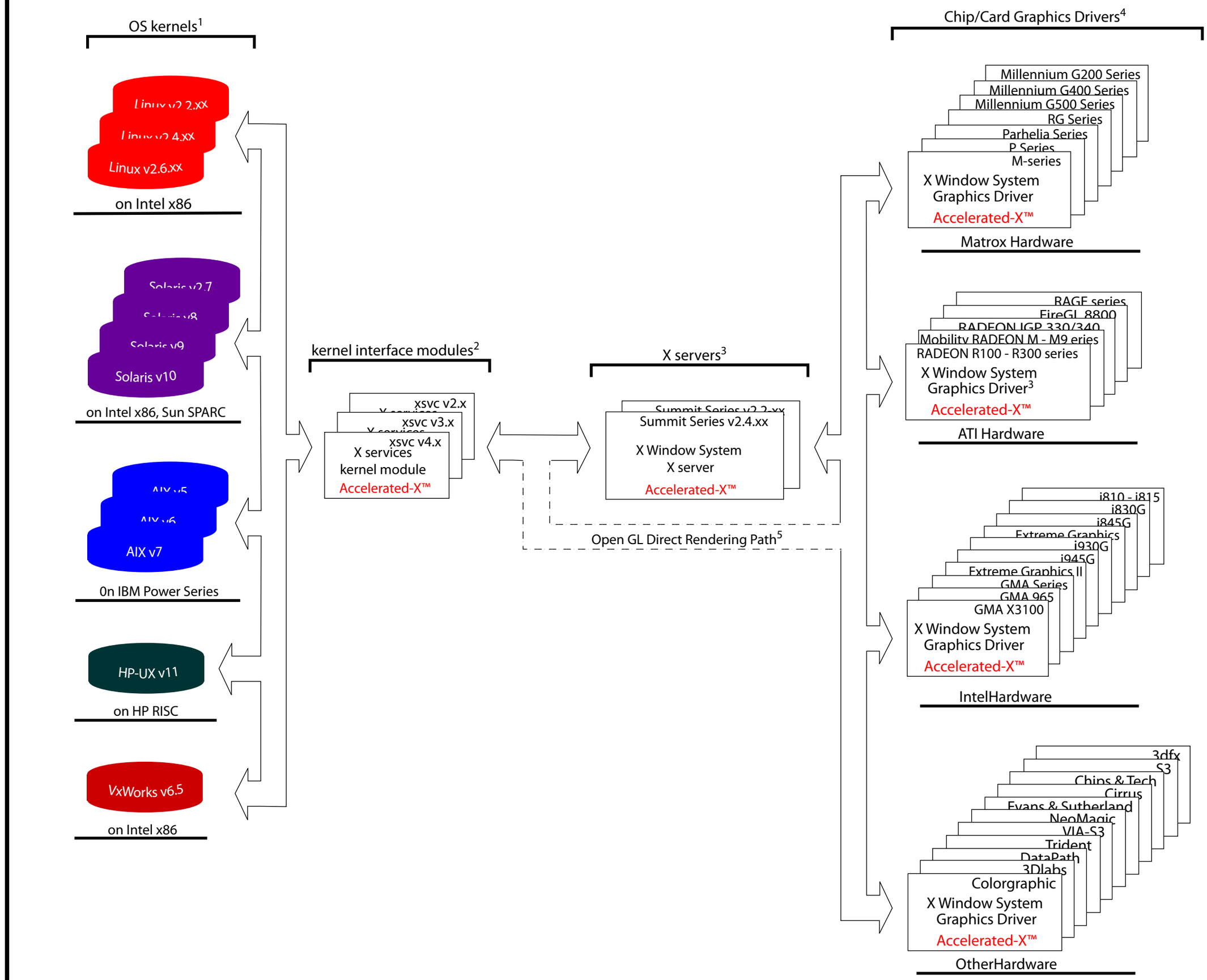
The Premier X Window System For UNIX/Linux

Xi Graphics' Accelerated-X™ Summit Series Software



Notes:

1. The architecture design of Accelerated-X begins with the philosophy that the X Window System is essentially an application sub-system, rather than a kernel module or utility. This has the effect of isolating Acceleerated-X from most changes that occur in the kernel from bug fixes, and minor revisions, and reduces the effect of major changes in the particular brand of the UNIX kernel. In the case of Linux, the X.org/Open Source/Linux kernel crew has adopted the approach that some of the graphics of X should be moved into the kernel under the guise that graphics performance or system security will be enhanced. In doing so, small changes to the kernel can (and do) cause problems in the large, and highly complicated X Window System SW, seriously complicating system SW maintenance.

2. Since the UNIX kernels (we include Linux in this group) vary among the various brands - Solaris on SPARC, Solaris on Intel, Linux on Intel, AIX on Power Series, and so on - Accelerated-X includes an interface module, called the "X Services" module, or "xsvc", to glue the X server to the particular brand and version of kernel being installed. Kernel changes that effect the kernel interface are normally confined to this module, avoiding many maintenance issues that the X.org approach (moving some graphics operations into the kernel) must handle.

3. X servers in the Accelerated-X products are produced "from scratch" by Xi Graphics developers. This large body of work has undergone major changes over the past fifteen years at Xi Graphics as the use of graphics has exploded in small and large computer systems. Where the initial design of the X Window System envisioned the use of a graphics chip and a monitor to produce two-dimensional displays, modern graphics-intensive systems employ multiple, sometines many, monitors/projectors, 3D graphics, "stretched desktops" across multiple graphics chips/cards - some with one or more "video windows" showing live action from remote cameras - and so on. Most of the infrastructure necessary to accomplish such features with acceptable performance, while keeping SW maintenance at low levels, is found in the X server. X servers produced by a myriad of Open Source part-time contributors have shown to be lacking when the graphics requirements of a system is non-trivial and important to the successful performance of the system.

4. Graphics Drivers in the Accelerated-X products are also produced "from scratch" by Xi Graphics developers. To do so, the developers must have the complete logic design documentation for the graphics chips/cards - sometimes referred to as "register level" documentation - which is normally confidential IP of the graphics chip/card manufacturer. Without the cooperation of the graphics hardware manufacturer in providing Xi Graphics this documentation (under NDA, of course), Xi Graphics (nor anyone, for that matter) can produce a "hardware-accelerted" graphics driver that can take full advantage of the hardware capabilities of the graphics chip/cards. The Accelerated-X graphics drivers are designed to work only with X servers in the Accelerated-X Summit Series products. The interface between the graphics drivers and X servers is proprietary to Xi Graphics, and has been designed for high-performance, robust operation, ease of porting, and very low maintenance.

5. OpenGL implementations can include the ability of a graphics application to largely bypass the X server and communicate directly with the graphics card(s) in the system, known as "Direct Rendering". This is a non-trivial piece of work to accomplish correctly while maintaining system security and "distance" from the kernel while preventing the graphics applications creating havoc within the X Window System SW, providing high-performance (which is the reason for direct rendering in the first place), and still have a robust/low maintenance system. The X.org approach to OpenGL Direct Rendering requires certain 3D graphics operations to be carried out *inside* the kernel (see the DRI/DRM design), which leads to numerous complications for the overall system

Wm.E. Davis 2011

— 0 —.