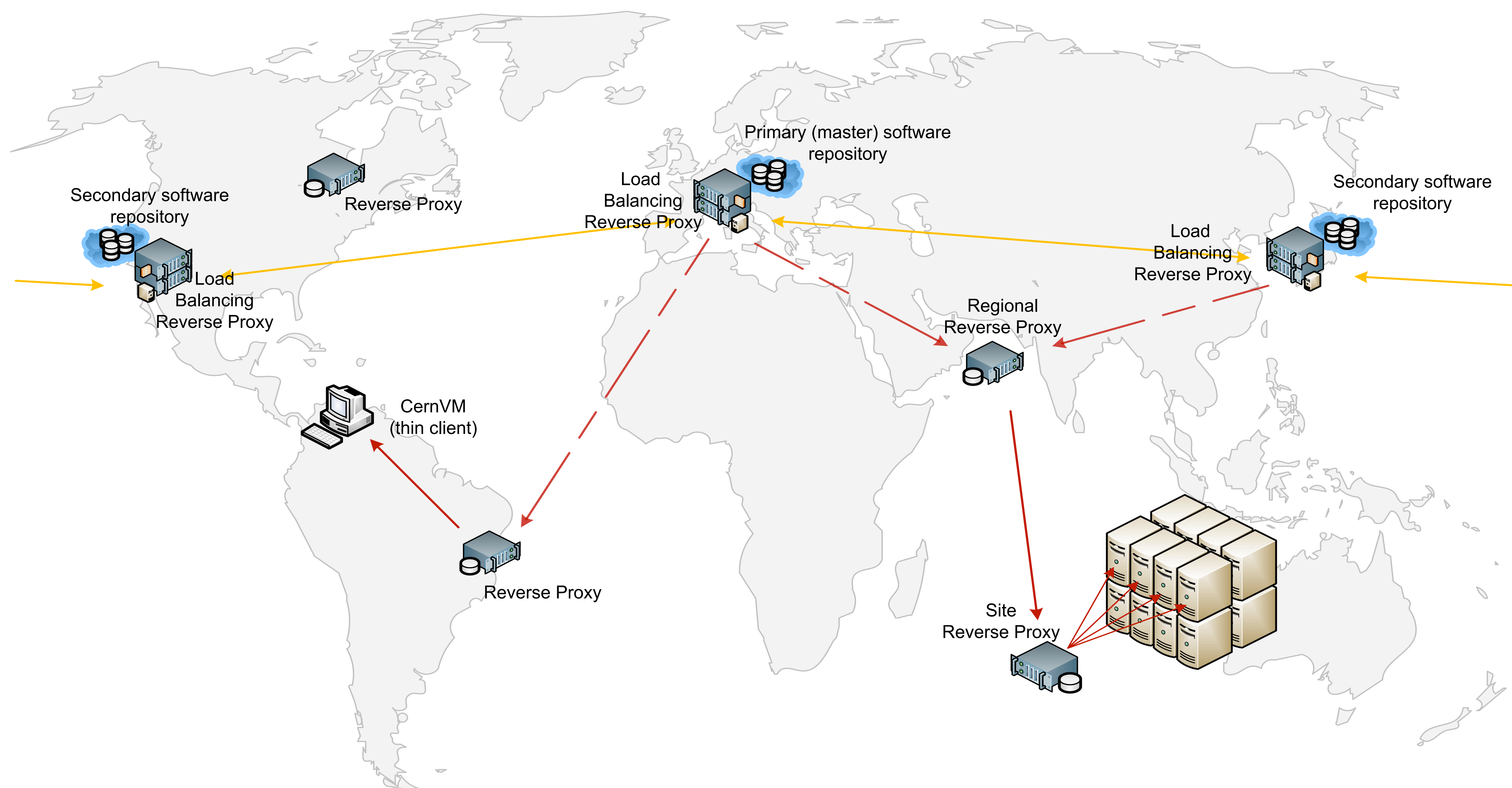


C. Aguado-Sanchez¹⁾, P. Buncic¹⁾, L. Franco¹⁾, S. Klemer¹⁾, P. Mato¹⁾

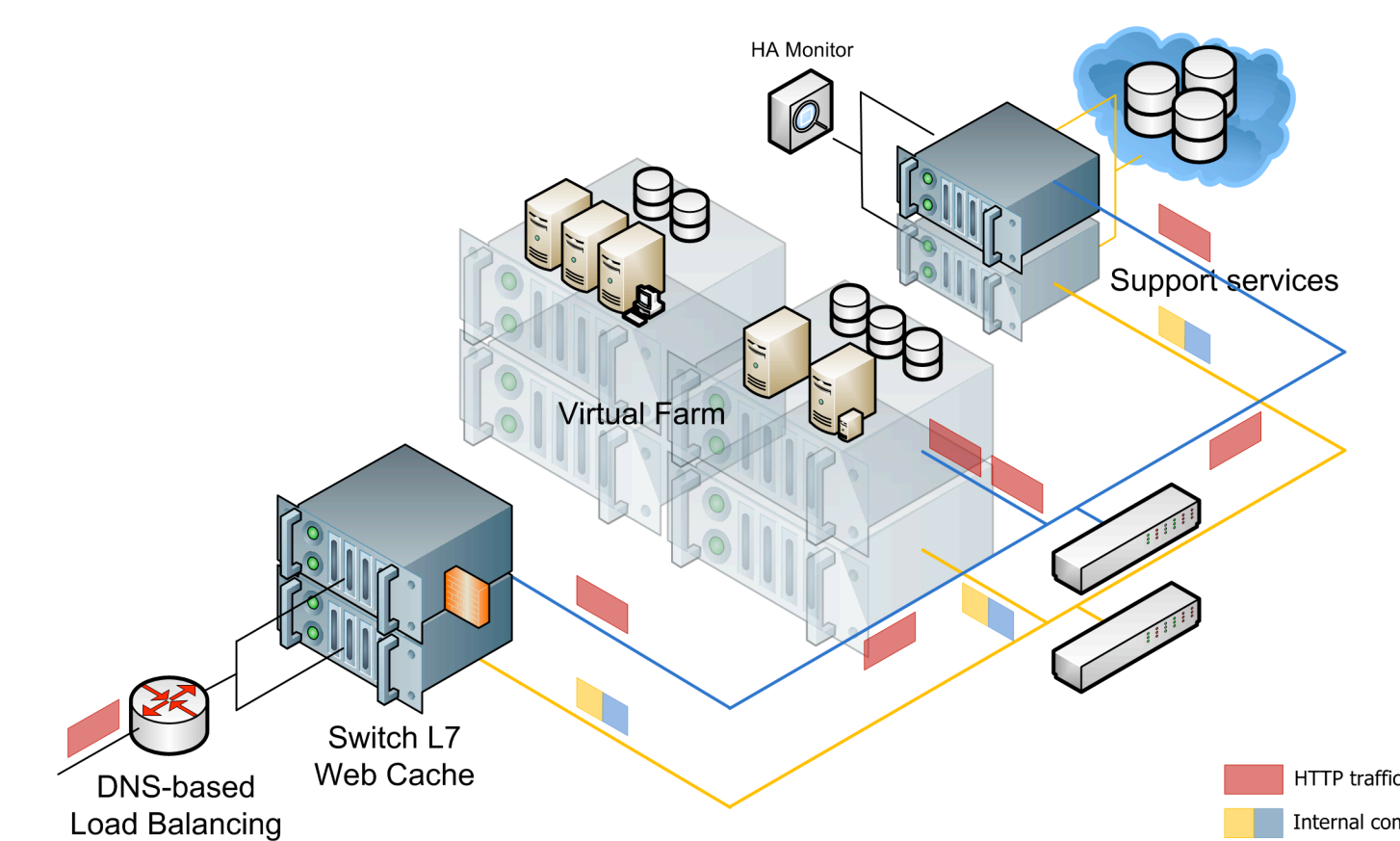
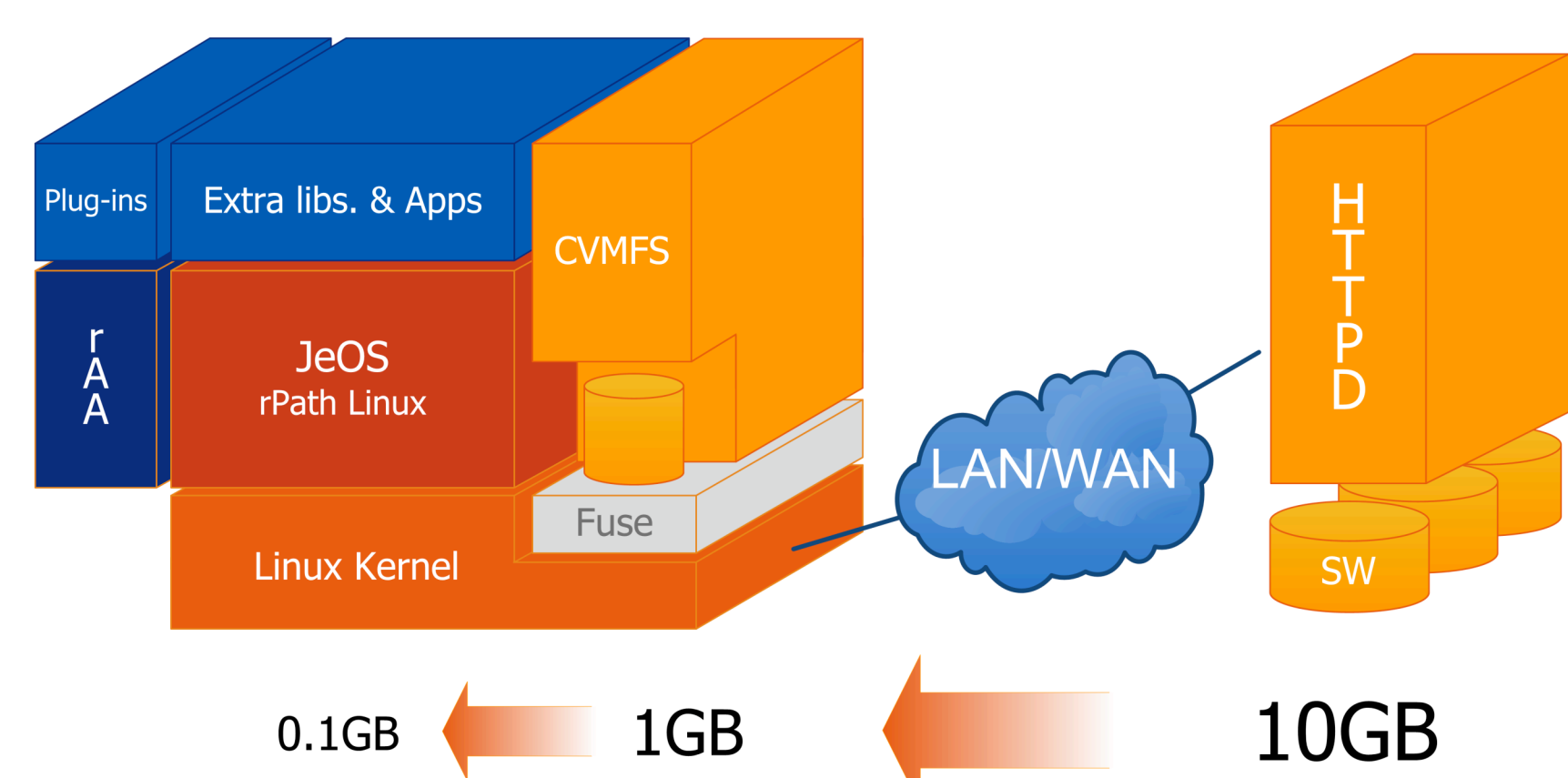
¹⁾ CERN, Geneva, Switzerland

CernVM¹⁾ is a Virtual Software Appliance to run physics applications from the LHC experiments at CERN. The virtual appliance provides a complete, portable and easy to install and configure user environment for developing and running LHC data analysis on any end-user computer (laptop, desktop) and on the Grid independently of operating system software and hardware platform (Linux, Windows, MacOS). The aim is to facilitate the installation of the experiment software on an user computer and minimize the number of platforms (compiler-OS combinations) on which experiment software needs to be supported and tested thus reducing the overall cost of LHC software maintenance.



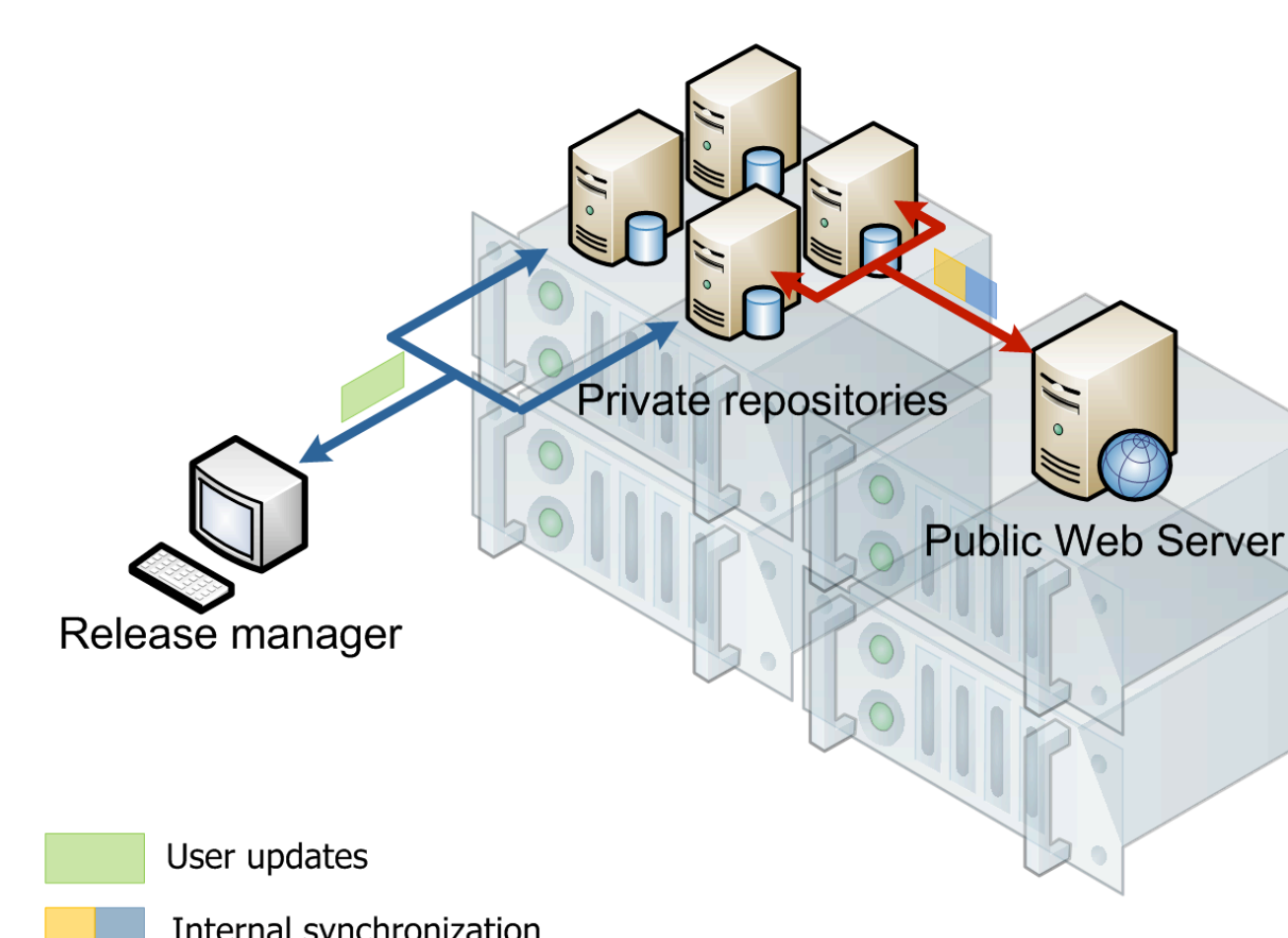
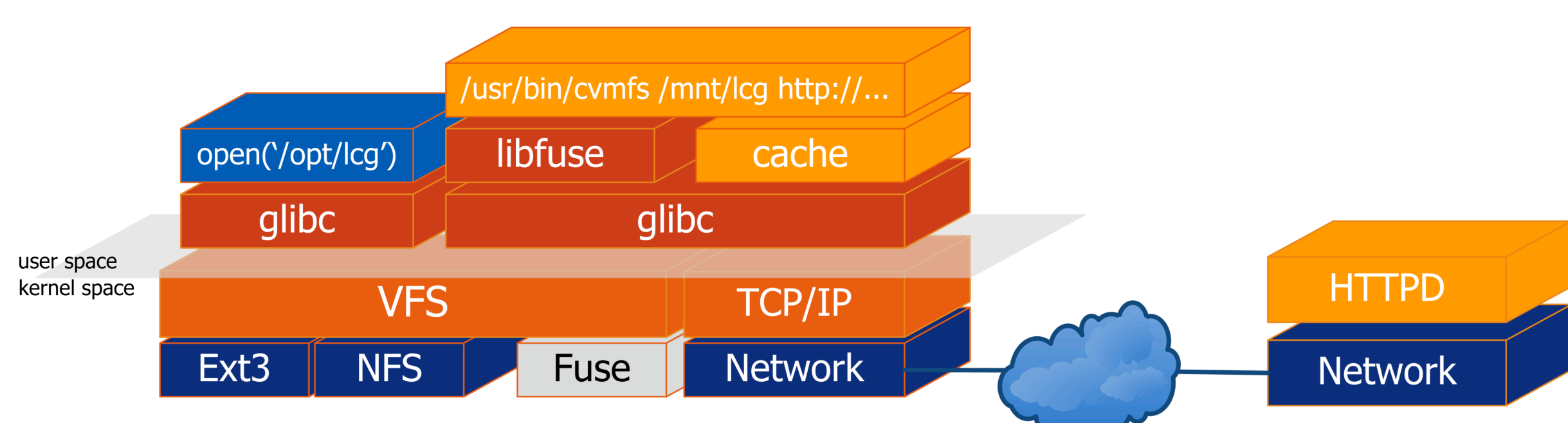
The CernVM Virtual Software Appliance is a thin Virtual Machine that contains just enough of Operating System to run the application frameworks of the four LHC experiments. Its Operating System, based on rPath Linux²⁾, fits into a compressed file smaller than 100 MB. The experiment software stack is brought into appliance by means of a file system specifically designed for an efficient and 'just in time' software distribution. In this model, the client downloads only necessary binaries and libraries as they get referenced for the first time. By doing that, the amount of software that has to be downloaded in order to run the typical experiment tasks in the Virtual Machine is reduced by an order for magnitude.

To minimize impact on central file distribution point, the access to CernVM file servers can be channeled through a chain of proxy servers until it reaches the central Web server. In order to provide scalable central services in support of CVMFS, we have build an infrastructure that can be expanded by adding more front end servers (reverse proxies) operating in DNS load balancing configuration and forwarding request based on URL content to the backend servers that run again as Virtual Machine on top of the physical servers. Another pair of the servers, running in a highly available configuration, is providing iSCSI targets for the rest of the infrastructure and allowing for live migration of Virtual Machines between physical nodes.



Practical experience from running a large scale data processing in HEP environment³⁾ shows that one of the biggest problems, an efficient distribution of code to many thousands of worker nodes around the world, can be solved by introducing a dedicated file system⁴⁾ for software distribution. Global Read-Only Web file system (GROW-FS) is designed to make a directory tree stored on a web server accessible over the wide area network, with aggressive caching and end-to-end integrity checks. It is designed to maximize the cacheability of all components of the file system. Both files and file metadata can be cached on local disk without remote consistency checks, as well as cached on shared proxy servers, allowing the file system to scale to a very large number of clients.

In CernVM model the experiment application software and its dependencies are built independently from CernVM Virtual Machine. The procedures for building, installing and validating each software release remains in the hands and responsibility of each user community. We provide the tools to synchronize pre-built and configured experiment software releases with our central distribution point that we plan to evolve into fully decentralized content delivery network.



The original GROW-FS file system was embedded in Parrot⁵⁾ framework and meant to run completely in user space. Since in Virtual Machine we have a full access to kernel and kernel modules, we constructed CernVM File System (CVMFS) from Parrot and GROW-FS code base and adapted them to run as a FUSE kernel module while adding the extra features like:

- possibility to use multiple (hierarchy of) file catalogues on the server side
- transparent file compression under given size threshold
- dynamical expansion of environment variables embedded in the symbolic links

References

1. CernVM Project, <http://cernvm.cern.ch>
2. rPath, <http://www.rpath.org>
3. Pagan Griso, D. Lucchesi, G. Compostella, I. Sfiligoi, D. Cesini, *CDF experience with Monte Carlo*, CHEP'07, IOP Publishing Journal of Physics: Conference Series 119 (2008) 072025
4. Moretti C, Sfiligoi I, Thain D, *Transparently Distributing CDF Software with Parrot*, Feb 13-17 2006 Presented at CHEP06, Mumbai, India, 26.
5. Cooperative Computing Tools, <http://www.cctools.org>