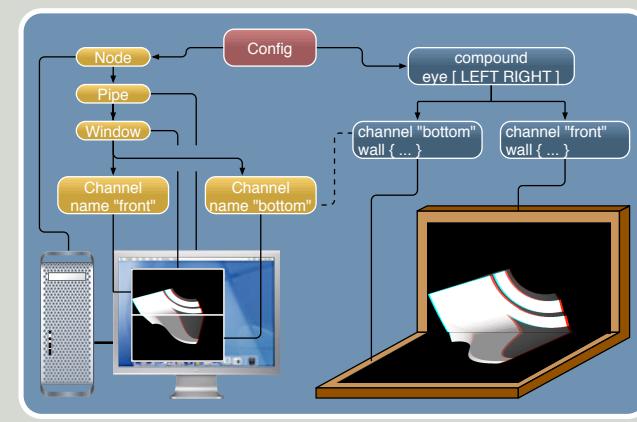


## Scalability

Applications build with Equalizer are scalable, using multiple graphics cards, processors and computers to render a single or multiple views. The application is rendering in parallel at the optimal performance, in contrast to other solutions which operate on the OpenGL command stream.

Equalizer provides a comprehensive set of algorithms to parallelize and load-balance the rendering of demanding data sets. The rendering is distributed across all available resources (decomposition) and the results are assembled on the final view (recomposition).

Equalizer supports screen-space, database, time-multiplex, pixel and stereo task distribution.



[www.equalizergraphics.com](http://www.equalizergraphics.com)  
[info@equalizergraphics.com](mailto:info@equalizergraphics.com)



Equalizer is a product of Eyescale Software GmbH.

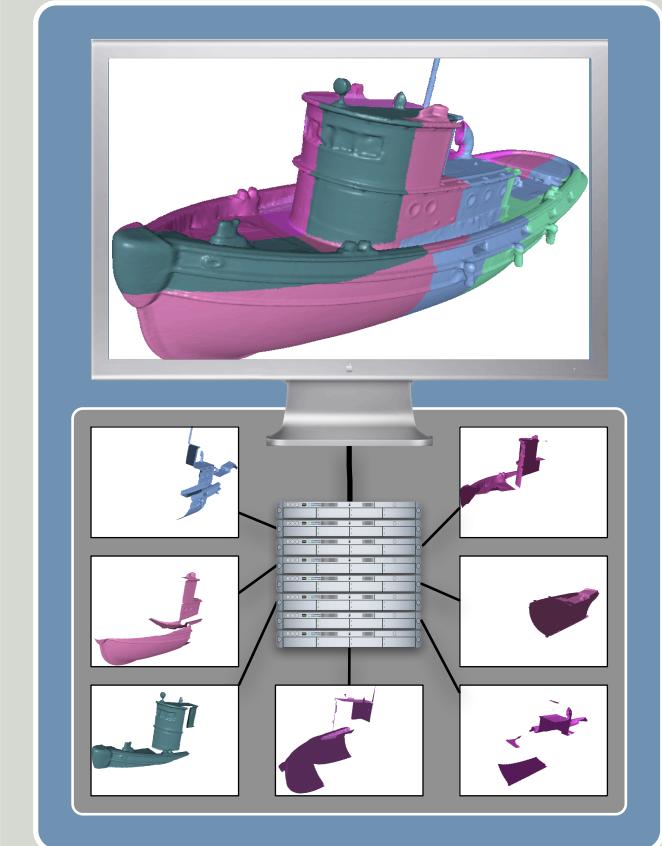
[www.eyescale.ch](http://www.eyescale.ch)  
[info@eyescale.ch](mailto:info@eyescale.ch)

## Contributors:



# Equalizer

## Parallel Rendering



Equalizer is the standard middleware to create parallel OpenGL®-based applications. It enables applications to benefit from multiple graphics cards, processors and computers to scale the rendering performance, visual quality and display size. An Equalizer-based application runs unmodified on any visualization system, from a simple workstation to large scale graphics clusters, multi-GPU workstations and Virtual Reality installations.

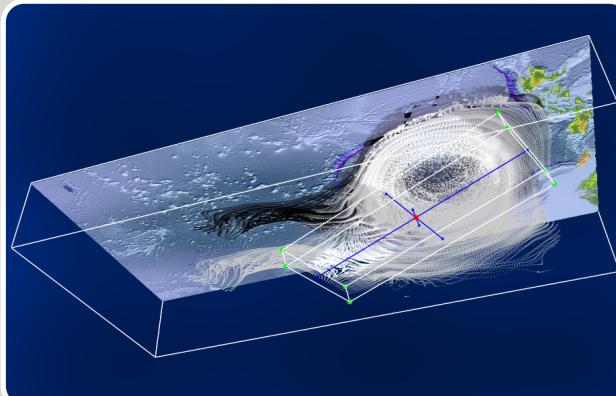
## **Benefits**

**Parallel Rendering Know-How:** Equalizer contains the essence of 10+ years of experience in parallel and scalable rendering, easily integrated into your application.

**Fast Path for Scalable OpenGL Applications:** Equalizer provides the natural parallel execution model to exploit the parallelism of multicore, multi-GPU workstations and graphics clusters.

**Feature-Rich Framework:** Equalizer contains state-of-the-art scalable rendering algorithms, and its open development model ensures constant improvement. Equalizer applications are flexible and deployable in many rapidly changing environments.

-based Flow Visualization (Courtesy University of Siegen, 2008)



## Highlights

Cross-Platform: Linux, Windows XP and Mac OS X

C++ API with a modern, high-performance design

GPL license: free commercial and open source use

**Portability:** 32 and 64 bit, little- and big-endian  
**Interconnects:** Ethernet, Infiniband

Cross-Platform: Linux, Windows XP and Mac OS X

## Features

**Runtime Configurability:** An Equalizer application can run optimally on any configuration of hardware, from laptops to large scale visualization clusters. The application is configured at runtime using a simple configuration file.

**Distributed Execution:** Equalizer applications can be written to support cluster-based execution. The task of distributing the application data is simplified through a solution-based distributed objects.

**Support for Immersive Environments:** Equalizer  
supports both active and passive stereo rendering, as  
well as head tracking used in Virtual Reality installs.

In application written with `Equalizer` can be deployed in many different ways, without any modification or compilation, for example:

## Use Cases

**Multi-GPU Workstations** are an affordable way to scale the rendering performance and display size. And graphics cards by running multiple renderers and qualizers applications can exploit multiple processors reads on a single computer.

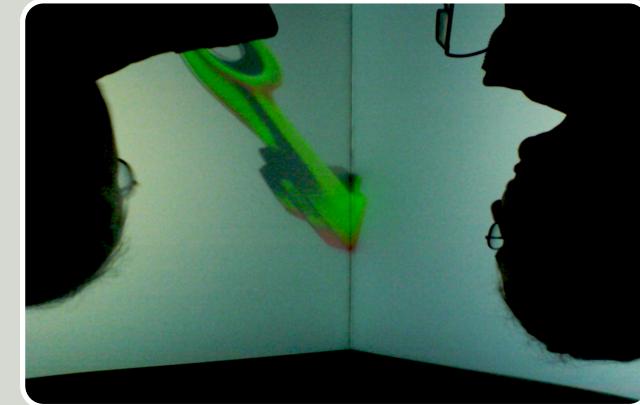
**Renderable Graphics** aggregate multiple graphics cards, processors and potentially computers to render more data faster.

*Virtual reality installations use passive or active stereo rendering with head tracking, both supported by Equalizer.*

One instance of the application clusters' renderers' code is executed locally on each node.

**Equalizer** can be deployed in application written with `Quasar` in many different ways, without any modification or compilation, for example:

A virtual reality installation using passive stereo and head tracking



The application data structures, scene graph and rendering logics are unchanged and keep using the OpenGL hardware for maximum performance and compatibility.

The Equalizer API is a GLUT-like, minimally invasive parallel programming interface. Most of the application is undifferentiated, only the application's rendering code is plugged into the Equalizer framework for parallel execution.

Dequazizer distills the expertise of three generations of parallel rendering APIs, delivering multi-view, cluster-based applications to the fastest porting path to create parallel, scalable OpenGL-based applications.

Parallel Programming Interface

A display wall running an Equalizer-based terrain rendering application

