Ray Casting

Hao Li

Applied Geometry Group
ETH Zurich

Ray casting is not always ray tracing

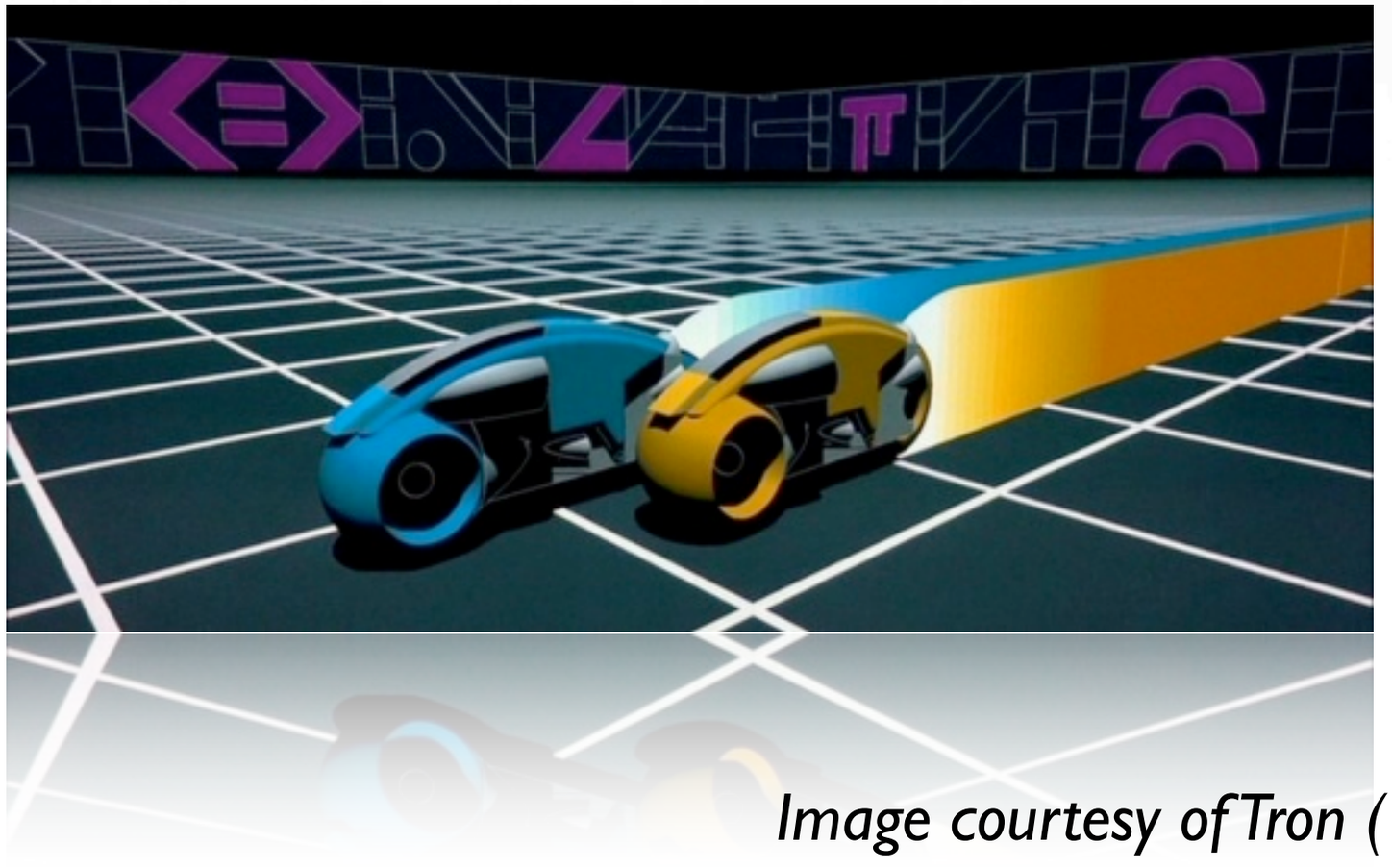
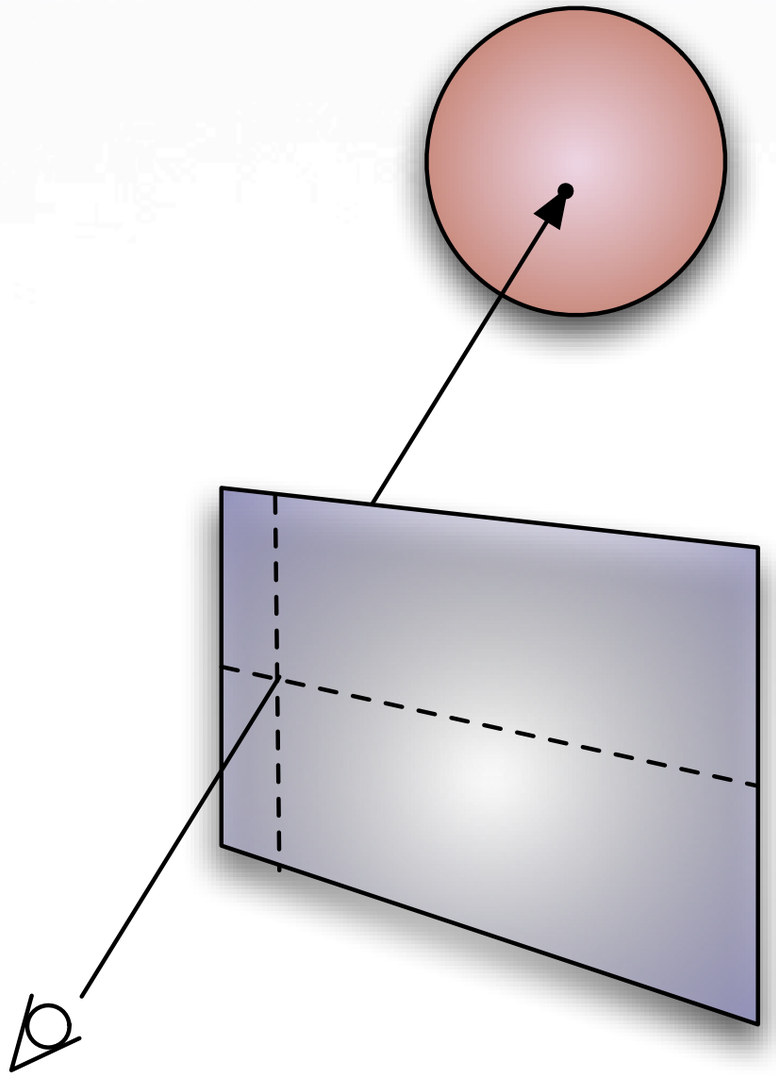


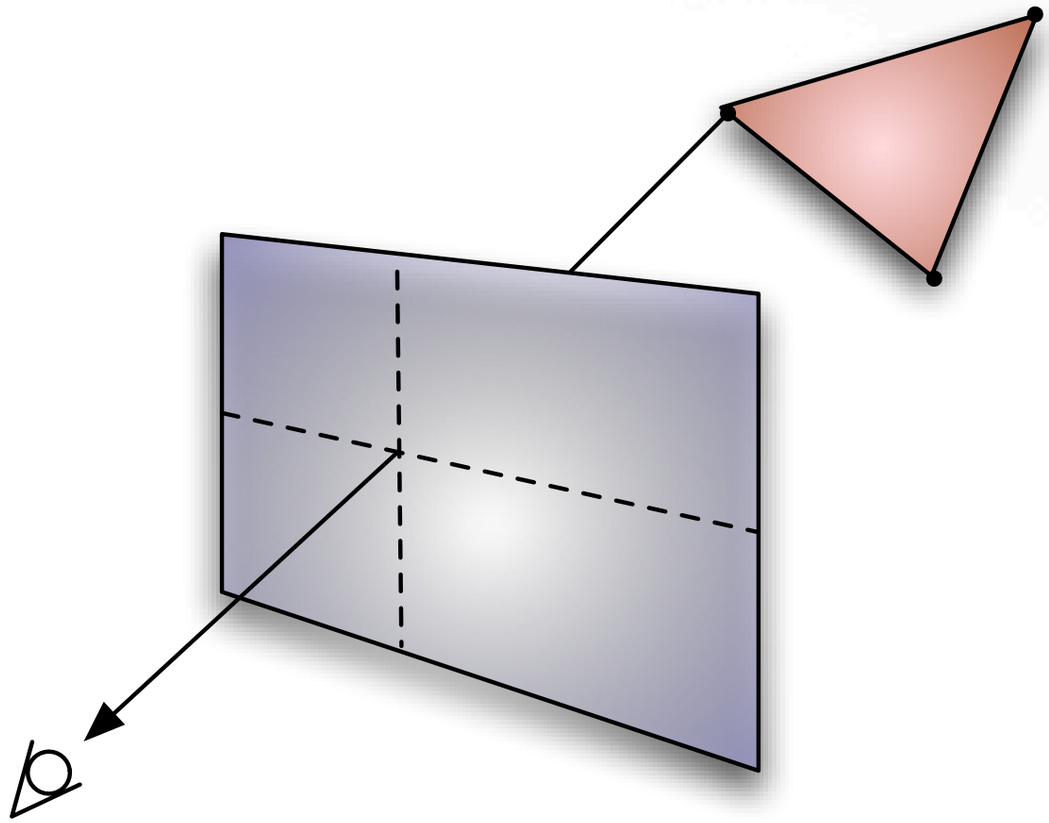
Image courtesy of Tron (1982)

Ray casting is not always ray tracing

Image courtesy of Wolfenstein3D (1992)

Ray casting is not rasterization





Exercise 1

Tasks

Tasks

- Understanding the framework

Tasks

- Understanding the framework
- Ray generation

Tasks

- Understanding the framework
- Ray generation
- Ray-surface intersection

Tasks

- Understanding the framework
- Ray generation
- Ray-surface intersection
  - plane, sphere, triangles

Tasks

- Understanding the framework
- Ray generation
- Ray-surface intersection
  - plane, sphere, triangles
- Constant shading

Tasks

Basic ray tracer

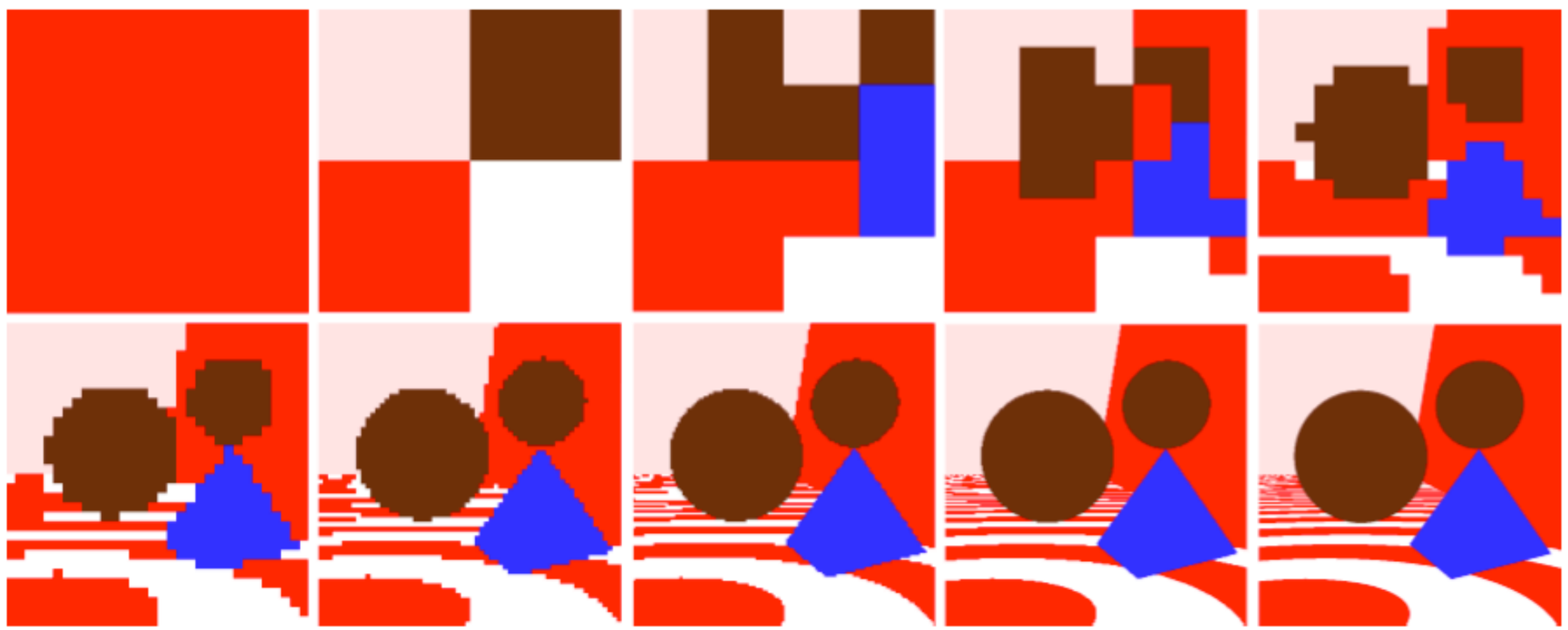
- Understanding the framework
- Ray generation
- Ray-surface intersection
  - plane, sphere, triangles
- Constant shading

Constant shading can be cool

Image courtesy of Apple Inc.

Optional Tasks

Progressive sampling



Optional Tasks

Quadrics (quadratic surface)

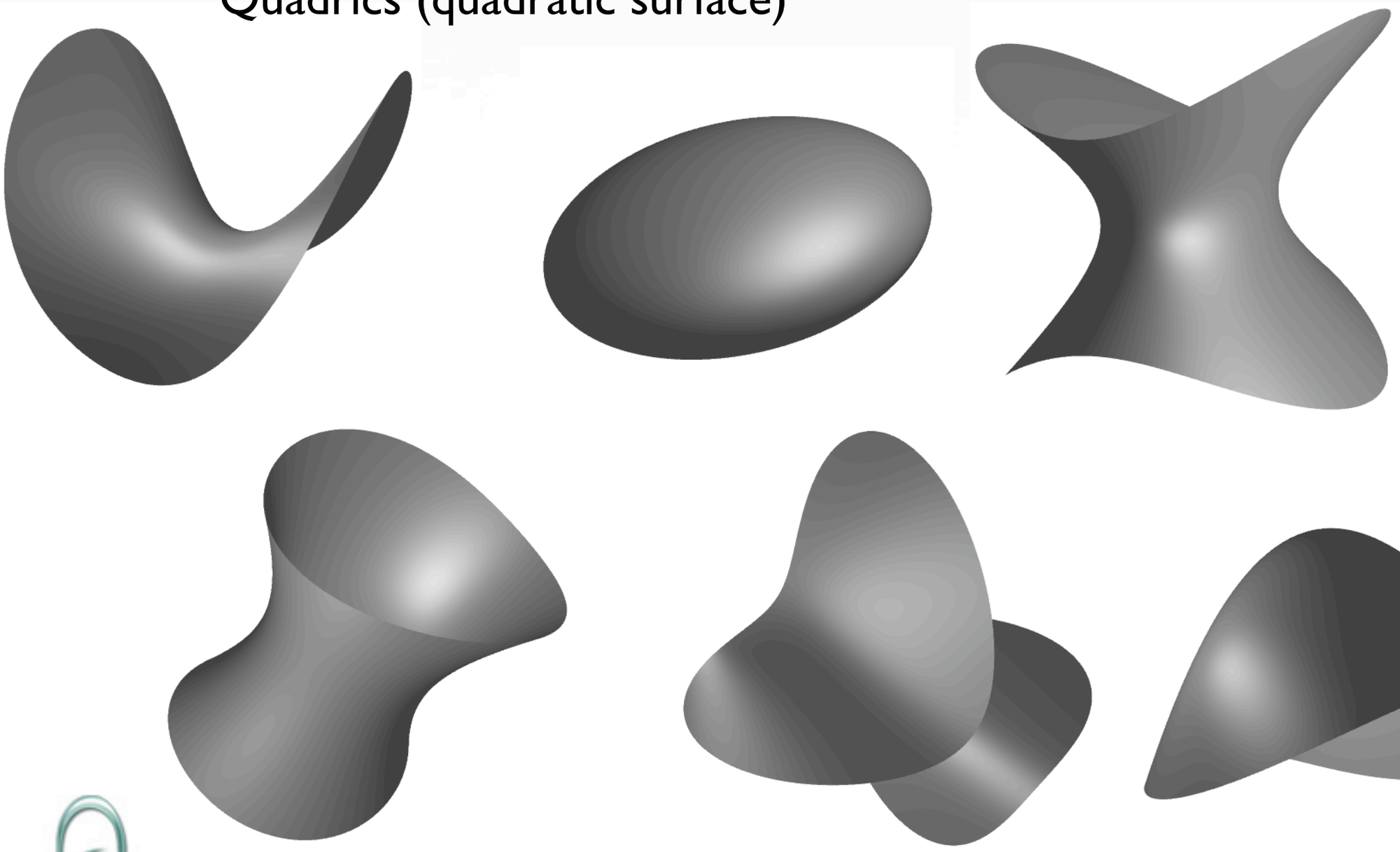


Image courtesy of DGP Toronto

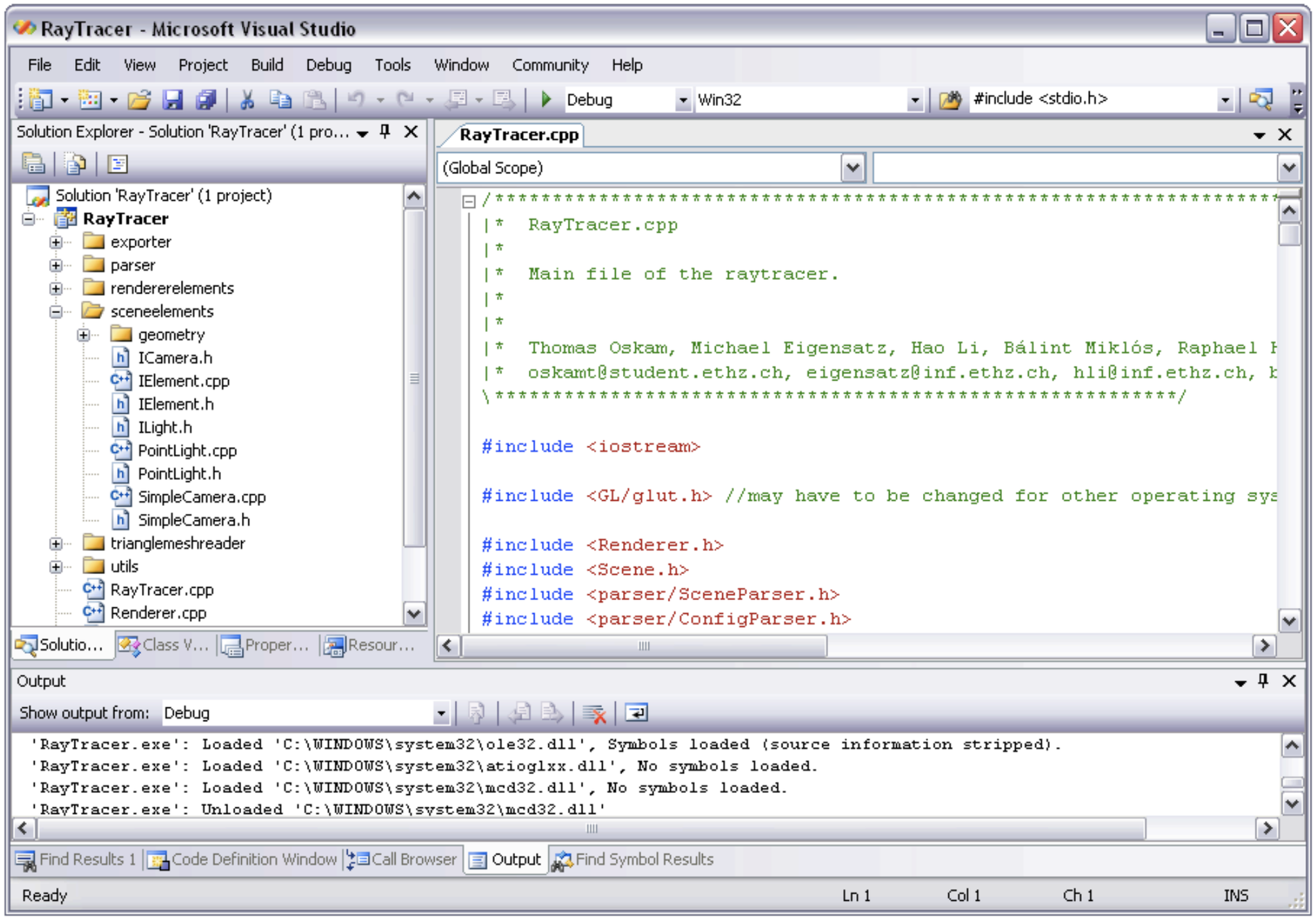
Optional Tasks

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0$$

Quadrics

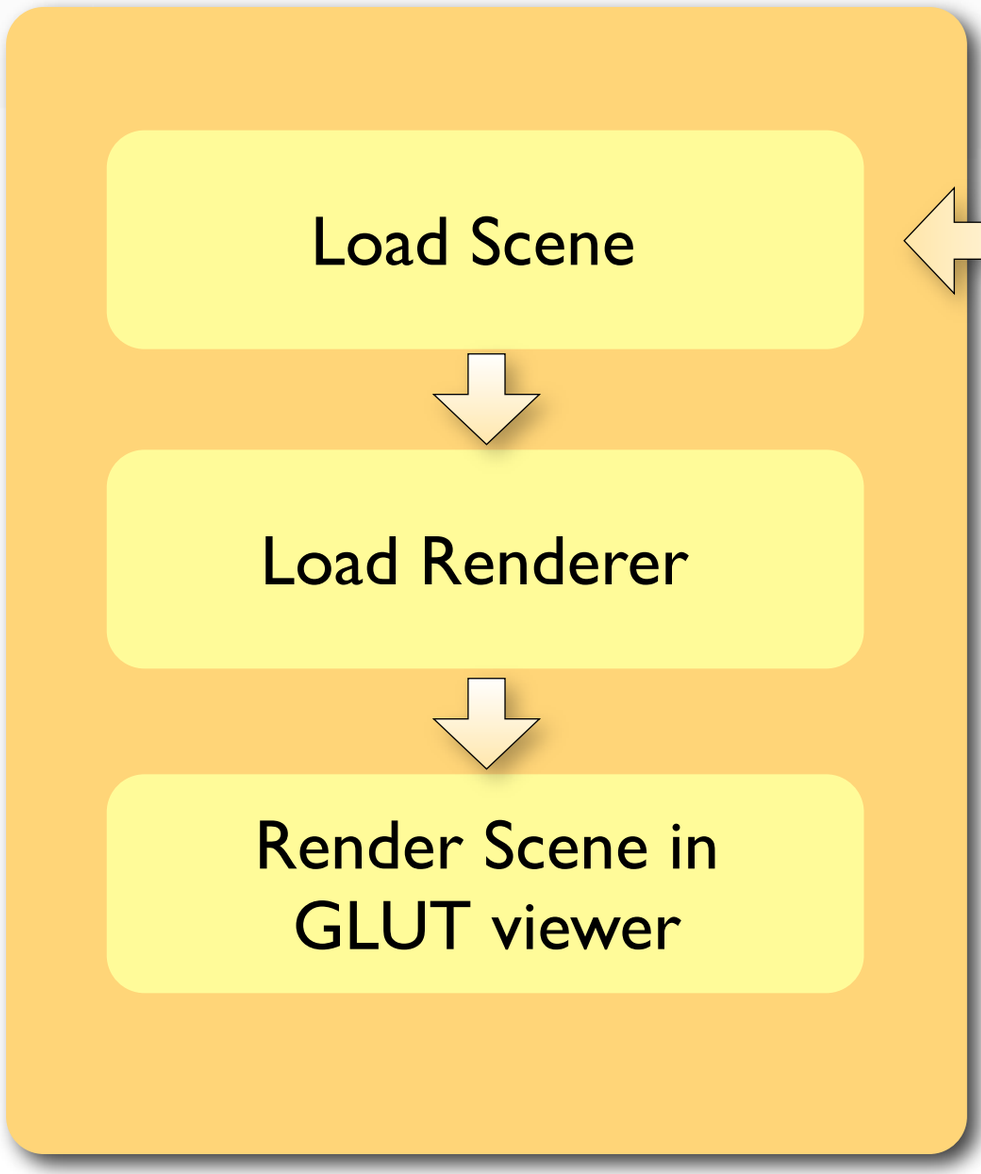
The Framework

Ray Tracing Framework



Ray Tracing Framework

Data flow



via SceneParser.cpp

SceneDescription.xml

contains:

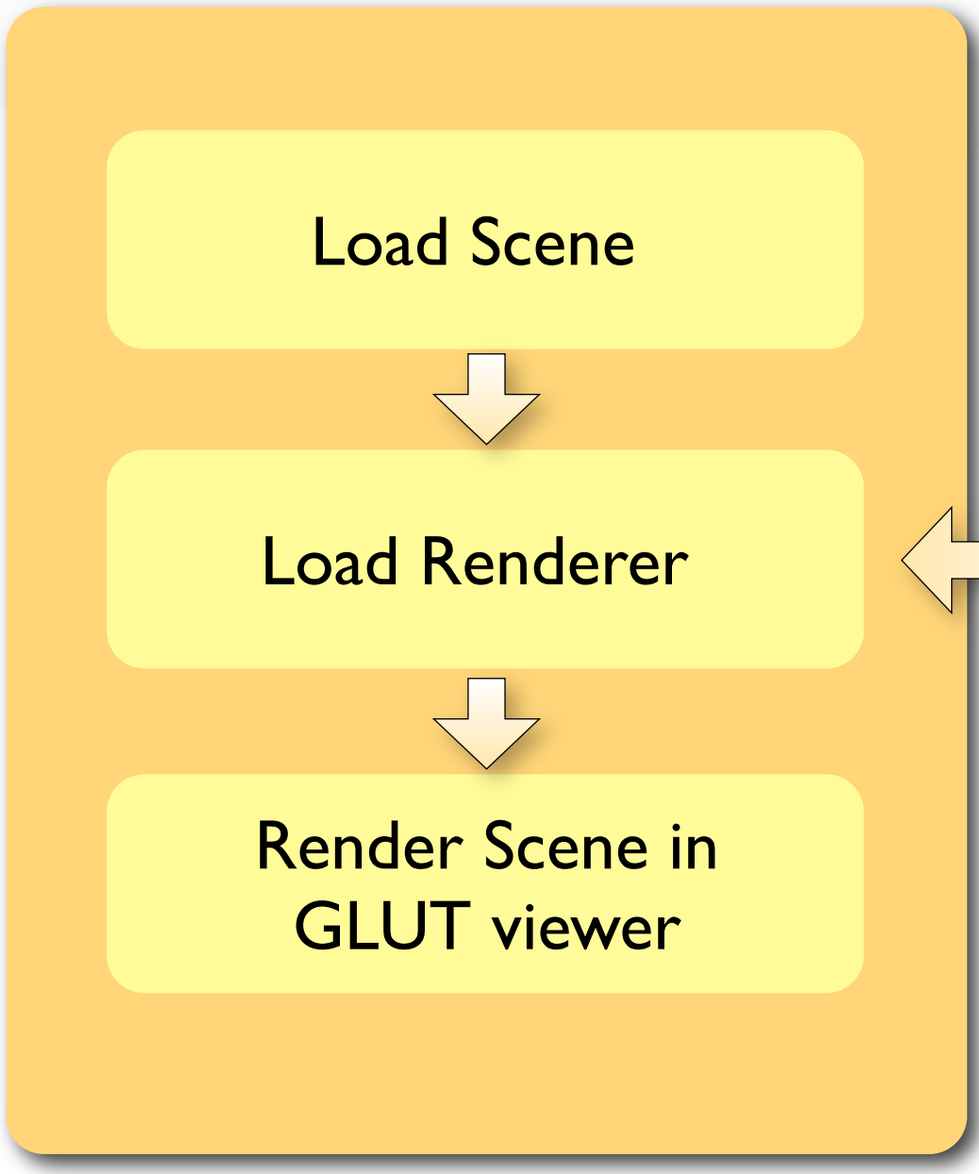
1 camera

n elements

m light sources

Ray Tracing Framework

Data flow



via ConfigParser.cpp

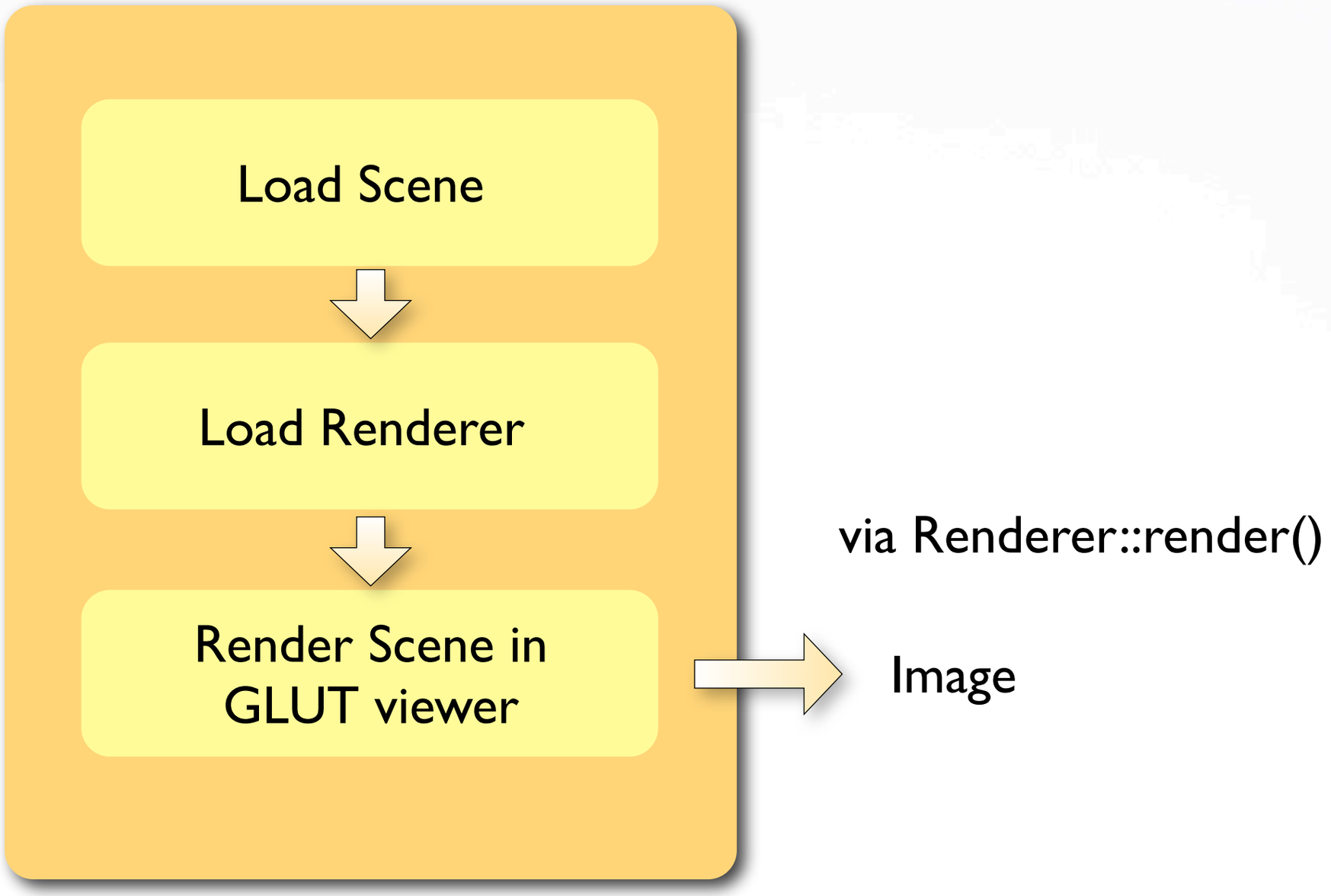
Config.xml

contains:

Sampler

Ray Tracing Framework

Data flow



Rendering Pipeline

Renderer

Sampler

Shader

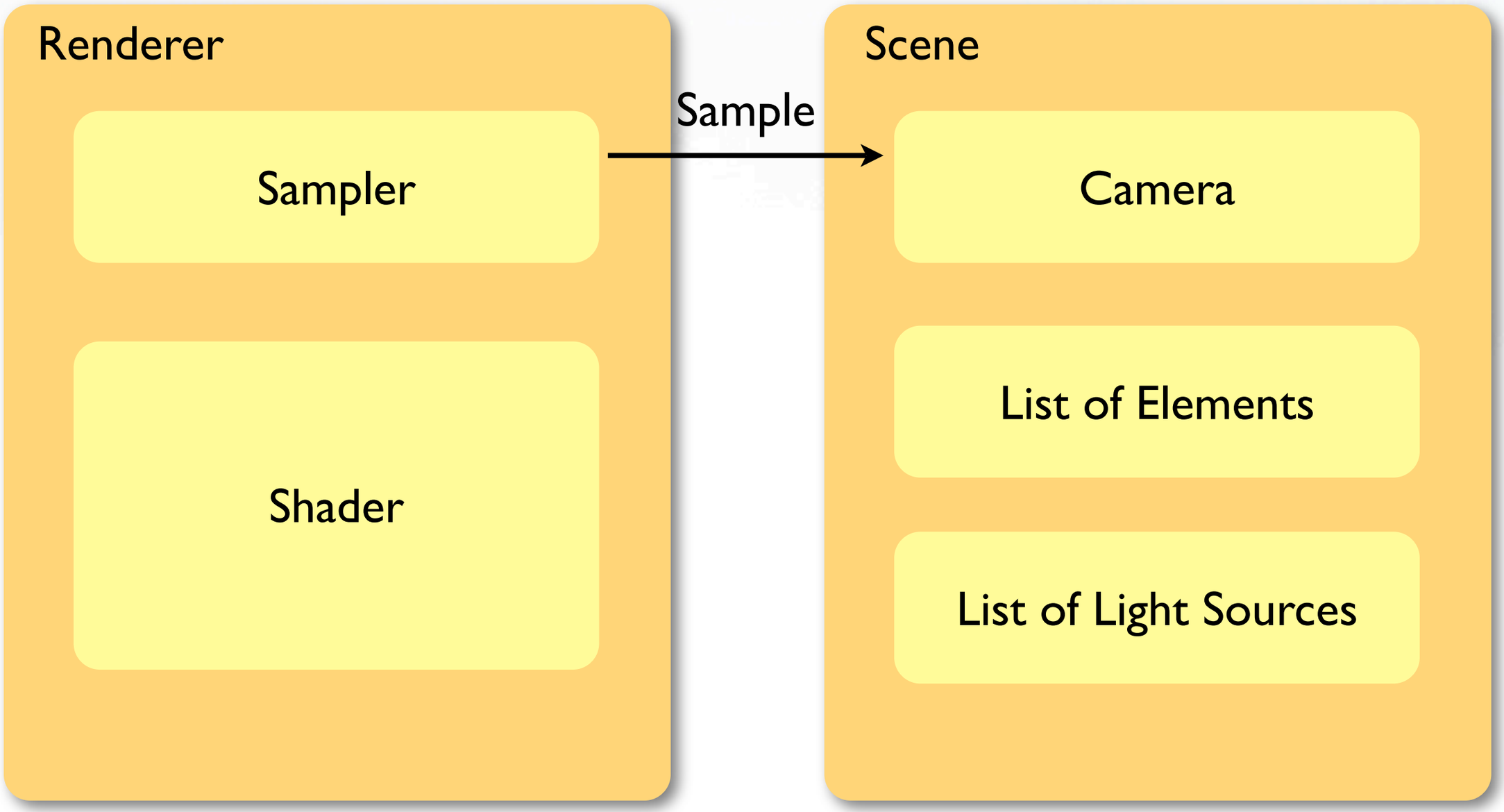
Scene

Camera

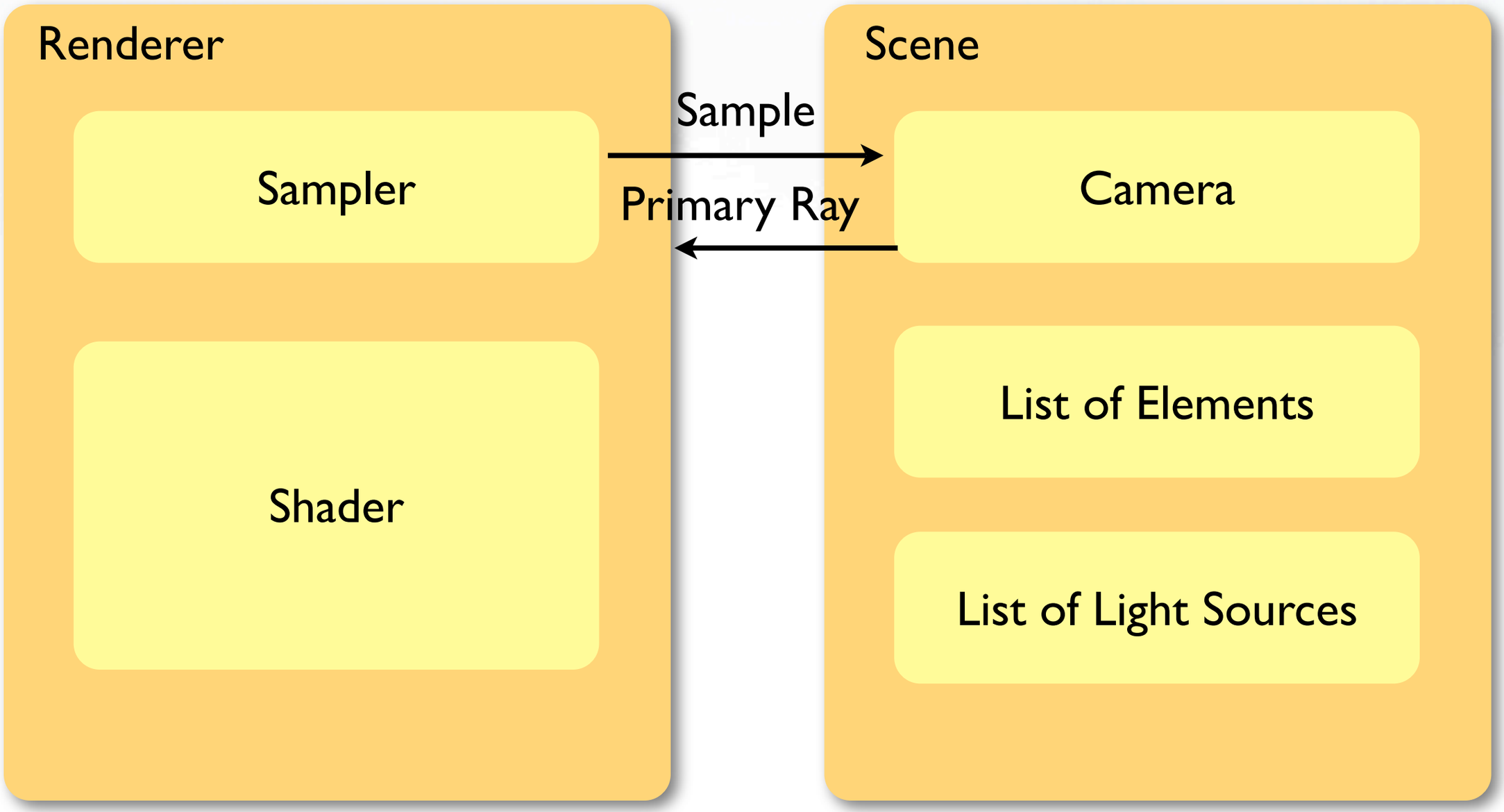
List of Elements

List of Light Sources

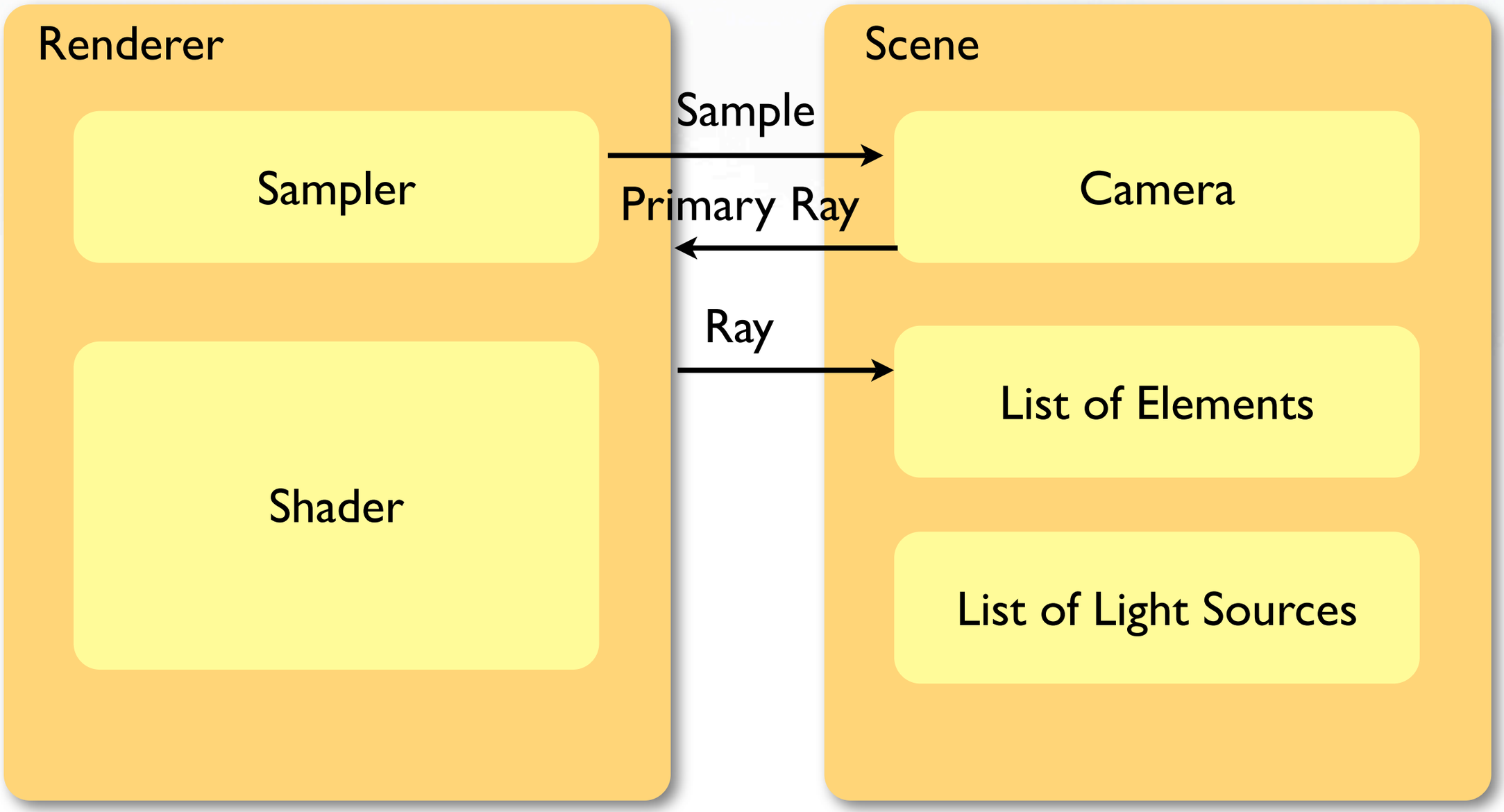
Rendering Pipeline



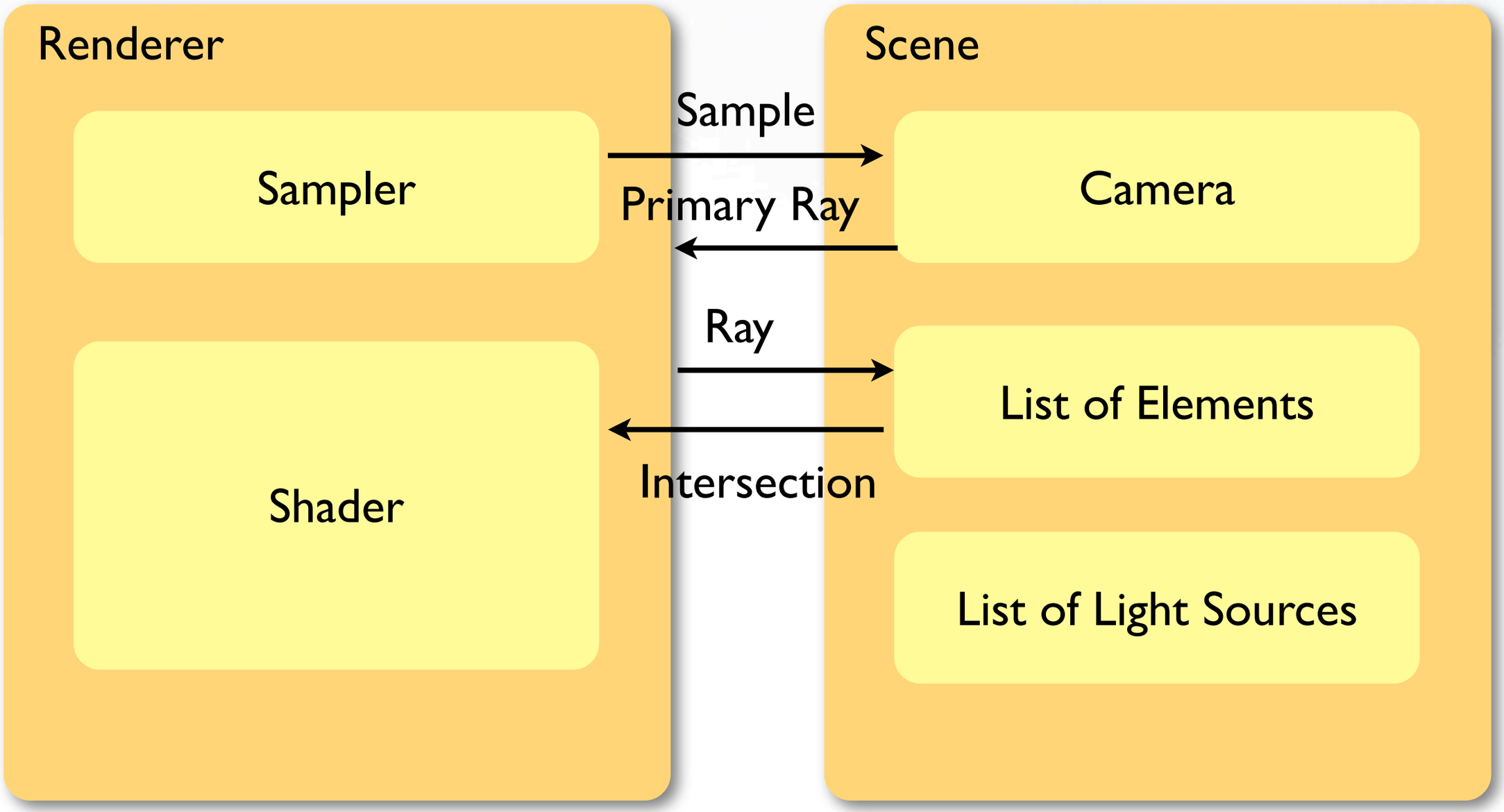
Rendering Pipeline



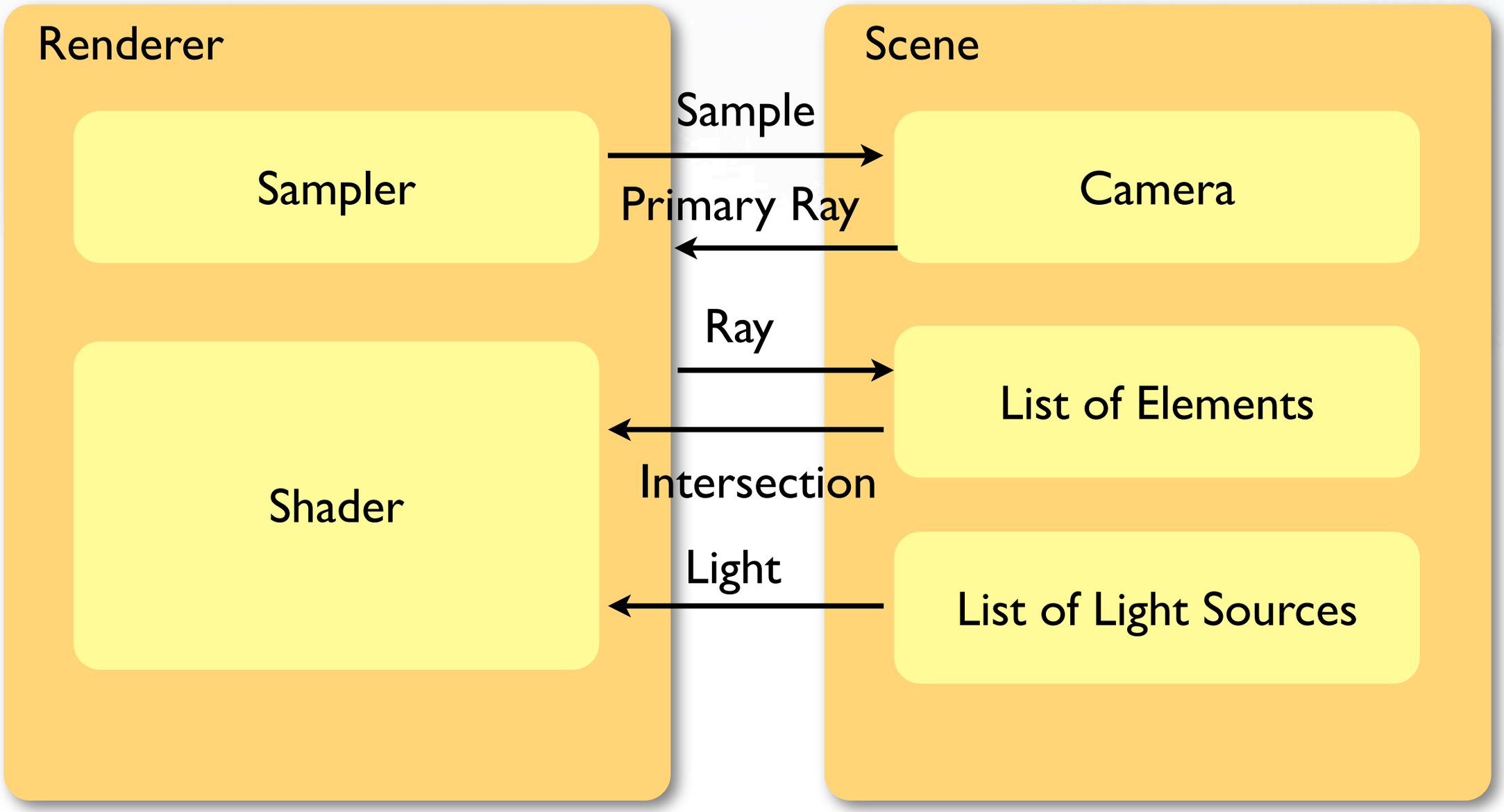
Rendering Pipeline



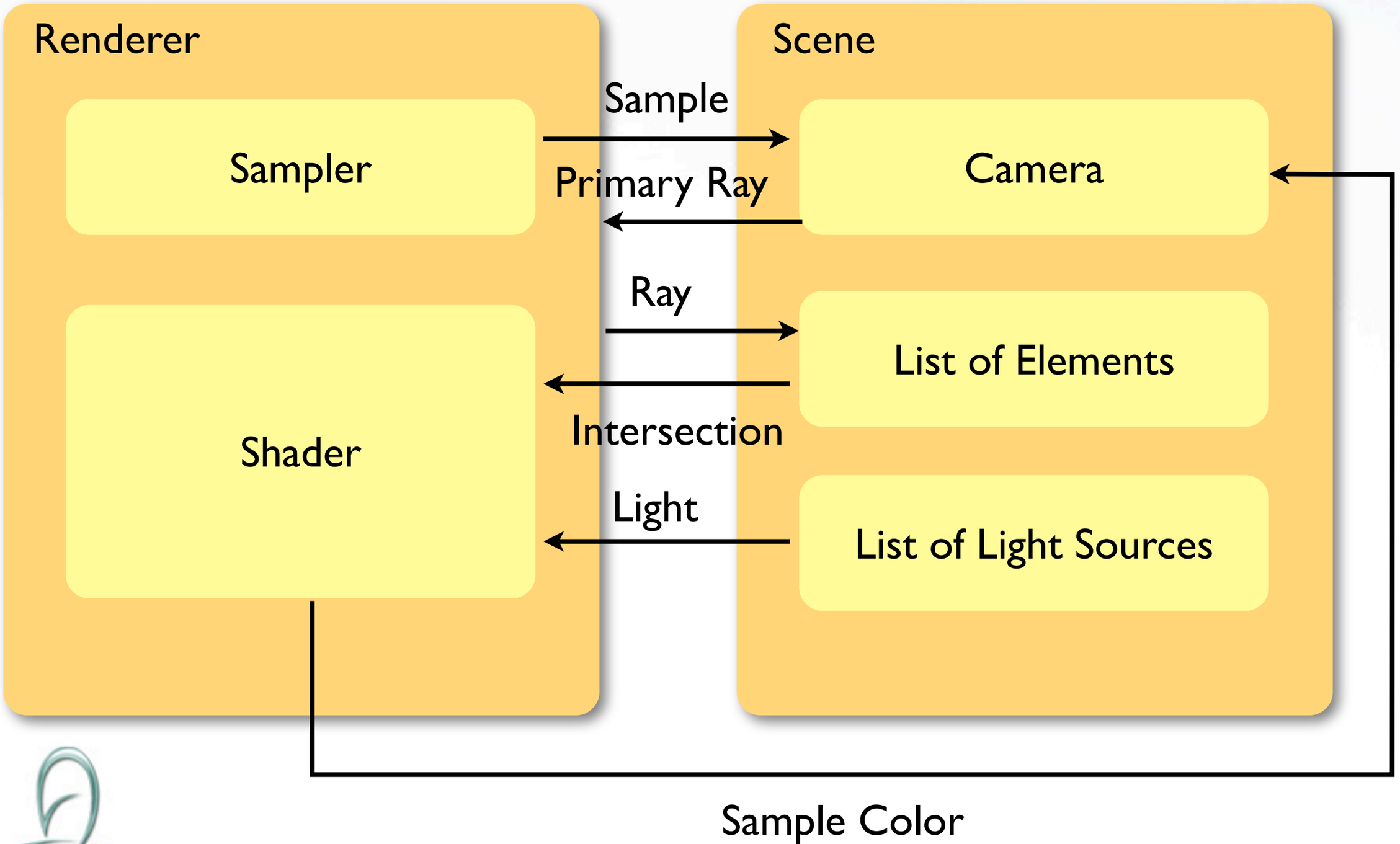
Rendering Pipeline



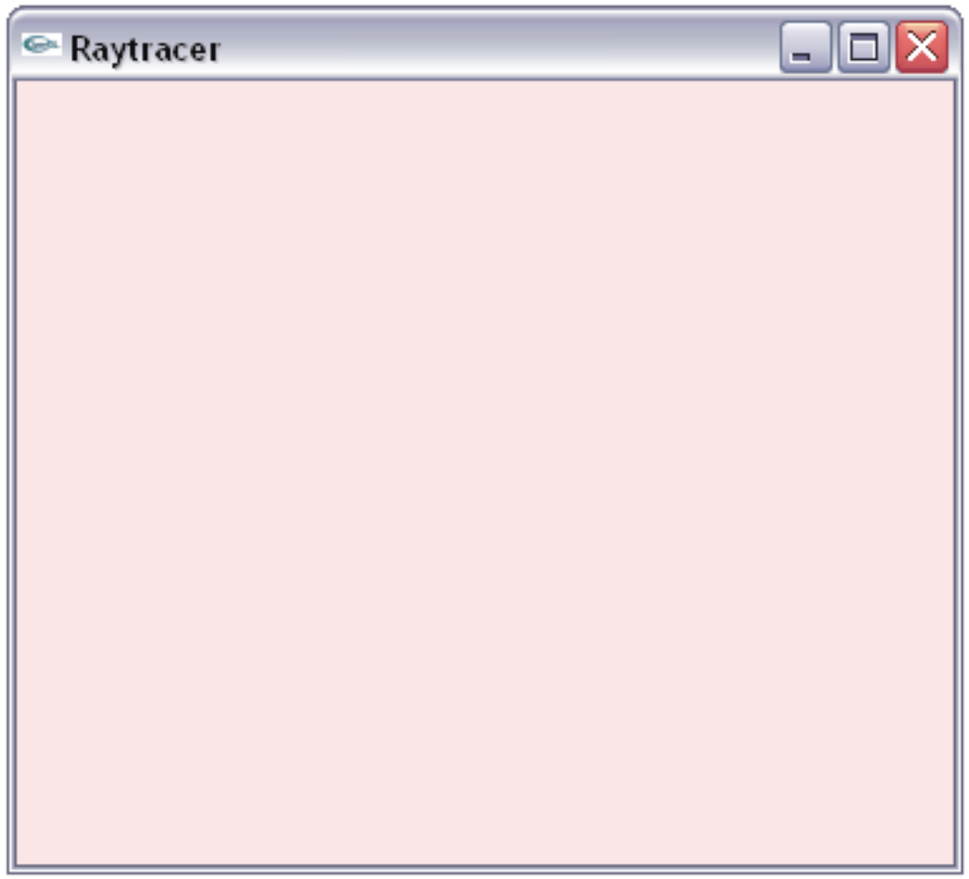
Rendering Pipeline



Rendering Pipeline



First Build and Run



Orientation

Orientation

- main() is in RayTracer.cpp

Orientation

- main() is in RayTracer.cpp
- Rendering function is Renderer::render()

Orientation

- main() is in RayTracer.cpp
- Rendering function is Renderer::render()
- Utility classes (vector, ray, etc...) in utils/

Orientation

- main() is in RayTracer.cpp
- Rendering function is Renderer::render()
- Utility classes (vector, ray, etc...) in utils/
- Geometric elements in sceneelements/geometry/

Orientation

- main() is in RayTracer.cpp
- Rendering function is Renderer::render()
- Utility classes (vector, ray, etc...) in utils/
- Geometric elements in sceneelements/geometry/
- Sampler, shader and other classes in rendererelements/

What to hand in?

What to hand in?

- Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)

What to hand in?

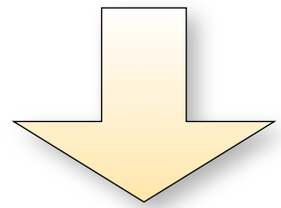
- Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)
- Custom scene description files

What to hand in?

- Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)
- Custom scene description files
- Html readme file (follow template)

What to hand in?

- Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)
- Custom scene description files
- Html readme file (follow template)



introcg@inf.ethz.ch

Please, only submit
VS 2005 projects

hao@inf.ethz.ch

hao@inf.ethz.ch

?