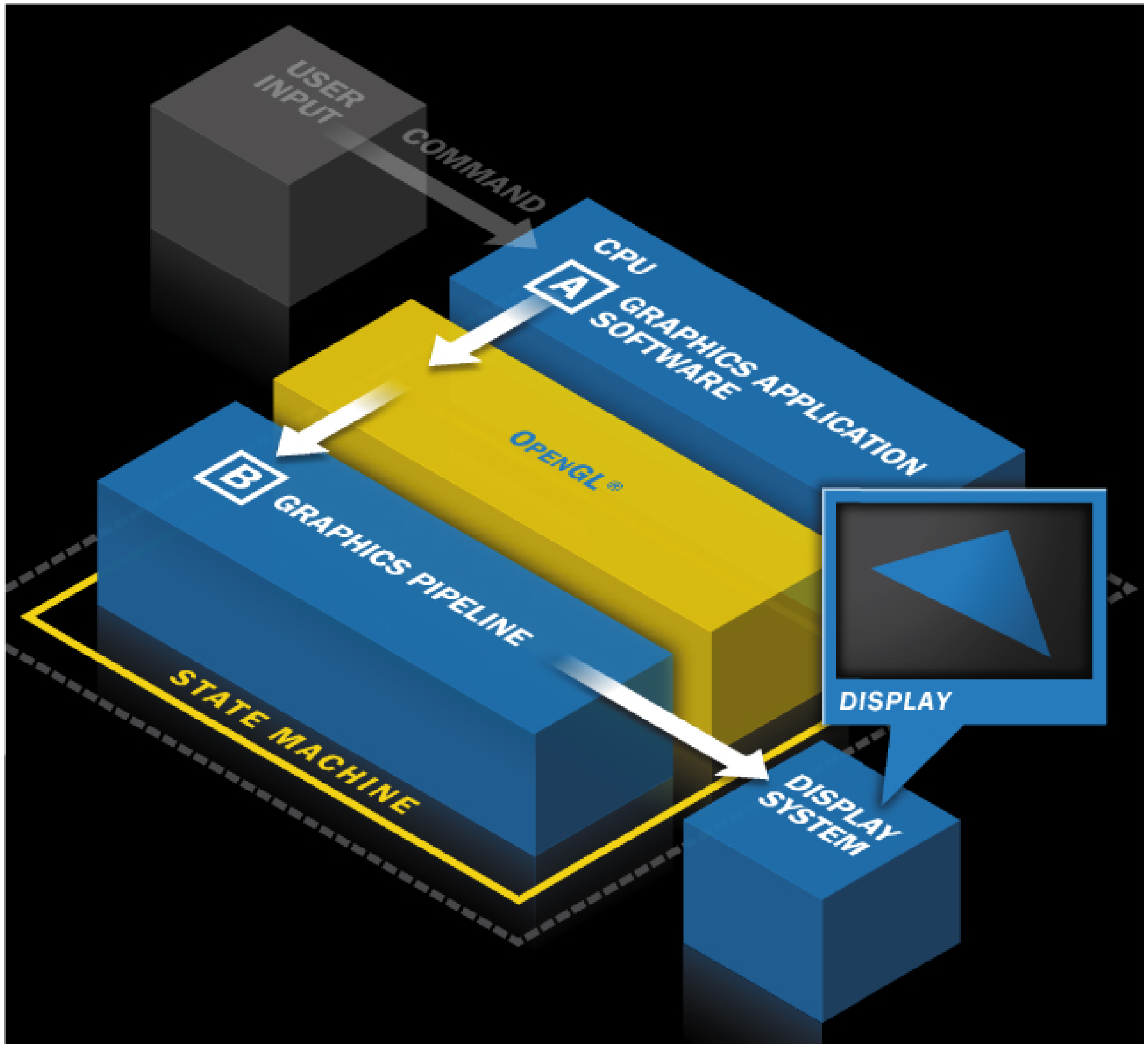
’327 Patent

SGI’s OpenGL® Architecture

A Graphics Pipeline **CAN EXECUTE COMMANDS** *generated by a wide variety of graphics application programs*



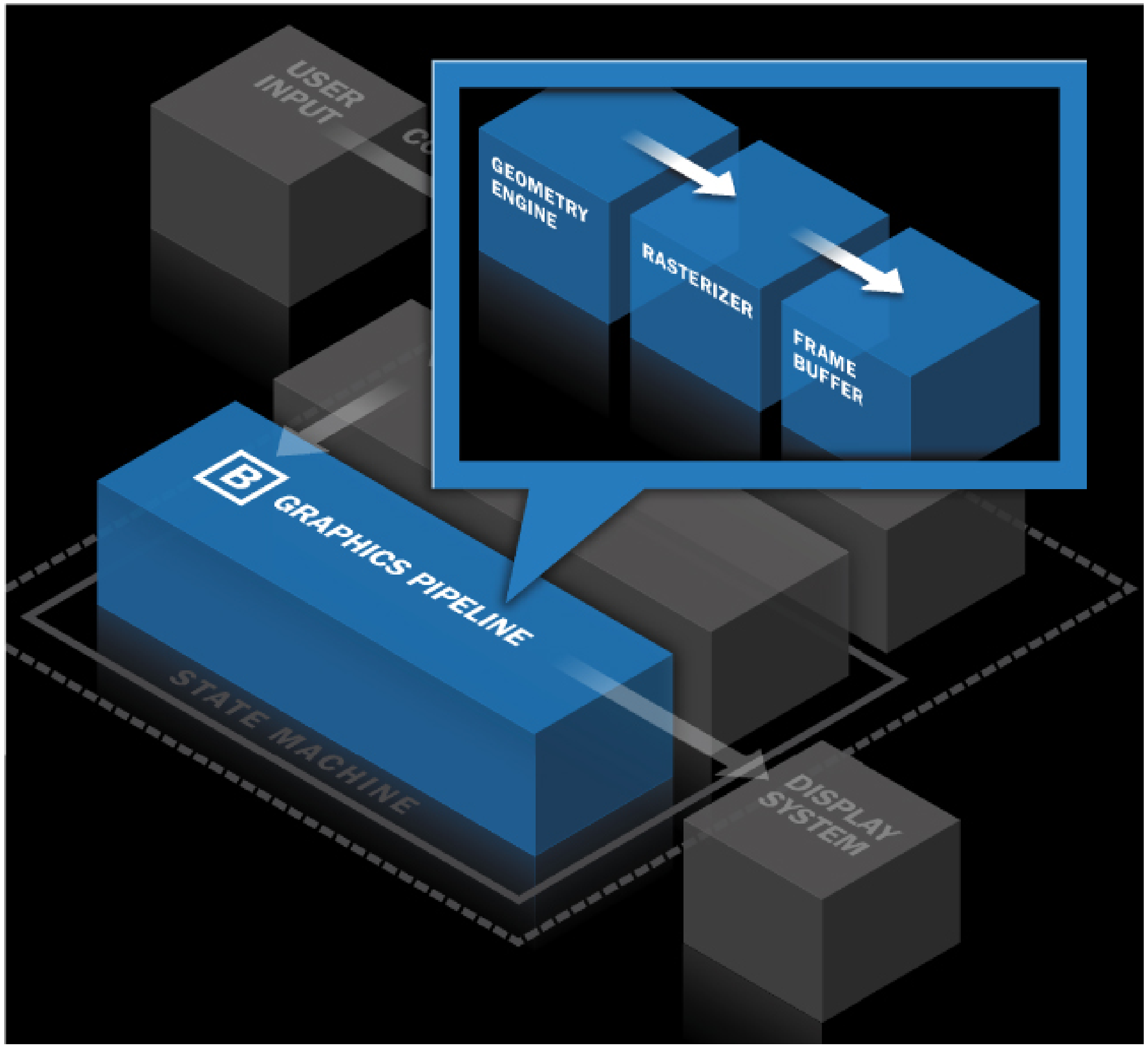
’327 Patent

SGI’s OpenGL® Architecture

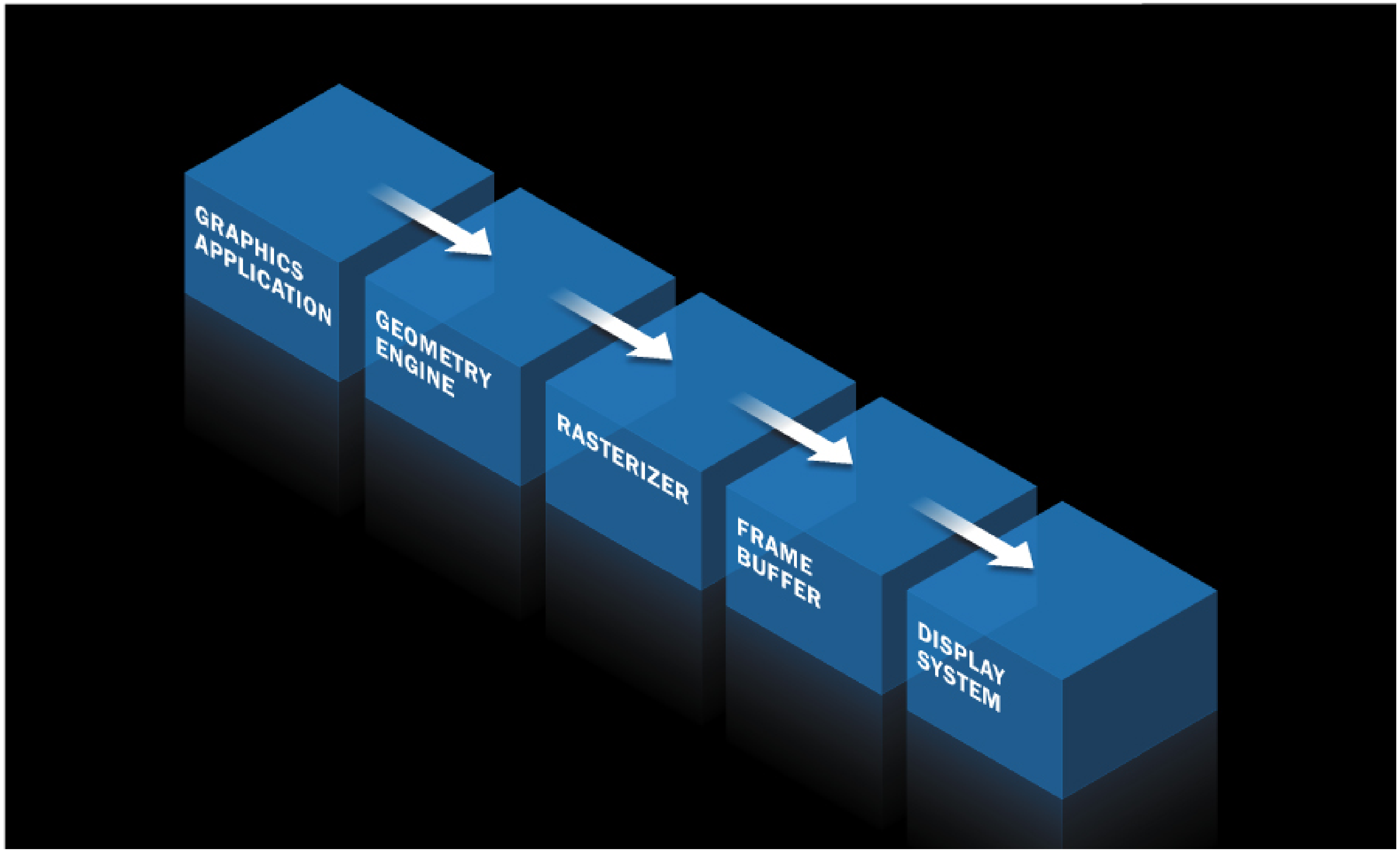
Graphics Pipeline B

is comprised of a

- ***Geometry Engine***
- ***Rasterizer***
- ***Frame Buffer***



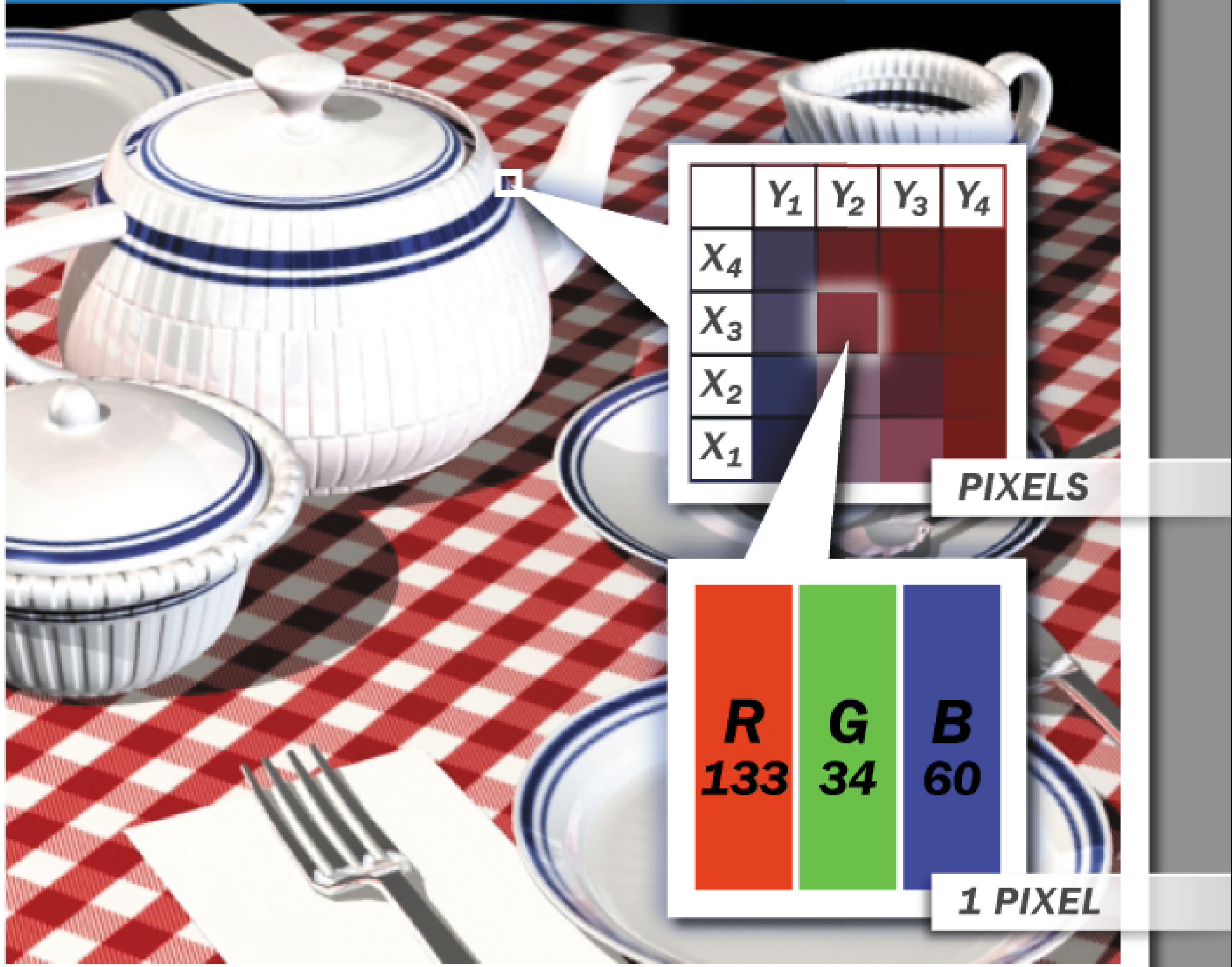
General Depiction of a Graphics Pipeline



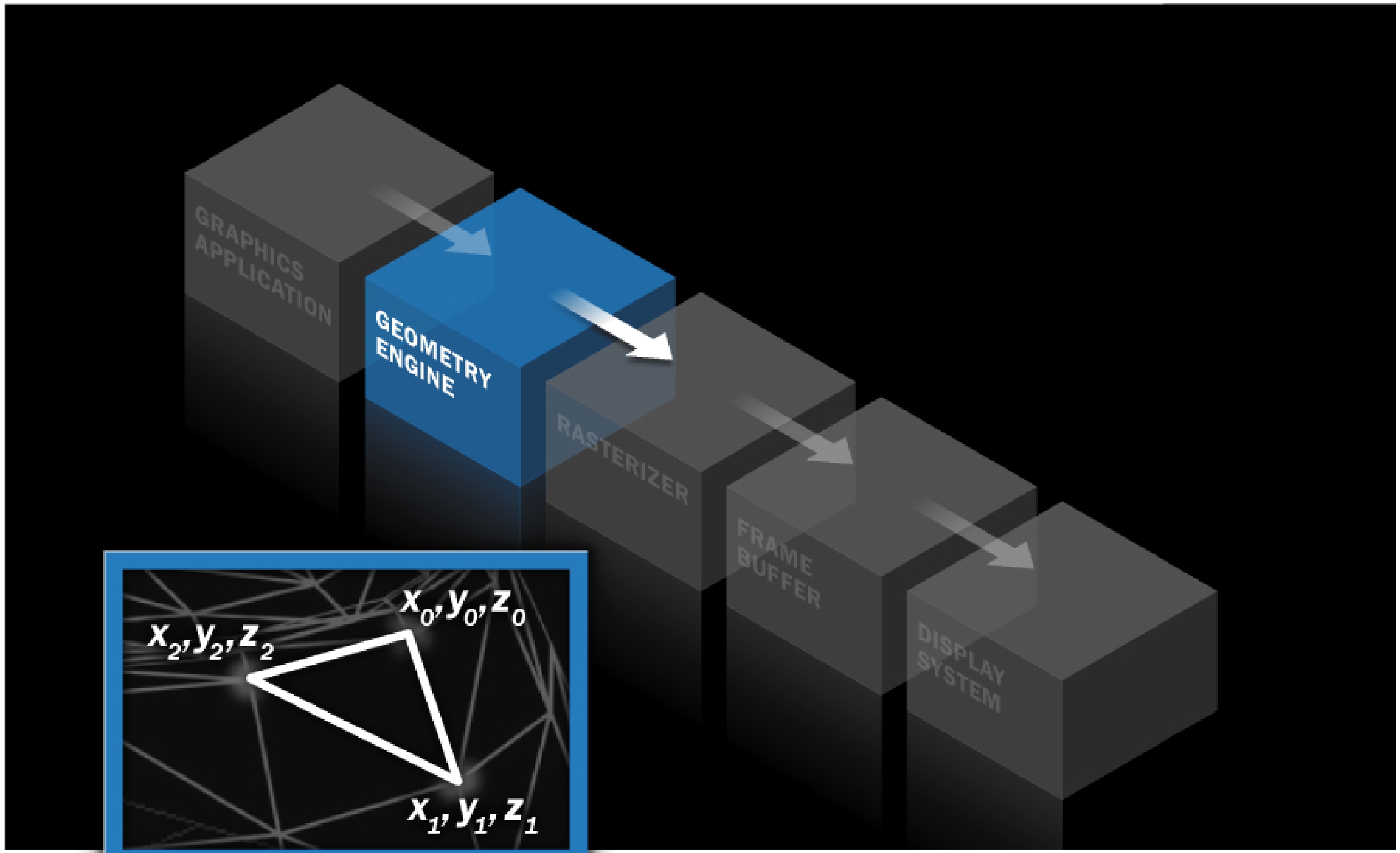
’327 Patent

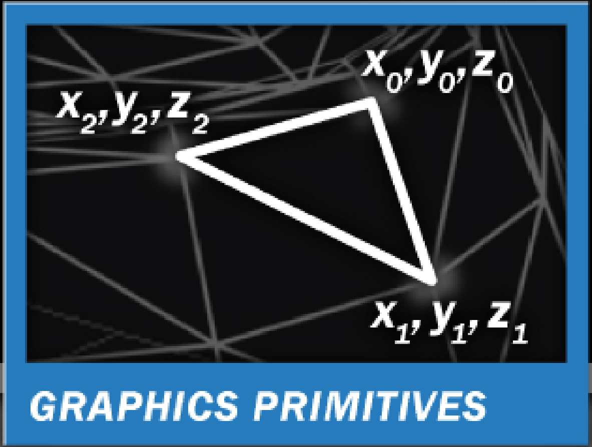
Pixels or Picture Elements

RENDERED IMAGE



General Depiction of a Graphics Pipeline

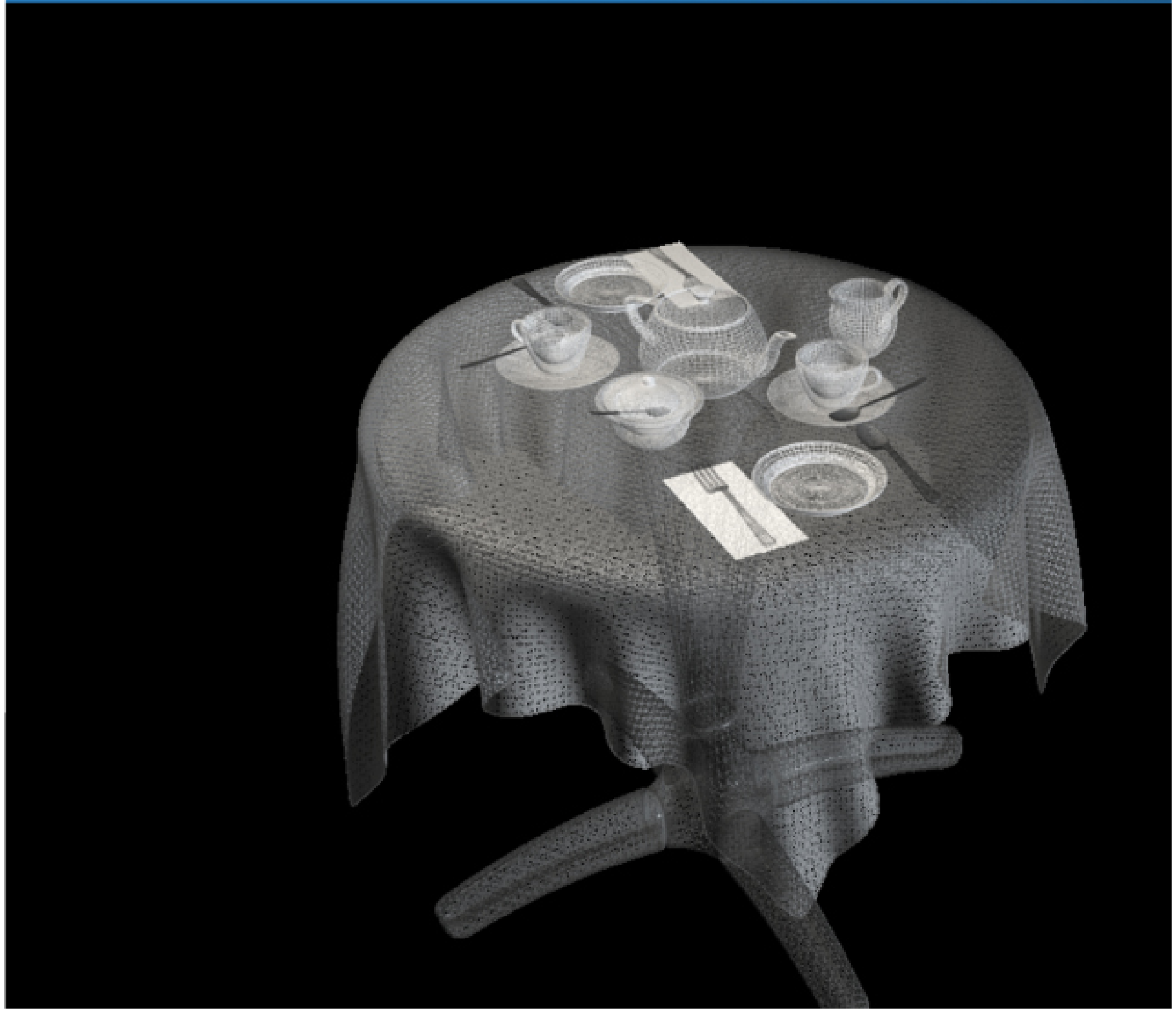




’327 Patent

3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

3D MODEL

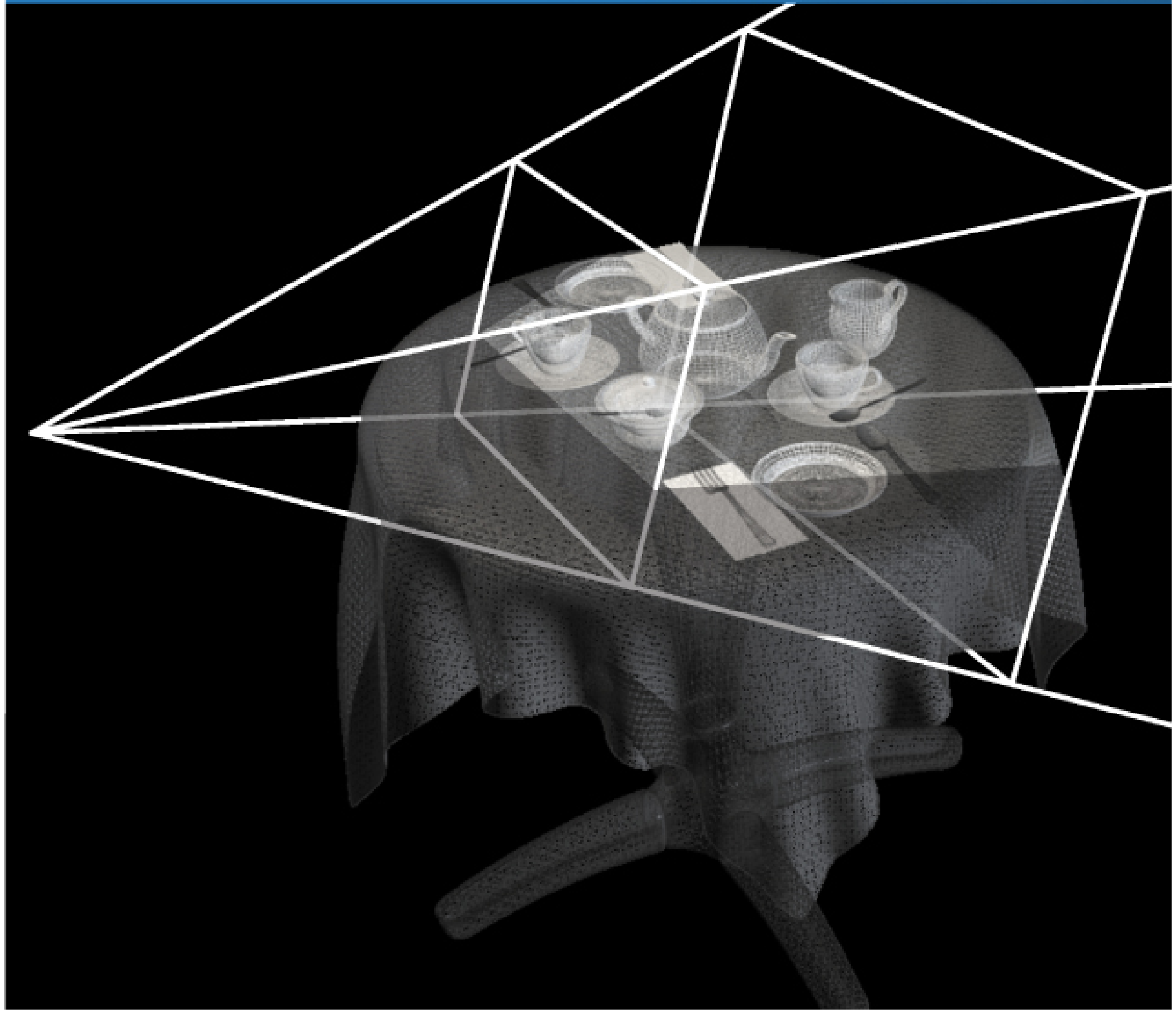


’327 Patent

3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

User chooses view

3D MODEL

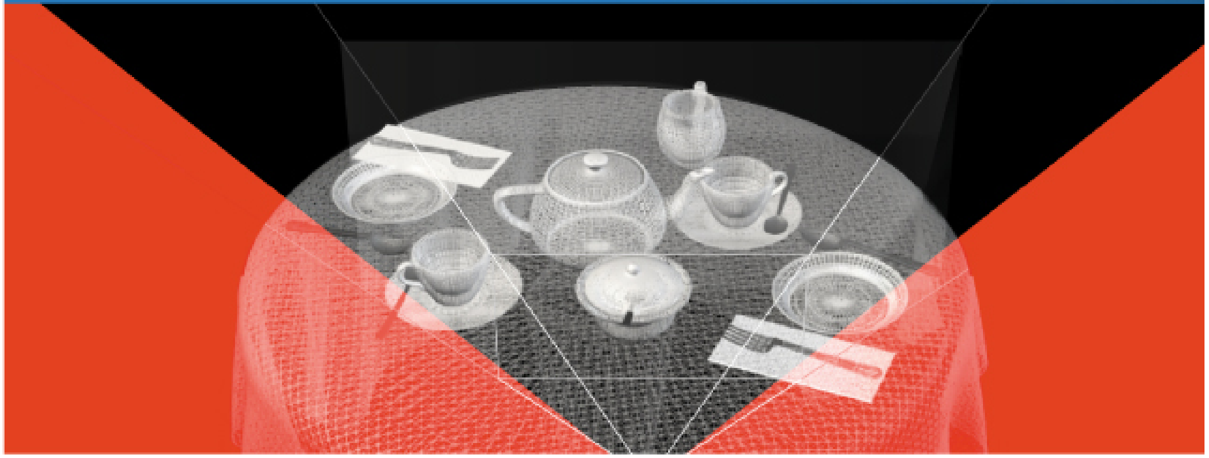


’327 Patent

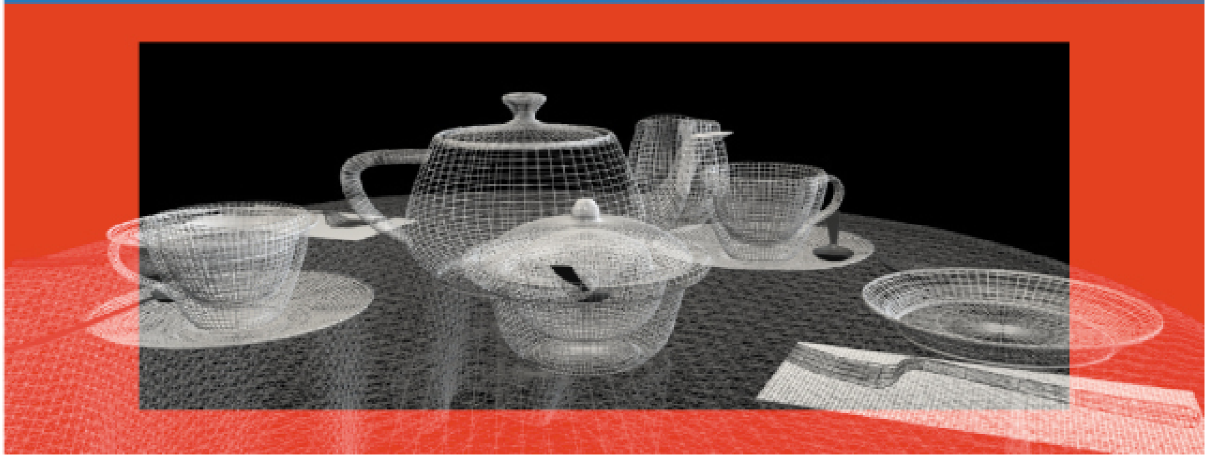
3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

Objects outside the field of view are not displayed

3D MODEL – “TRANSFORM” VIEW



3D MODEL – USER VIEW

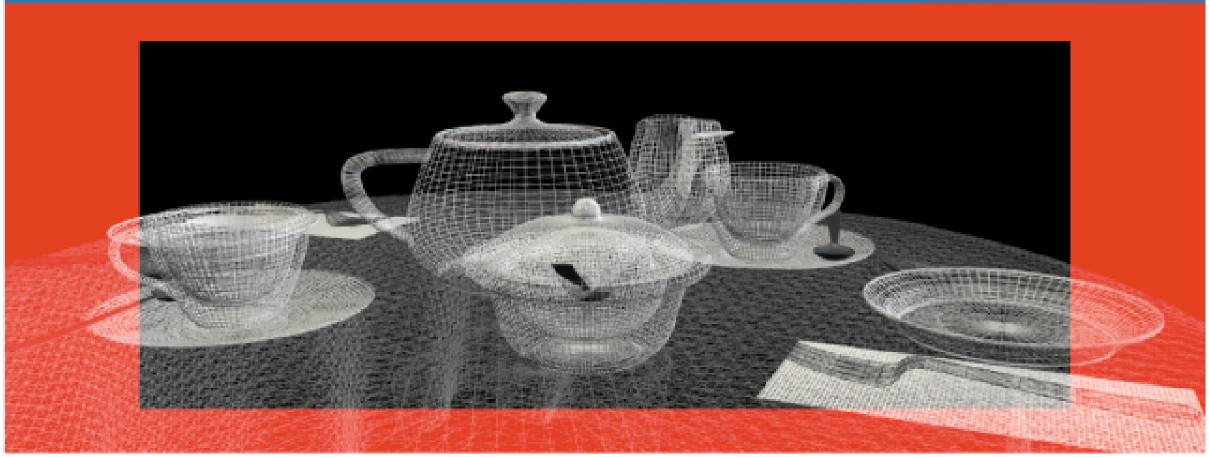


’327 Patent

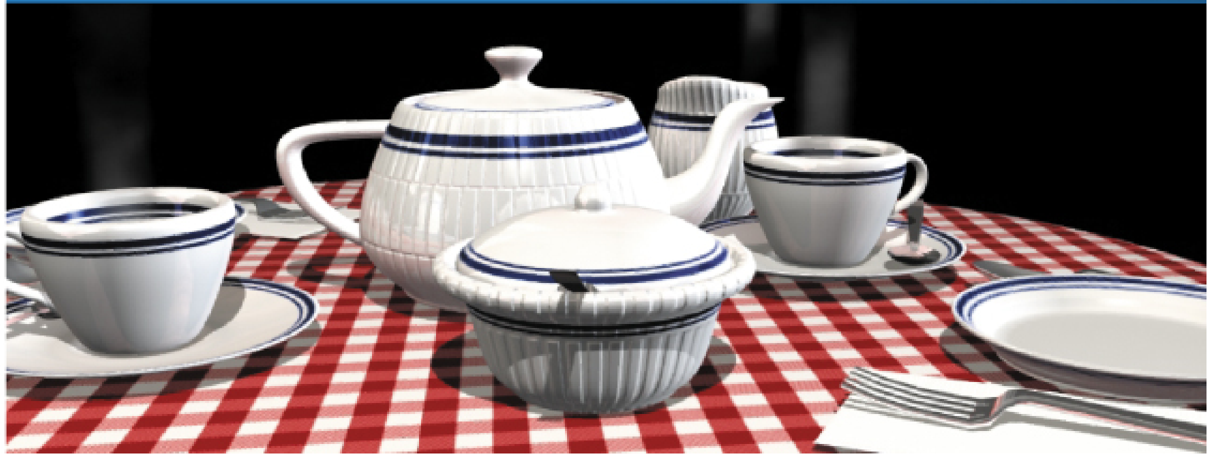
3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

Objects outside the field of view are not displayed

3D MODEL – USER VIEW



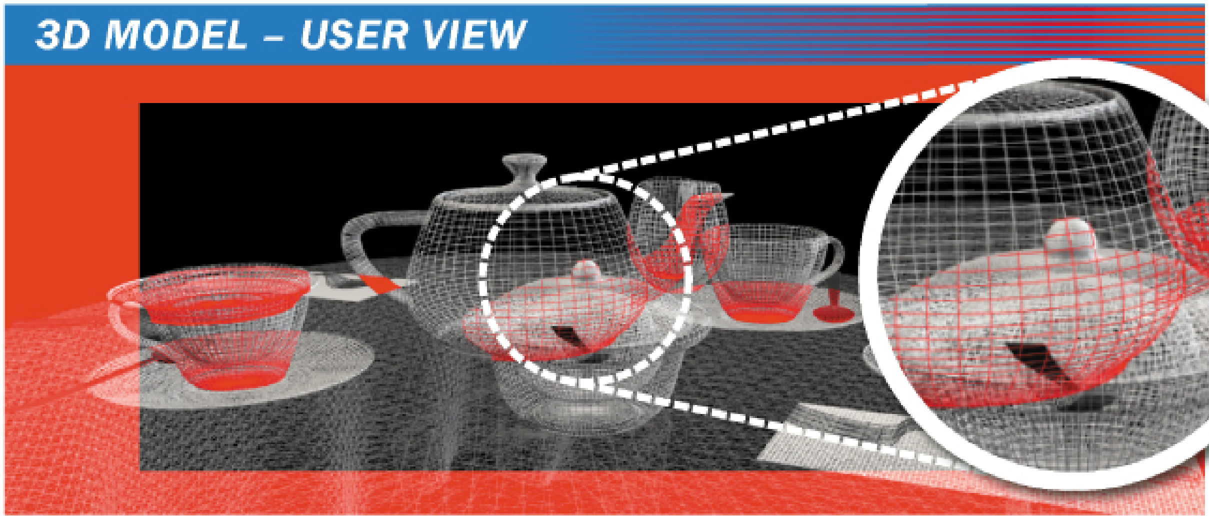
2D RENDERED USER VIEW



’327 Patent

3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

Object primitives blocked by other objects in the chosen view are not displayed



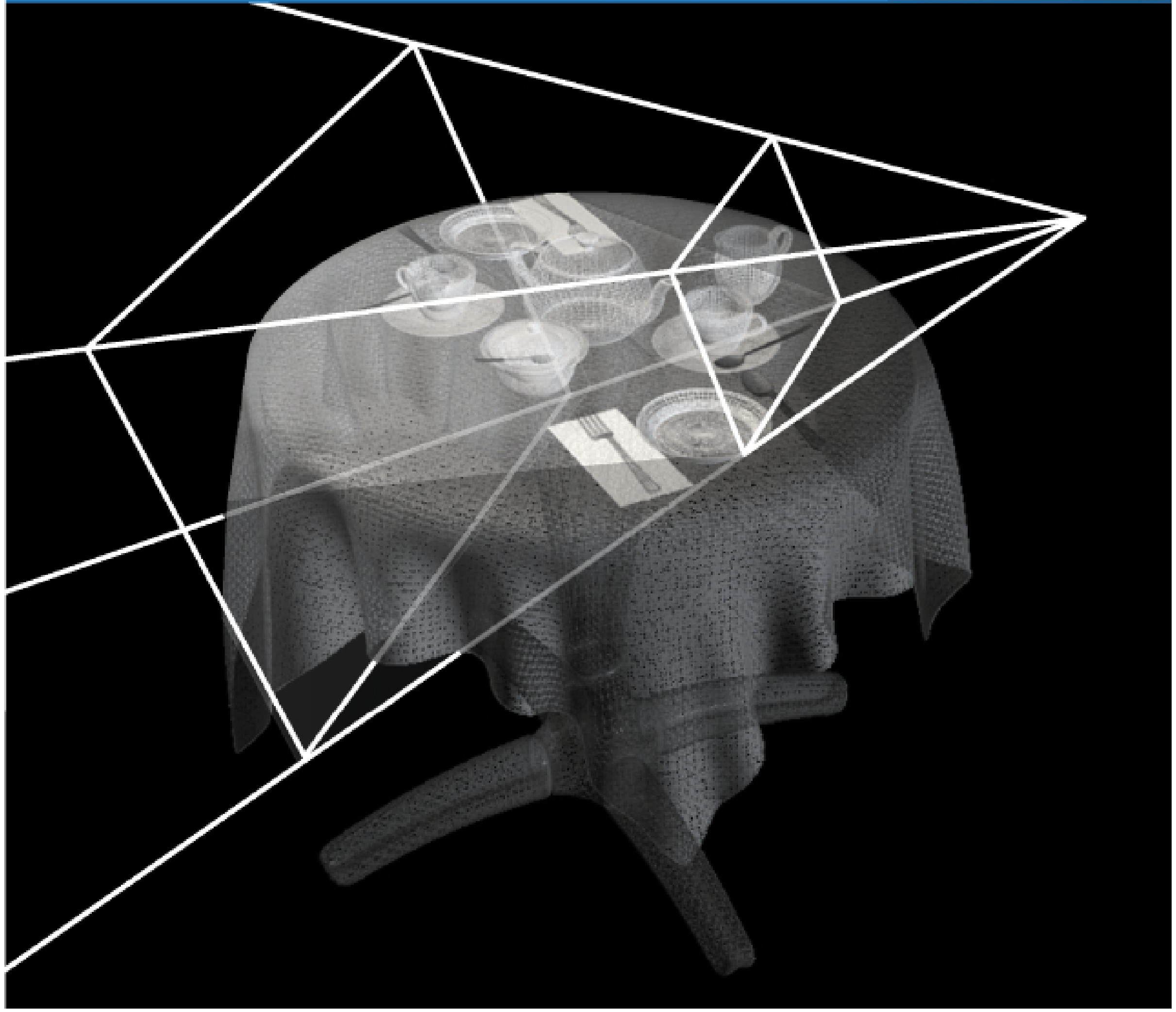
2D RENDERED USER VIEW

'327 Patent

3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

User chooses second view

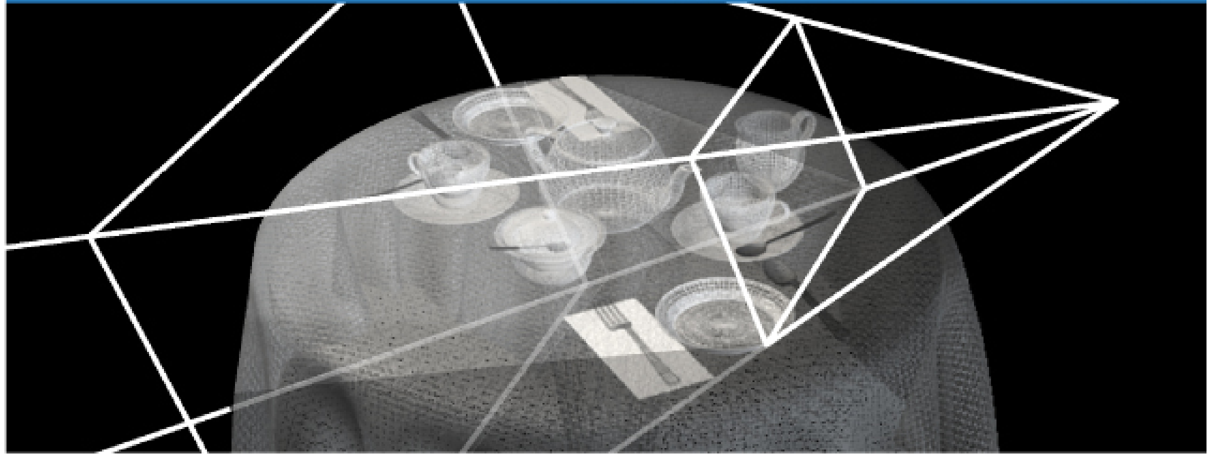
3D MODEL – SECOND VIEW



’327 Patent

3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

3D MODEL – SECOND VIEW

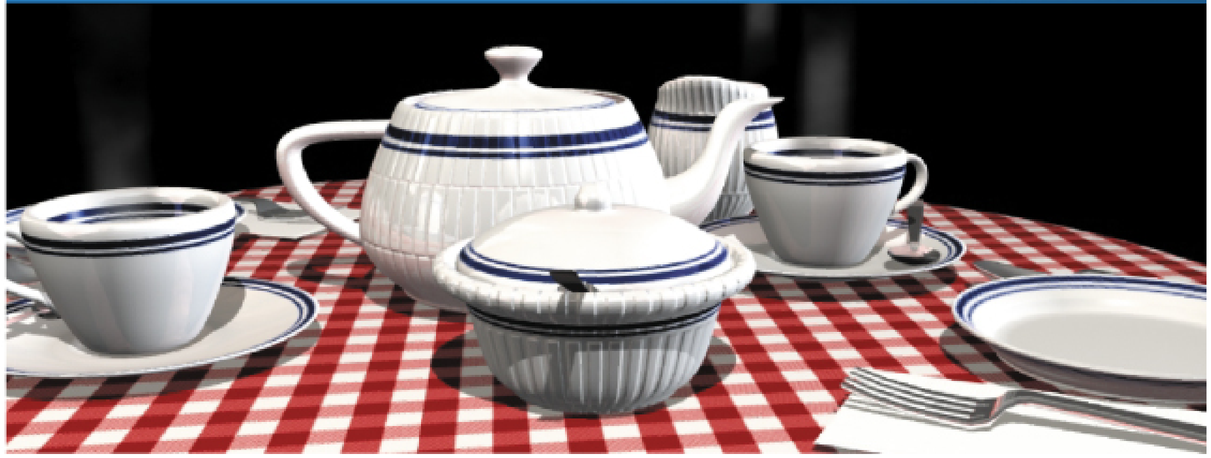


2D RENDERING – SECOND USER VIEW

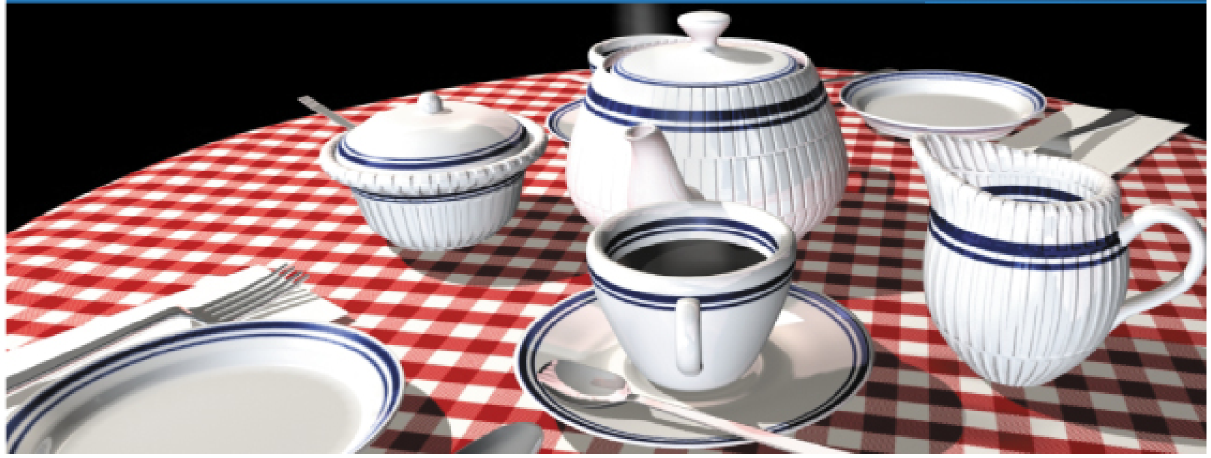
’327 Patent

3-Dimensional World versus 2-Dimensional Representation of a Particular Viewpoint

2D RENDERING – FIRST USER VIEW

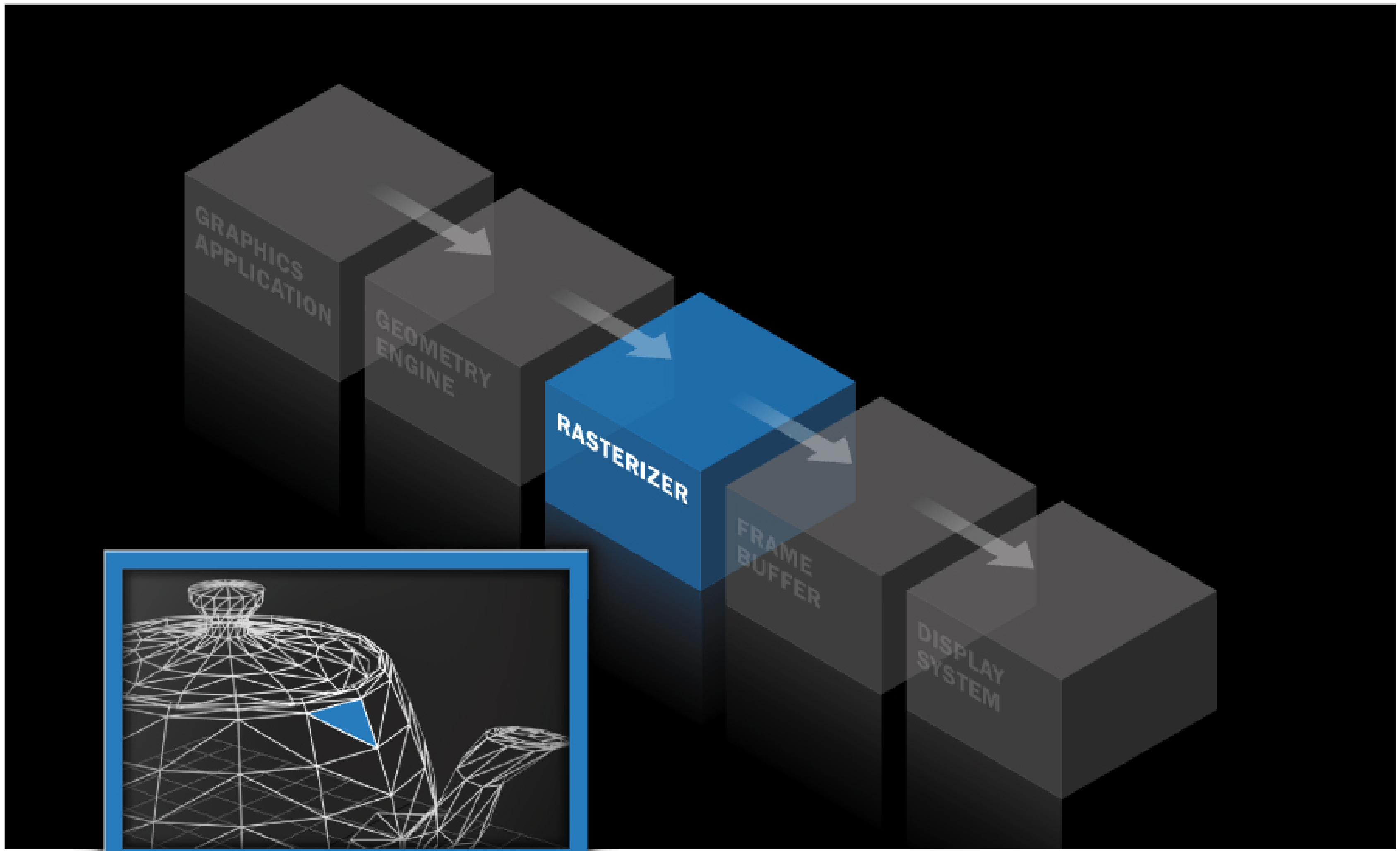


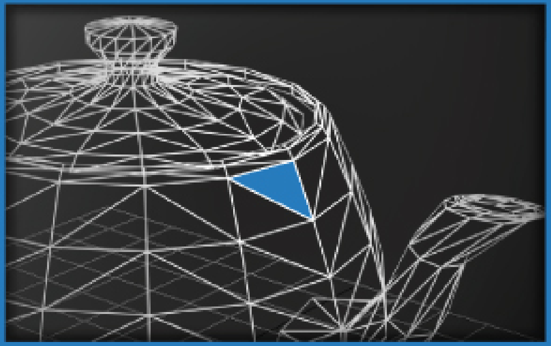
2D RENDERING – SECOND USER VIEW



'327 PATENT

General Depiction of a Graphics Pipeline

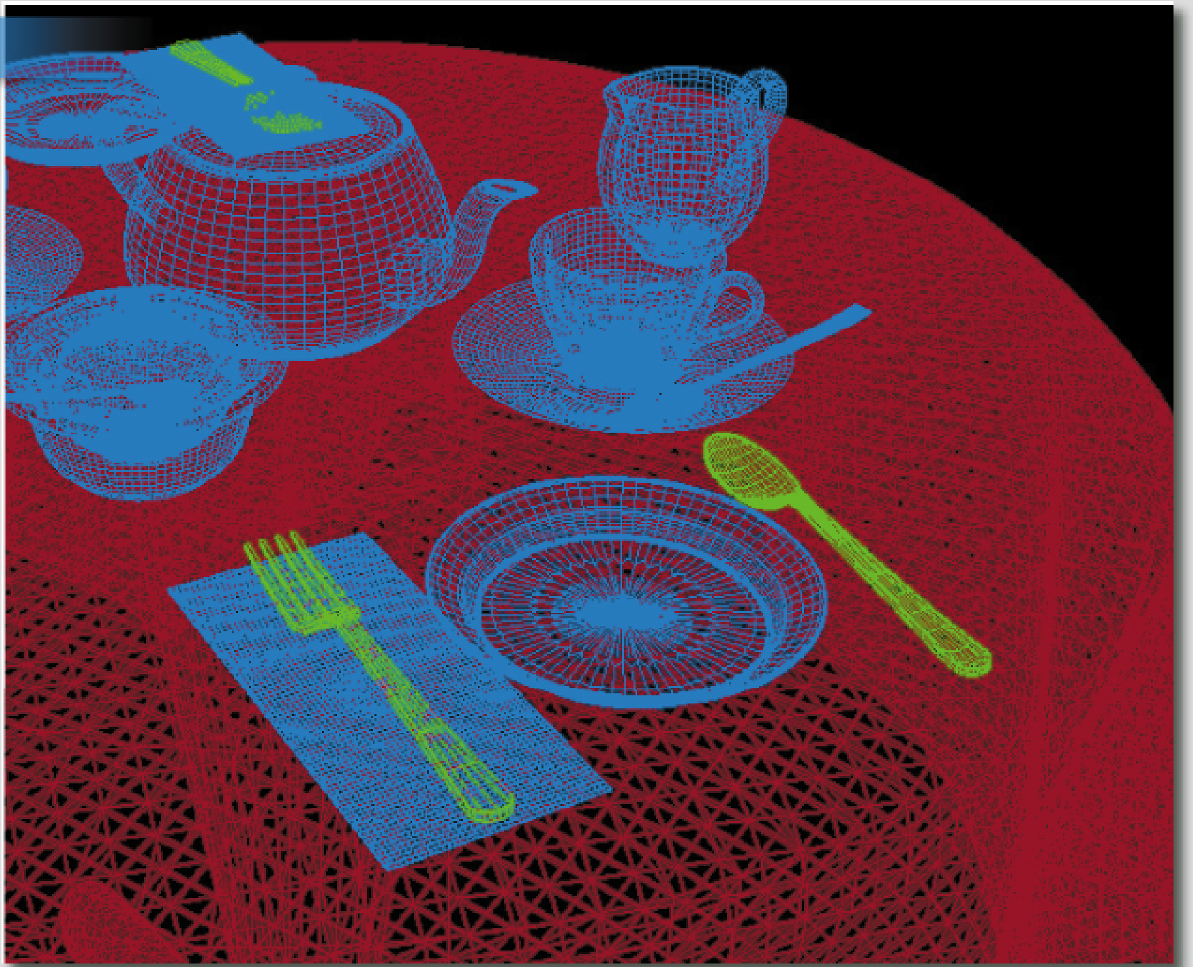




RASTERIZER

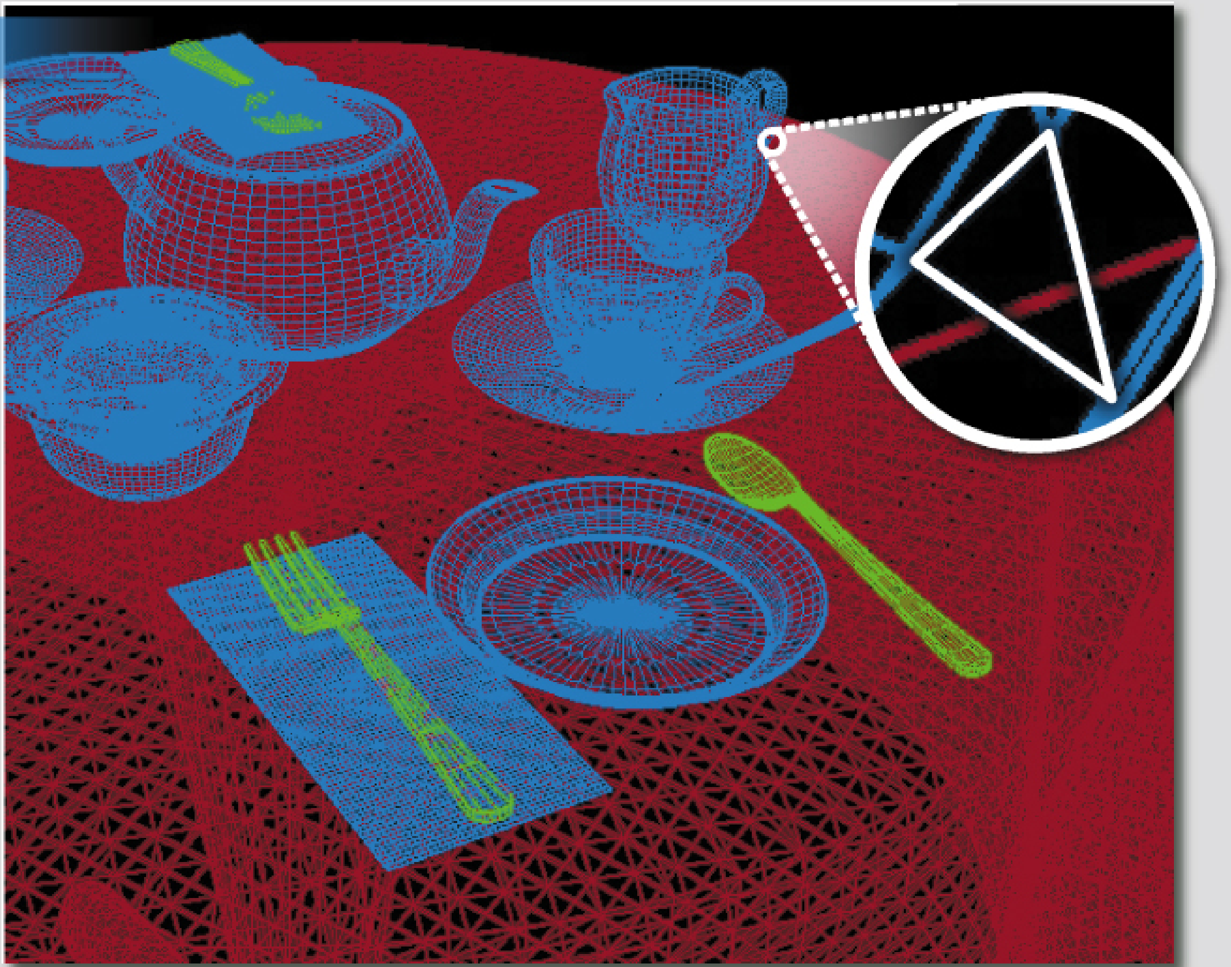
Rasterizing Processes

- PRIMITIVES



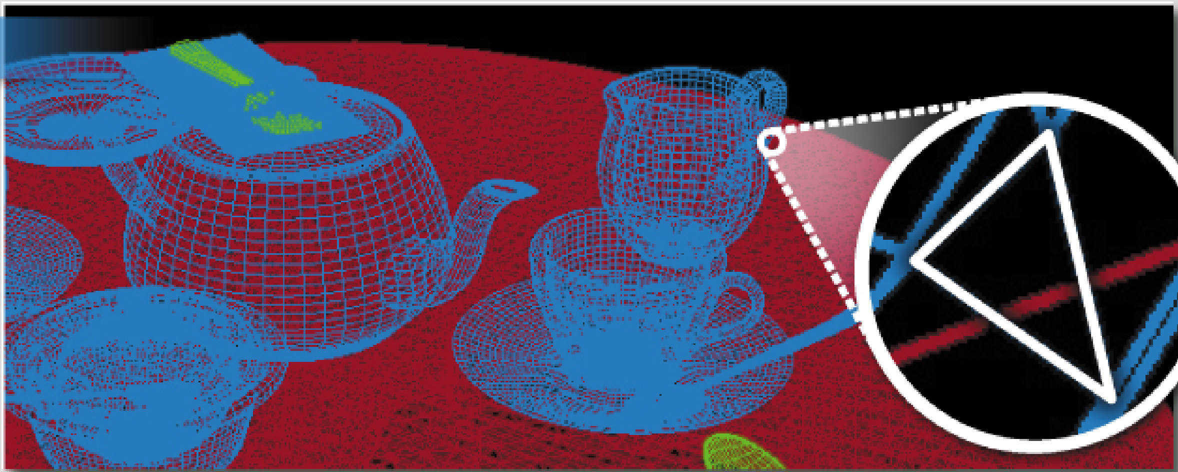
Rasterizing Processes

- PRIMITIVES



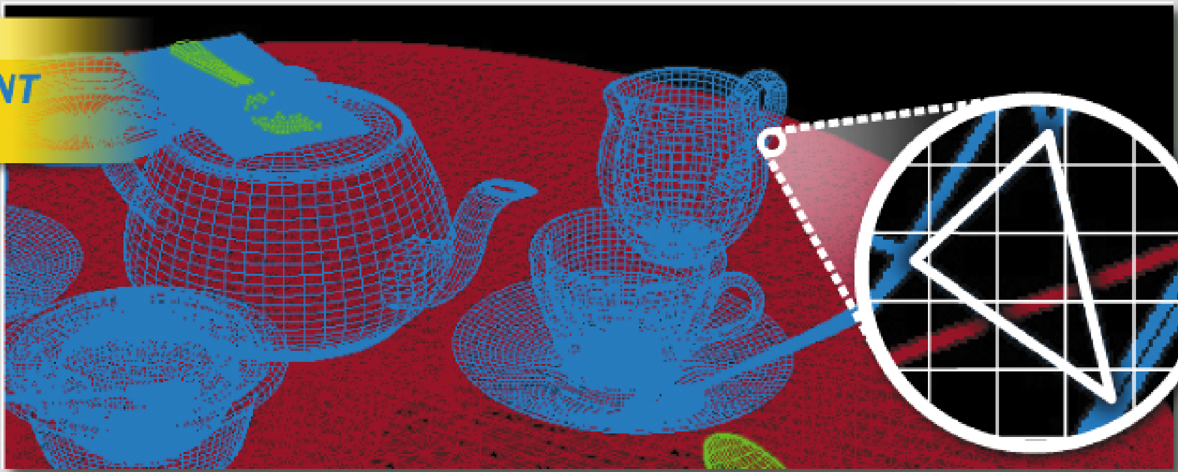
Rasterizing Processes: SCAN CONVERSION

• PRIMITIVES



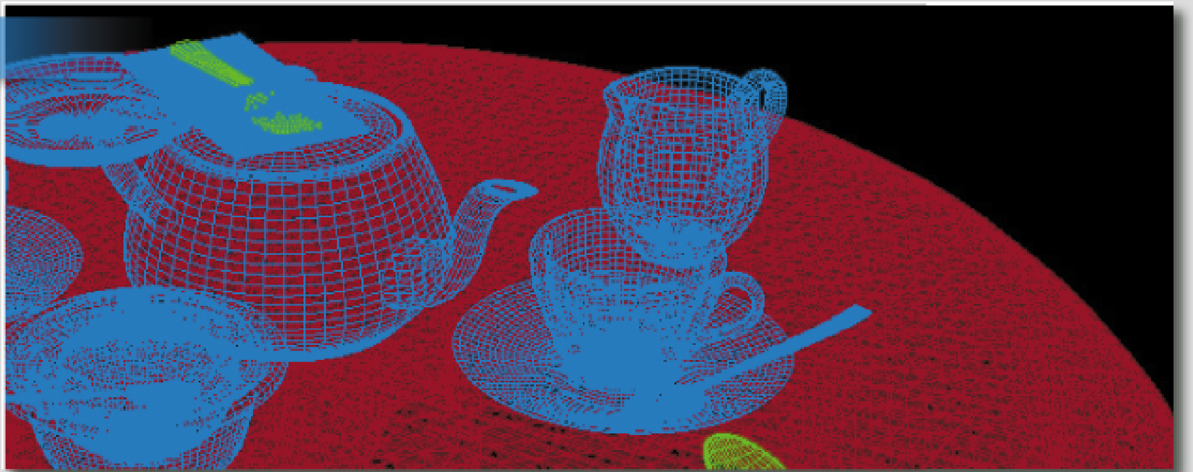
RASTERIZED IMAGE:

• PIXEL ASSIGNMENT TO PRIMITIVES



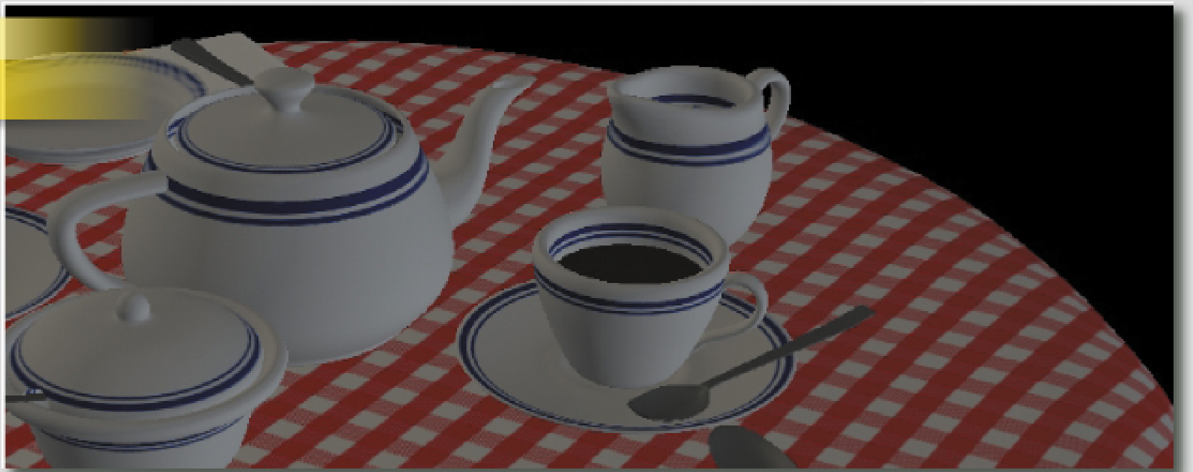
Rasterizing Processes: COLOR

• PRIMITIVES



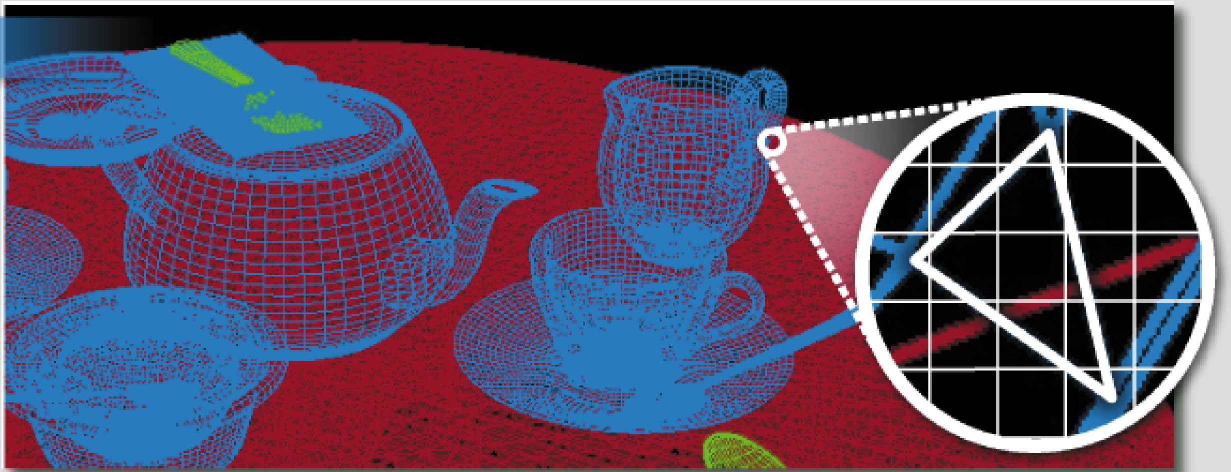
RASTERIZED IMAGE:

• COLOR



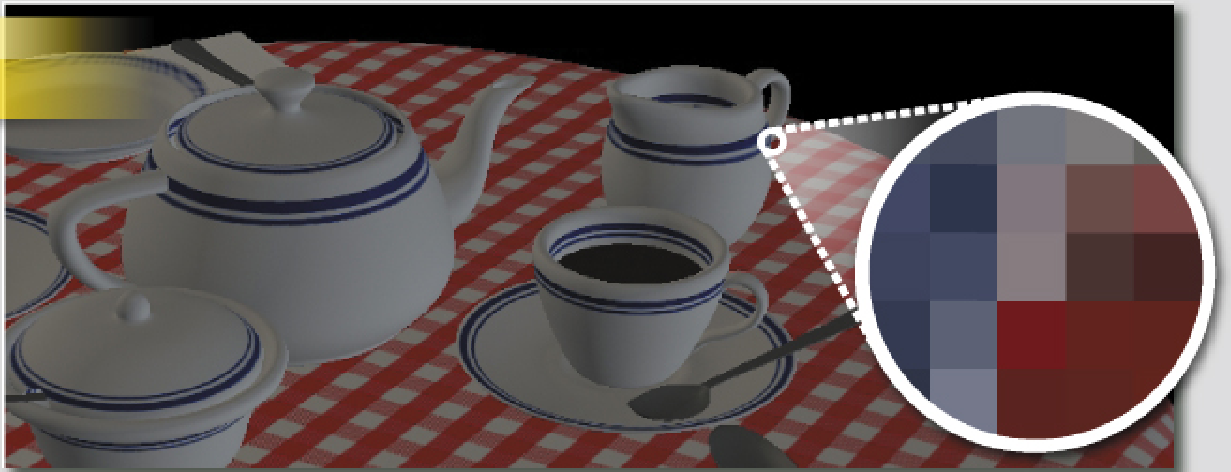
Rasterizing Processes:
COLOR

• PRIMITIVES



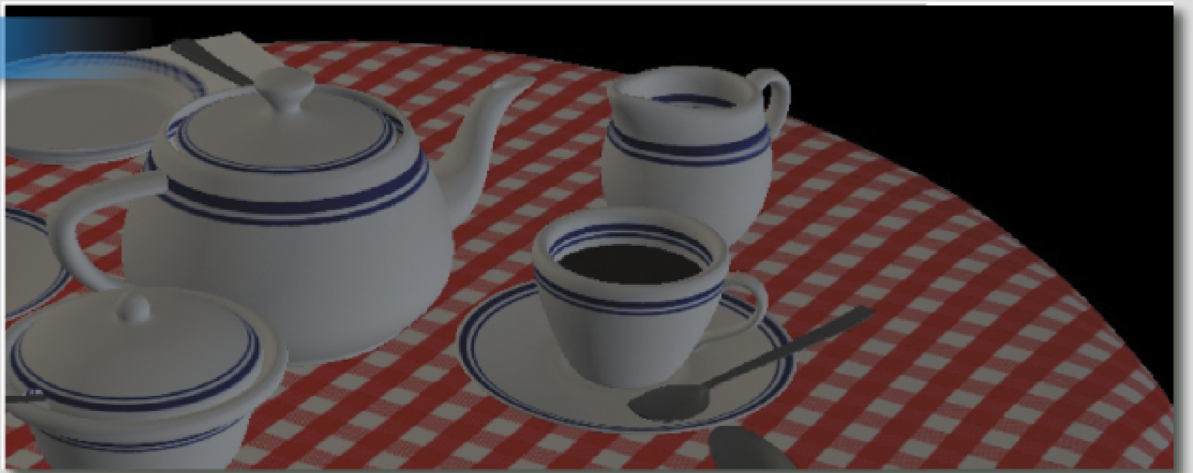
RASTERIZED IMAGE:

• COLOR



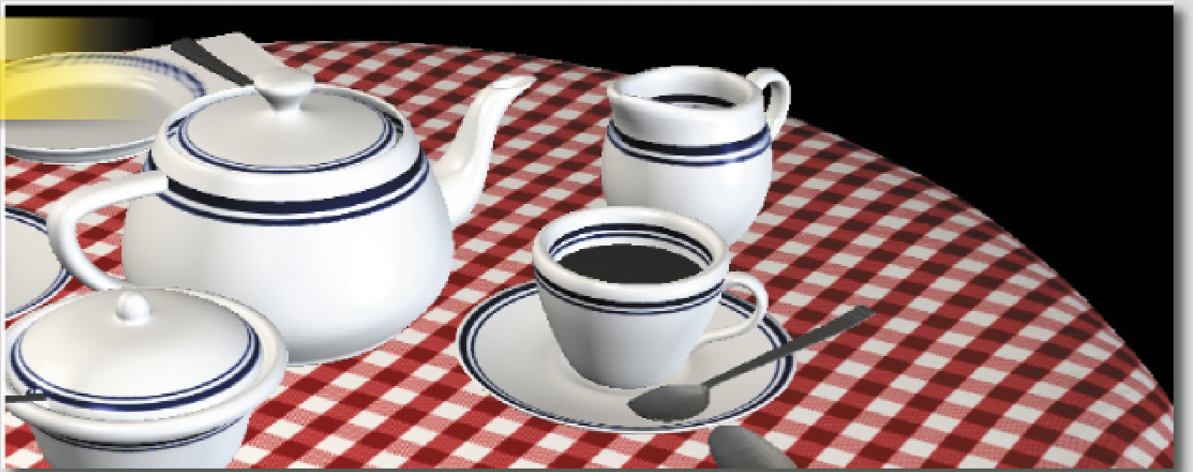
Rasterizing Processes:
LIGHTING

• COLOR



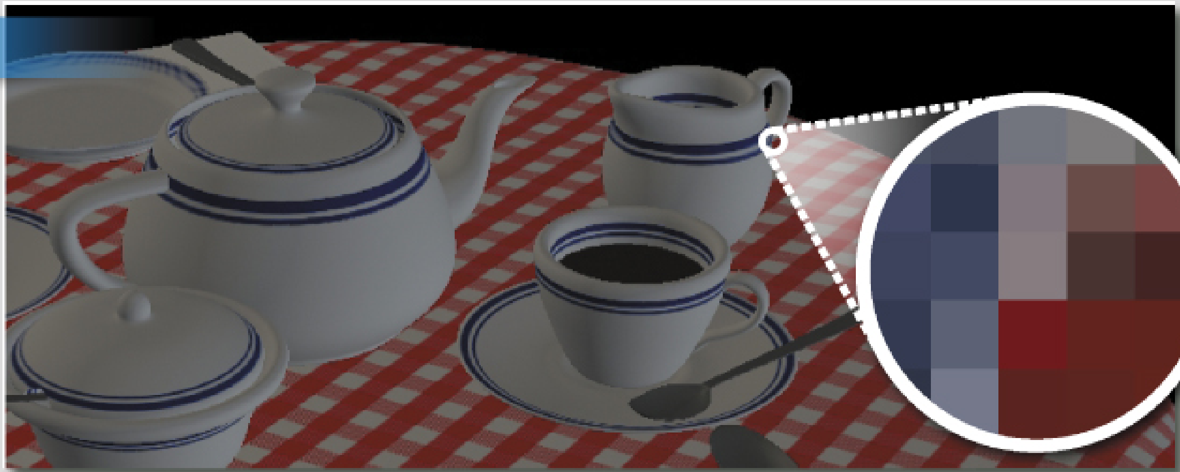
RASTERIZED IMAGE:

• LIGHTING ADDED



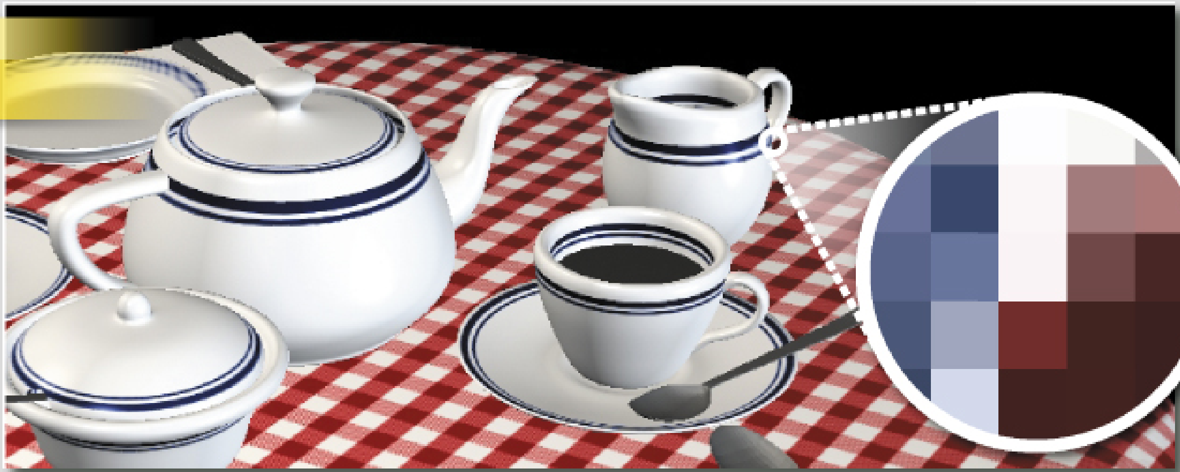
Rasterizing Processes: LIGHTING

• COLOR



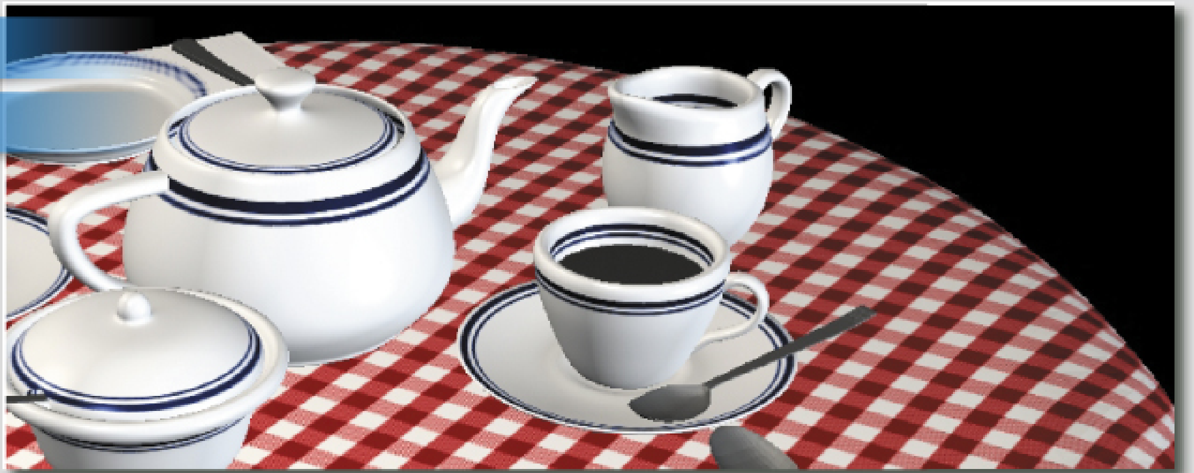
RASTERIZED IMAGE:

• LIGHTING ADDED



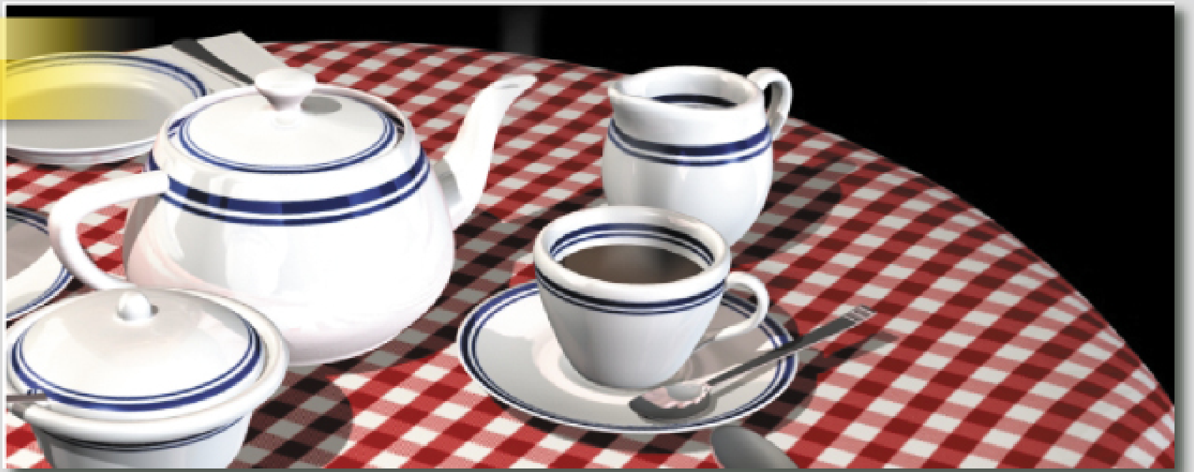
Rasterizing Processes:
SHADOW

- COLOR
- LIGHTING



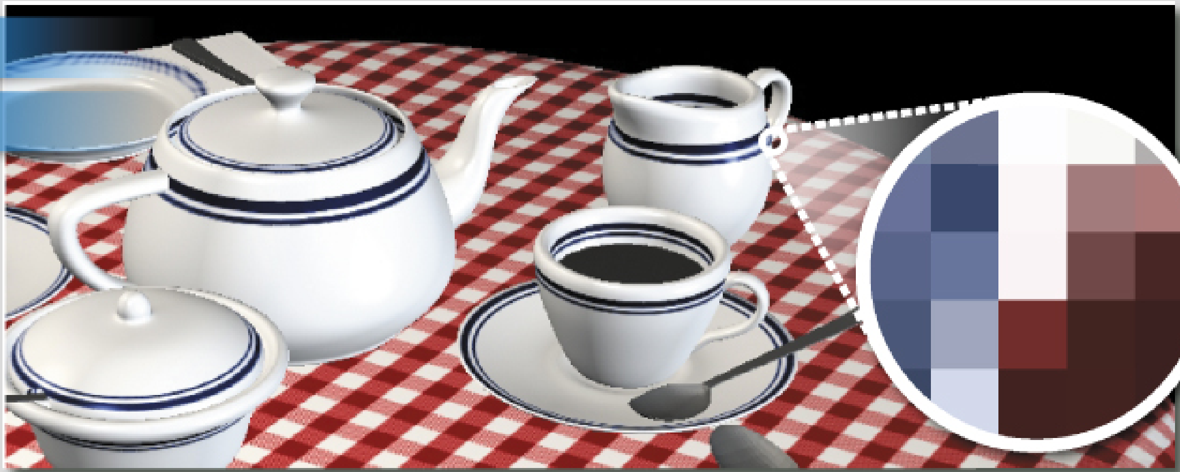
RASTERIZED IMAGE:

- SHADOW ADDED



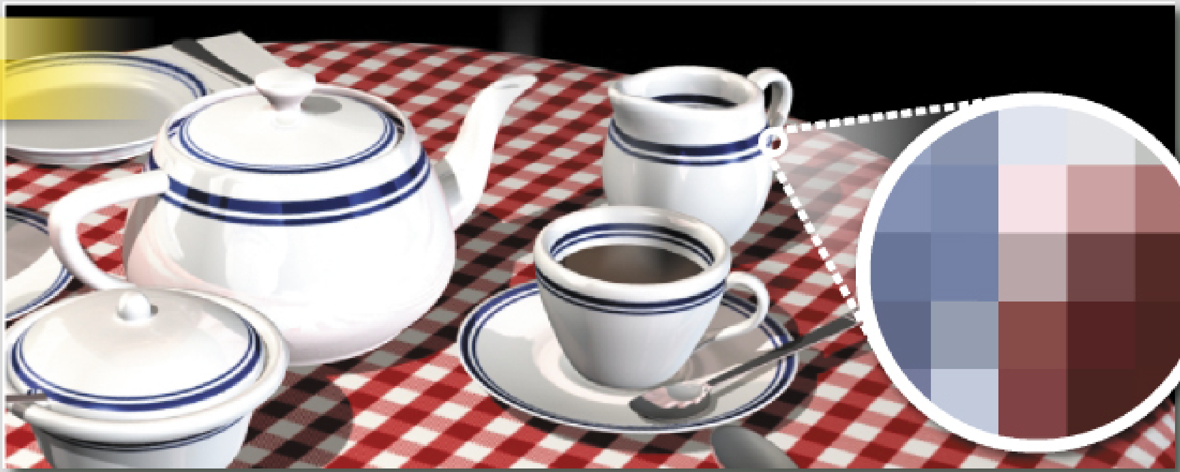
Rasterizing Processes:
SHADOW

- COLOR
- LIGHTING



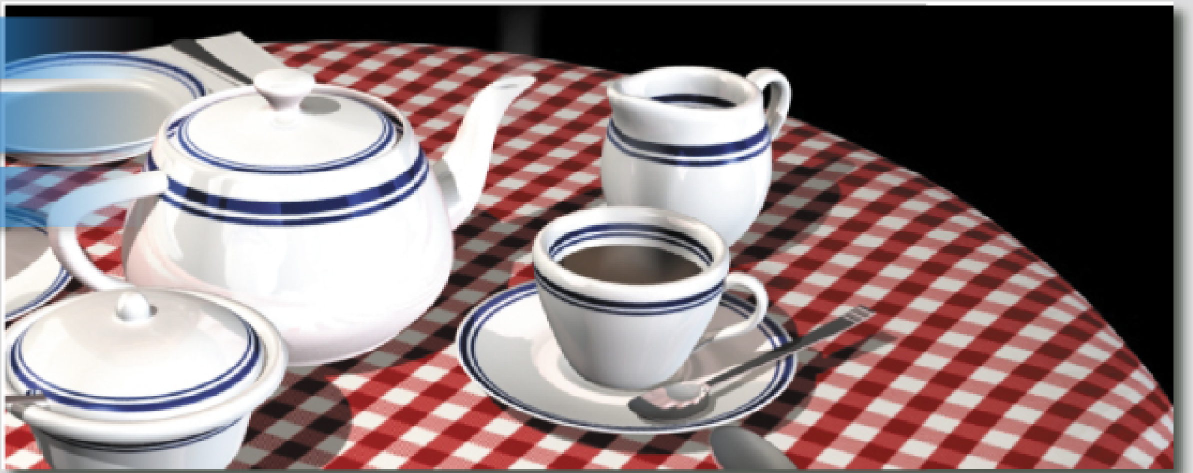
RASTERIZED IMAGE:

- SHADOW ADDED



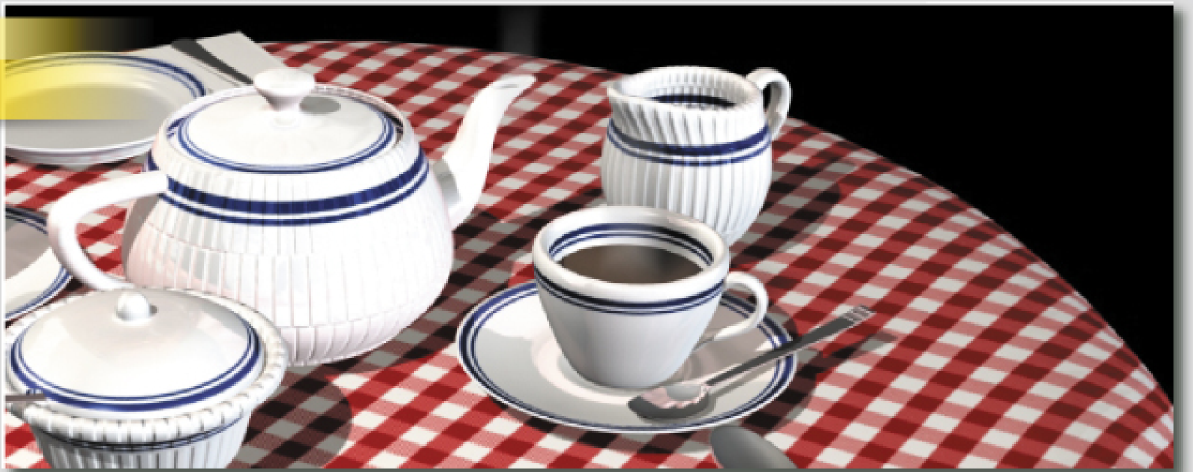
Rasterizing Processes: TEXTURE

- COLOR
- LIGHTING
- SHADOW



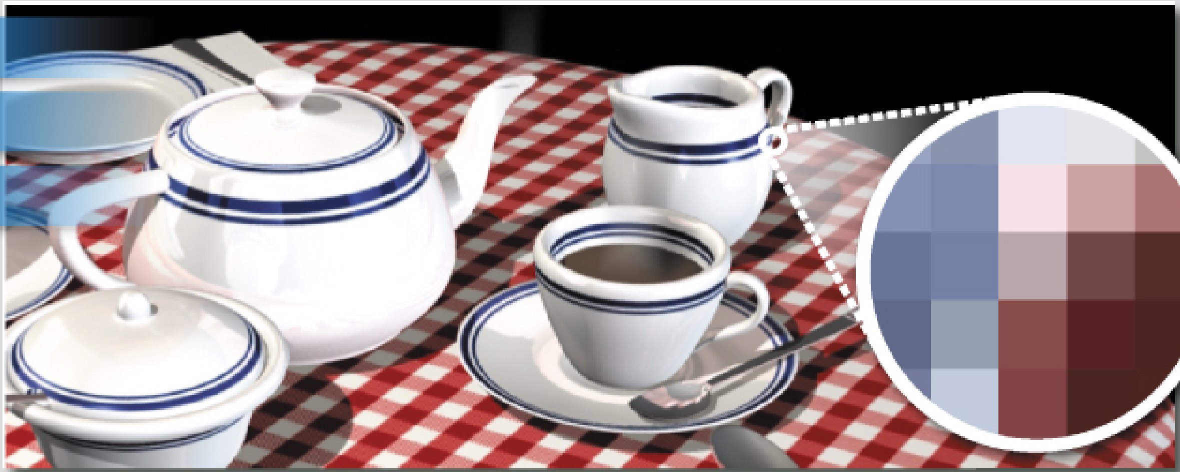
RASTERIZED IMAGE:

- TEXTURE ADDED



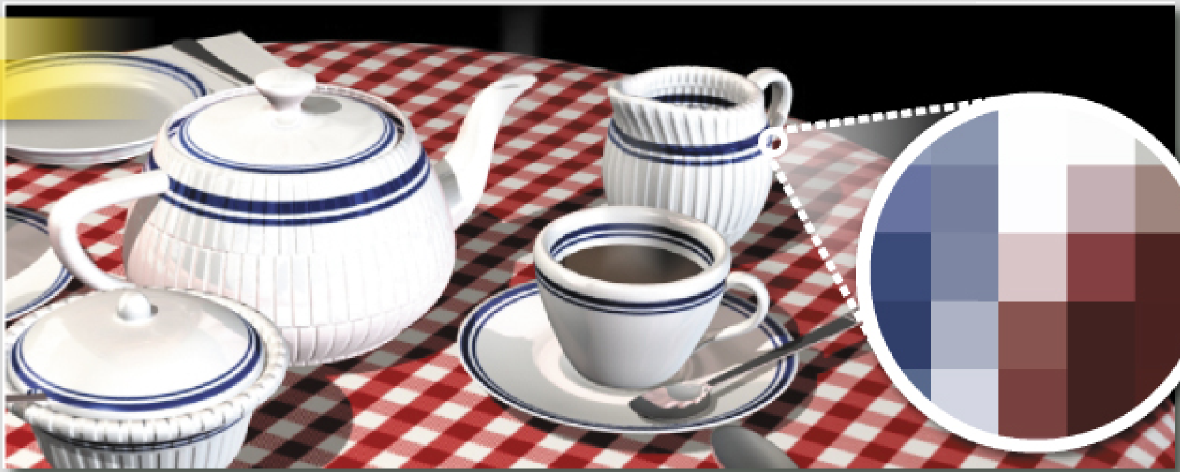
Rasterizing Processes: TEXTURE

- COLOR
- LIGHTING
- SHADOW



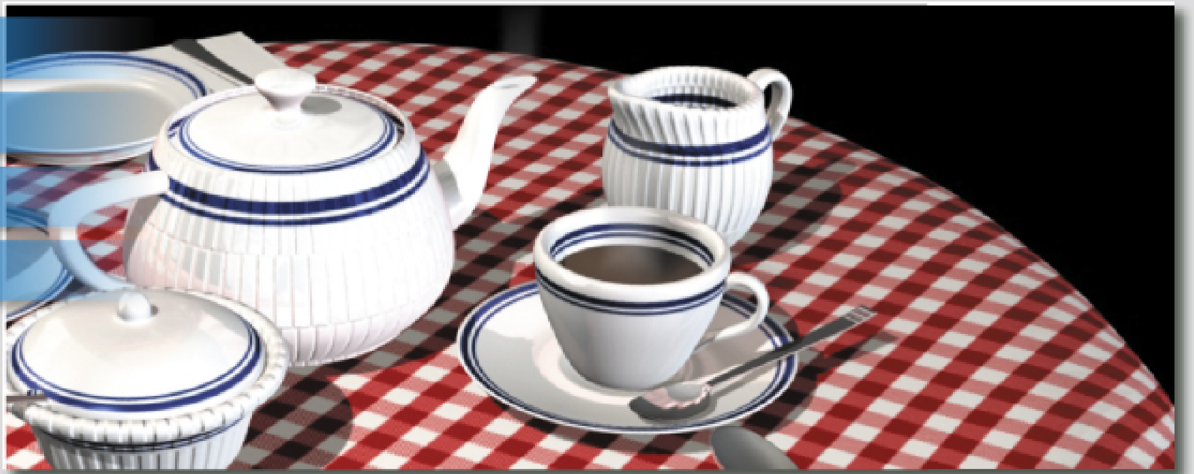
RASTERIZED IMAGE:

- TEXTURE ADDED



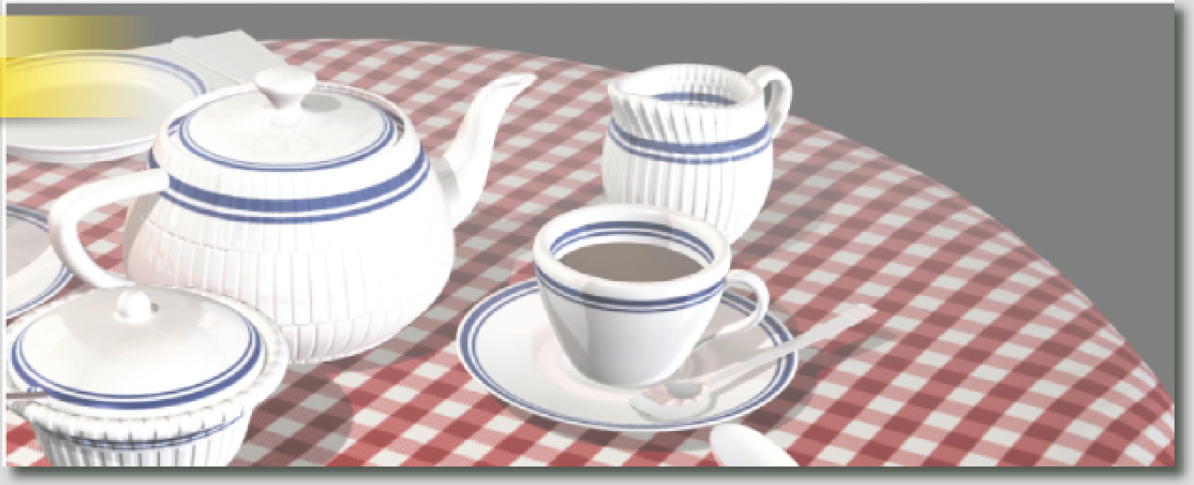
Rasterizing Processes: FOG

- COLOR
- LIGHTING
- SHADOW
- TEXTURE



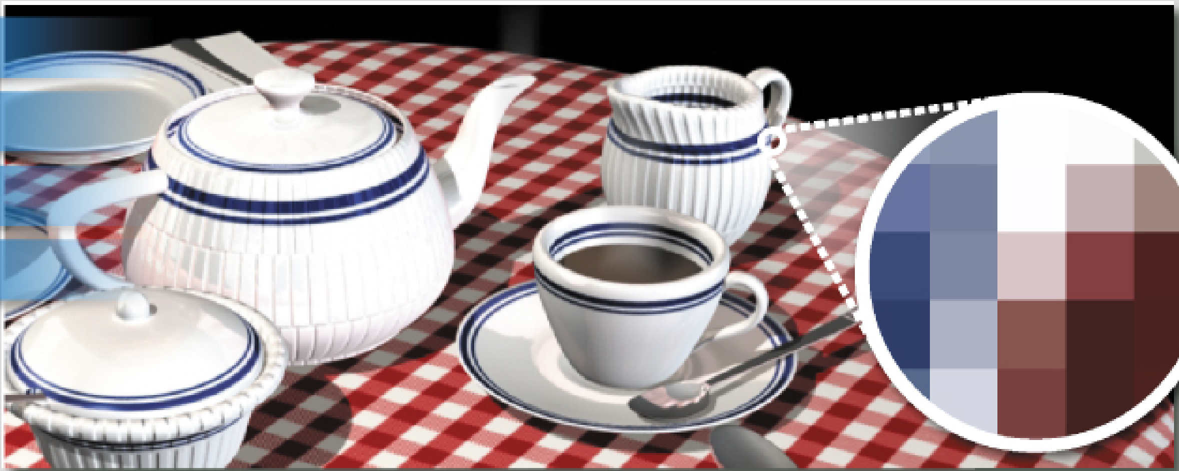
RASTERIZED IMAGE:

- FOG ADDED



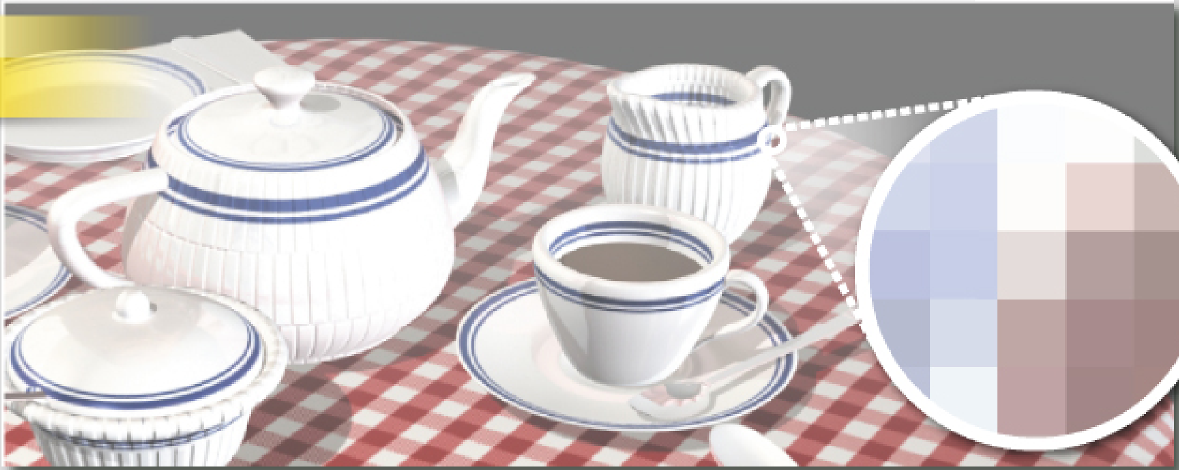
Rasterizing Processes: FOG

- COLOR
- LIGHTING
- SHADOW
- TEXTURE



RASTERIZED IMAGE:

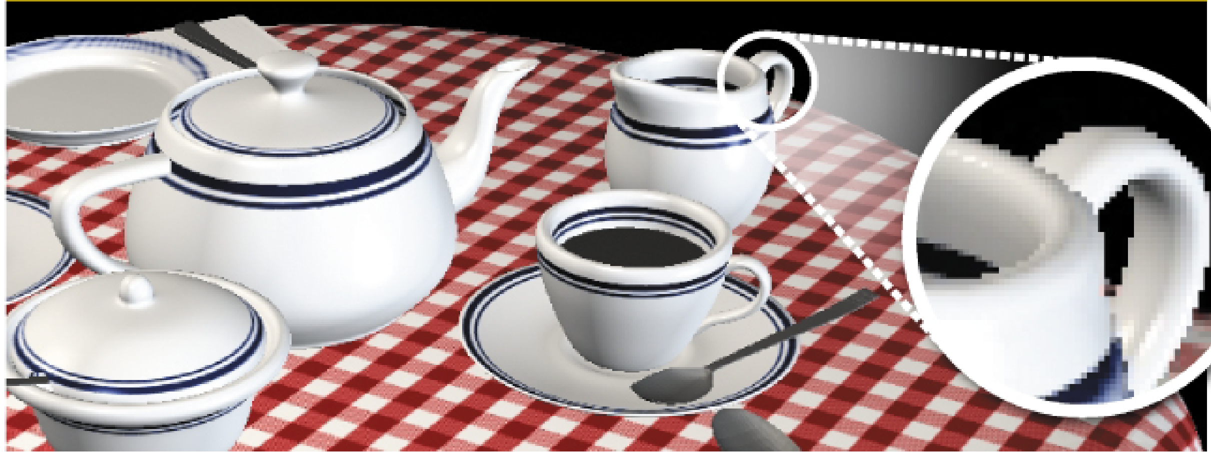
- FOG ADDED



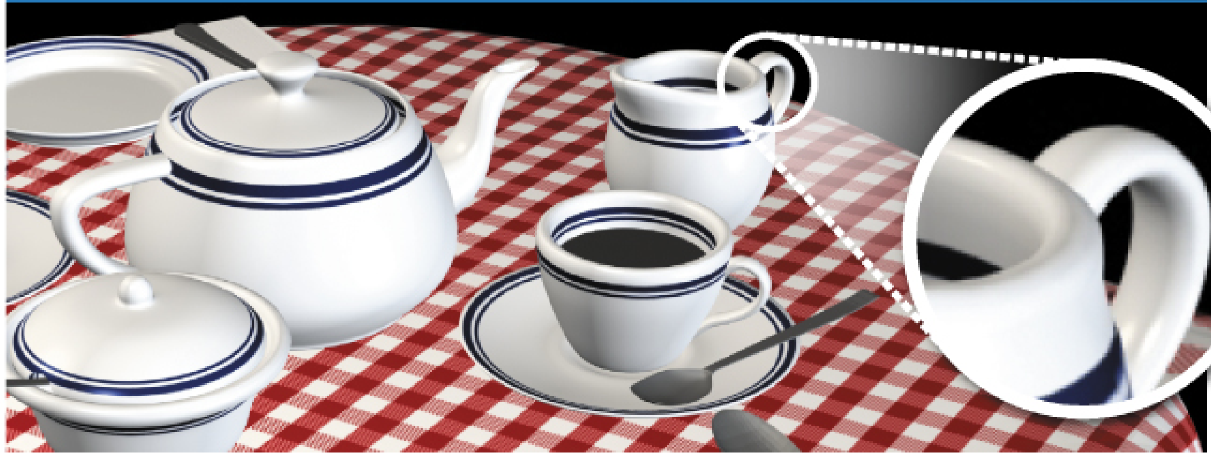
’327 Patent

Rasterizing Processes: Anti-Aliasing

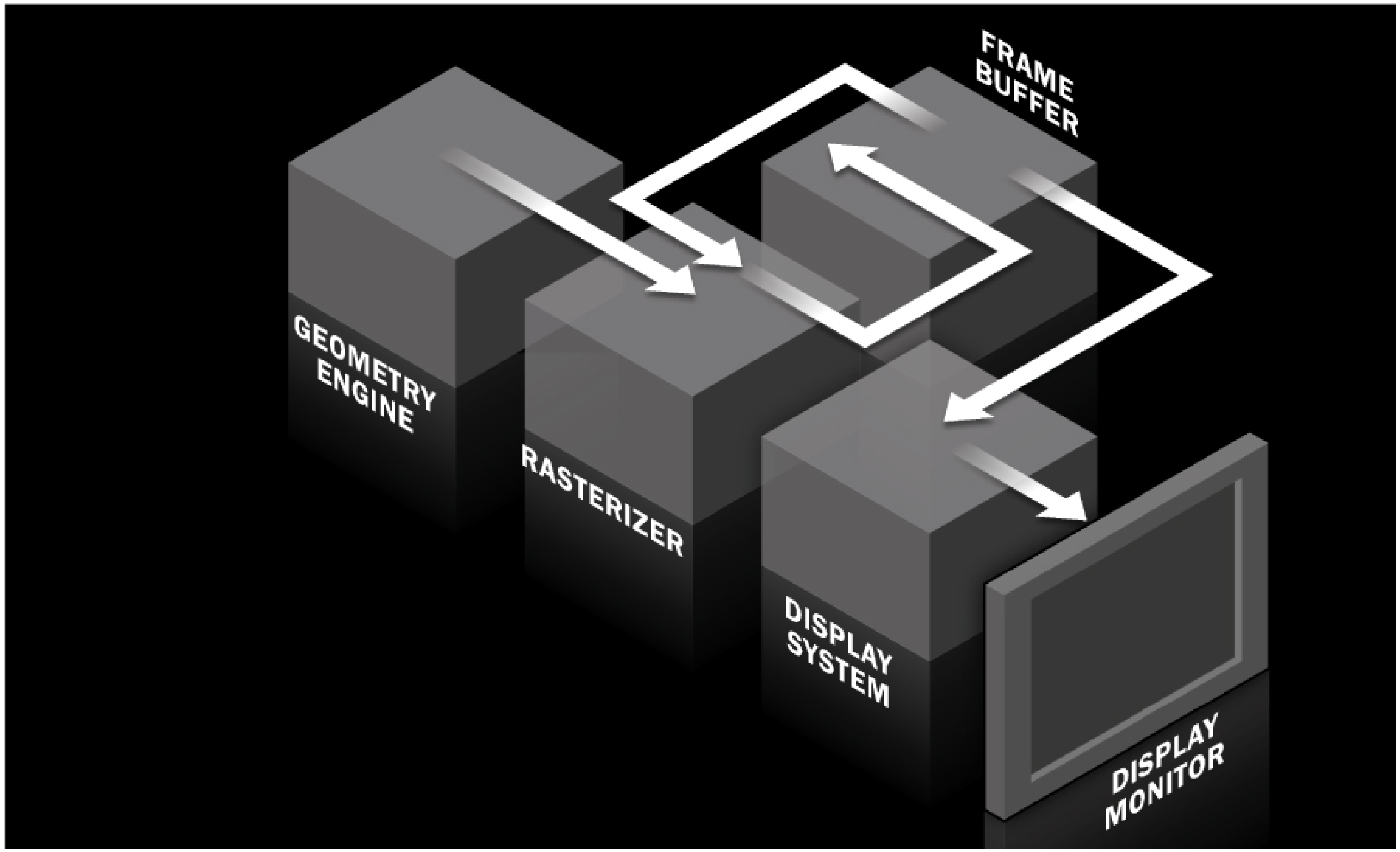
NOT ANTI-ALIASED



ANTI-ALIASED



The Inventive Pipeline



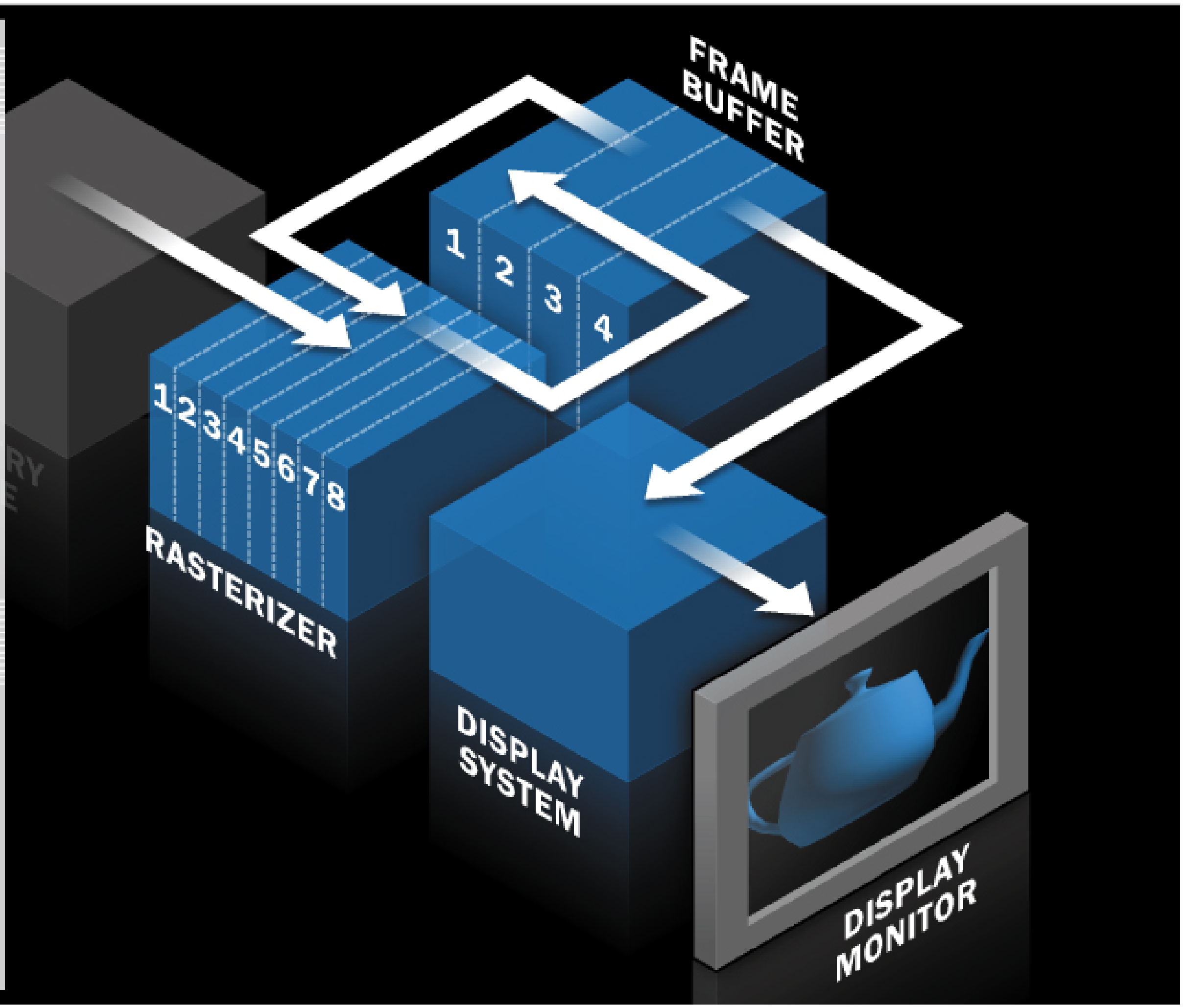
The Inventive Pipeline

RASTERIZATION PROCESSES:

1: Scan Conversion
2: Assigning Color
3: Lighting
4: Applying Texture
5: Applying Fog
6: Blending
7: Shading
8: Antialiasing

FRAME BUFFER FUNCTIONS:

1: Receive from Rasterizer
2: Available to Rasterizer
3: Store Fragments
4: Store Pixels



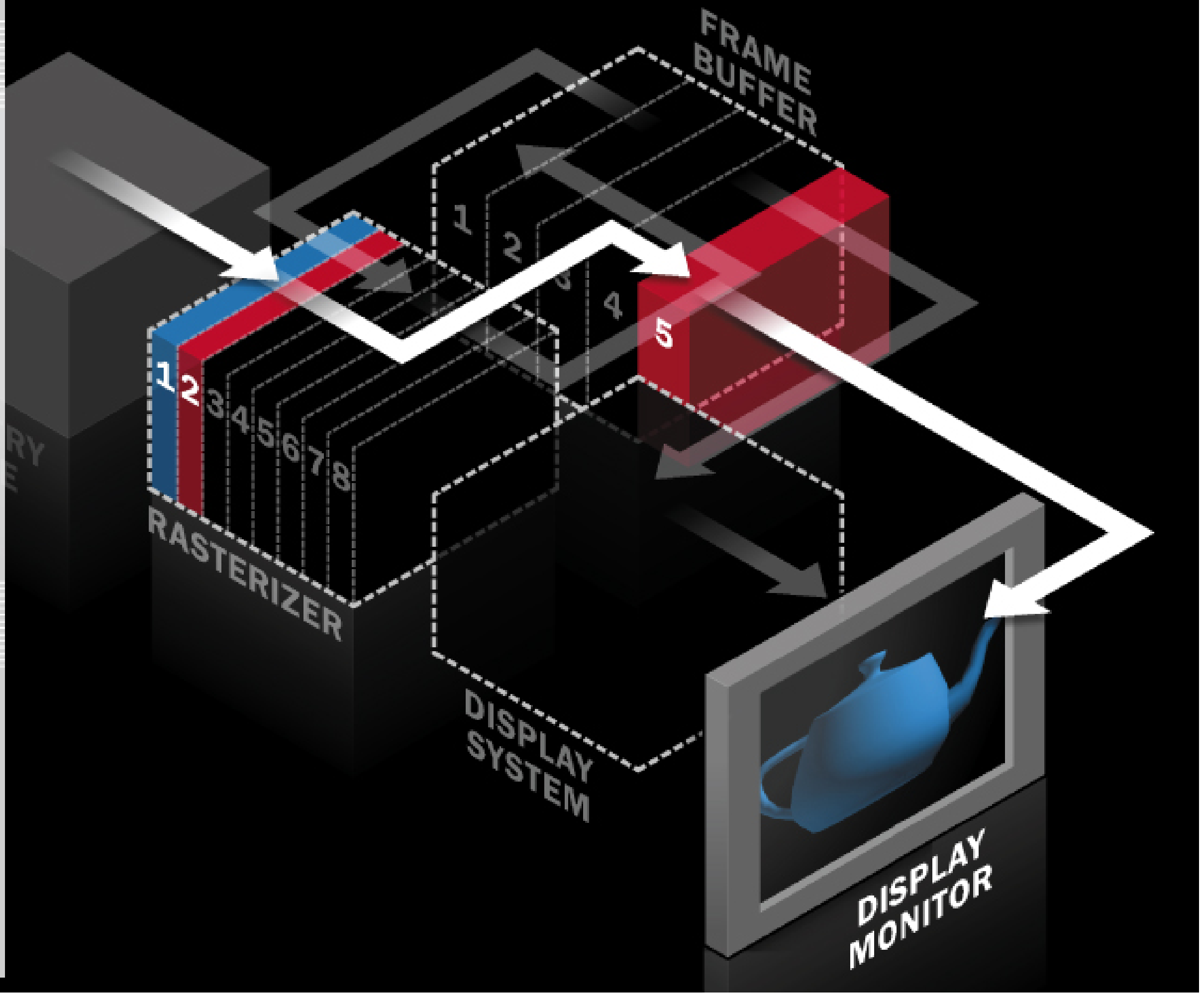
AMD's Proposed Construction Cuts the Heart Out of the Pipeline as Envisioned by the Inventors

RASTERIZATION PROCESSES:

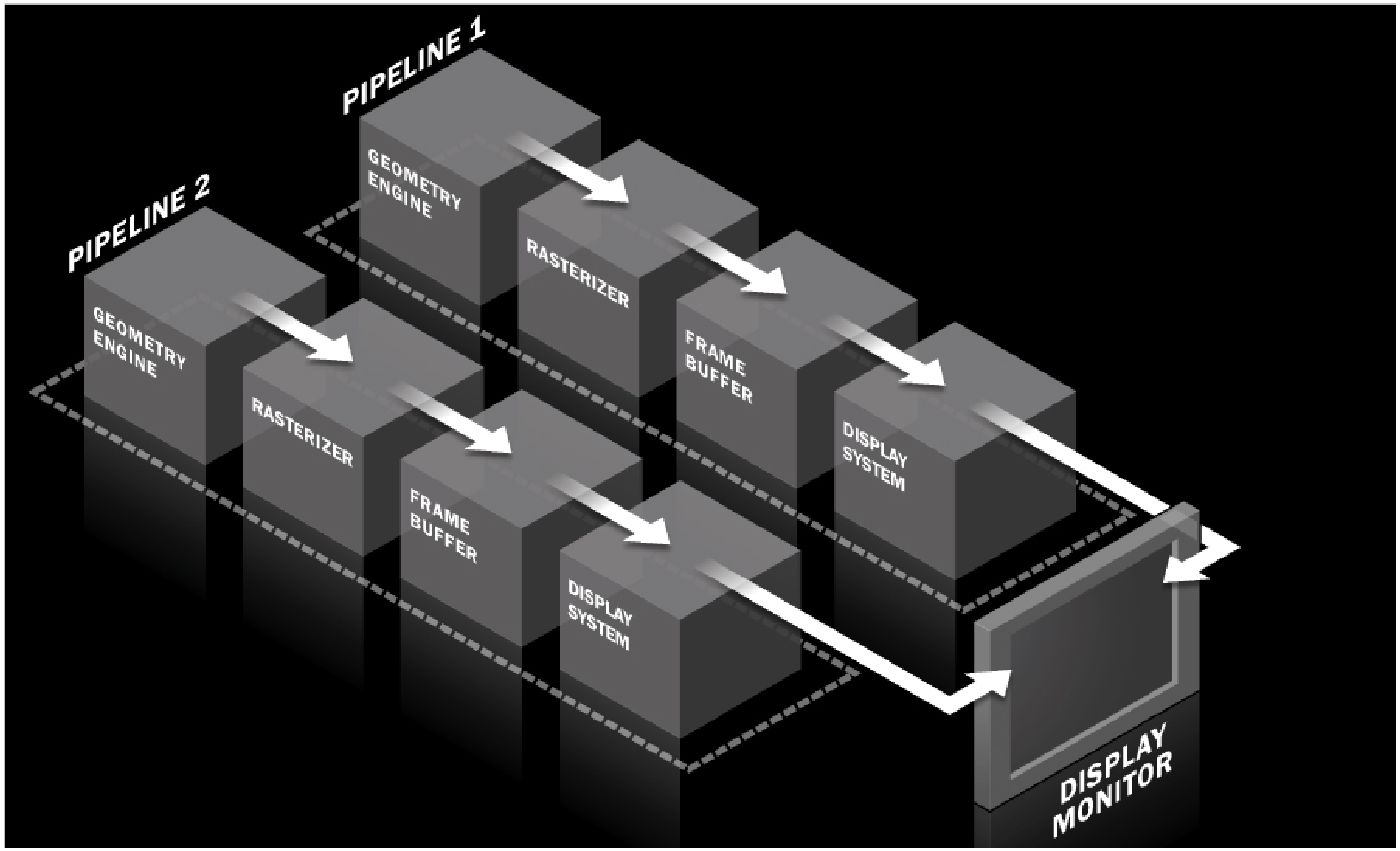
1: Scan Conversion
2: Assign Base Color
~~3: Lighting~~
~~4: Applying Texture~~
~~5: Applying Fog~~
~~6: Blending~~
~~7: Shading~~
~~8: Antialiasing~~

FRAME BUFFER FUNCTIONS:

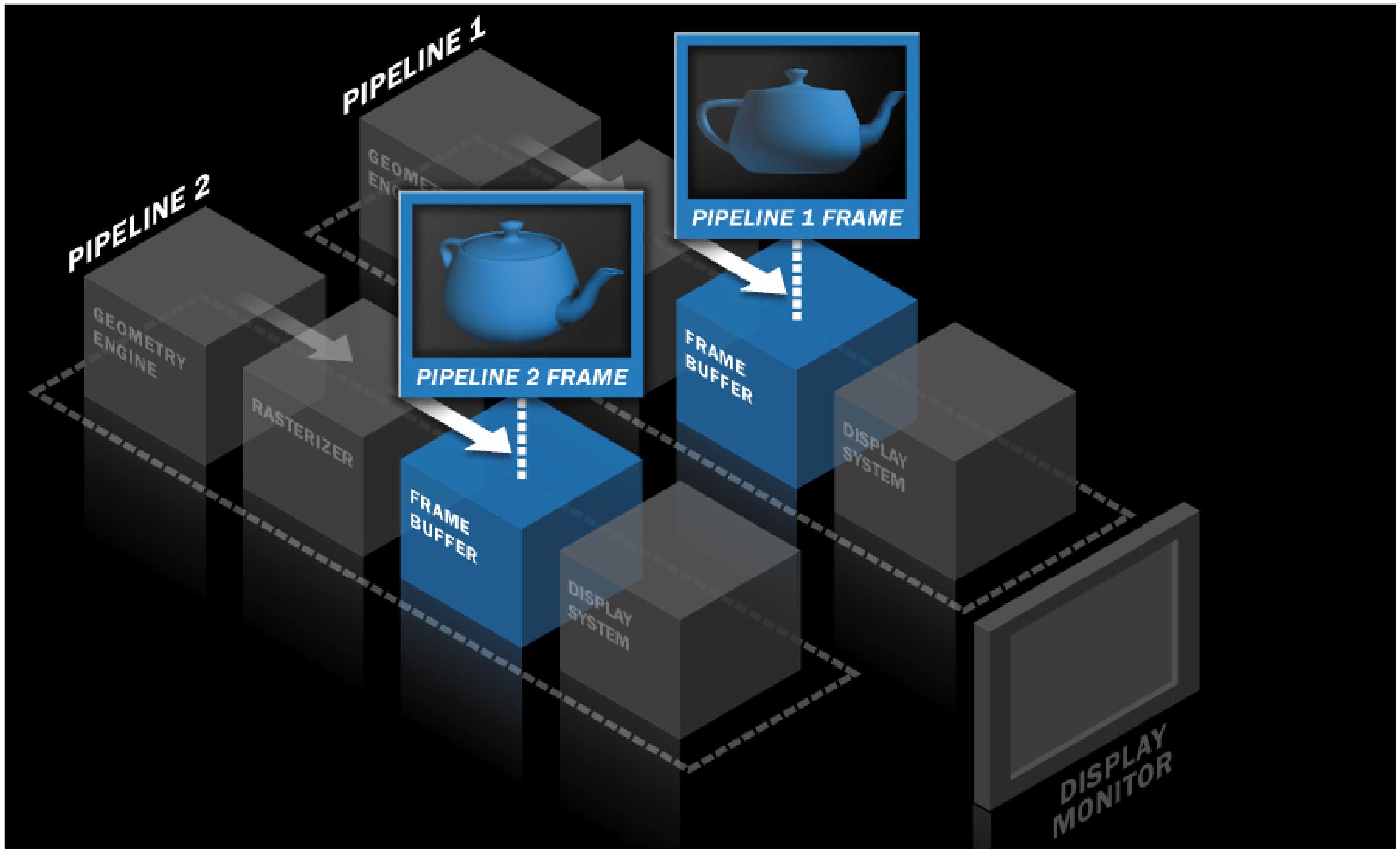
~~1: Receive from Rasterizer~~
~~2: Available to Rasterizer~~
~~3: Store Fragments~~
~~4: Store Pixels~~
5. Scanner



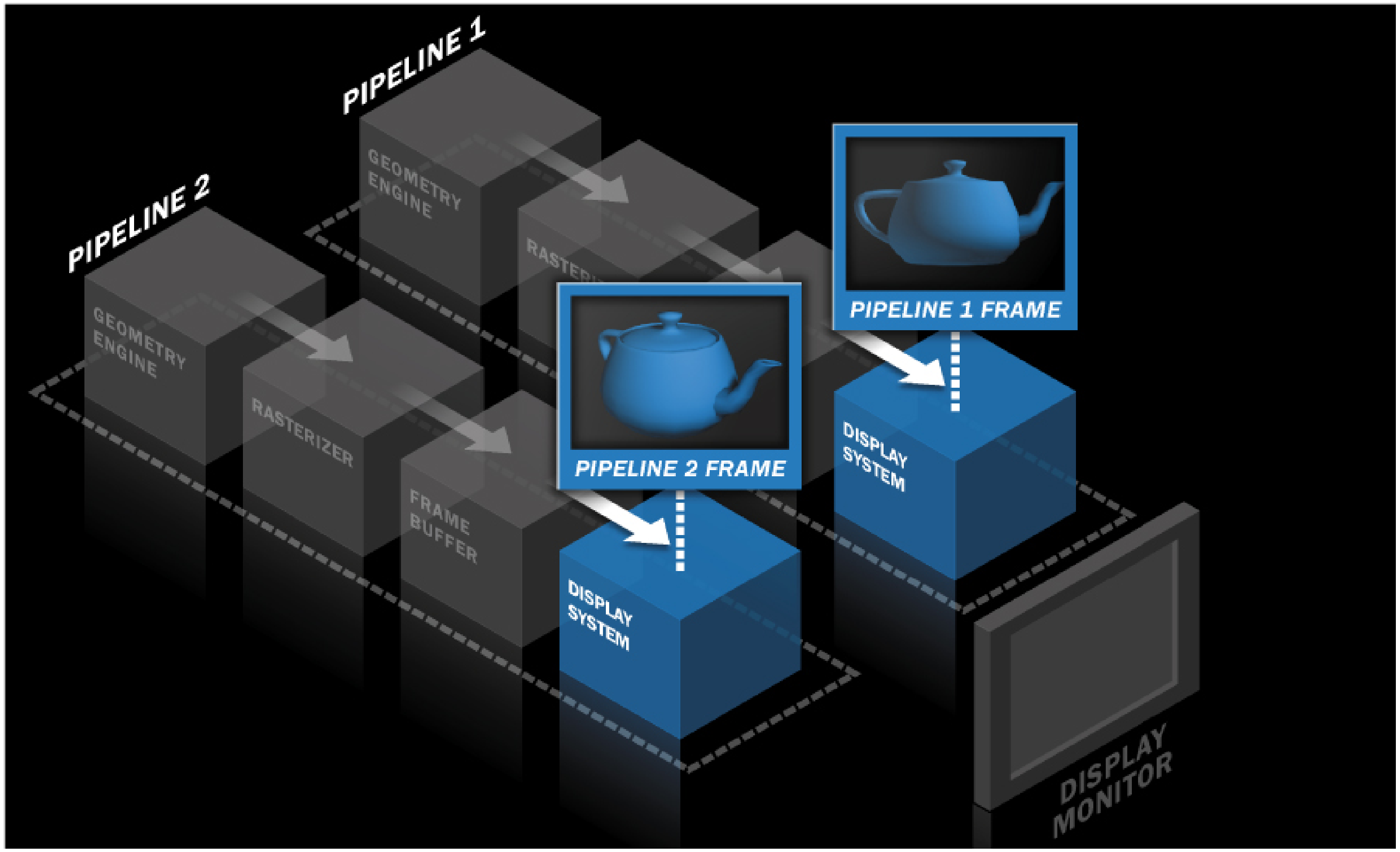
Prior Art


PIPELINE 1
GEOMETRY ENGINE
RASTERIZER
FRAME BUFFER
DISPLAY SYSTEM
PIPELINE 2
GEOMETRY ENGINE
RASTERIZER
FRAME BUFFER
DISPLAY SYSTEM
DISPLAY MONITOR

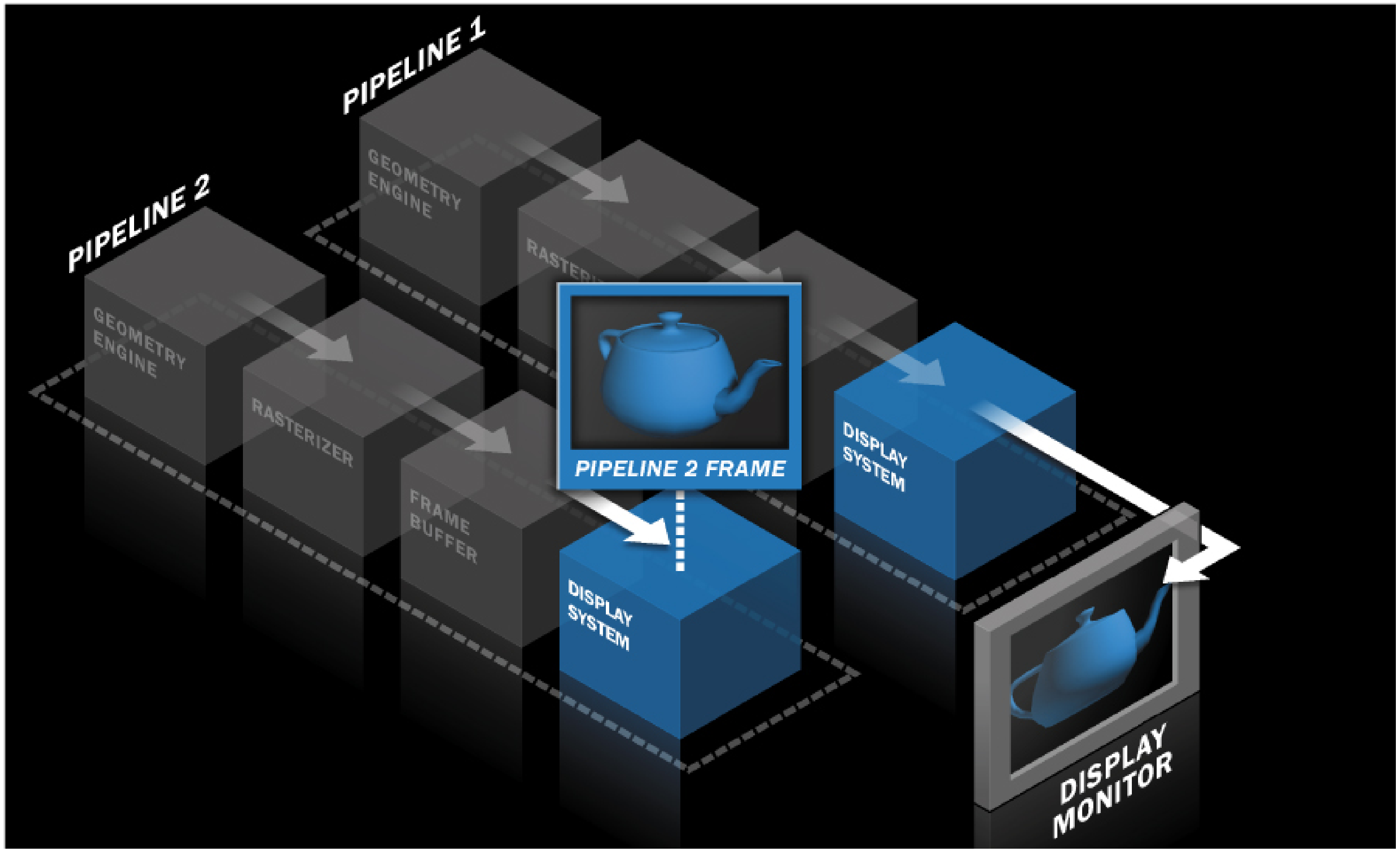
Prior Art


PIPELINE 1
PIPELINE 2
GEOMETRY ENGINE
RASTERIZER
FRAME BUFFER
DISPLAY SYSTEM
PIPELINE 1 FRAME
PIPELINE 2 FRAME
FRAME BUFFER
DISPLAY SYSTEM
DISPLAY MONITOR

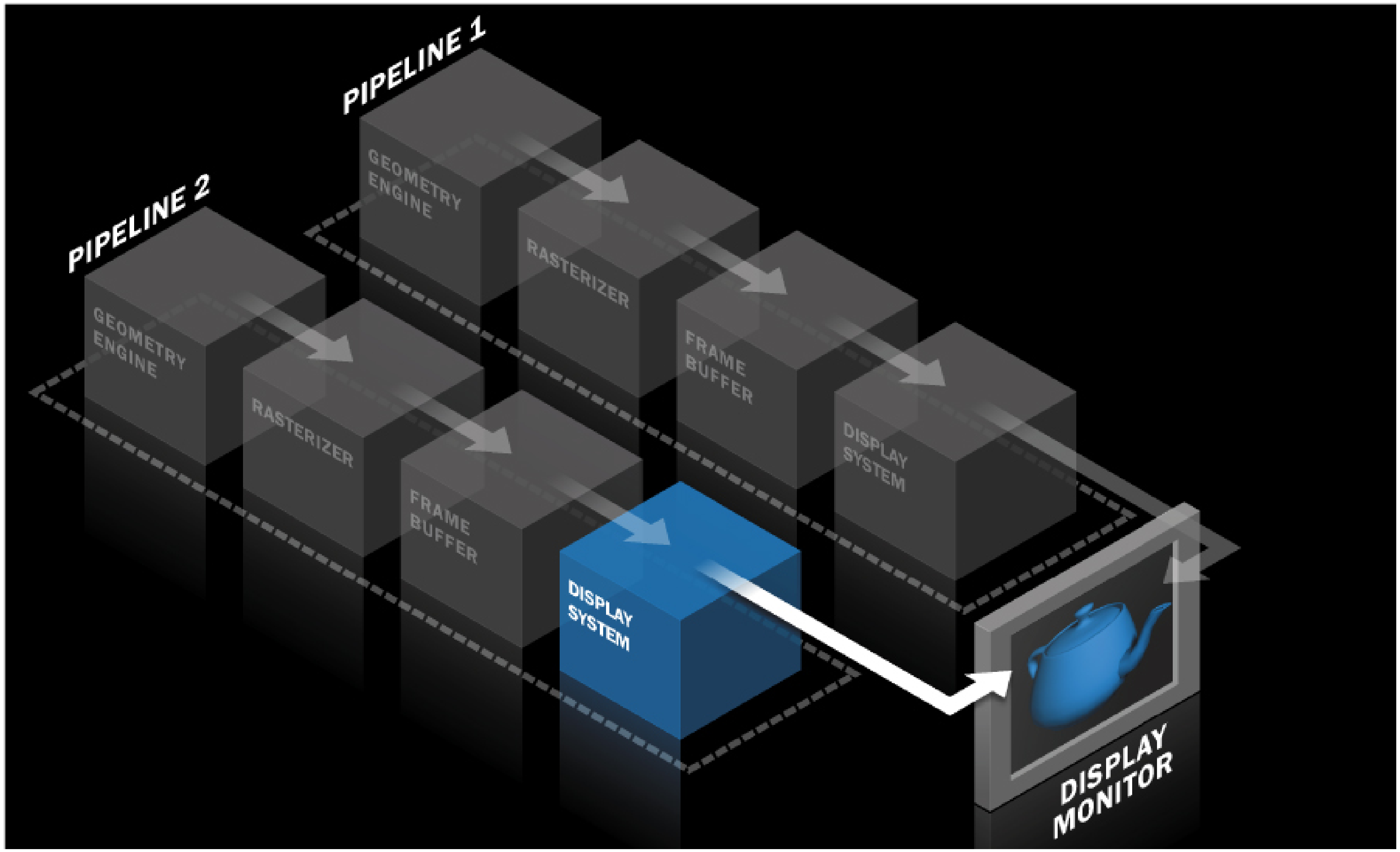
Prior Art


PIPELINE 1
PIPELINE 2
GEOMETRY ENGINE
RASTERIZER
GEOMETRY ENGINE
RASTERIZER
FRAME BUFFER
PIPELINE 1 FRAME
PIPELINE 2 FRAME
DISPLAY SYSTEM
DISPLAY SYSTEM
DISPLAY MONITOR

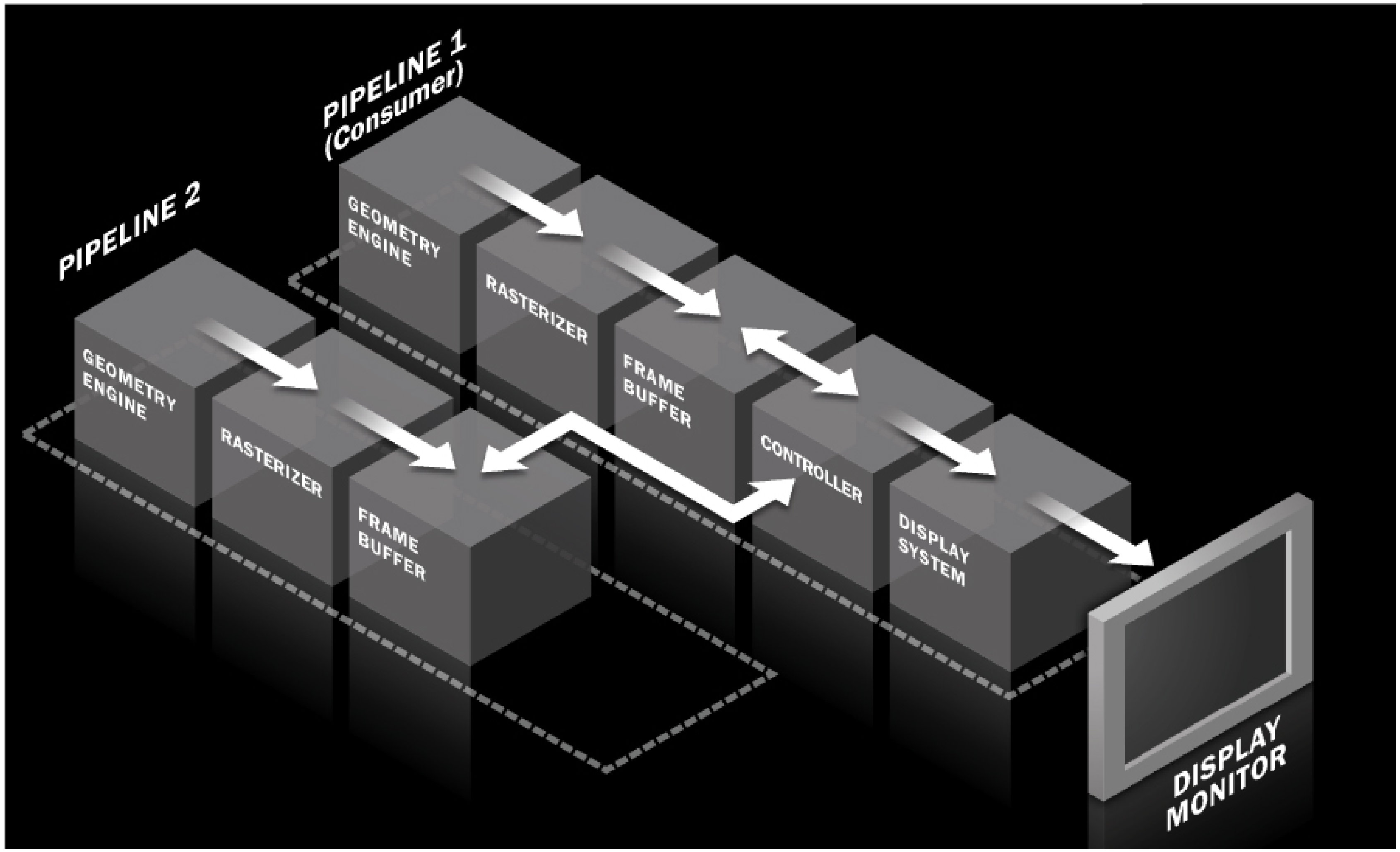
Prior Art


PIPELINE 1
PIPELINE 2
GEOMETRY ENGINE
RASTERIZER
GEOMETRY ENGINE
RASTERIZER
FRAME BUFFER
PIPELINE 2 FRAME
DISPLAY SYSTEM
DISPLAY SYSTEM
DISPLAY MONITOR

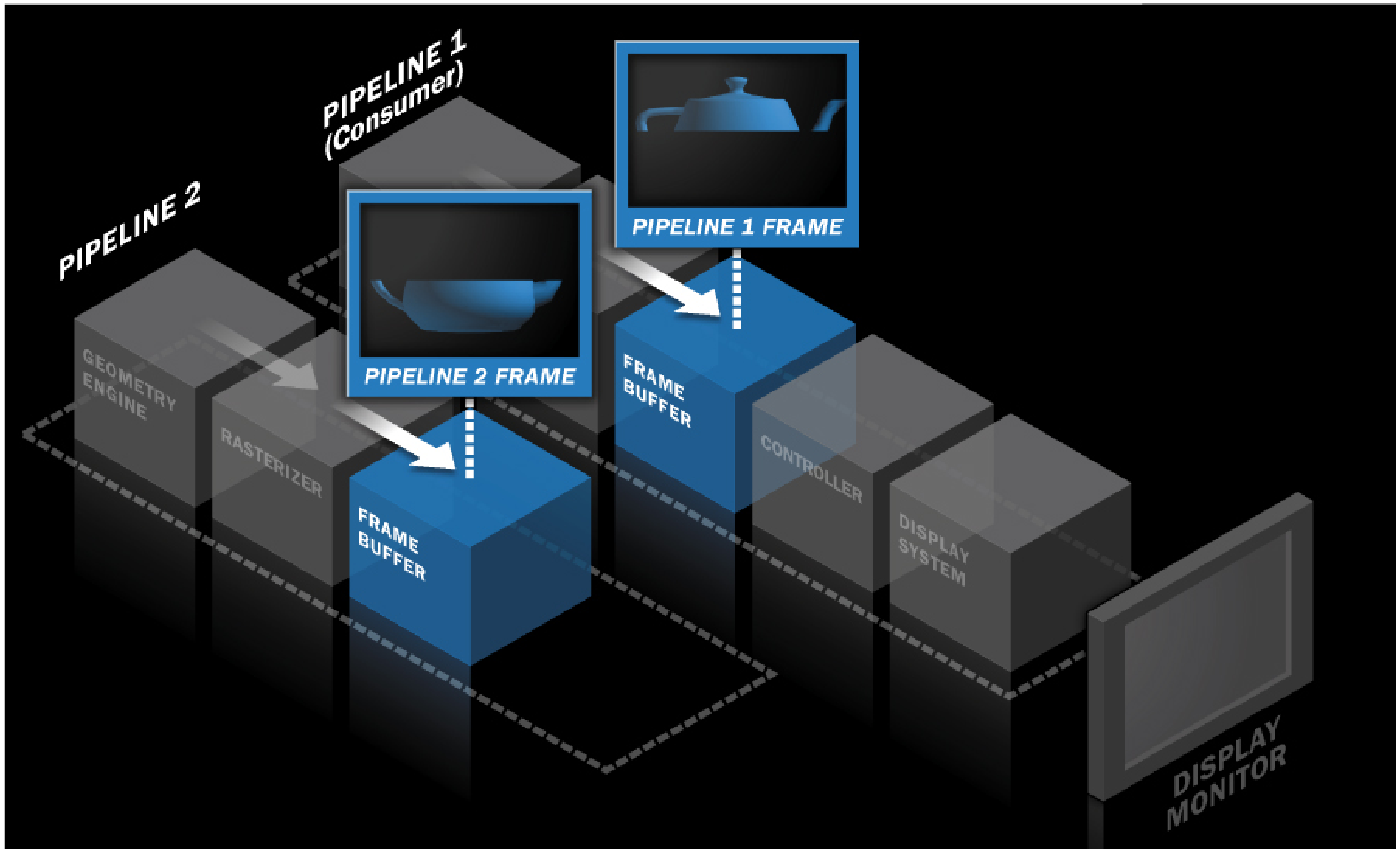
Prior Art


PIPELINE 1
PIPELINE 2
GEOMETRY ENGINE
RASTERIZER
FRAME BUFFER
DISPLAY SYSTEM
GEOMETRY ENGINE
RASTERIZER
FRAME BUFFER
DISPLAY SYSTEM
DISPLAY MONITOR

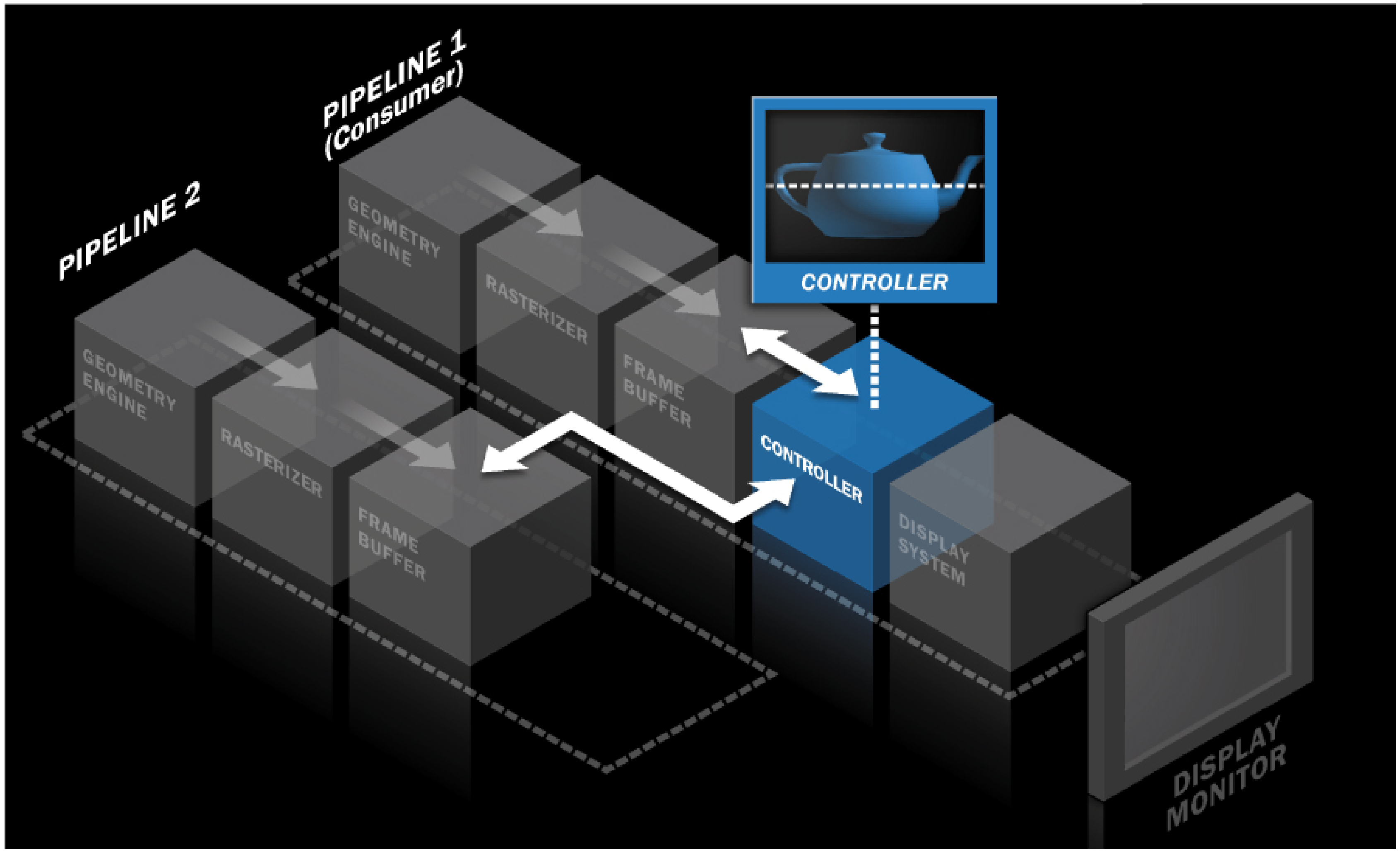
Multiple Rendering Pipelines Increase Processing Speed



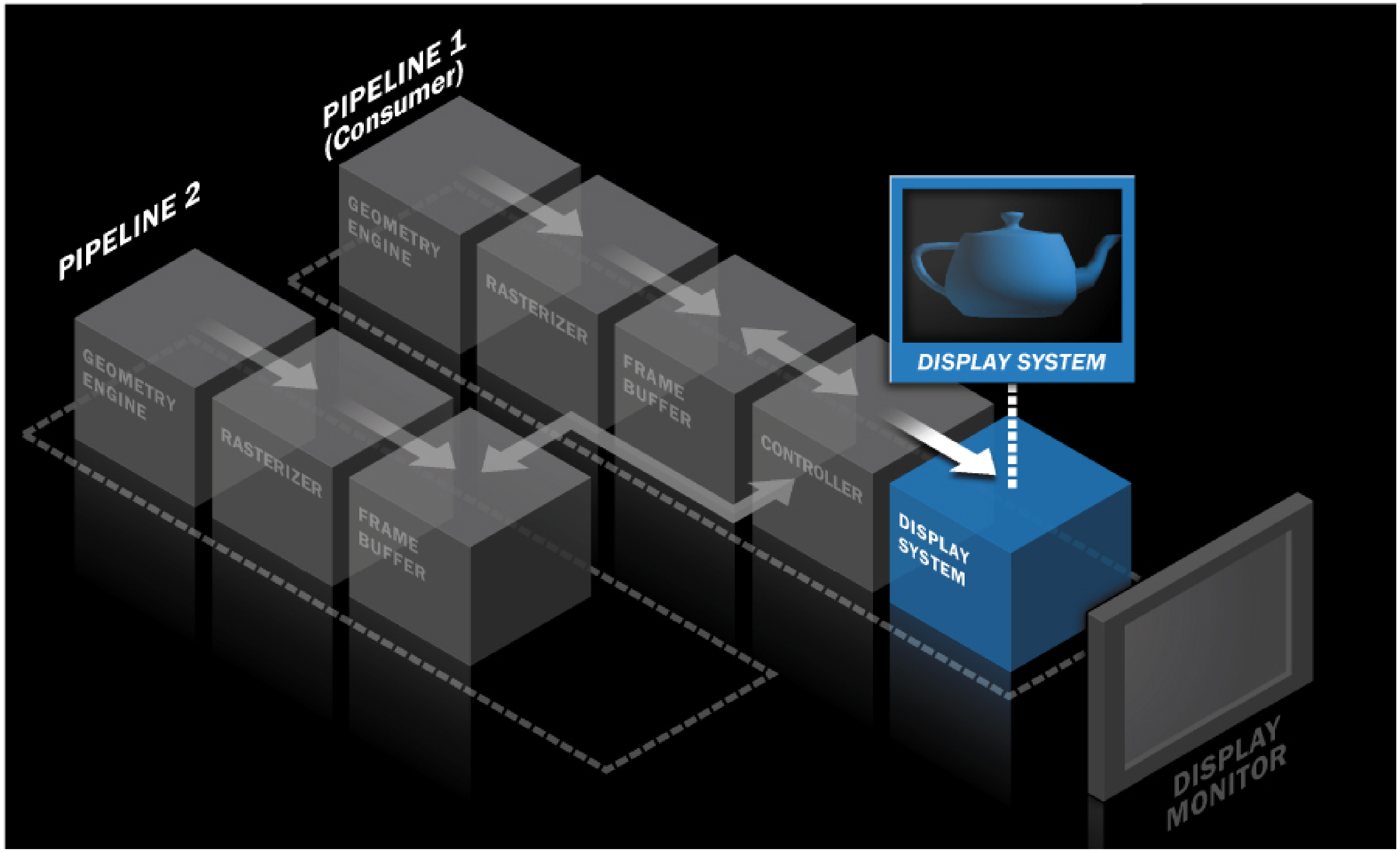
Multiple Rendering Pipelines Increase Processing Speed



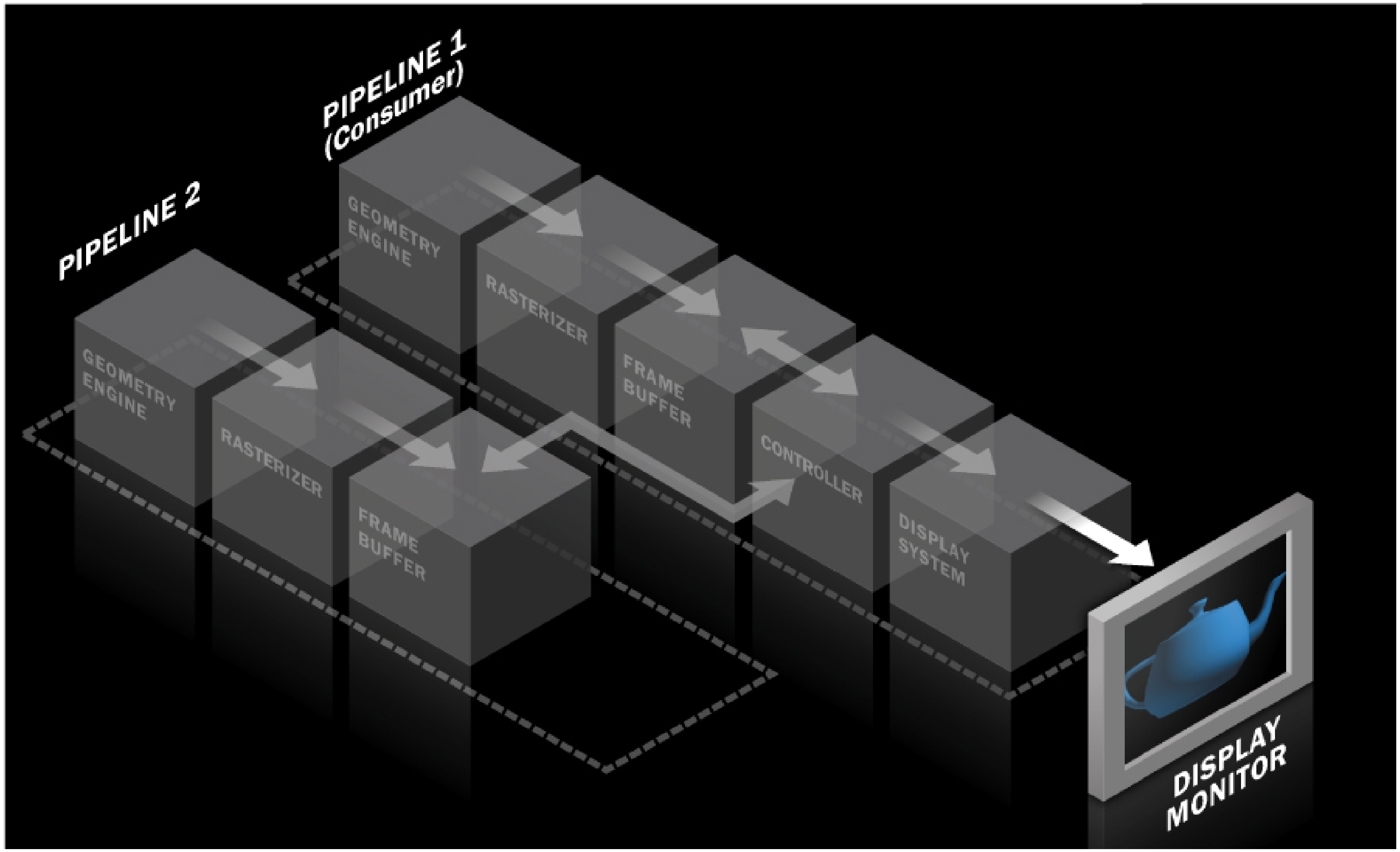
Multiple Rendering Pipelines Increase Processing Speed



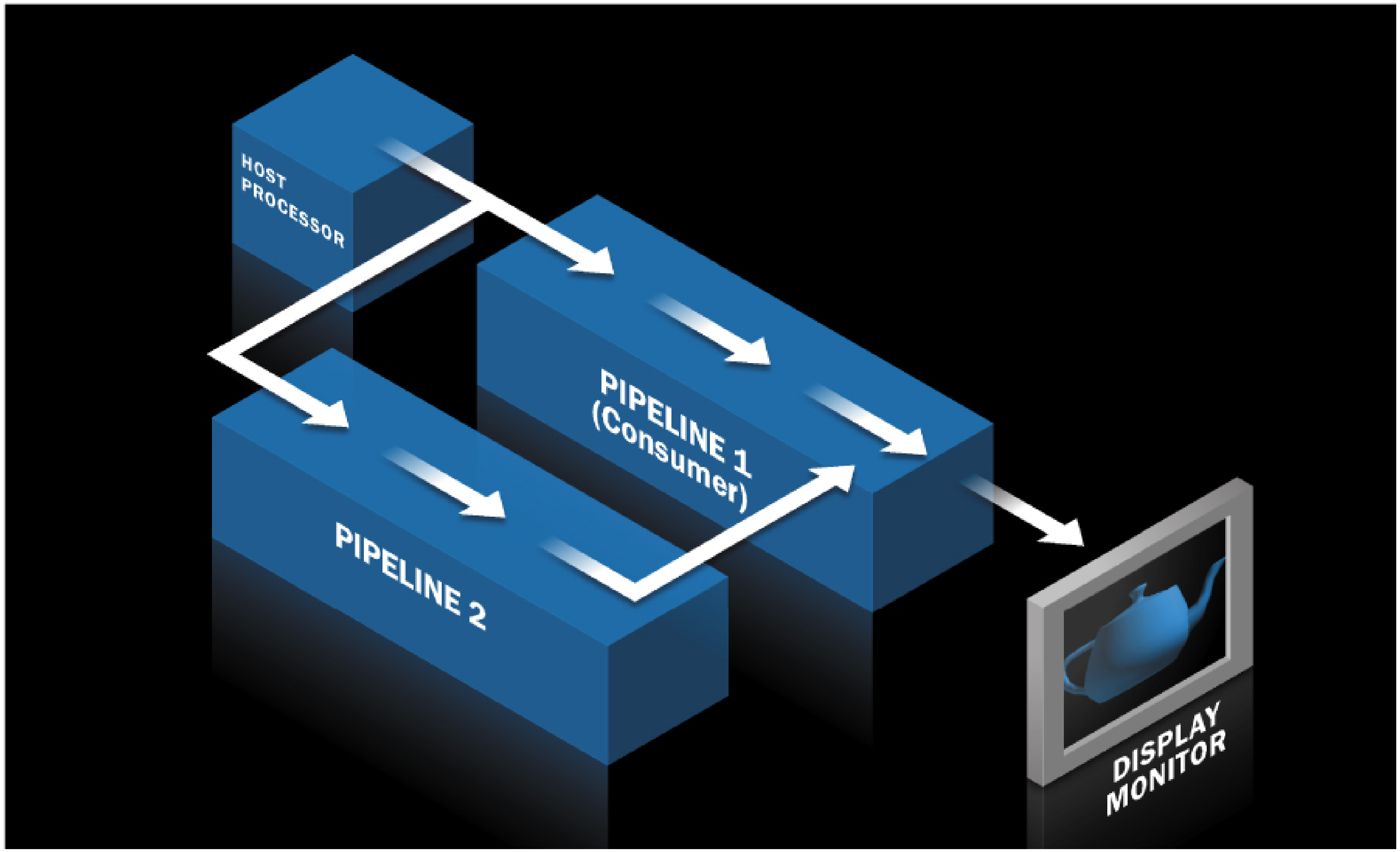
Multiple Rendering Pipelines Increase Processing Speed



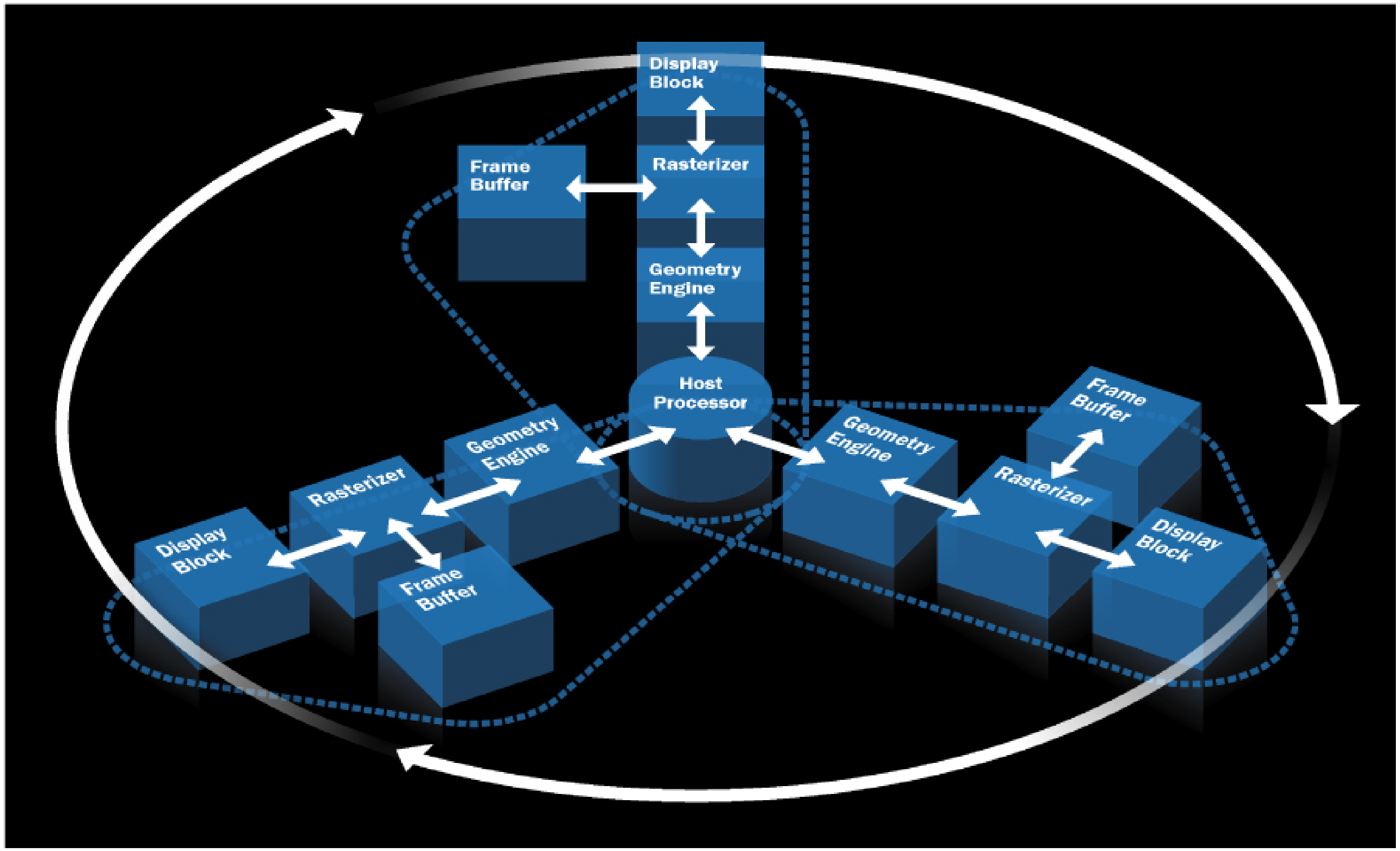
Multiple Rendering Pipelines Increase Processing Speed



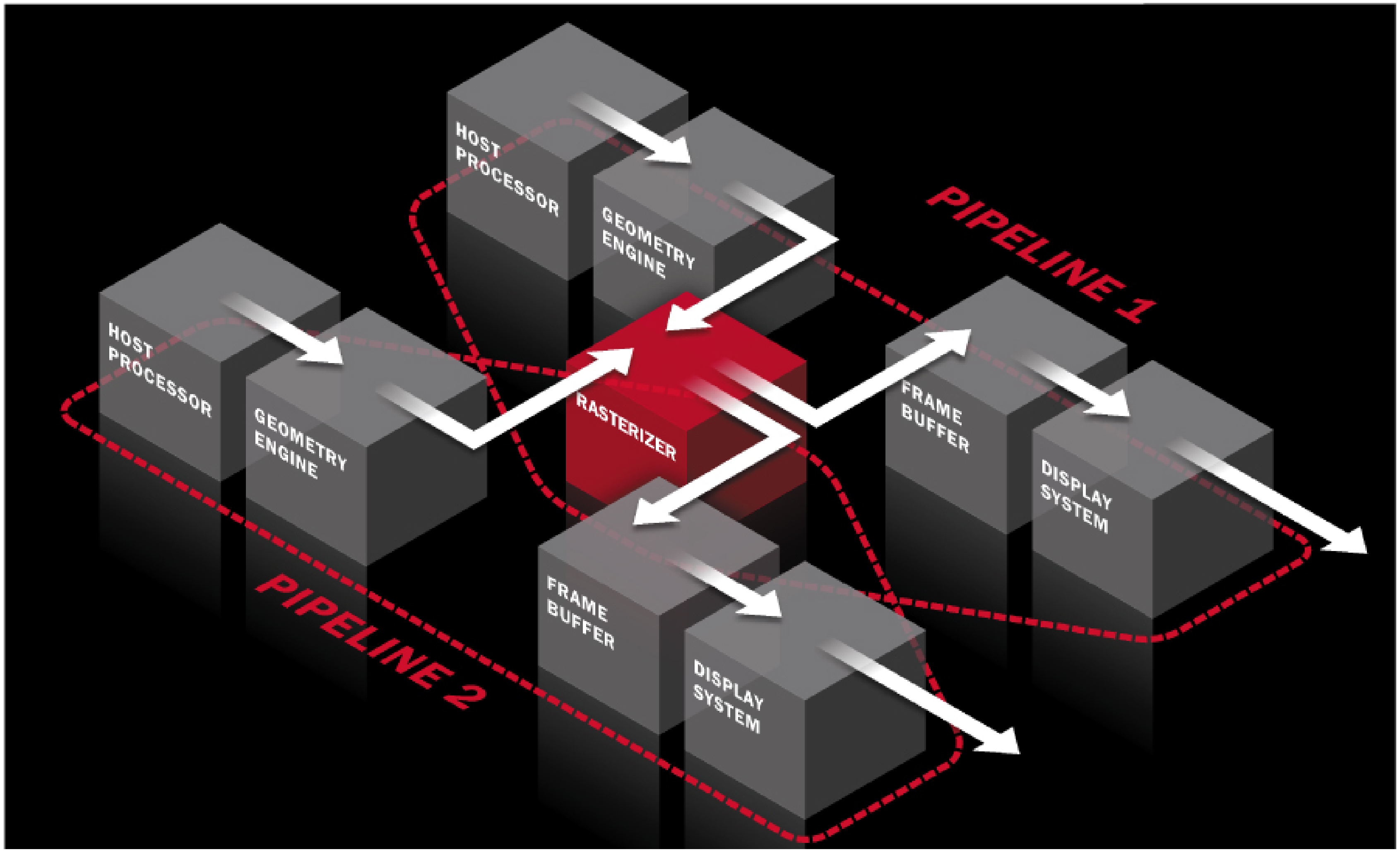
Multiple Rendering Pipelines Receive Commands from a Host Processor



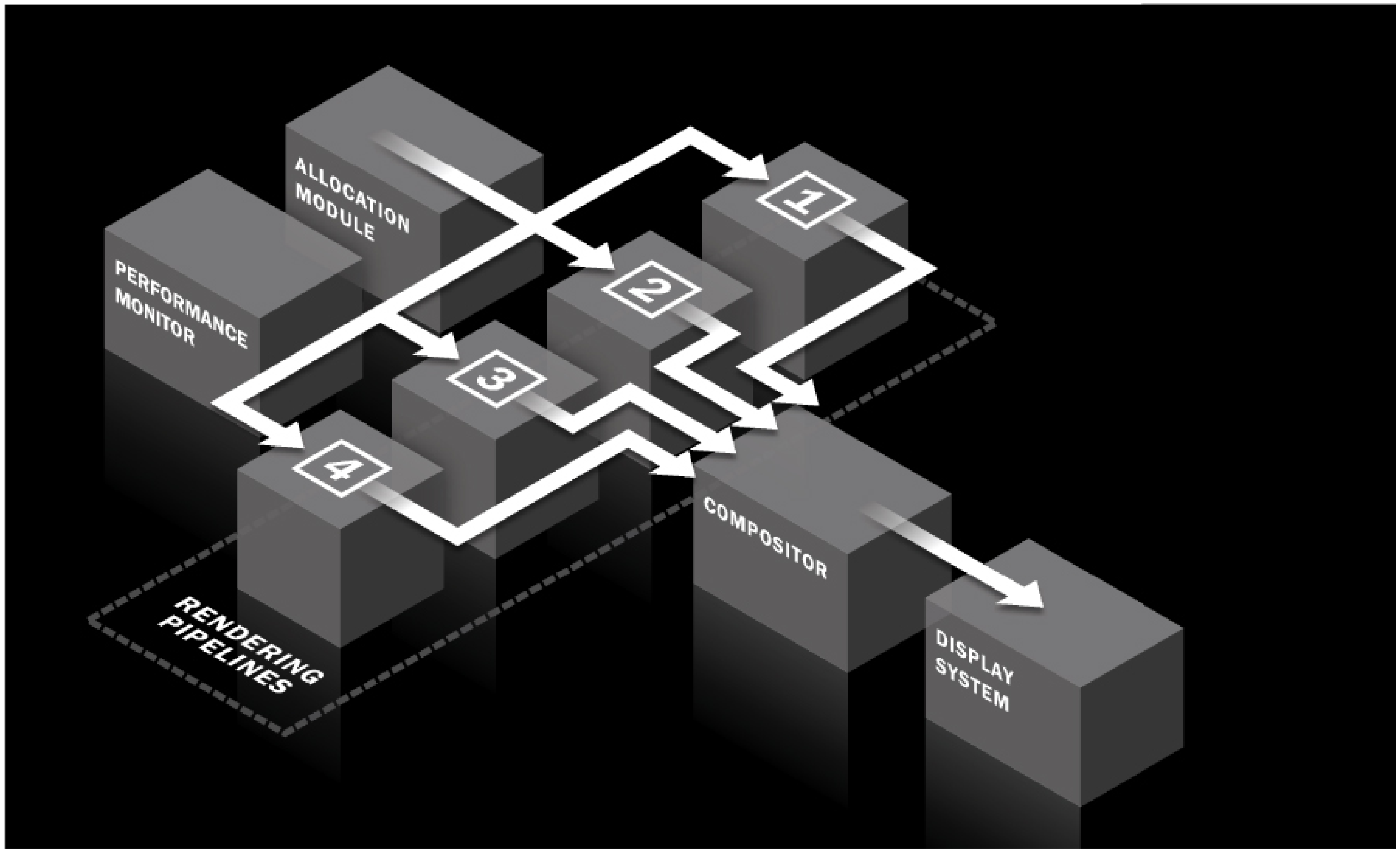
Multiple Rendering Pipelines Receive Commands from a Host Processor



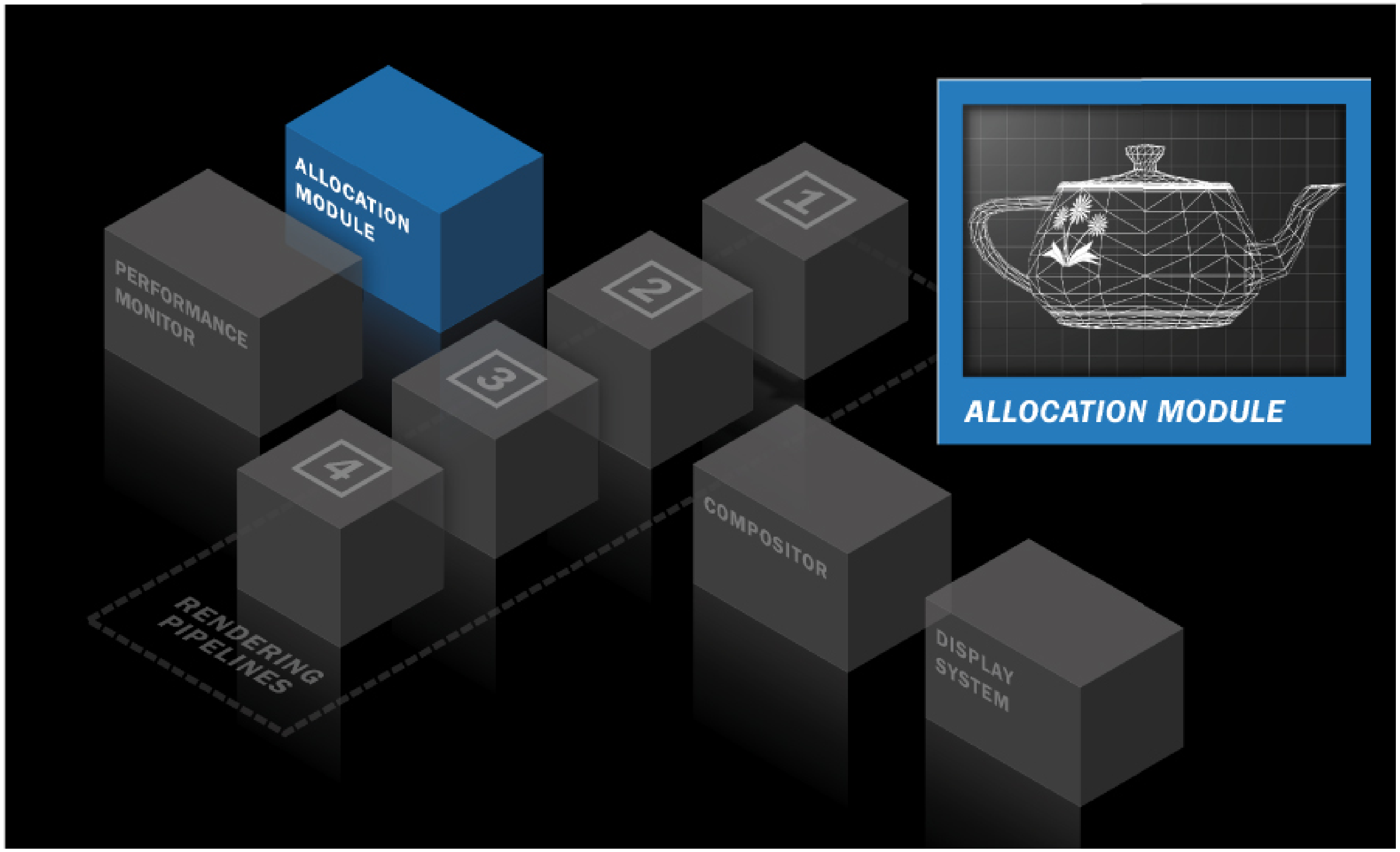
Sharing a Rasterizer Creates a Bottleneck Undermining the ’200 Patent



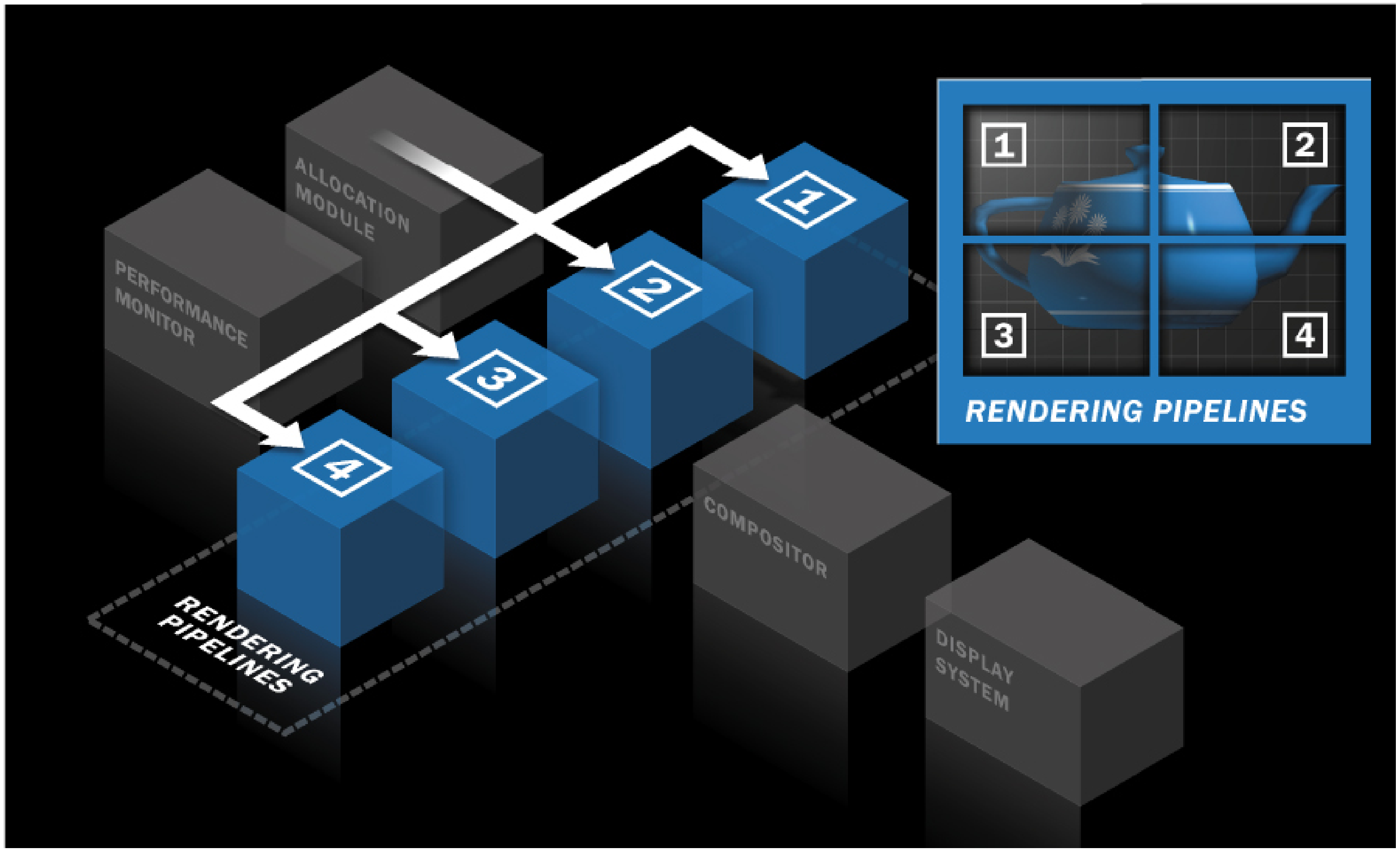
Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



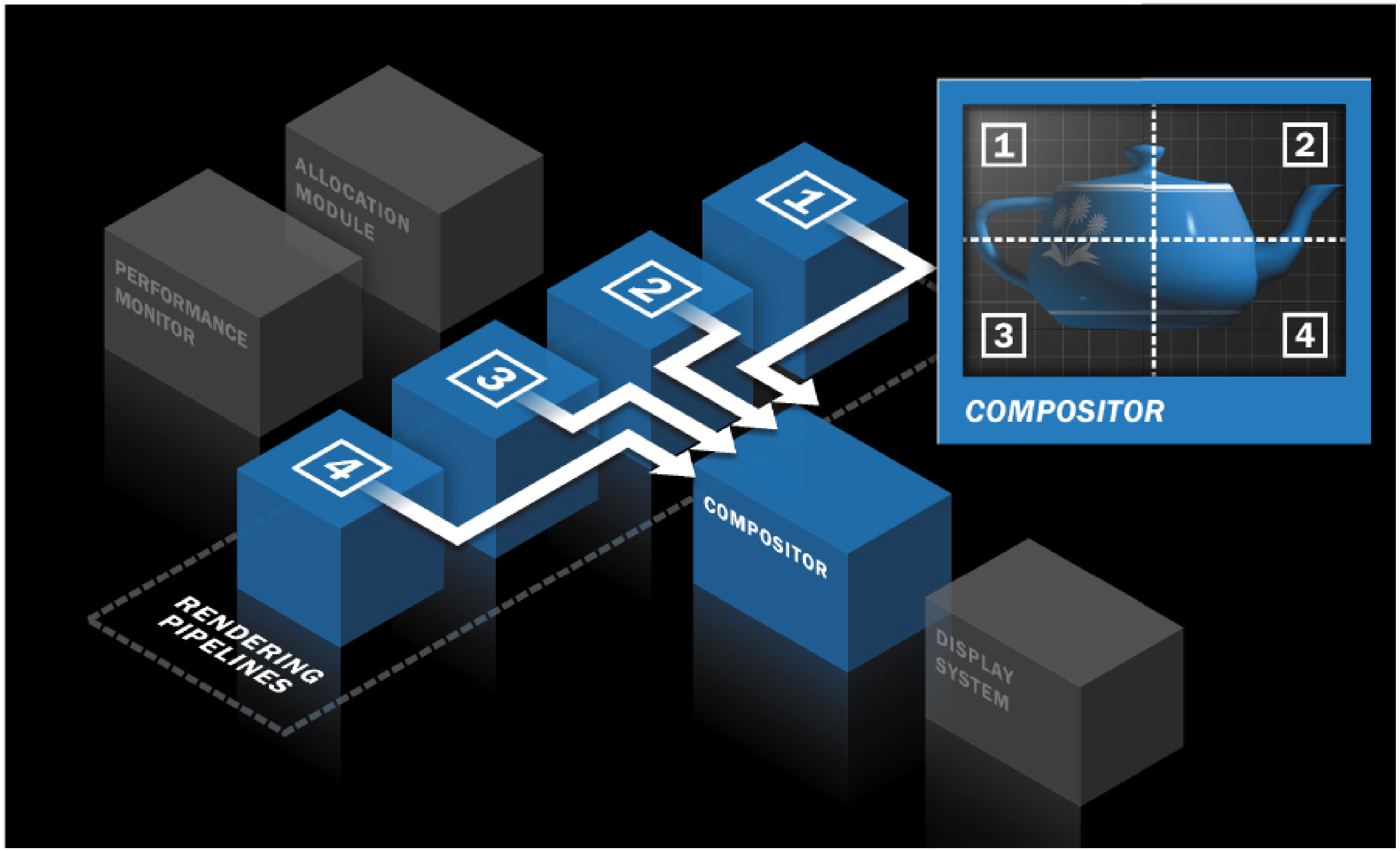
Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



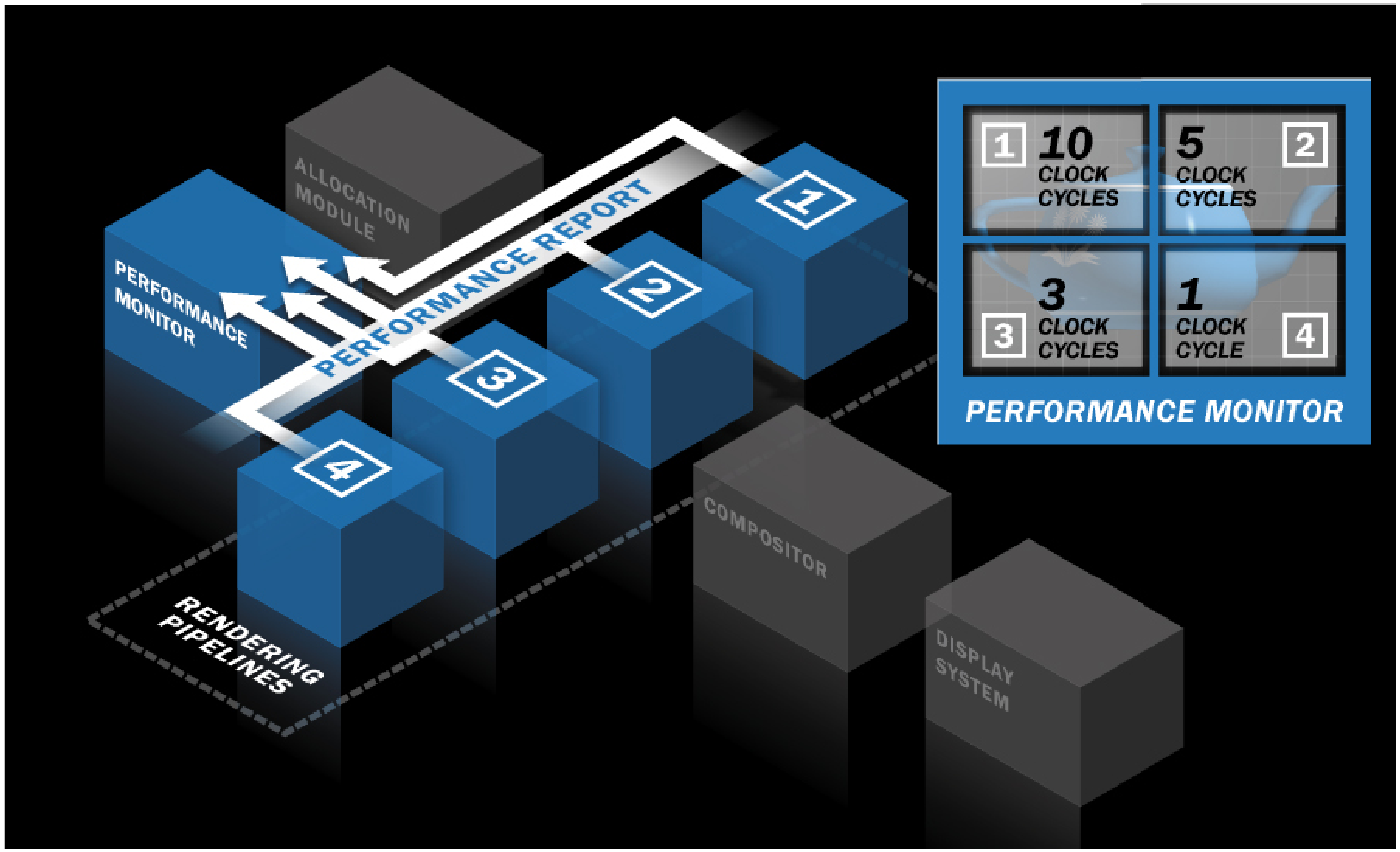
Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



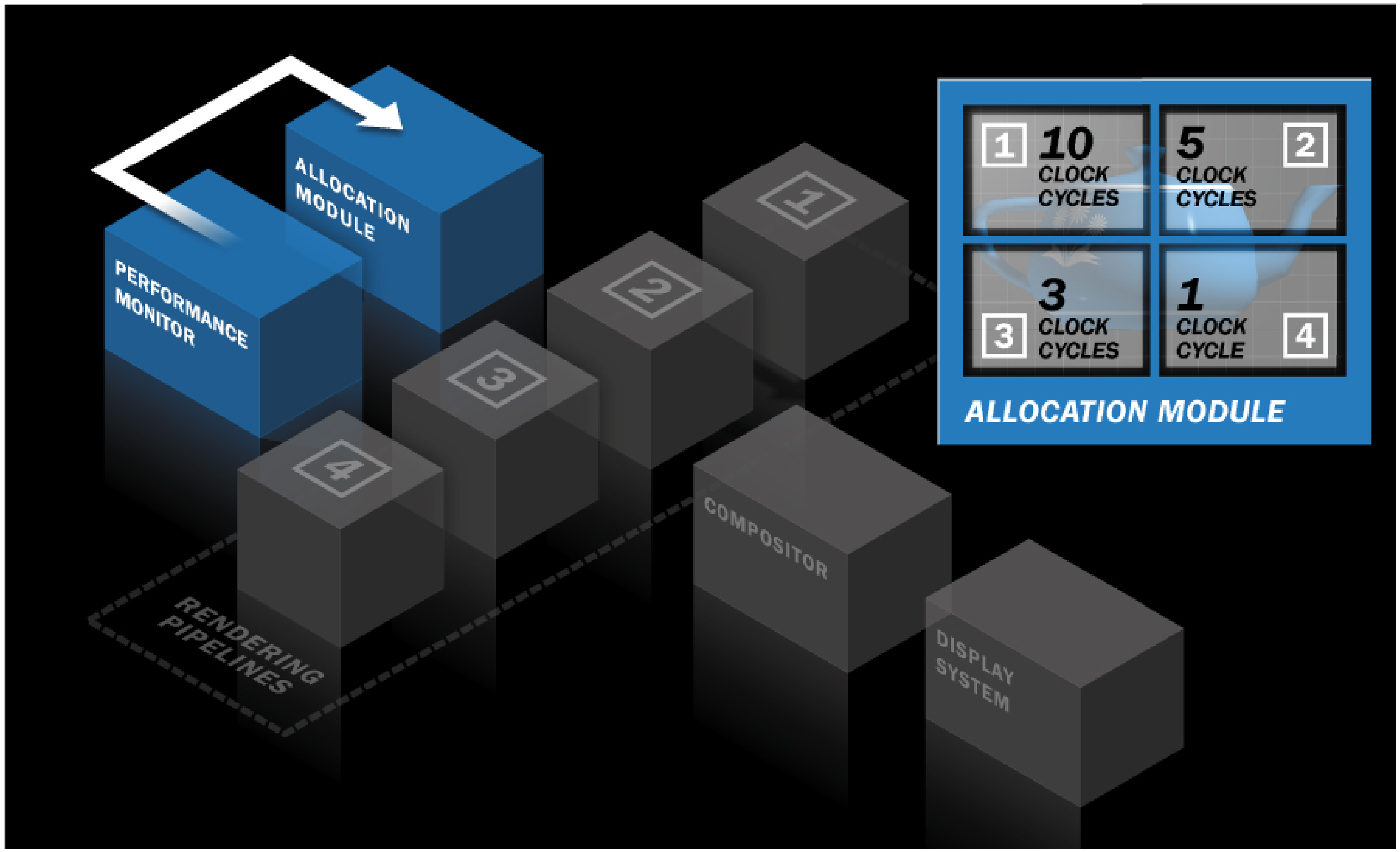
Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



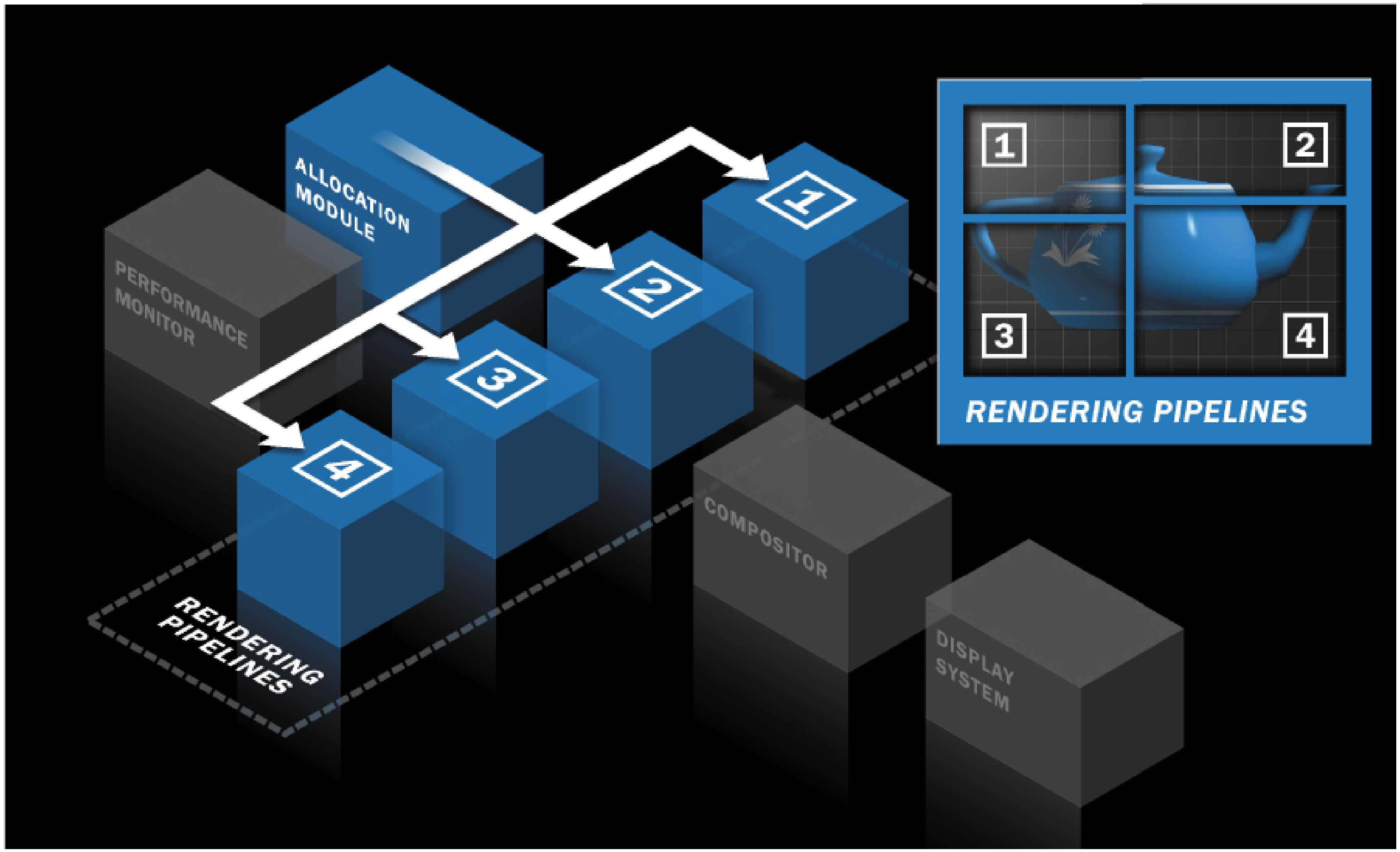
Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



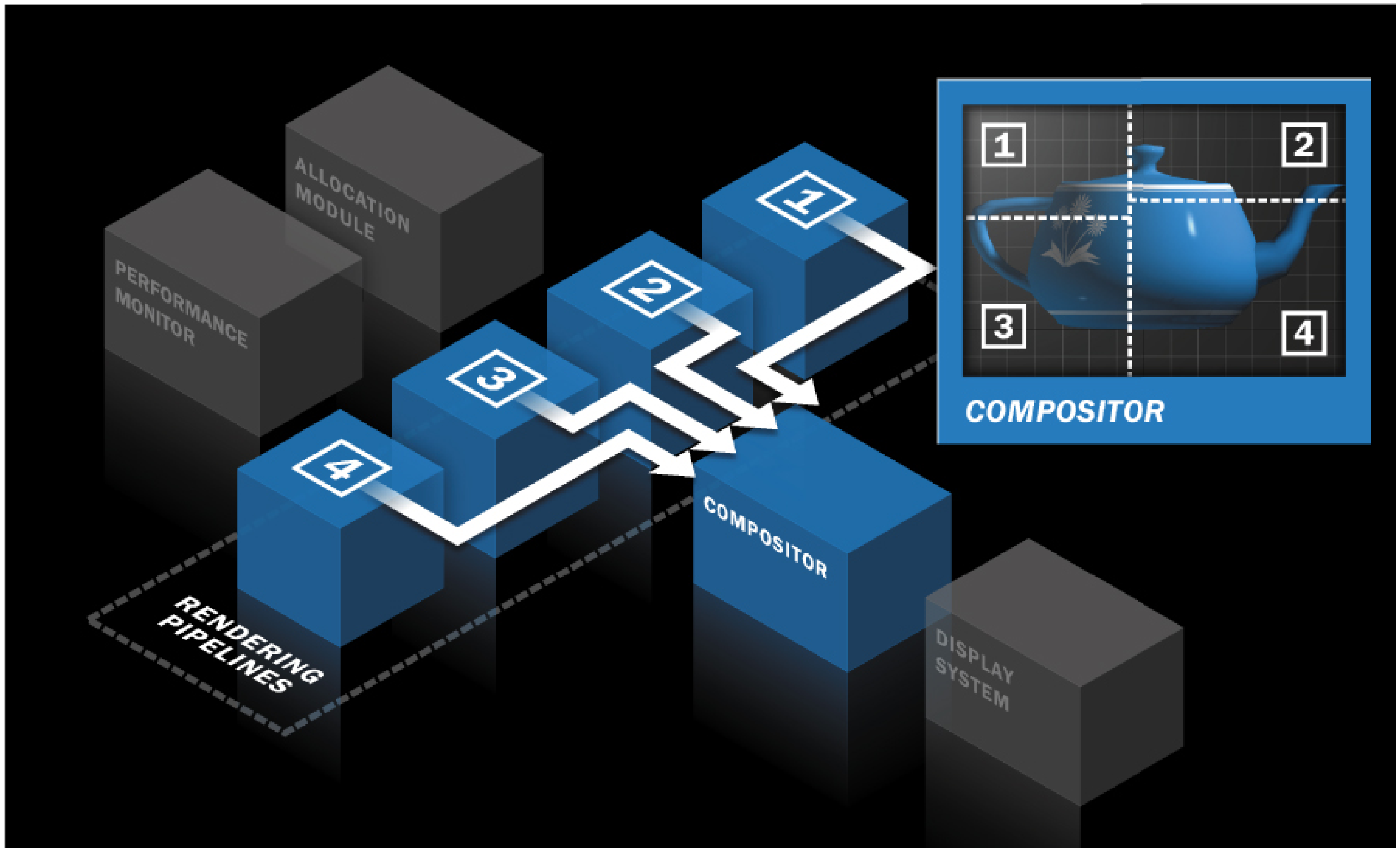
Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



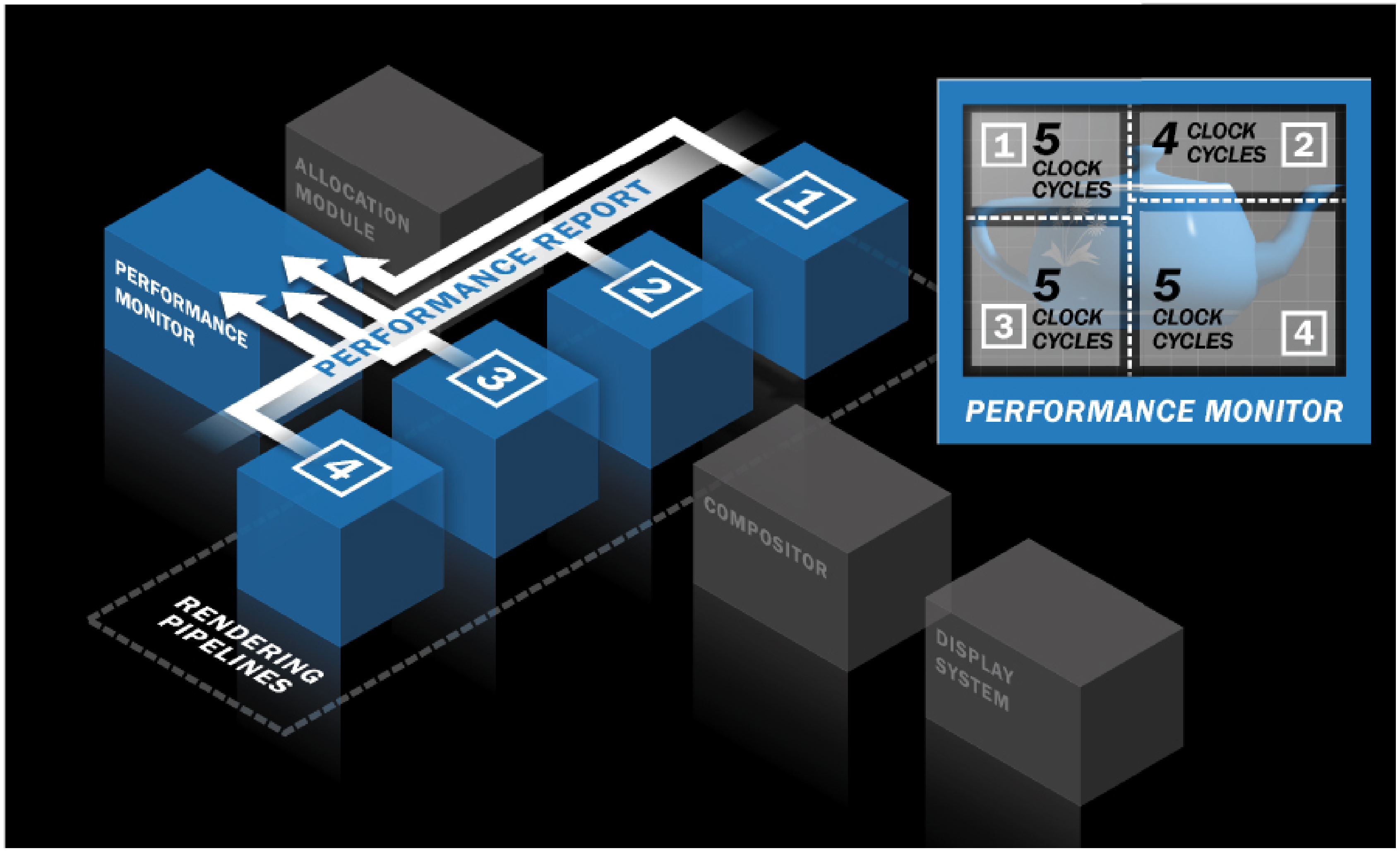
Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance



Balancing the Load Across Multiple Rendering Pipelines Increases Rendering Efficiency and Performance


ALLOCATION MODULE
PERFORMANCE MONITOR
PERFORMANCE REPORT
1
2
3
4
RENDERING PIPELINES
COMPOSITOR
DISPLAY SYSTEM
1 5 CLOCK CYCLES
4 CLOCK CYCLES 2
5 3 CLOCK CYCLES
5 CLOCK CYCLES 4
PERFORMANCE MONITOR