SIGGRAPH2015
Xroads of Discovery

Engine

- XGS - Exient Gaming Software

- Multiplatform modern game engine

- Fast data builds

- 200 million users

Presentation agenda

- Identify the technical constraints

- Environment rendering with LOD and face culling techniques

- Strategies for optimising shaders

- Power management strategies

Technical constraints

- Storage - Initial download 100 mb

- Platform hardware - 4 API's, countless handsets

- Maintainability - Long term free to play product

- Nature of the product - Animated and fast

Environments

Factors effecting Environments

- Limited storage

- Draw call limited

- Library based asset system

- One model to rule them all

Existing Env Pipeline

- PVS visibility system

- 3d tools allow artists to place assets from a library.

- Baked vertex colour lighting with perspective shadow maps

Environment Lods

- Albedo and lighting baked into vertex colour

- Redundant model data is removed at build time

- All distance geometry can be batched in a single draw call

- Pro: No geometry change means no silhouette pop

- Con: Maintaining lighting and geometry between the two LOD's

Environment: Geometry Culling

- Backface removal - storage gain

- Store invisible geometry per PVS section - performance gain

- Fidelity - the results can be improved with more samples
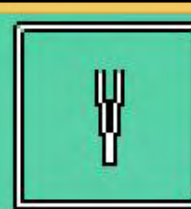
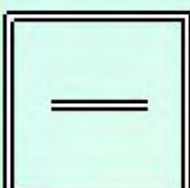- Savings - 10 - 20 % reduction in faces

FLY

HIDE

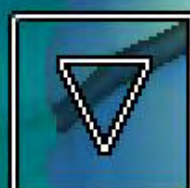Camera Coords: -405.21, 1558.83, -122.99

EXIT

RUN

X : -292.0
Y : -52.4
Z : 784.4
H : -65.1
P : -43.8
FOV : 68.8

DROP

Shaders

- Shaders are on EVERYTHING !

- Not just for visual quality - We can use them for visual communication

- Shader efficiency is vital to providing good performance

- Different shader techniques are needed for lower spec devices

Shader Lodding

- Designate every device a shader LOD index (1 to 4)

- Create a set of shaders for each LOD

- The device lod index forces the game to render with the appropriate shader alternative.

- If a shader alternative is missing, the next highest index is used

Multiple shaders per scene

- Compiler generates a unique shader set per scene

- Uniforms change to constants

- Pro: Improved performance
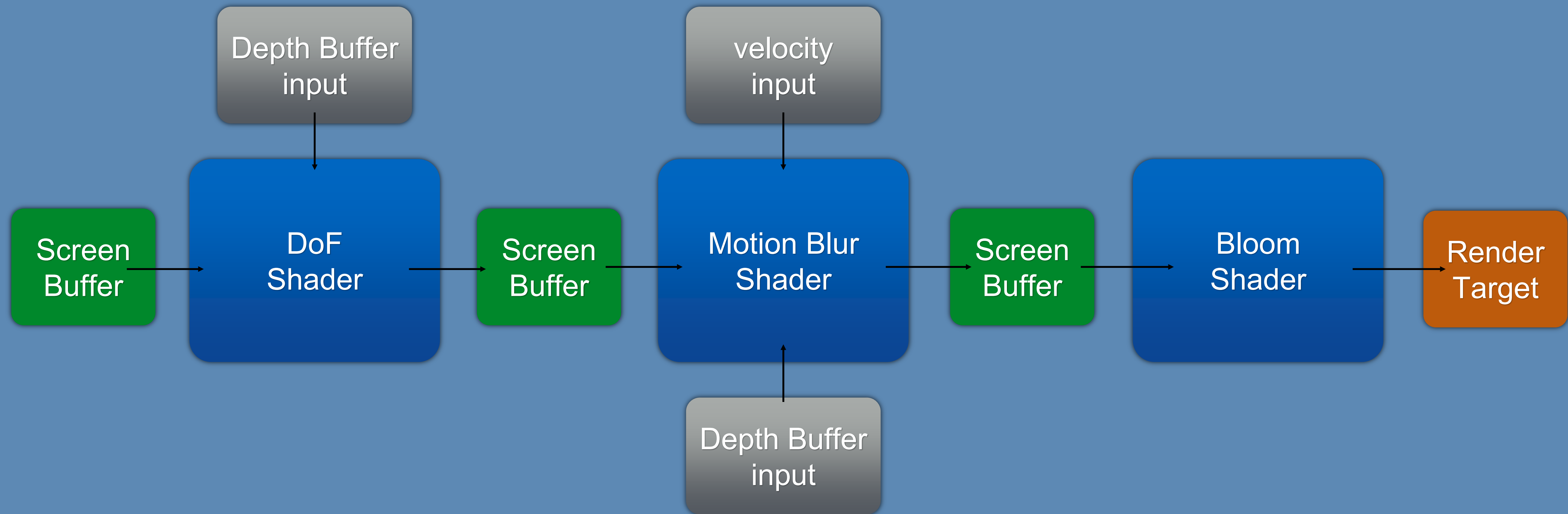
- Con: Large database footprint

Post Processing

- Bloom, Depth of Field, Motion Blur.

- Provide final-stage visual styling per scene.
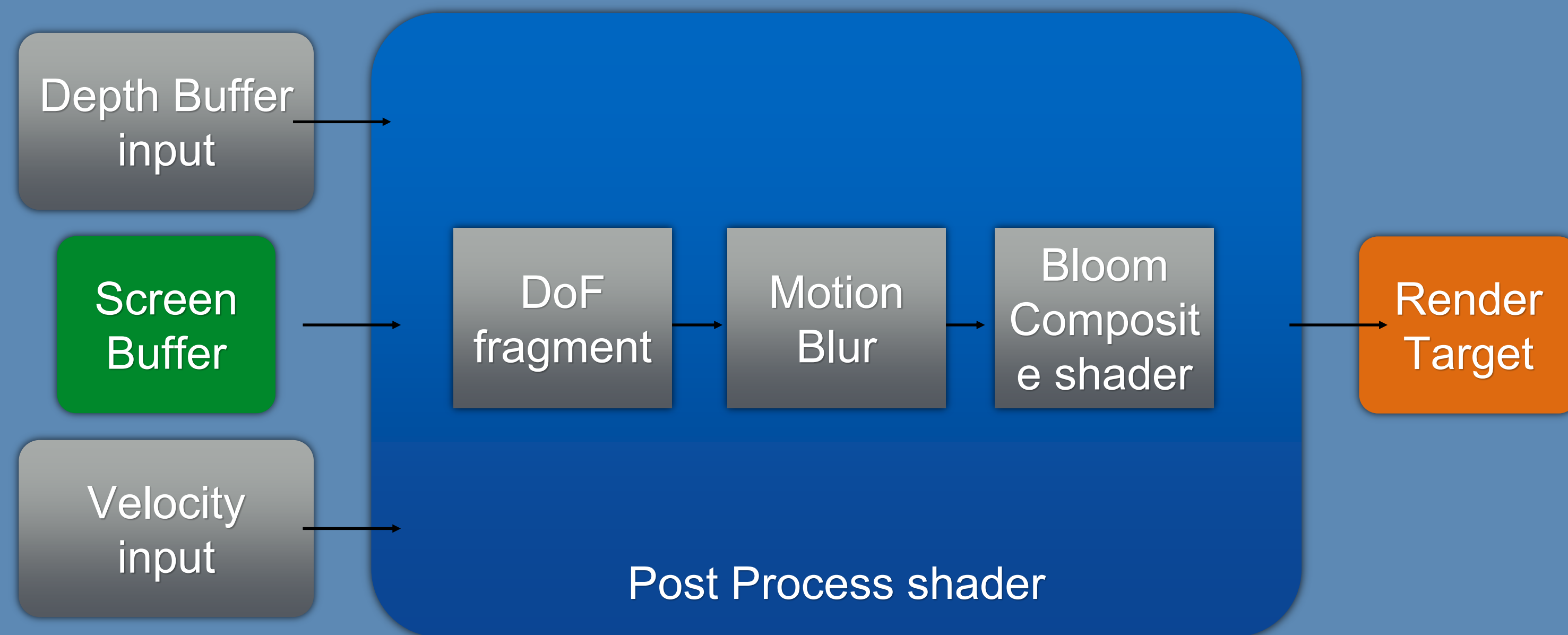
- Full screen per pixel effect

Traditional post processing

Composite Post Processing

Power management strategy

- Low/High graphics options

- Lower frequency of tasks - animation skinning/ Physics

- Make savings when the game is paused

- Rendering Front ends to rendertarget, update elements as required

- Do we really need 60 fps on all screens?

Take aways

- Taking a straightforward approach to rendering

- Find ways to optimise that will give the biggest impact for the least disruption of the pipeline

- Breaking tradition - current rendering trends aren't always the best

- Trust and teamwork

Thank you!

Visit us at www.exient.com

simon@exient.com