



FABRICENGINE®

It's Time For Real-Time



Introduction

What will pipelines look like 5–10 years from now?

How do studios get there?



State of DCC tools

Poor performance of commercial tools

- a limitation of older software, not modern hardware
- too many black boxes

Maya/Max/Softimage/Nuke Are Fixtures

- Artists don't like change
- Years of R&D investment



Studios are addressing this themselves.

Building custom frameworks:

- High cost of development and maintenance
- Often creates a new bottleneck between TDs and R&D
- Frameworks often become production–specific
- Massive duplication of effort across the industry

It's not even their core business!



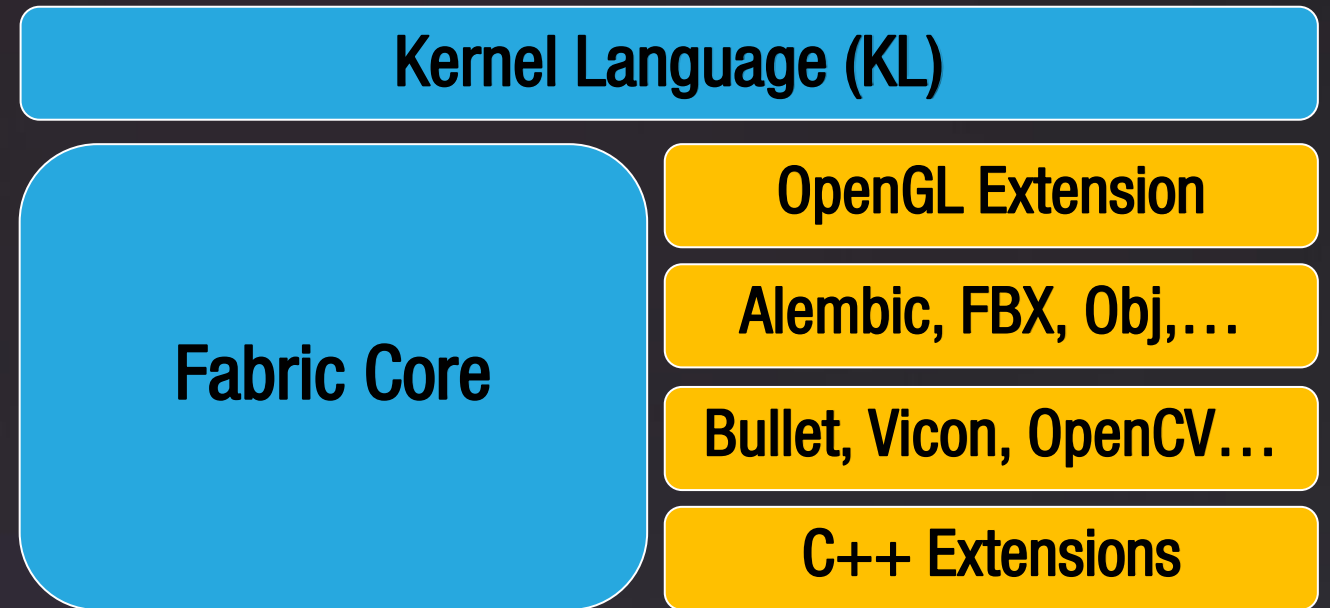


FABRICENGINE®

Core Architecture

FABRICCORE

- Compute Engine
 - Multi-threaded
 - Cross-Platform
 - Uses LLVM
 - Extensible
- Kernel Language
 - Accessible to TDs
 - Tightly scoped
 - As fast as multithreaded C++



The DCC performance problem

PYTHON

Dynamic

Rapid Iteration

But Slow

C++

Compiled

Slow iteration

But Fast (sometimes)



What we wanted

~~PYTHON~~

Dynamic

Rapid Iteration

~~But Slow~~

~~C++~~

~~-Statically Compiled~~

~~-Slow iteration~~

Fast



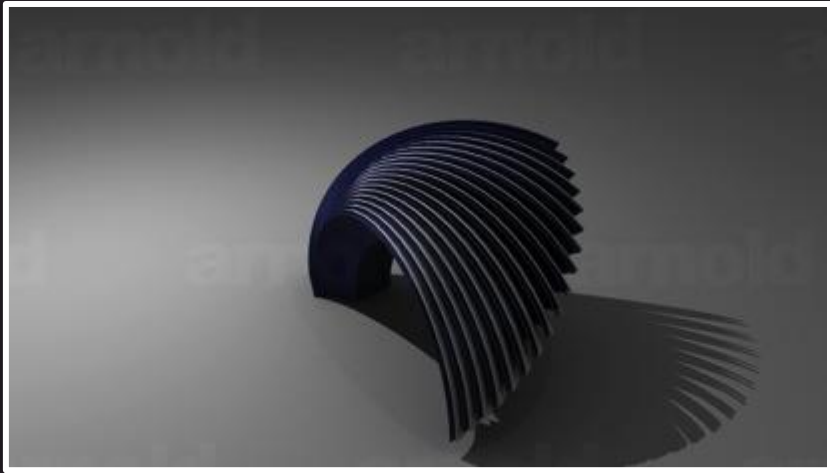
Kernel Language (KL)

Dynamically Compiled (LLVM)

Easy to learn (if you can write Python, you can write KL)

As fast as highly optimized C++





FABRICSPLICE

R&D Encapsulated

Splice IDE

KL Operator

KL Operator

KL Operator

Splice Plugin – DCC, Custom DG Framework

Host DCC

Host Dependency Graph

Native Code
(CPU)

Creation Core

Kernel
Language

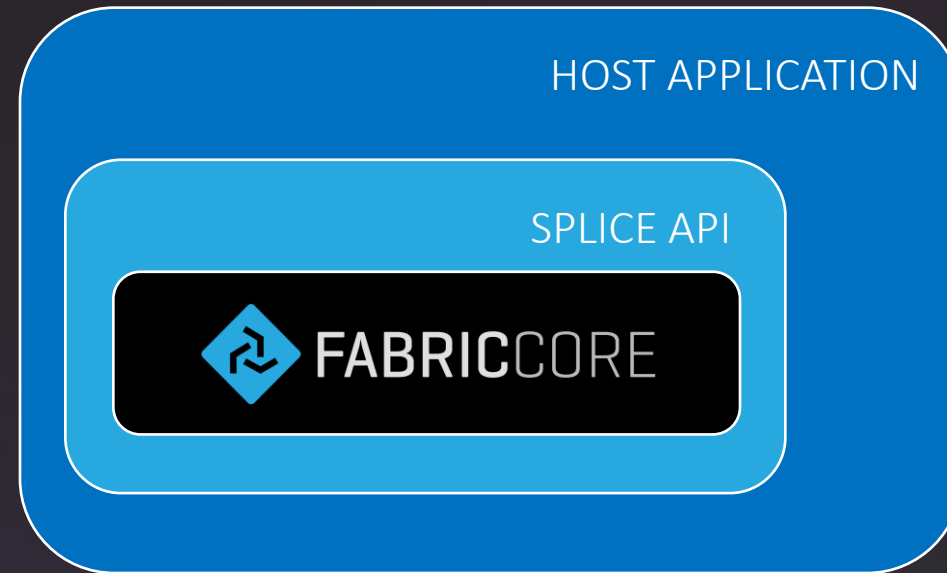
Dependency
Graph

OpenGL Extension

Data Format Extensions

Third–Party Lib Extensions

Custom Extensions





- Scripted Operators as fast as C++
- TDs can write KL code

- Deformers
- Procedural Geometry
- Procedural Rigging
- Solvers
- Simulation
- ...



FABRICSPLICE

