

Game Development

Harder Than You Think

Historie

Før: Kode som kjører raskt på ønsket maskin

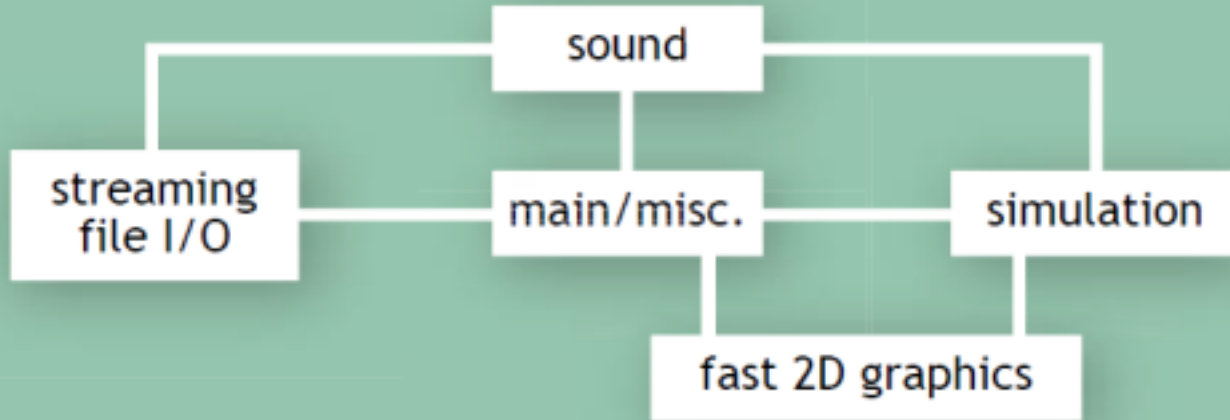
Nå: Kode som kjører

Utfordringer

- Prosjektstørrelse og kompleksitet
- Domenespesifikke krav

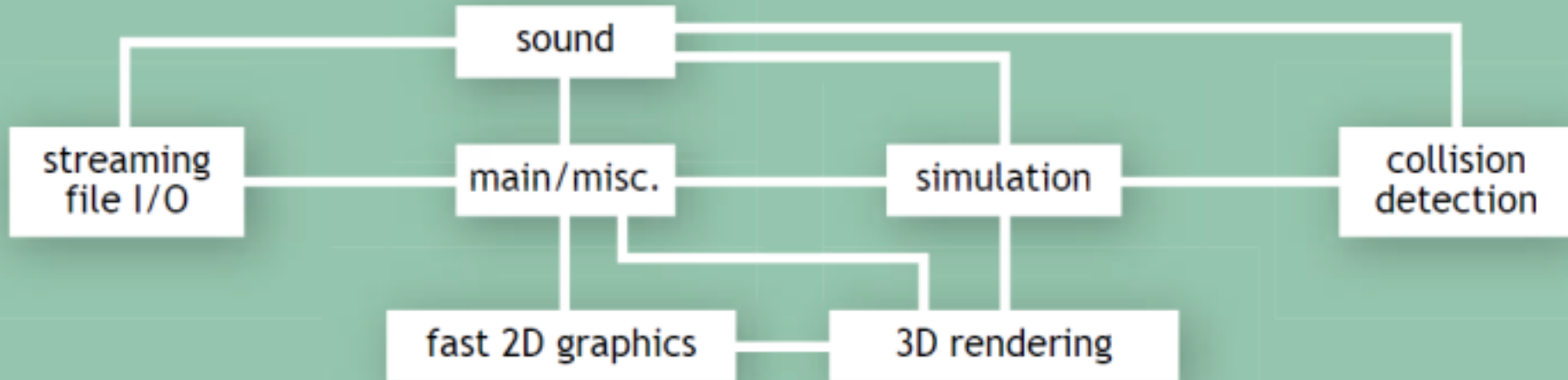
Prosjektstørrelse og kompleksitet

A 2D Game Circa 1994

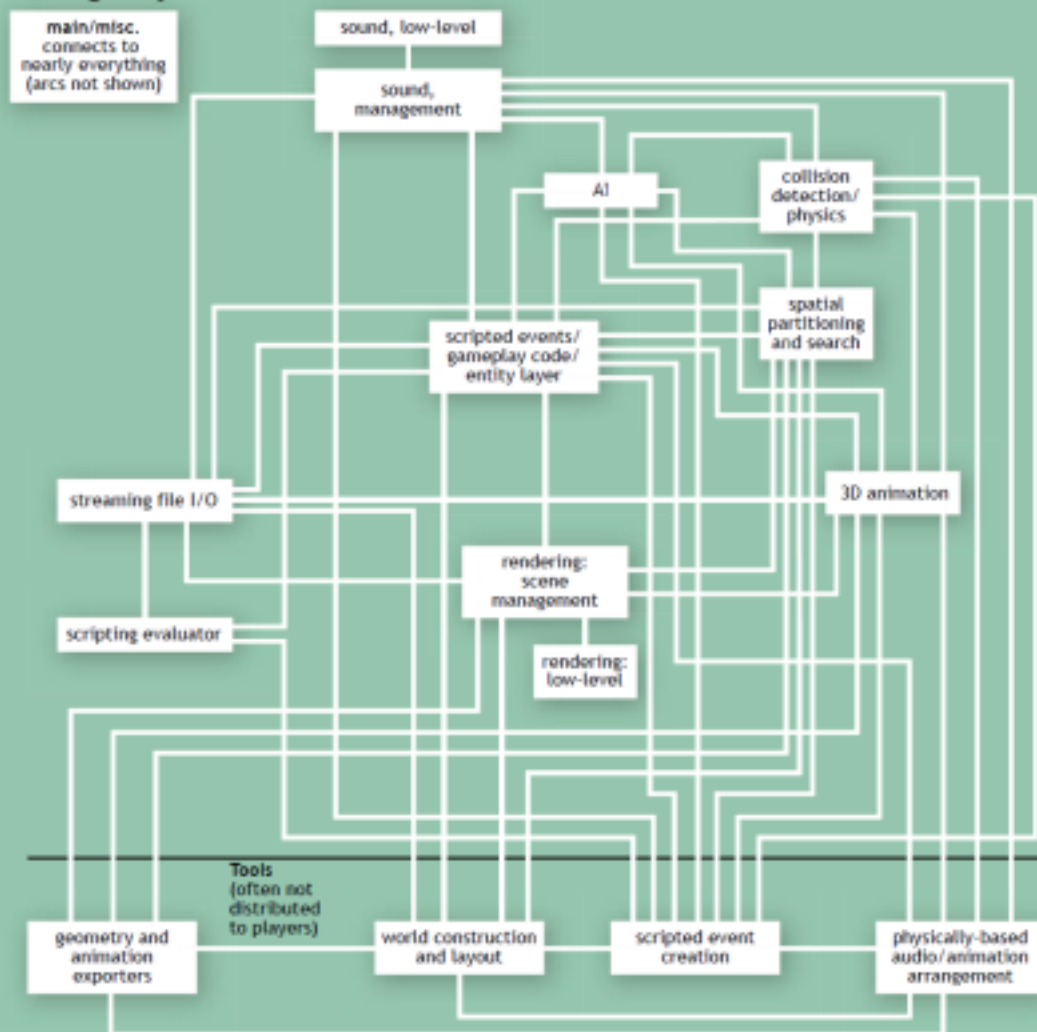


Prosjektstørrelse og kompleksitet

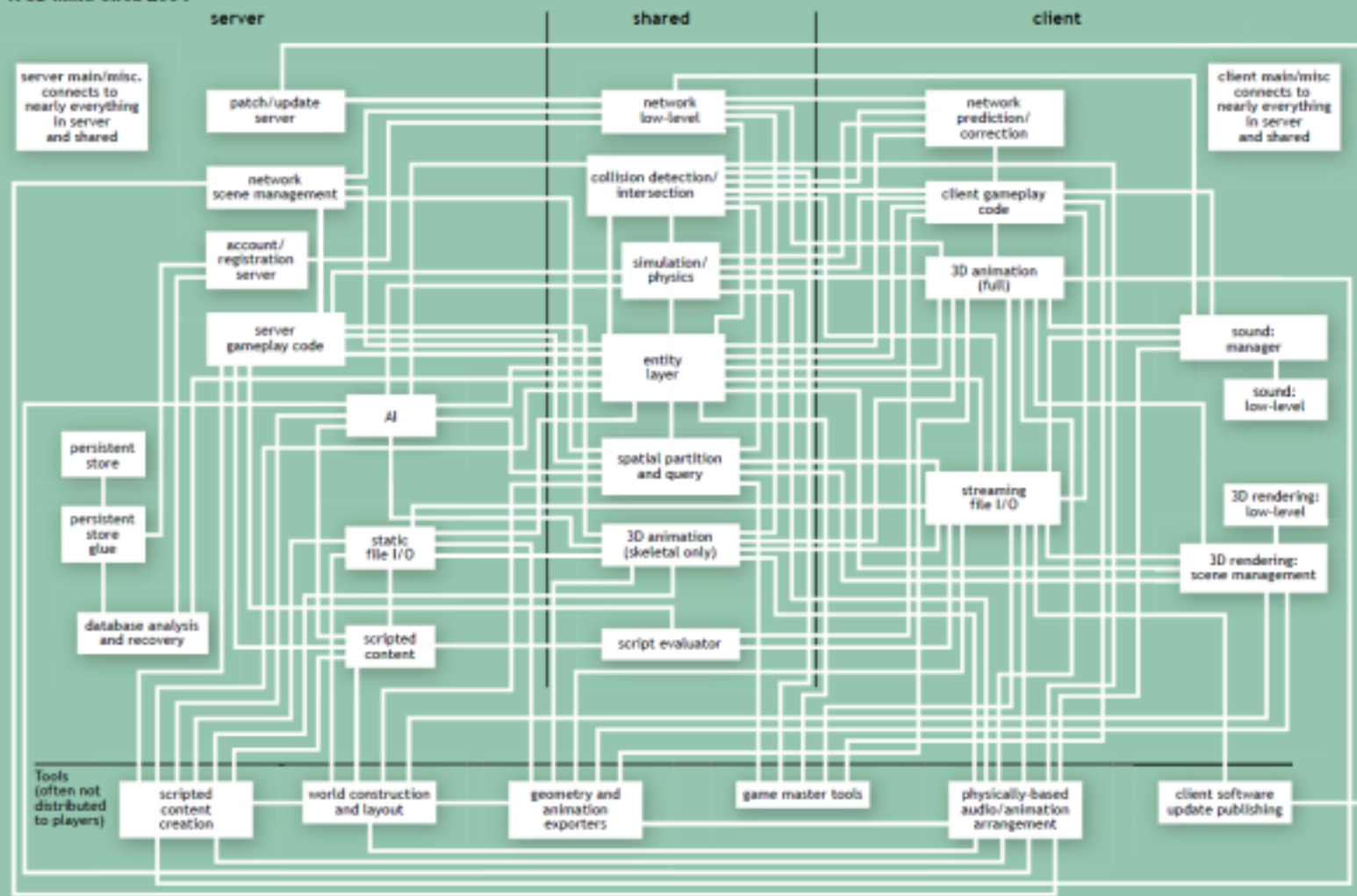
A 3D Game Circa 1996



A 3D Single-Player Game Circa 2004



A 3D MMG Circa 2004



Verktøy

Microsoft Visual Studios (C++)

Ikke utviklet med tanke på spill

Konsoller vanskelige pga. kort livssyklus

3D programmer laget for film (Maya, 3D Studio MAX)

Mer fokus på spill nå

Workflow

Lange kompiler/rediger/debug-sykluser

- C++ har lang kompileringstid
- Ofre kjøretid for raskere kompilering

Mange plattformer

- Kompilere alle tar tid

Tredjepartskomponenter

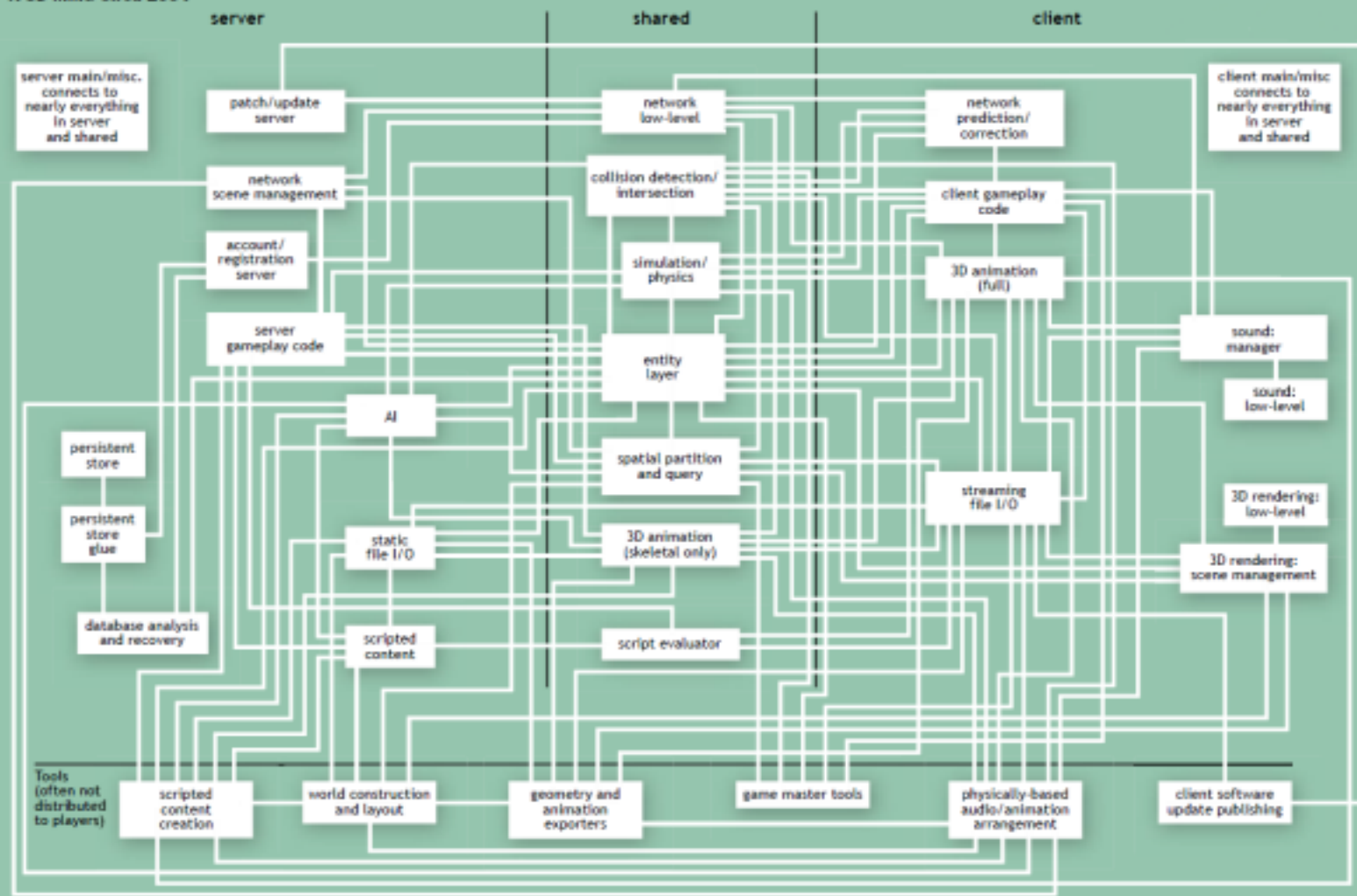
Eksempler

- Lyd, rendering (grafikk)
- Animasjon (skeletal, morph targets)
- Kollisjonsgjenkjenning og fysikk (vanskelig å lage selv)

Utfordringer

- Integrering av komponentene
- Ikke perfekte
- Er det verdt det?

A 3D MMG Circa 2004



Tredjepartskomponenter

Eksempler

- Lyd, rendering (grafikk)
- Animasjon (skeletal, morph targets)
- Kollisjonsgjenkjenning og fysikk (vanskelig å lage selv)

Utfordringer

- Integrering av komponentene
- Ikke perfekte
- Er det verdt det?

Tredjeparts spillmotor

Fordeler:

- Komponentene passer sammen
- Vanskelig å lage selv

Ulemper:

- (Ikke mange valg)
- (Dyrt)
- Ikke sikkert det passer til spillet

Domenespesifikke krav

Spillmotor

- Høyere krav til kvalitet og ytelse
- Matematisk kunnskap (lineær algebra, 2D+3D geometri, etc)
- Algoritmisk kunnskap

Simuleringsdybde (AI og fysikk)

Profileringsverktøy

Risiko

Konklusjon

“Games are hard”

Stadig utvikling

Mer og mer komplekse systemer

Krever mye av programvaren og utviklerne

Dette som gjør det interessant og gøy!