

1

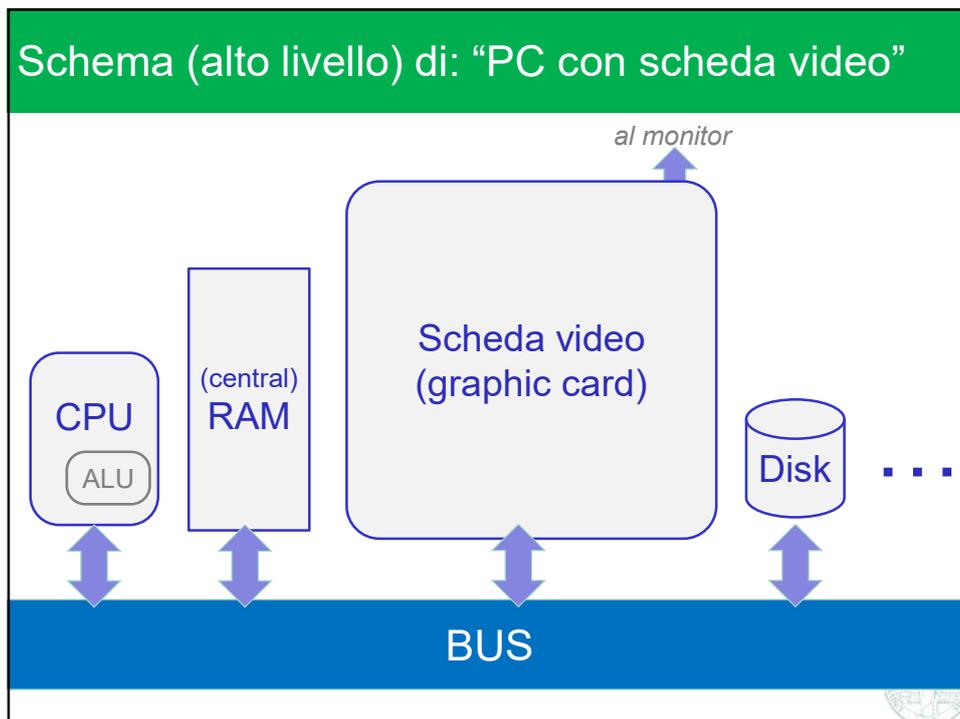
### Hardware specializzato per il rendering

#### Visione di insieme:

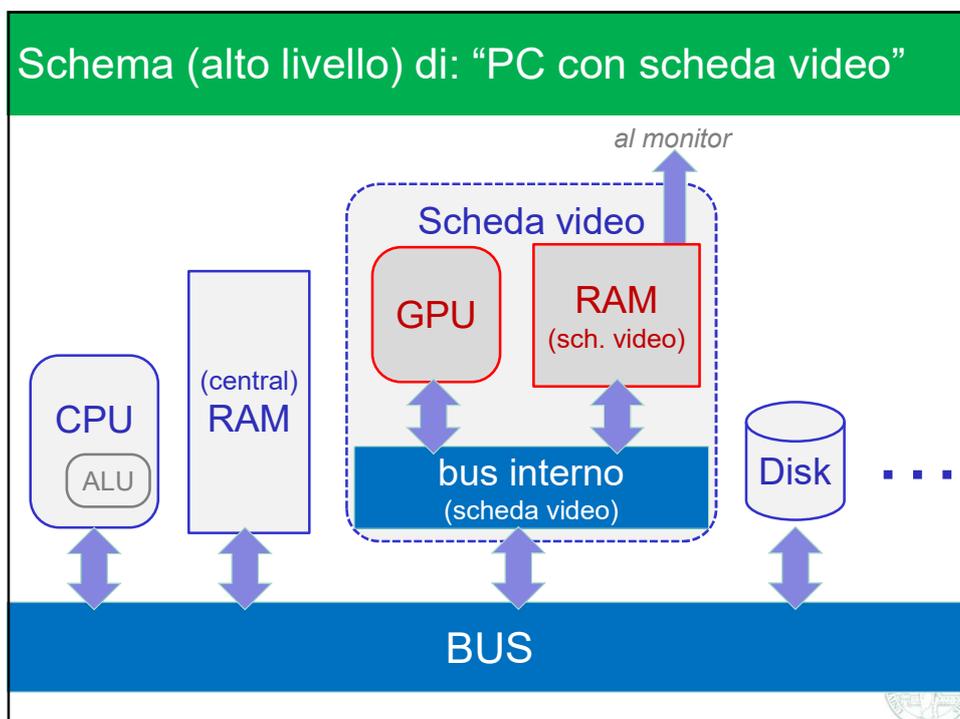
- ✓ **"GPU":**
  - ⇒ Graphics Processing Unit
  - ⇒ La CPU della scheda video
  - ⇒ *Instruction Set* specializzato!
- ✓ Architettura a **pipeline**
  - ⇒ a "catena di montaggio"
- ✓ Modello di computazione **SIMD**
  - ⇒ sfrutta l'alto grado di parallelismo insito nel problema
- ✓ Possiede la **propria** memoria RAM a bordo
  - ⇒ "RAM CPU" vs "RAM GPU"
  - ⇒ grandi copie di memoria da una all'altra dispendiose

1

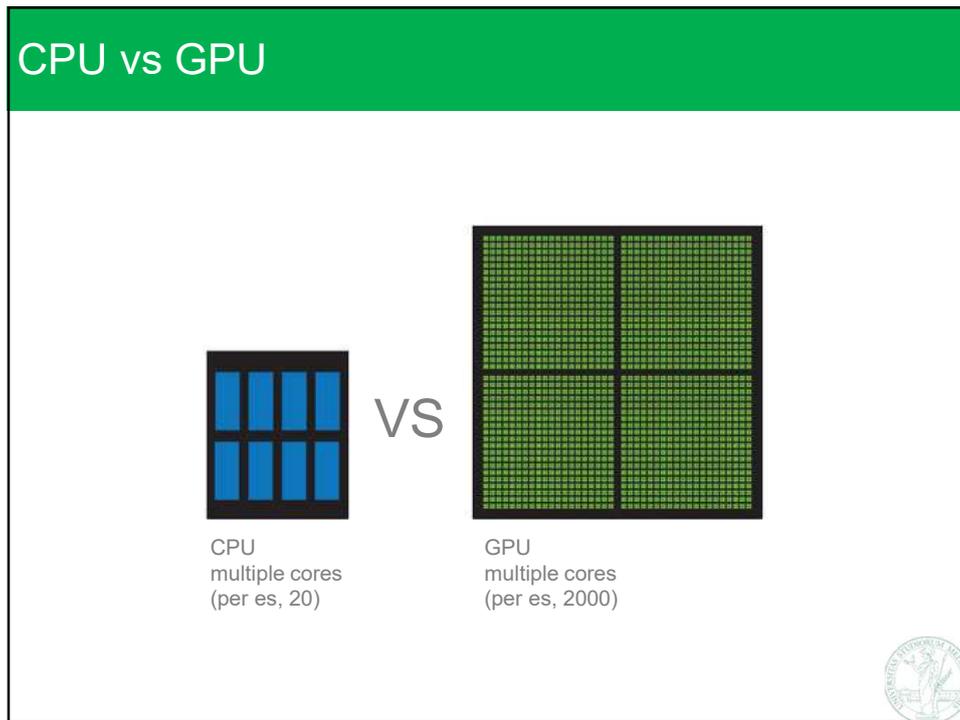
3



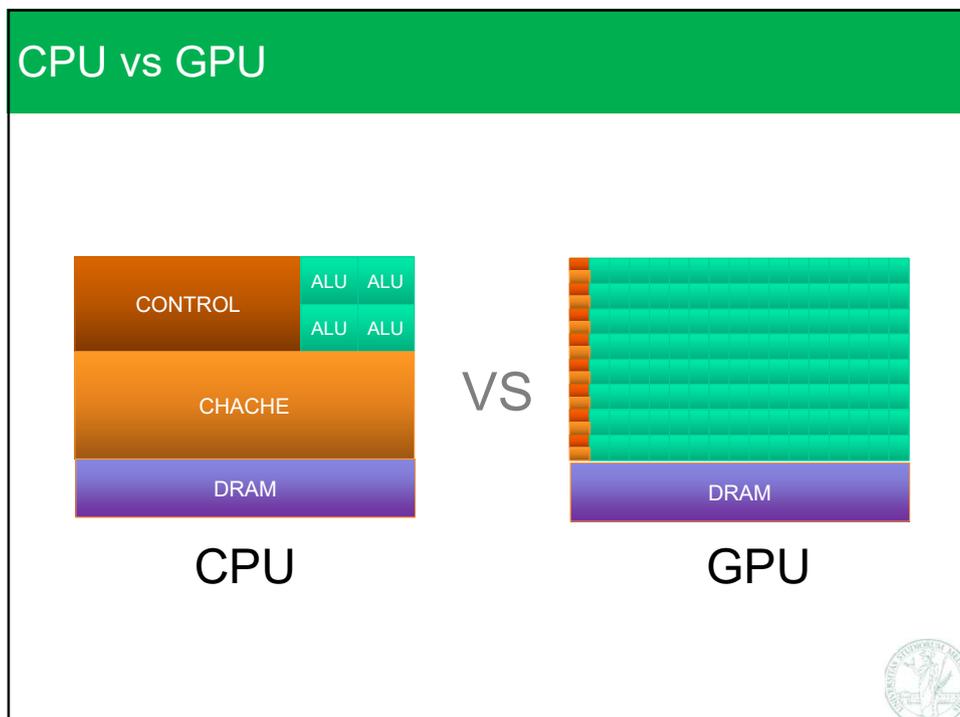
4



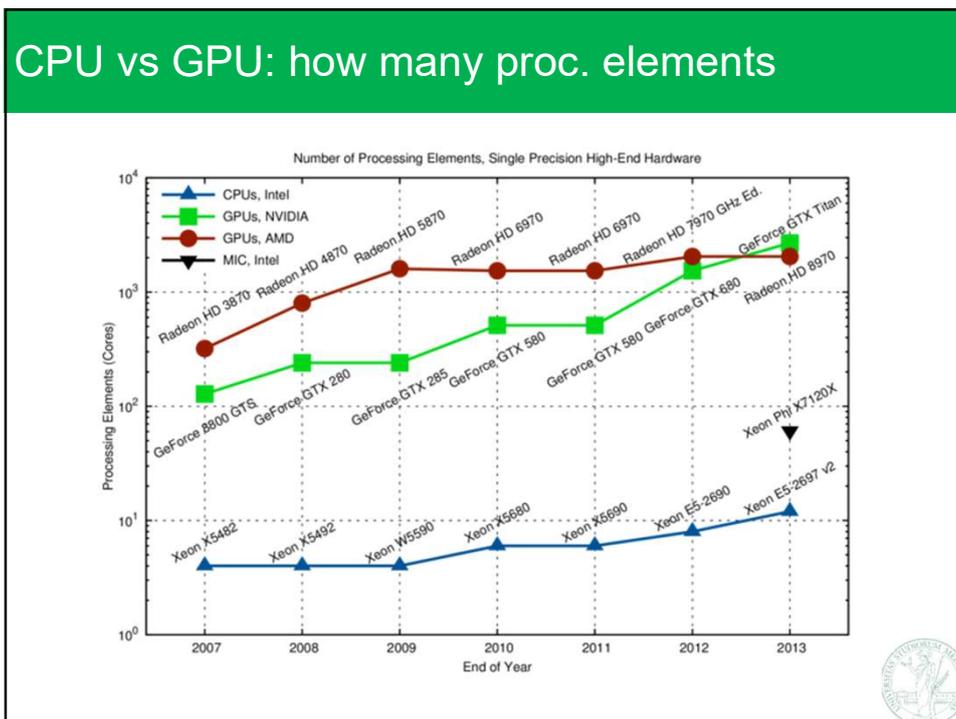
5



6



7



8

### CPU vs GPU

✓ Transistors:  
 >80%  
 control+cache

→ flessibilità

**CPU**

VS

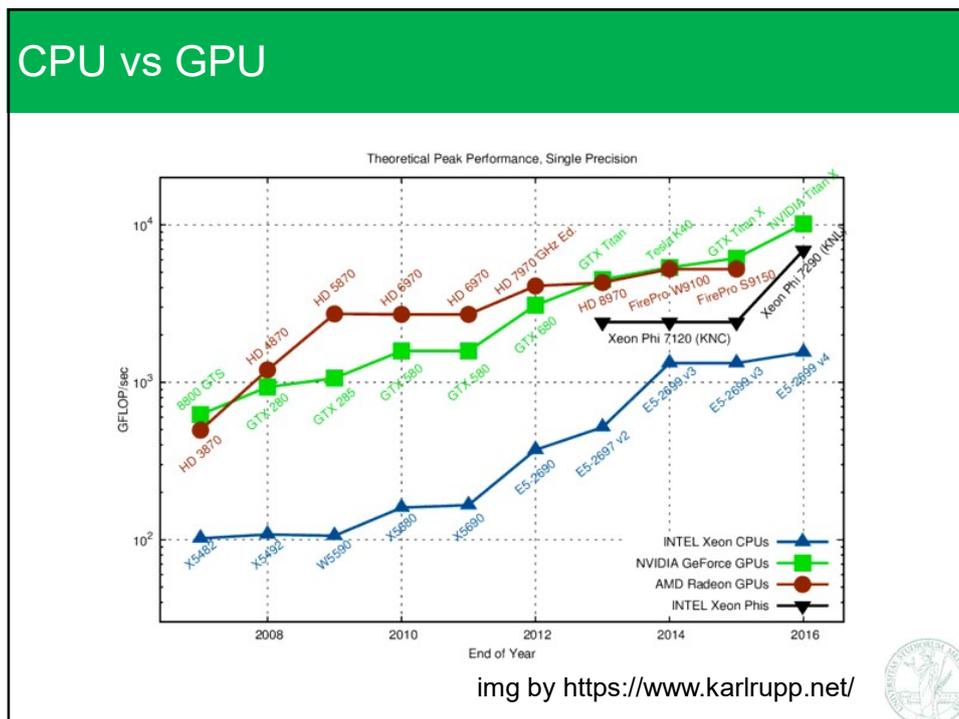
✓ Transistors:  
 ~80% ALU

→ potenza (FLOPS)

**GPU**

1.4 G transistors → 1 TeraFLOP

9

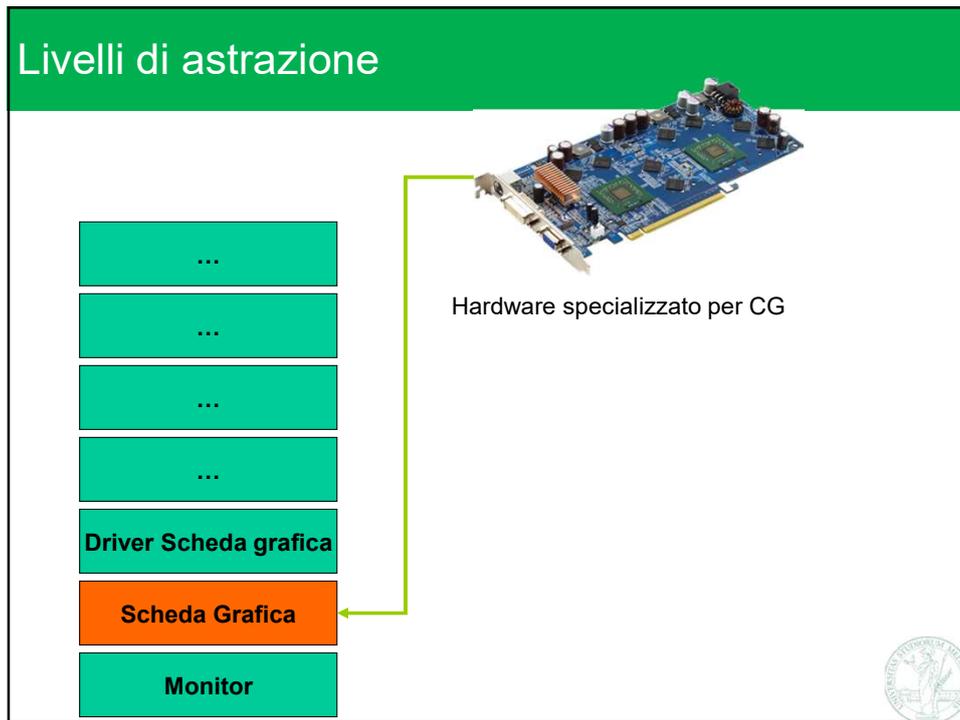


10

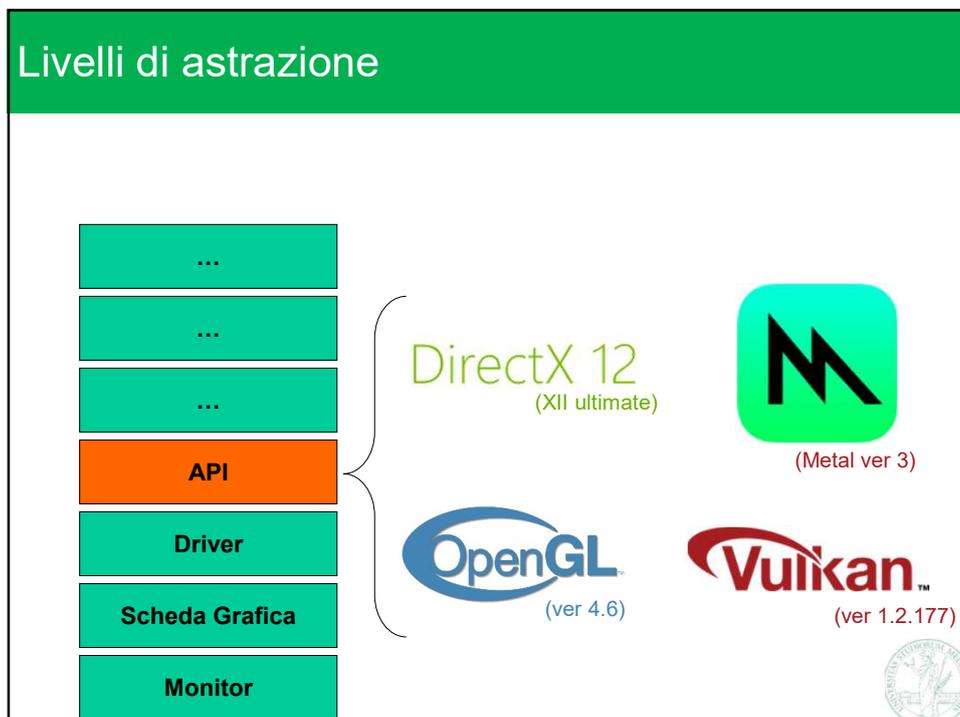
### Hardware specializzato per il rendering

- ✓ **Vantaggio: efficienza**
  - **instruction set** specializzato
    - (computazioni più comuni sono **hard-wired**)
  - computazioni in **parallelo**:
    1. fra CPU e GPU
      - » rendering demandato alla scheda grafica
      - » resto dell'applicazione libera di utilizzare la CPU e RAM base
    2. a volte: fra GPU distinte (es. più schede sullo stesso BUS)
    3. fra le fasi del pipeline (vanno tutte in parallelo)
    4. dentro *ogni fase* del pipeline (più sottoprocess. per fase)
    5. instruction level: operazioni operano su vettori di 4 operandi
- ✓ **Svantaggio: rigidità**
  - scelta quasi obbligata dell'approccio al rendering utilizzato

12



19



20

## API grafiche diffuse

### ✓ Direct3D

⇒ Microsoft

- (proprietario, e **non cross platform**)

⇒ DirectX è il nome collettivo

⇒ Stessi scopi di OpenGL

- una API per usare le stesse GPU
- struttura non dissimile
  - di solito, meno elegante, più macchinoso
- C (e C++)

⇒ E' l'alternativa più comune a OpenGL

- Grossomodo:
  - Direct3D = industrial standard (e **XBOX**)
  - OpenGL = industrial + academic standard
  - Metal = l'API usata su Mac



21

## API grafiche diffuse



⇒ by Khronos (again)

⇒ versione più a basso livello delle API OpenGL

- "bytecode" for the shaders
- better debugging
- unified mobile / embedded / desktop

⇒ Simile alla vers  $\geq 12$  di Direct3D



22

## API grafiche diffuse

- ✓ Metal (Apple Inc.)
  - ⇒ Funzionalità simili a OpenGL + OpenCL
  - ⇒ Basso livello
  - ⇒ Solo per iOS, (e macOS, e tvOS)



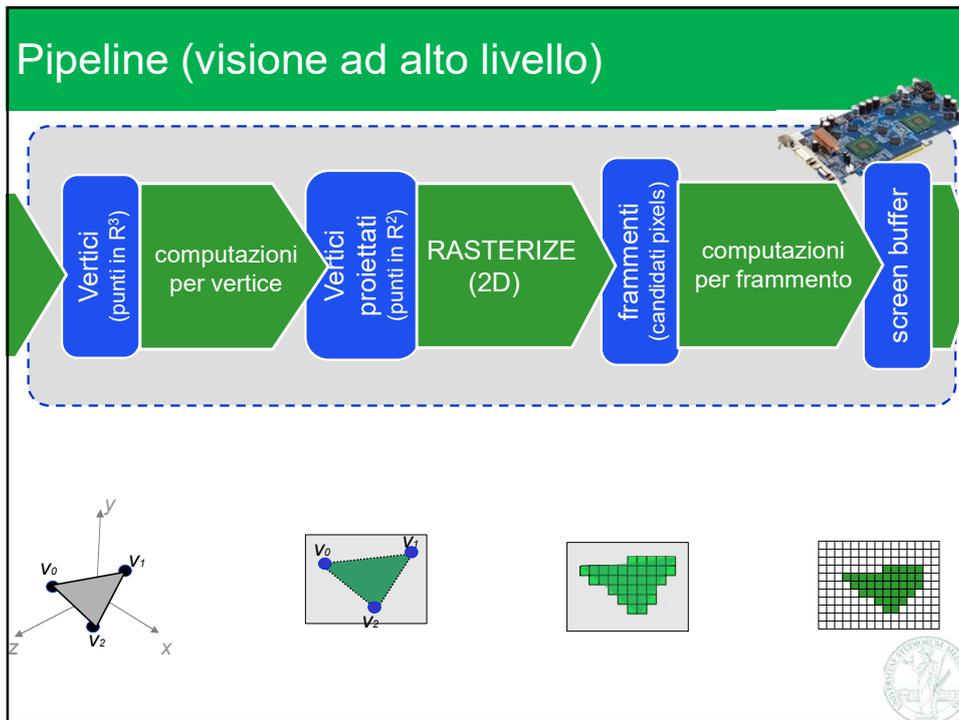
23

## OpenGL : storia

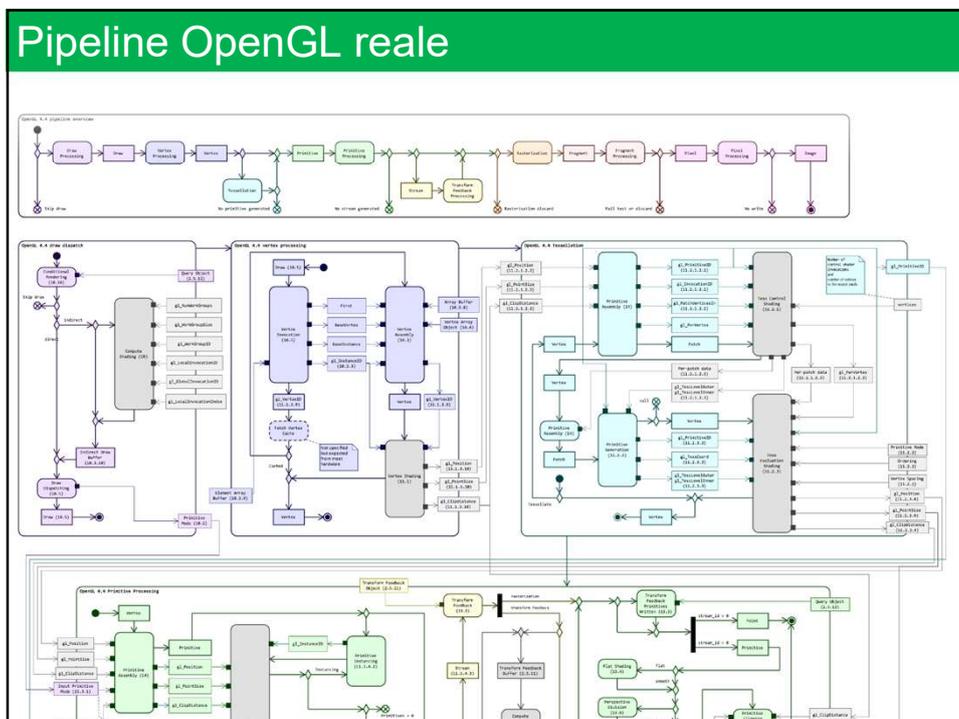
- ✓ inizialmente sviluppato da Silicon Graphics 
- ✓ dal 2002 al 2006:  
OpenGL **A**rchitecture **R**eview **B**oard
  - ⇒ mantiene e aggiorna le *specifiche*
  - ⇒ industria 90%, accademia 10%
  - ⇒ ogni compagnia / gruppo, un voto
- ✓ dal 2006: ARB si evolve nel Khronos Group



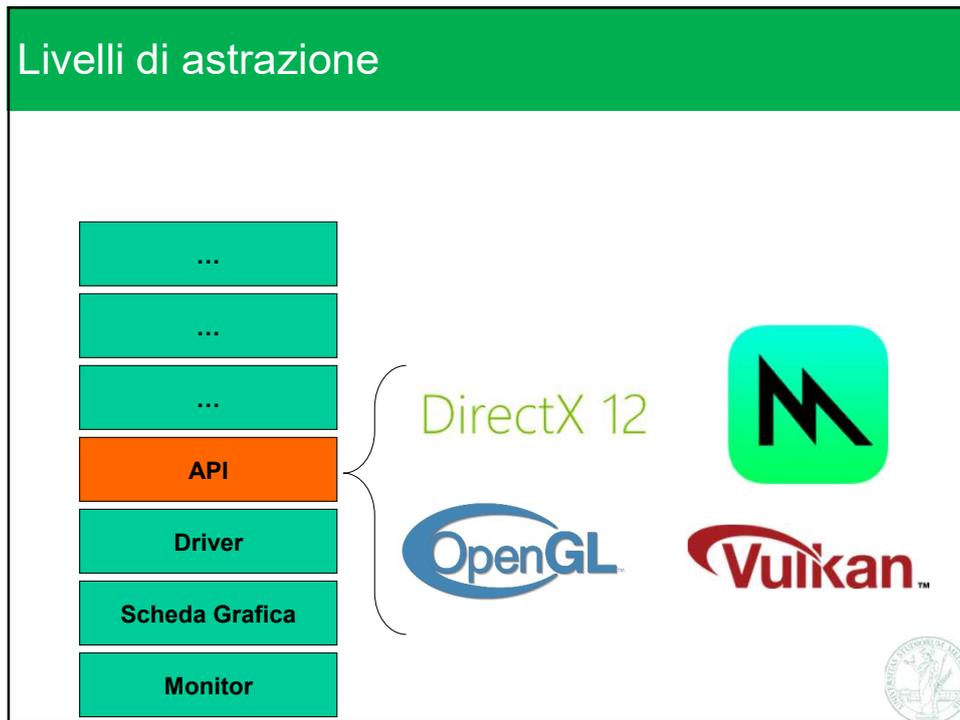
24



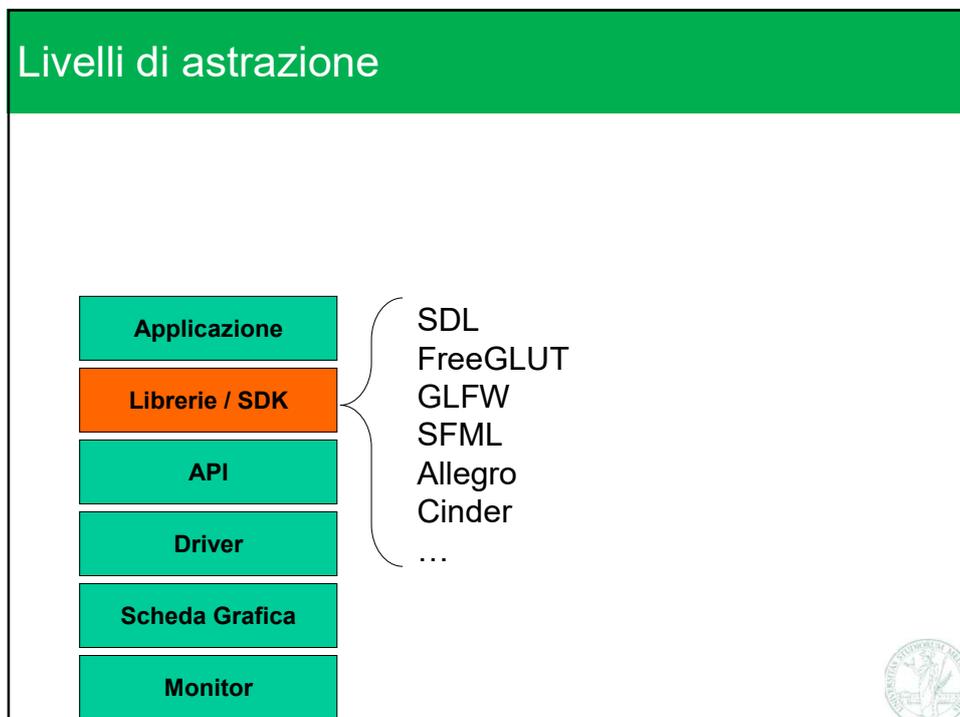
25



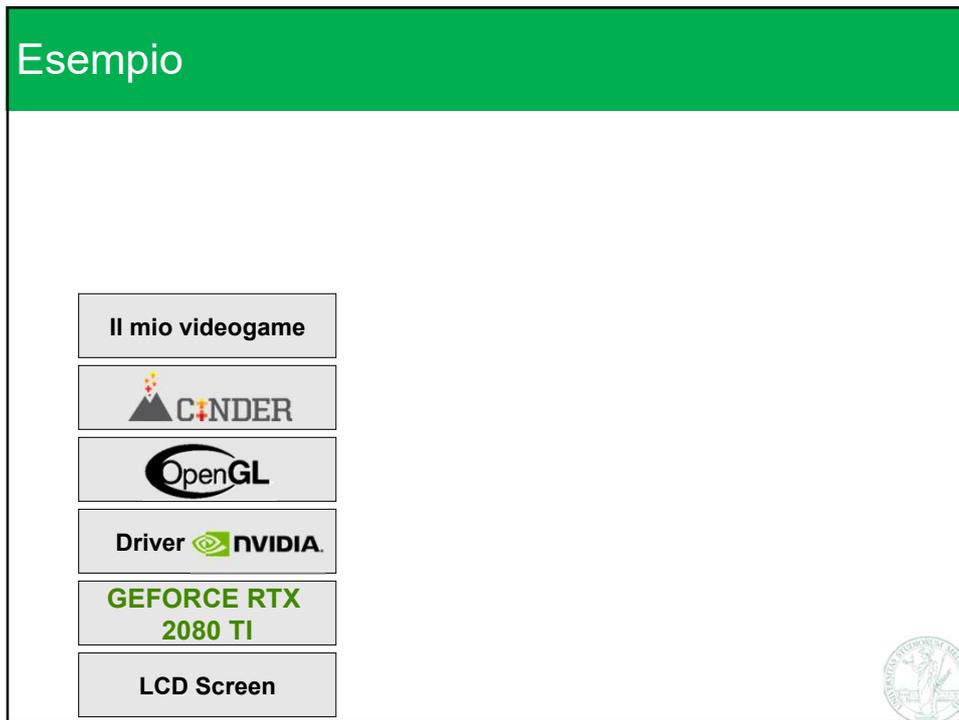
26



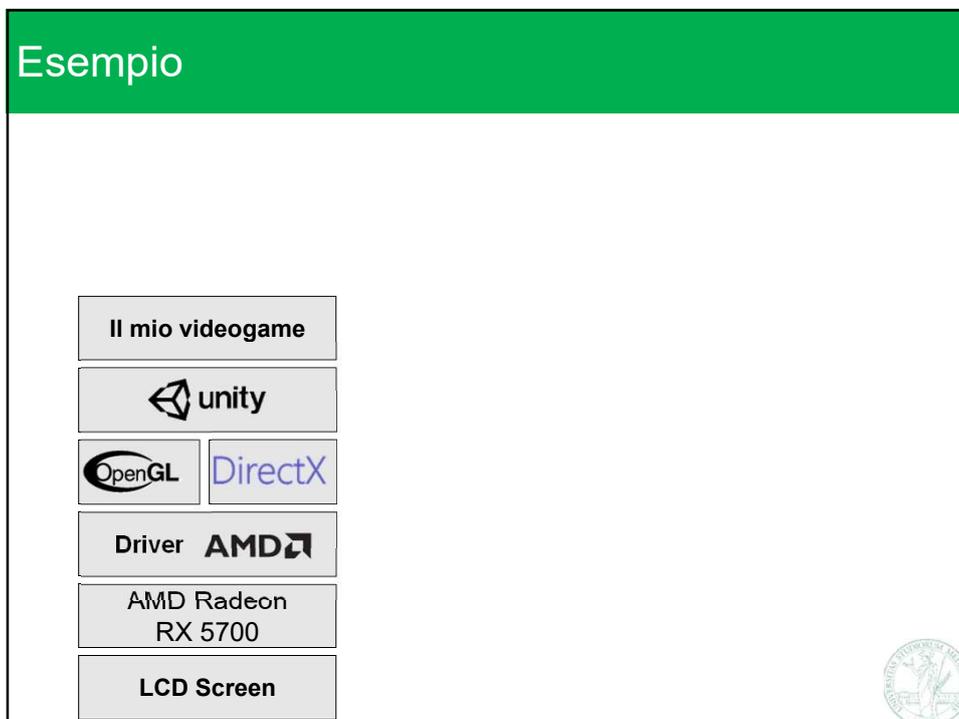
27



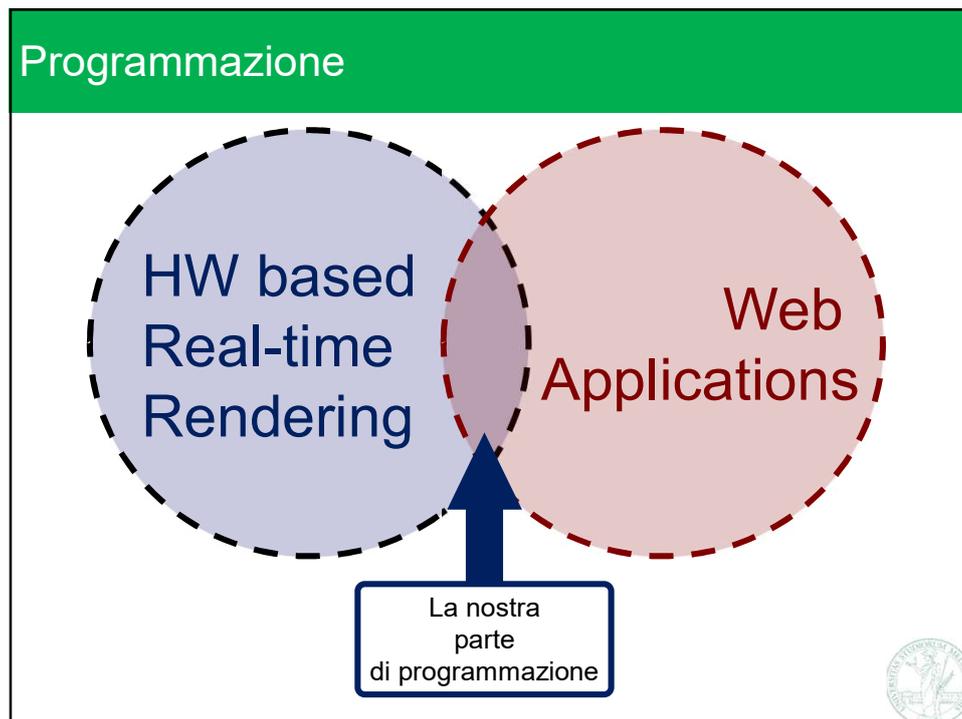
28



29



30



31

### 3D sul web: tentativi di soluzione

✓ Soluzioni:

- ⇒ Remote rendering
  - Il client richiede, il server renderizza e manda immagini
  - Esempio: **Stadia** by Google (per games)
  - (fra l'altro: ottimo per DRM!)
- ⇒ Rendering come preprocessing
  - Esempio: **QTVR** by Apple, Inc. 
- ⇒ Web-oriented formats for 3D interactive scenes
  - Esempi: **VRML**, **X3D**, **Unity bundles**   
- ⇒ Plugins:
  - **Flash** by Adobe 
  - **Silverlight** by Microsoft 



32

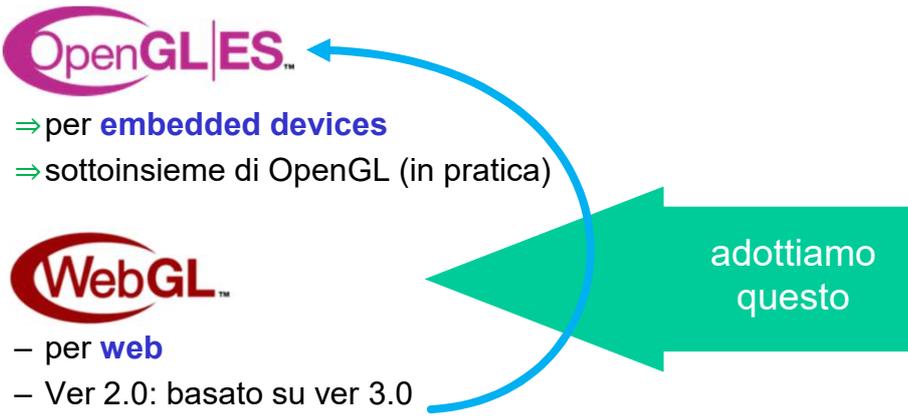
### 3D sul the Web: le soluzioni più usate oggi

- ✓ **WebGL** by Khronos (API)
  - ⇒ Graphics Hardware acceleration from browsers
  - ⇒ Javascript based
  - ⇒ Native! – no plugin
  - ⇒ **GLES 2.0**  $\subset$  **OpenGL**
- ✓ **three.js**
  - ⇒ Higher level
  - ⇒ OpenSource, *free*, MIT licensed
- ✓ **Emscripten**
  - ⇒ transpiler: C++ + OpenGL  $\rightarrow$  JavaScript + WebGL
- ✓ **Unity** (game engine)
  - ⇒ exports projects as web applications
- ✓ Ad hoc dev-tools:
  - ⇒ **3DHOP** - 3D Heritage Online Presenter
  - ⇒ **Google Earth Engine**
  - ⇒ ...



33

### OpenGL : sotto API



**OpenGL|ES™**

- ⇒ per **embedded devices**
- ⇒ sottoinsieme di OpenGL (in pratica)

**WebGL™**

- per **web**
- Ver 2.0: basato su ver 3.0
- **HTML5**
- un language binding in **JavaScript**
- Ver 2.0
- *soluz emergente per il 3D sul Web (senza plug-in!)*



34

## I nostri progettini

A vertical stack of six rectangular boxes representing the components of a WebGL project. From top to bottom: 'Javascript + HTML page', 'three.js', 'WebGL' (with the red and white logo), 'Driver', 'Scheda video', and 'Schermo'. A small circular logo of the University of Milan is in the bottom right corner of the slide.

35

## Three.js

- ✓ Grafica su Web made easy!
- ✓ Un API a livello più alto basato su WebGL
  - ⇒ Cross platform, cross browser, cross vendor (davvero)
  - ⇒ Moltissime utili funzioni grafiche, supportano
    - Mesh poligonali
    - Camera (macchina fotografica)
    - Luci, materiali, e lighting
    - Animazioni
    - Shaders (in GLSL)
    - Importers
    - Virtual reality
  - ⇒ Supporto allo sviluppo:
    - debuggers, esempi, documentazione...

A small circular logo of the University of Milan is located in the bottom right corner of the slide.

36