



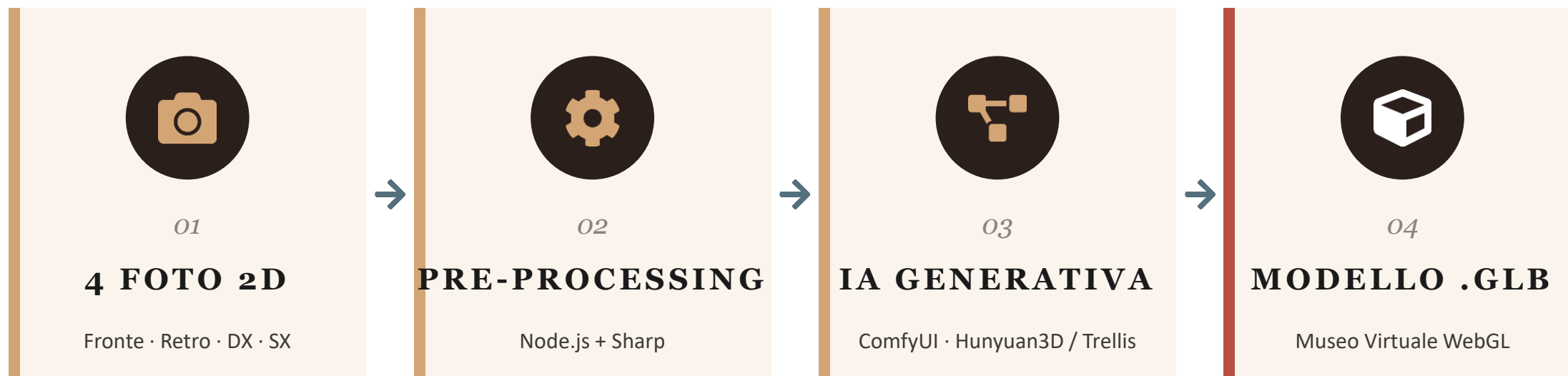
Generazione 3D via ComfyUI

Dall'immagine 2D al museo virtuale interattivo

Un sistema di modellazione 3D assistita dall'IA per la digitalizzazione di reperti archeologici in modelli tridimensionali navigabili in WebGL.

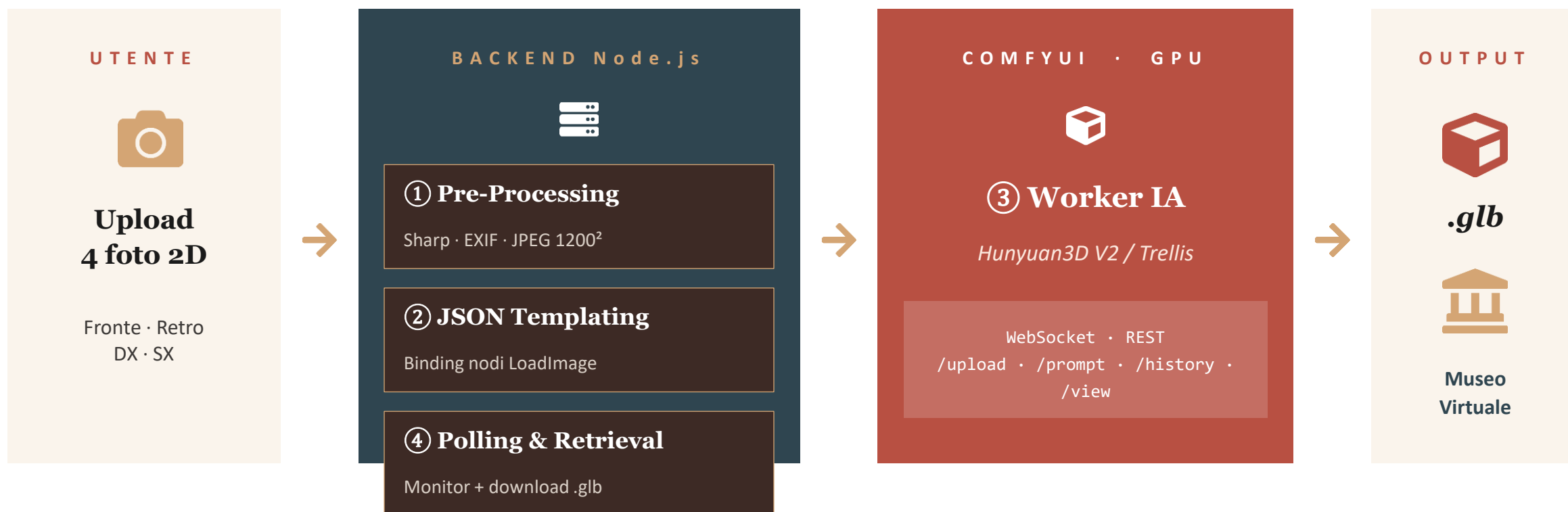
Da fotografia a reperto digitale

Il sistema trasforma una serie di fotografie 2D di un reperto archeologico in un modello 3D interattivo in formato .glb (formato open binario), pronto per essere esposto nel museo virtuale.



Il sistema opera in modo completamente automatico: calibrazione, upload, ricostruzione e recupero dei dati 3D.

Flusso end-to-end in 4 fasi



Il server orchestra tutte le fasi 1 · 2 · 4 · l'IA esegue la ricostruzione 3D nella fase 3

Cos'è ComfyUI

Un motore open-source per costruire pipeline di IA generativa attraverso un editor visuale a nodi. Ogni nodo rappresenta un'operazione (caricamento immagine, inferenza, post-processing) e i nodi si collegano come un grafo eseguibile. Il workflow può essere esportato in JSON e pilotato via API, che è esattamente ciò che viene fatto nel backend.



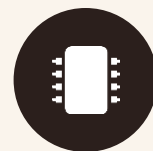
Open Source

Python-based, community-driven,
estensibile con custom nodes.



Workflow Visuali

Pipeline a nodi esportabili in JSON
per automazione.



Modelli Flessibili

Stable Diffusion, Flux, Hunyuan3D e
modelli custom.



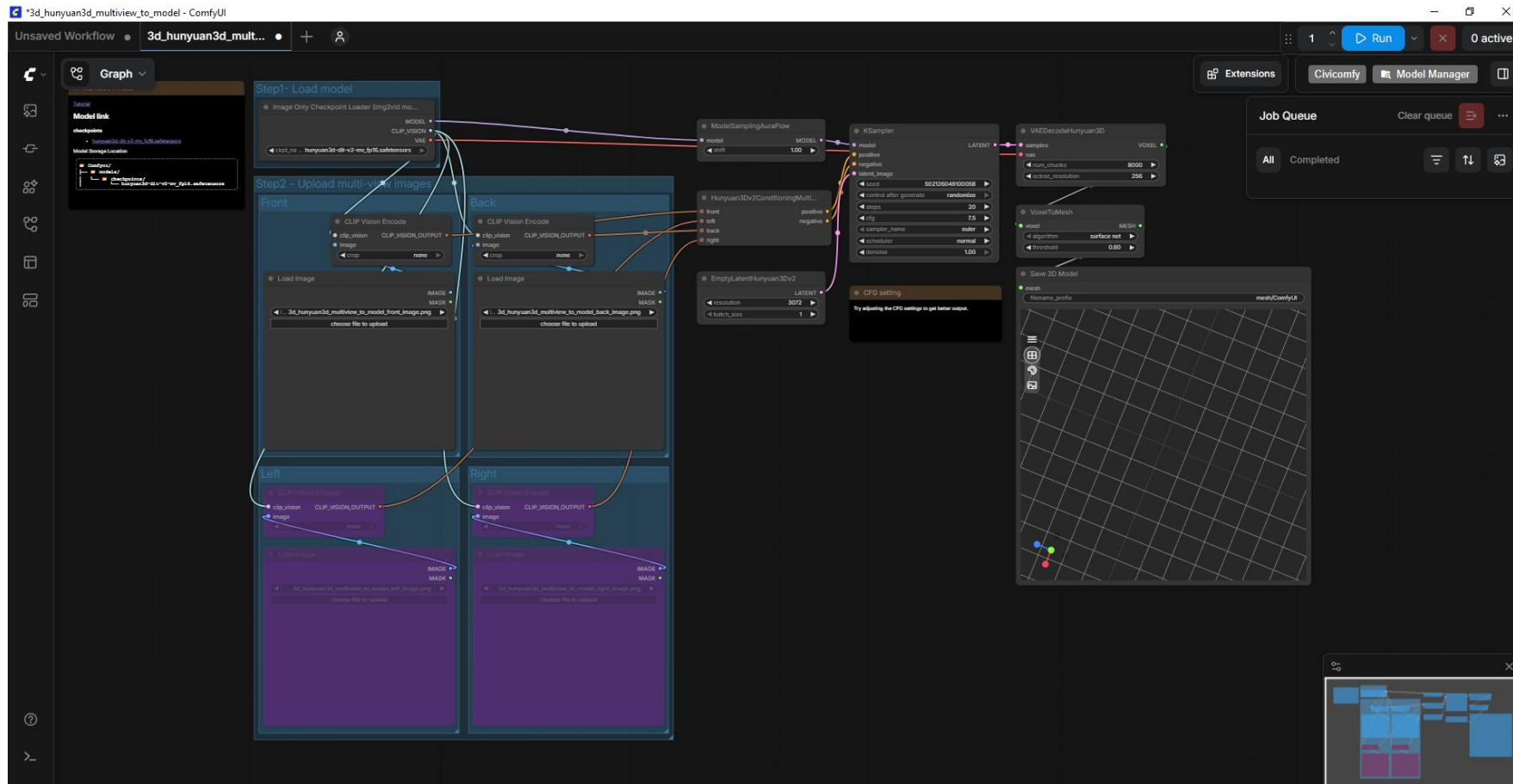
API Robuste

REST + WebSocket per eventi real-
time e stato persistente.

Nel caso di studio: il workflow ufficiale di Hunyuan3D V2 / Trellis ricostruisce il modello 3D partendo dalle 4 viste ortogonali.

ComfyUI Workflows

I workflow possono essere creati, osservati, modificati e configurati da un'interfaccia completamente visuale.



L'intero workflow è stato configurato come una diagramma a blocchi dall'interfaccia visuale di ComfyUI per poi essere esportato ed utilizzato dal Backend.

ComfyUI API in pratica

I quattro endpoint che il backend Node.js invoca durante la generazione di un modello 3D.



A · UPLOAD

POST /upload/image

```
const form = new FormData();
form.append("image", fs.createReadStream(path));
await fetch(`${COMFY}/upload/image`, {
  method:
"POST", body: form
});
```



B · PROMPT

POST /prompt

```
const workflow = require("./workflow.json");
const res = await fetch(`${COMFY}/prompt`, {
  method: "POST",
  body: JSON.stringify({ prompt: workflow })
});
const { prompt_id } = await res.json();
```



C · POLLING

GET /history/{id}

```
while (true) {
  const h = await fetch(
    `${COMFY}/history/${prompt_id}`
  ).then(r => r.json());
  if (h[prompt_id]?.status?.completed) break;
  await sleep(2000);
}
```



D · DOWNLOAD

GET /view

```
const url = `${COMFY}/view?filename=
  ${name}.glb&type=output`
;
const glb = await fetch(url)
  .then(r => r.arrayBuffer());
fs.writeFileSync("Vaso.glb", Buffer.from(glb));
```

Tutto il flusso è implementato in ~200 righe di Node.js, con retry e logging dedicati per ogni endpoint.

1

Ingestion e Pre-Processing

Normalizzazione delle immagini per garantire il successo della generazione IA



Auto-orientamento

La libreria Sharp raddrizza le immagini basandosi sui metadati EXIF.



Quadratura (Squaring)

L'IA lavora meglio su matrici quadrate: vengono aggiunti bordi senza distorcere la foto.



Ottimizzazione risoluzione

Tutte le immagini sono ridimensionate a 1200×1200 px e convertite in JPEG ottimizzato.



Standardizzazione naming

File rinominati con schema fisso (es. Vaso_front_1.jpg) e archiviati in cartelle dedicate per essere poi esportate automaticamente nel data product.

Il server non inoltra le immagini grezze: ogni foto è normalizzata prima di essere inviata.

2

Orchestrazione del Workflow

Il json rappresentante il Workflow — il "cervello" dell'integrazione con ComfyUI

A

Caricamento del template

Il server legge il json, il workflow ComfyUI basato su Hunyuan3D V2 esportato in formato API.

B

Binding dinamico degli input

Il sistema cerca i nodi LoadImage e mappa ciascuna vista (front/back/left/right) sul nodo corrispondente.

C

Configurazione dei parametri

Seed, Steps e Resolution definiti nel template vengono applicati prima dell'invio alla GPU.

```
{
  "nodes": [
    {
      "class_type": "LoadImage",
      "inputs": { "image": "Vaso_front_1.jpg" }
    },
    { ... retro, dx, sx ... },
    { "seed", "steps", "res" }
  ]
}
```

Il template è la singola fonte di verità del workflow generativo.

3

Comunicazione con il Worker IA

WebSocket + REST API · la GPU assume il controllo della generazione

STEP A

Caricamento

POST /upload/image

Il server invia fisicamente le immagini processate all'endpoint di upload di ComfyUI.

Questo passaggio è critico: permette al worker IA di trovare i file localmente prima dell'esecuzione.

STEP B

Esecuzione

POST /prompt

Il server invia il workflow JSON completo all'endpoint /prompt.

Da questo istante ComfyUI prende il controllo della GPU ed esegue la ricostruzione fotogrammetrica assistita dall'IA.

Il canale di comunicazione combina WebSocket (eventi real-time) e chiamate REST (stato persistente).

4

Monitoraggio e Finalizzazione

La generazione richiede minuti: il server gestisce l'attività in background

POLLING



`/history`

Monitoraggio ciclico

Il server interroga periodicamente l'endpoint `/history` di ComfyUI per verificare lo stato del Job ID.

DOWNLOAD



`/view`

Recupero del modello

A generazione completata, l'IA produce un file `.glb` che il server scarica dall'endpoint `/view` e archivia nella cartella dell'opera.

PUBBLICAZIONE



stato → Pronto

Disponibilità immediata

L'opera viene resa visibile nel Catalogo (ispezione tecnica) e posizionata su un piedistallo nel Museo Virtuale.

Il modello è istantaneamente renderizzato in tempo reale tramite Three.js nel portale pubblico.

Cosa rende il sistema distintivo



Automazione totale

L'utente carica solo le immagini: calibrazione, upload e recupero dei dati 3D avvengono in completa autonomia.



Scalabilità asincrona

Il server gestisce più richieste contemporaneamente, accodandole nel sistema di polling senza bloccare l'utente.



Tracciabilità

Il logging dedicato (logs/comfyui.log) monitora ogni fase della comunicazione tra portale e IA per il debug.



Integrazione WebGL

Il risultato non è un file statico ma un'entità interattiva renderizzata in tempo reale con Three.js.

Un sistema che unisce l'affidabilità del backend alla potenza creativa dell'IA generativa.

Grazie.

Domande · Discussione · Demo

Ogni fotografia entra come pixel; esce come un reperto digitale navigabile.

Il ponte è un workflow che automatizza quel passaggio, end-to-end.