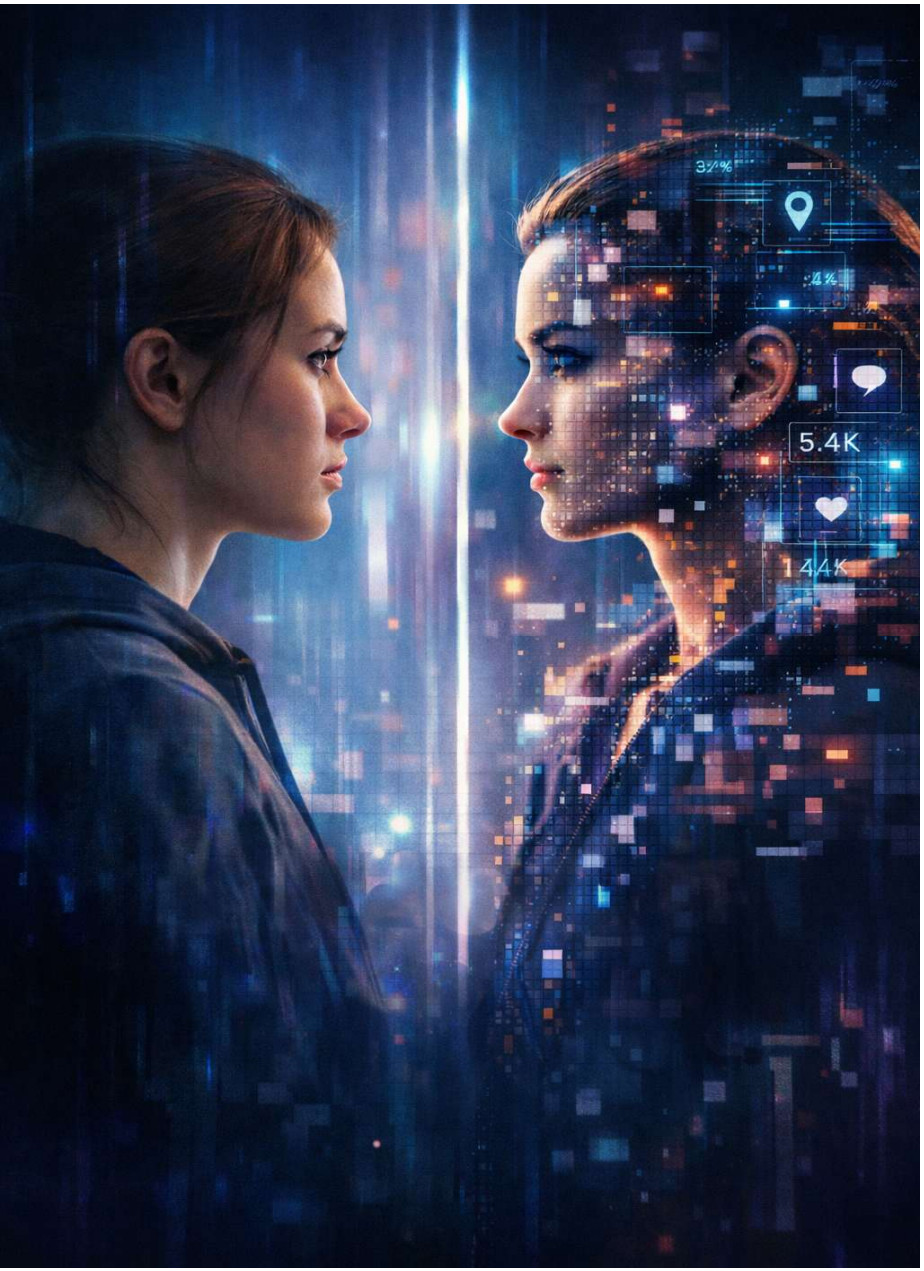
Build your Idol

Hands-on



Workflow/ Pipeline

- Goal
- Options
  - Simple
  - Semi
  - Pro
- The Chosen one

Problem definition

The practical goal:

→ modeling and design of a virtual idol as a preliminary stage to an autonomous, embodied AI entity

Introduction (1)

The fundamental question:

Standardization & Interoperability
vs.
Visual Quality & Individuality

Introduction (2a)

Platform-specific avatar systems
→ very low effort

Examples:
→ Meta Avatars SDK
→ VRChat (Avatare via Unity)
→ Roblox

Characteristics:
→ strong integration into the respective platform
→ very quick to use
→ good animation and tracking

Conclusion:
Good for specific platform projects, but ...
... hardly portable
... limited control over data and pipeline

Open Avatar Standards & Creator-Tools

Examples:
→ ~~Ready Player Me~~
→ VRM (based onglTF)
→ Tools (in VRM- ecosystem)
→ VRoid Studio
→ Blender (VRM-Plugins)

Charakteristics:
→ Open Standard
→ Integrated Humanoid-Rig and Facial Expressions
→ GoodInteroperability (Unity, Web, XR)

Conclusion:
Acceptable results with low to medium effort

High-quality character systems

Examples :
→ MetaHuman Creator
→ Character Creator

Charakteristics :
→ very high visual quality (realistic or stylized)
→ professional pipelines
→ good animation and facial capture

Conclusion:
Relevant for professional virtual idols or for research on digital humans.
But...
... complex workflow!
... not very “lightweight”

Introduction (2b)

Platform-specific avatar systems
→ very low effort

Examples:
→ Meta Avatars SDK
→ VRChat (Avatare via Unity)
→ Roblox

Characteristics:
→ strong integration into the respective platform
→ very quick to use
→ good animation and tracking

Conclusion:
Good for specific platform projects, but ...
... hardly portable
... limited control over data and pipeline

Web/AI-based avatar generators
→ quite new & dynamic

Examples:
→ Luma AI
→ Avaturn
→ and many others

Characteristics:
→ fast generation
→ partially customizable via photo
→ API-first approach
→ still fragmented
→ quality difficult to compare

Conclusion:
Exciting tech stacks that, with usually little effort, lead to initial, quick, and acceptable results.
But...
... no clear standard yet.

High-quality character systems

Examples :
→ MetaHuman Creator
→ Character Creator

Charakteristics :
→ very high visual quality (realistic or stylized)
→ professional pipelines
→ good animation and facial capture

Conclusion:
Relevant for professional virtual idols or for research on digital humans.
But...
... complex workflow!
... not very “lightweight”

The chosen one ?

The more fundamental question:

Standardization & Interoperability
vs.
Visual Quality & Individuality
vs.
Complexity & Usability

Your choice: Semi-Pros vs. Pro

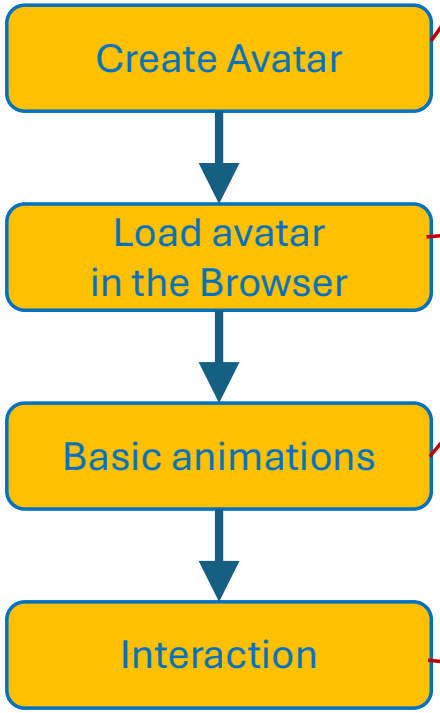
For semi-pros:

→ Web alternative without Unity
(based on VRM + Three.js)
→ VRM: → format based on glTF 2.0
for humanoid 3D avatars
→ freely usable!

For pros:

→ engine-based pipelines (e.g. Unity, Unreal)
→ choose an avatar standard that is as open as
possible to avoid dependence on a single platform

Your choice? The Semi-Pro Pipeline?



Avatar created e.g. in
VRoid Studio
and exported as
VRM-Datei

(VRoid Studio supports VRM export of characters)

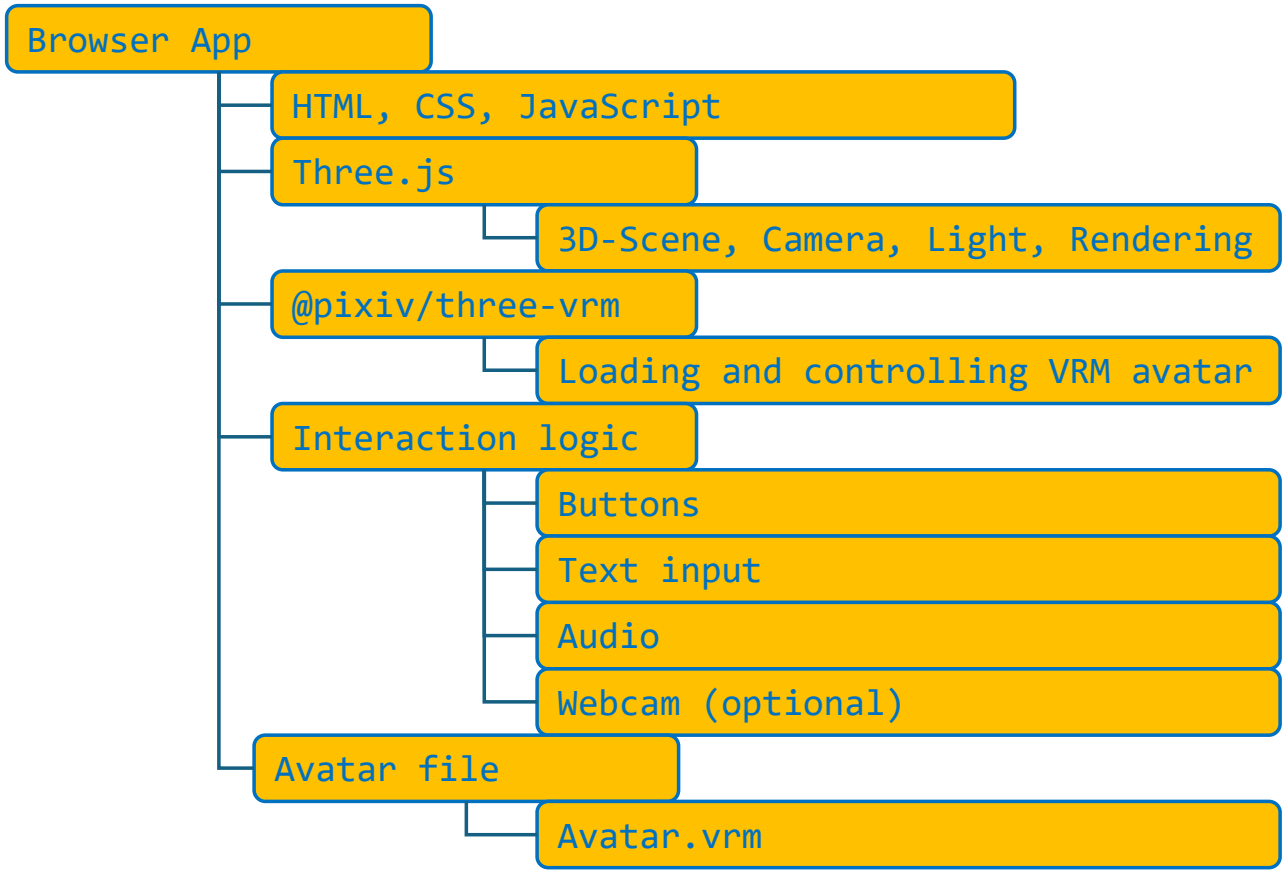
Avatar is rendered In the
Browser
using
Three.js

(The library @pixiv/three-vrm makes VRM models directly usable in Three.js)

Idle, waving, speaking, or head movements, for example, via:
- pre-made animation clips
- simple bone rotation
- blend shapes (for faces)
- ...

Interaction is entirely **web-based**
- Buttons for the avatar to wave, etc.
- Text input for communication
- Audio for lip movements
- Mouse position for the avatar's gaze direction
- WebXR for VR/AR in the browser

Your choice? The Semi-Pro Pipeline?

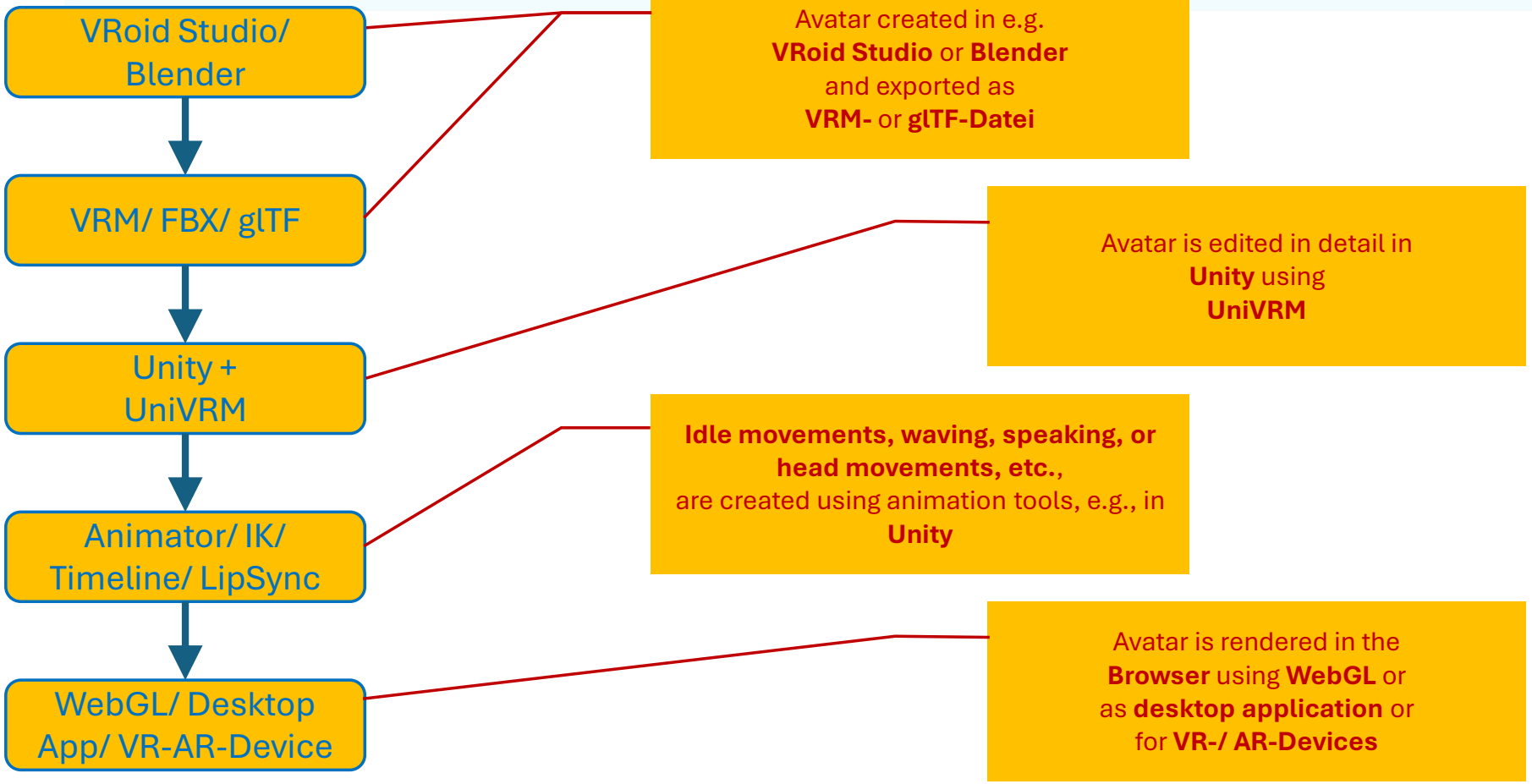


Your choice? Semi-Pros estimated efforts

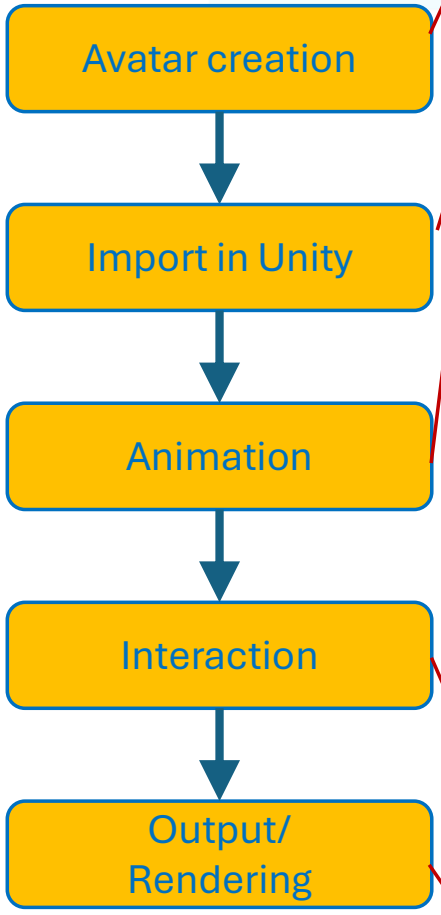
Estimated time required for a simple prototype

→Generate VRM avatar in browser:	1-2 hours
→Display VRM avatar in browser:	1-3 hours
→Simple animations:	2-6 hours
→Button interactions:	1-3 hours
→Facial expression/lip movement:	½-1 day
→Simple AI dialogue:	1-3 days

Your choice? The Pro Pipeline?

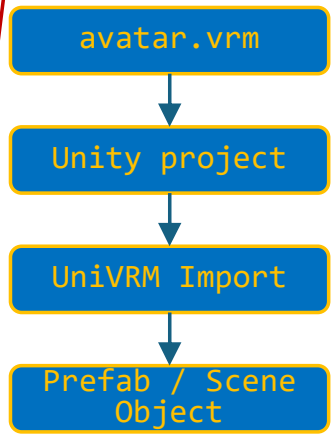


Your choice? The Pro Pipeline?



Avatar creation e.g. using **VRoid Studio**, **Blender** or other **character tools** (most obvious alternative is VRoid Studio, due to its direct VRM export.)

Avatar import using **UniVRM**
→ VRM/ glTF implementation for **Unity**
→ allows import, editing, and export of VRM avatars



(simple) animations with **Unity's Humanoid/ Mecanim-System**

Humanoid avatar can adopt the animations of other humanoid characters
→ **Retargeting**

Typical initial animations:
Idle, **Winken**, **walking, pointing, bowing, dancing**
(**Asset Stores**, **motion-capture libraries, custom animations**)

Interaktion in **Unity** über **Skripte**, **Animator-Parameter**, **Trigger**:
Button Click → Avatar winkt
Text Input → Avatar spricht
Mikrofon → LipSync
Nähe/ Kamera → Avatar schaut Nutzer an
Dialogstatus → Emotion / Geste/ Antwort

Techn Optionen für einen Prototypen:
Unity UI →Animator Controller
→einfache C#-Skripte
→Audioausgabe
→Blendshape-Steuerung

Export projekt as **WebGL** application
→ **JavaScript/ WebGL application**
(runnable in the browser)

Alternatively:
Desktop app
VR-/ AR application
lab installation
atreaming setup

TUS · Politécnico de Leiria · IPCA · HAMK Häme University of Applied Sciences · NHL Stenden · FHV Vorarlberg University of Applied Sciences · Universidad de Burgos · howest · Technical University of Liberec · Universitas Galatensis

Suggested: a web-based Semi-Pro-Pipeline

A Semi-Pro Web Pipeline

→Starting Point:	VRoid → VRMeb Pipeline (alternative: avaturn)	
→Step 1:	Basic Animations	(low effort)
→Step 2:	Face & Lip Sync	(low)
→Step 3:	Interactive Animation	(low to medium)
→Step 4 (opt.):	More complex animation	(medium)

Avatar moves	x	very low
Avatar speaks	xx	low
Avatar reacts interactively	xxx	moderate
Performance / Idol Level	xxxx	high

Goal:
An interactive, talking avatar in the browser within 1-2 days.

TUS · Politécnico de Leiria · IPCA Polytechnic University of Cávado and Ave · HAMK Häme University of Applied Sciences · NHL Stenden University of Applied Sciences · FHV Vorarlberg University of Applied Sciences · Universidad de Burgos · howest University of Applied Sciences · Technical University of Liberec · Universitas Galatensis

Making the Avatar speek (Web-based Semi-Pro)

- → Relatively low complexity
- → Good integration into VRoid/ VRM web pipeline

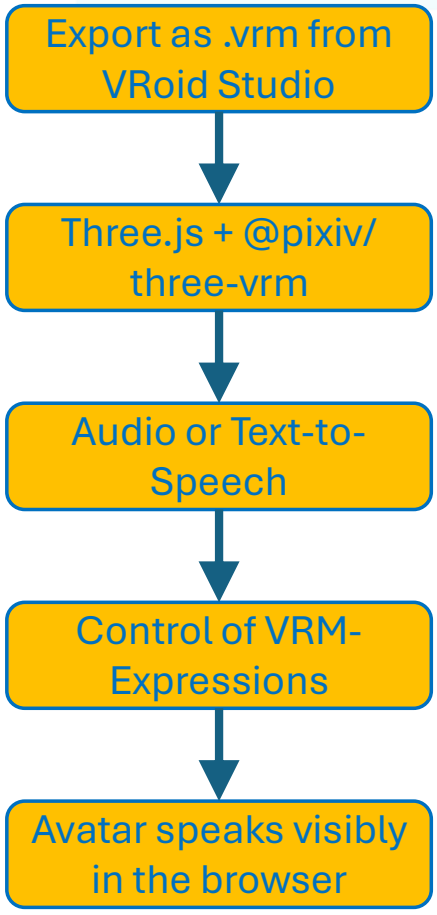
- → Technically, it consists of three layers:
  - → Text-to-Speech
  - → Audio output
  - → LipSync via VRM expressions

- → VRM offers standardized facial expressions and mouth shapes such as A, I, U, E, O
  - → Control for lipsync from audio or text

Goal:
An interactive, talking avatar in the browser within 1-2 days.

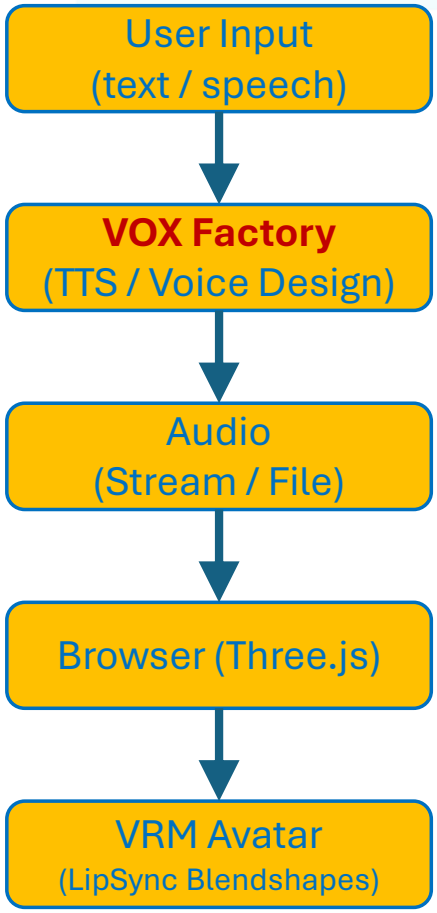
TUS · Politécnico de Leiria · IPCA · HAMK Häme University of Applied Sciences · NHL Stenden · FHV Vorarlberg University of Applied Sciences · Universidad de Burgos · howest · Technical University of Liberec · Universitas Galatensis

Making the Avatar speek (Web-based Semi-Pro)



Goal:
An interactive, talking avatar in the browser within 1-2 days.

Making the Avatar sing (Web-based Semi-Pro)



Goal (optional):

→ A consistent voice for the avatar with a web-based, integrated overall architecture

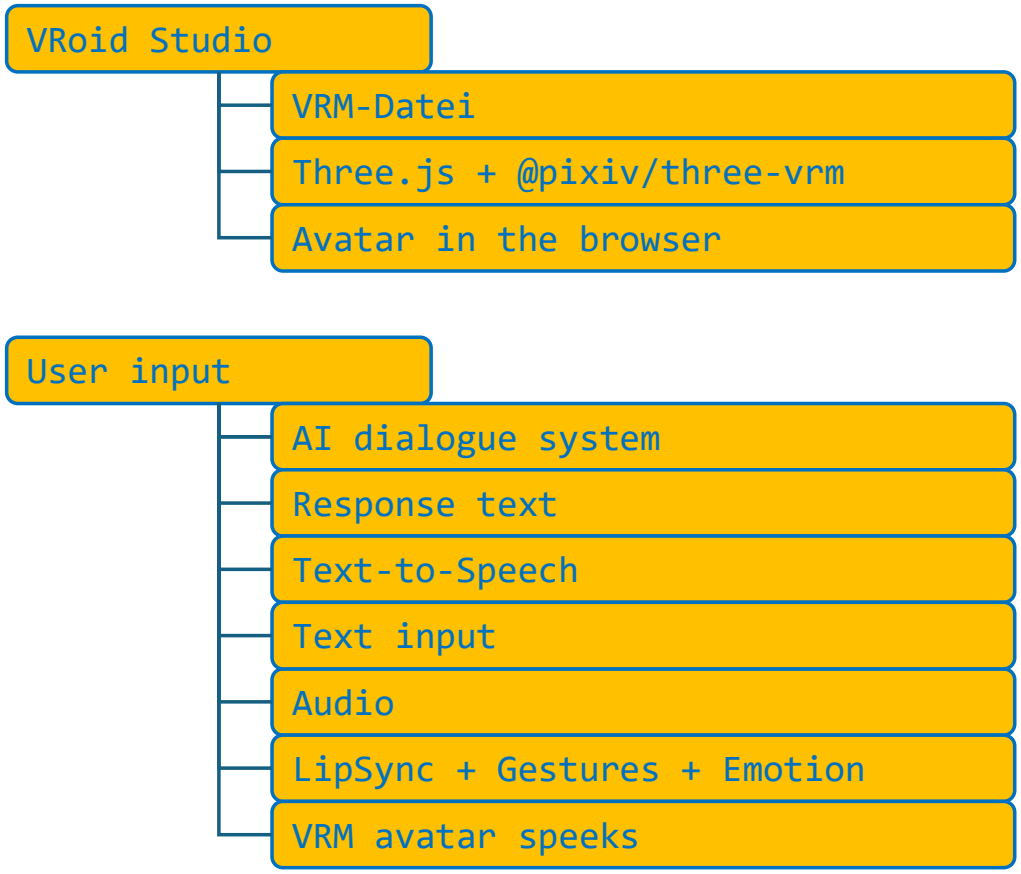
Goal:
An interactive, talking avatar in the browser within 1-2 days.

Making the Avatar think (Web-based Semi-Pro)

AI Dialogue System as Agent Layer between User and Avatar

1. User asks a question
2. Frontend sends input to backend
3. AI generates a response in the style of the avatar persona
4. Text-to-speech (TTS) generates audio
5. Browser plays audio
6. Lip-sync controller moves mouth expressions
7. Emotion controller sets emotions such as joy, neutral, and surprise
8. Gesture controller starts a corresponding animation

Making the Avatar think (Web-based Semi-Pro)



Pro!

TUS POLITÉCNICO DE LEIRIA IPCA HAMK Häme University of Applied Sciences NHL STENDEN FHV Vorarlberg University of Applied Sciences UNIVERSIDAD DE BURGOS howest TECHNICAL UNIVERSITY OF LIBEREC

Have fun :)

Thank you