# Technical Framework for Automated Low-Fidelity Character Animation in DaVinci Resolve Fusion

## 1. Introduction: The Intersection of Procedural Efficiency and "Scuffed" Aesthetics

In the contemporary landscape of digital content creation, the velocity of production often stands in direct tension with visual distinctiveness. For creators such as "Ovi English School," operating under a daily release schedule necessitates a production pipeline that radically minimizes manual labor without sacrificing the brand's unique visual identity. The requirement is specific: a daily animated podcast featuring a character named "Ku-chan" (a black cat) rendered in a "scuffed"—or low-fidelity, hand-drawn—style. This aesthetic, characterized by jittery lines, imperfect motion, and starkly simple lip-syncing, is not merely a stylistic preference but a strategic asset. It allows for the utilization of procedural animation techniques that obscure the lack of traditional frame-by-frame artistry.

This report outlines a comprehensive technical architecture for implementing this pipeline using **DaVinci Resolve Fusion**. While the user operates with a "beginner" skill set in Fusion and utilizes the Free version of the software on a MacBook Pro M5, the solution proposed herein leverages advanced proceduralism—specifically node-based compositing, audio-reactive modifiers, and Python scripting—to create a system where "animation" becomes an automated output of audio input.

The central thesis of this workflow is the transition from *keyframing* to *rigging*. By constructing a robust node graph that procedurally generates "boiling lines" (simulated hand-drawn jitter) and maps audio amplitude to mouth states, the operator effectively removes the need to "animate" daily. Instead, the workflow shifts to an asset-management and rendering task, capable of being executed in minutes rather than hours. This document serves as an exhaustive reference, detailing the theoretical underpinnings, step-by-step construction, and automation scripts required to turn a static asset library into a living, breathing animated series.

## 2. The Fusion Compositing Paradigm: A Theoretical Primer

To effectively utilize DaVinci Resolve for character animation, it is imperative to dismantle the mental models inherited from layer-based Non-Linear Editors (NLEs) like the Edit Page or Adobe Premiere Pro. Fusion operates on a **Flow Graph** principle, a distinction that is not merely cosmetic but structural, influencing every aspect of how the "Ku-chan" rig will be constructed and optimized.1

### 2.1 The Flow-Based Architecture vs. Layer-Based Systems

In traditional layer-based compositing, visual elements are stacked vertically on a timeline. The rendering order is implicit: layers higher in the stack obscure those below. Effects are applied as attributes of a layer, and nested compositions are required to group elements. This model, while intuitive for editing, becomes cumbersome for complex rigging where data—such as the audio amplitude from a TTS file—needs to drive multiple disparate parameters (e.g., mouth opening, head squashing, and line jitter) simultaneously.

In Fusion, the compositing environment is a spatial, resolution-independent 2D workspace where image data flows horizontally or vertically through **Nodes**. Each node represents a discrete mathematical operation—a file load, a color correction, a geometric transformation, or a pixel merge. The rendering order is explicit, defined solely by the connections (pipes) between nodes.

**Implications for the Daily Pipeline:**

* **Non-Destructive Branching:** A single "Head" asset can be branched into multiple paths. One path might feed into a shadow generation system, while another feeds into the main composite. This allows for complex looks without duplicating source files.
* **Resolution Independence:** Unlike a timeline fixed to 1920x1080, Fusion nodes process images at their native source resolution until specifically resized. This allows the "Ku-chan" character to be drawn at 4K and zoomed in for dramatic emphasis without pixelation, provided the scaling happens before the final rasterization.2
* **Concatenation:** Fusion is capable of concatenating adjacent transformation nodes (Transform, Merge, Size) into a single filtering operation. This preserves image quality by calculating the final pixel position once, rather than resampling the image multiple times at each step.

### 2.2 Anatomy of a Node: Inputs and Outputs

Understanding the input/output logic is prerequisite to building the rig. A standard node in Fusion, such as the Merge node (the equivalent of a "layer" combiner), possesses specific inputs that must be respected for the rig to function.

* **Yellow Input (Background):** This is the canvas. In the context of a character rig, this is the "body" or the element that sits behind.
* **Green Input (Foreground):** This is the element being composited "on top." For Ku-chan, the Head connects to the Green input of the Body's Merge node.
* **Blue Input (Effect Mask):** A grayscale channel determining the opacity of the operation. This is critical for the "scuffed" look, as we can use noise maps to mask parts of the character procedurally.
* **Red Output:** The processed image data flowing to the next operation.3

### 2.3 The Core Node Vocabulary

For the specific "scuffed" style requested, the following nodes form the essential vocabulary. Mastery of these specific operators is sufficient to build the entire pipeline.

| **Node** | **Acronym** | **Function** | **Application in Pipeline** |
| --- | --- | --- | --- |
| **Loader** | LD | Imports external media (PNG, WAV) from disk. | Loads the static cat drawings and daily TTS audio files. |
| **Merge** | Mrg | Combines two image streams. | Assembles Body, Head, and Mouth layers into a composite. |
| **Transform** | XF | Position, Rotation, Scale. | Animates the "bobbing" motion or head tilts. |
| **Displace** | Dsp | Warps pixels based on a map. | Creates the "wobbly" line effect (boiling lines). |
| **FastNoise** | FN | Generates procedural noise. | Drives the Displace node to randomize the wobble.4 |
| **TimeStretcher** | TS | Remaps frame timing. | Used to posterize motion (simulating 12fps).5 |
| **Dissolve** | Dx | Mixes two inputs. | Switches between "Mouth Open" and "Mouth Closed" states.6 |

## 3. Asset Preparation and Import Strategy

The "scuffed" style—reminiscent of *Dr. Katz* or early Flash animation—relies heavily on the quality of the initial assets. Since the user is "not an animator," the assets must be drawn once, statically, and designed for modular reassembly.

### 3.1 Deconstructing "Ku-chan"

To enable automated lip-sync and boiling lines, the character assets must be separated into discrete PNG files with alpha transparency. The rig requires the following breakdown:

1. **Ku\_Body\_Base.png:** The torso, limbs, and tail. This layer will be static or subject to simple global transforms (bobbing).
2. **Ku\_Head\_Base.png:** The head shape, ears, and eyes (if eyes are static). Importantly, the area where the mouth goes should be blank or have a neutral muzzle line.
3. **Mouth\_Closed.png:** A simple horizontal line, dot, or closed shape.
4. **Mouth\_Open.png:** A circle, oval, or open shape.
5. **Background\_Paper.png:** (Optional) A static texture, though we will generate this procedurally to save disk bandwidth.

**Resolution Strategy:**

For a 1080p output, it is recommended to author assets at approximately 2000x2000 pixels. This allows for reframing (zooming in/out) within the node graph without quality loss. The "scuffed" look often benefits from rough, aliased edges; therefore, disabling "Anti-aliasing" or using a "Pencil" tool in drawing software (Procreate, Photoshop) is advantageous.

### 3.2 The Fusion Interface Setup

Upon entering the Fusion Page in DaVinci Resolve, the interface presents two Viewers (1 and 2) and a Node Graph (Flow) at the bottom.

1. **Media Pool:** Drag the four PNG files into the Node Graph. They will appear as MediaIn nodes.
2. **Renaming:** Rigging requires precise organization. Select each node and press F2 to rename them logically: In\_Body, In\_Head, In\_MouthClosed, In\_MouthOpen.
3. **Color Space Management:** Ensure that the "Gamut" settings in Project Settings are consistent. Usually, for simple PNG graphics, sRGB is sufficient. Fusion operates in linear space, but MediaIn nodes usually handle the gamma conversion automatically based on file tags.

## 4. Construction of the Static Rig

The first phase of construction is assembling the static puppet. We build from the "back" (the background/body) to the "front" (the mouth).

### 4.1 Hierarchical Merging

The logic of the rig is hierarchical: The Mouth is a child of the Head, and the Head is a child of the Body.

1. **Body and Head Assembly:**
   * Drag the output (grey square) of In\_Body to the output of In\_Head. Fusion automatically creates a Merge node (let's name it Merge\_Head\_Body).
   * **Verification:** Ensure In\_Body connects to the **Yellow** (Background) input and In\_Head connects to the **Green** (Foreground) input. If they are swapped, the body will appear *over* the head. Use Ctrl+T (Command+T) to swap inputs on the Merge node.
2. **Pivot Point Adjustment:**
   * To allow the head to bob naturally, we must set the pivot point. Add a Transform node (XF\_Head) immediately after In\_Head but before Merge\_Head\_Body.
   * In the Viewer, move the Pivot (the crosshair widget) to the base of the neck. Now, if you adjust the Angle or Size in the Inspector, the head rotates from the neck, not the center of the face.

### 4.2 The Mouth State Switch

We need a mechanism to switch between the "Closed" and "Open" mouth states based on a single value (0 to 1). This is the foundation of the automated lip sync.

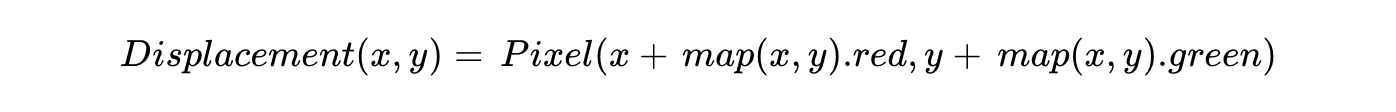
1. **The Dissolve Node:**
   * Add a Dissolve node (DX\_Mouth) to the graph.
   * Connect In\_MouthClosed to the **Background** (Yellow) input.
   * Connect In\_MouthOpen to the **Foreground** (Green) input.
2. **The Control Mechanism:**
   * Select the Dissolve node. In the Inspector, locate the **Background/Foreground** slider.
   * Dragging this slider from 0.0 to 1.0 creates a cross-fade between the closed and open mouth.
   * *Note on Style:* For a smooth animation, a cross-fade is fine. For the requested "scuffed" or anime style, we will eventually want this to snap instantly (0 or 1, no in-between). This will be handled in the automation section using thresholding expressions.
3. **Merging the Mouth:**
   * Connect the output of DX\_Mouth to the **Green** input of a new Merge node (Merge\_Face).
   * Connect the output of XF\_Head to the **Yellow** input of Merge\_Face.
   * *Hierarchy Check:* The flow is now: Mouths -> Dissolve -> Merge(over Head) -> Merge(over Body).

## 5. Procedural Imperfection: The Boiling Line Effect

The "Boiling Line" effect (often called Squigglevision) simulates the frame-by-frame variation of hand-traced animation. This is critical for the "low-fi" aesthetic, as it breathes life into static assets.4

### 5.1 The Mathematical Mechanism: Displacement

In Fusion, we do not redraw lines; we spatially warp the pixels of the static image using a randomized map.



### 5.2 Creating the Noise Map (The Driver)

1. **FastNoise Node:** Add a FastNoise node (FN\_Boil) to the graph. This node generates procedural Perlin noise.
2. **Configuration for "Boil":**
   * **Detail:** Set to 0. We want smooth, amorphous blobs, not grainy grit. High detail looks like static; low detail looks like paper warping.
   * **Contrast:** Increase to 2.0 or higher. We need distinct areas of black and white to push pixels in different directions.
   * **Scale:** Set to 10.0 - 20.0. This determines the frequency of the wobble. Too small makes the edges "fizz"; too large makes the character "swim."
   * **Seethe Rate:** This is the animation engine. Set to 0.5 (slow wobble) to 2.0 (nervous jitter). If this is 0, the effect is static.4

### 5.3 Applying the Displacement (The Warper)

1. **Displace Node:** Add a Displace node (DSP\_Boil) after the final Merge of the character rig.
2. **Connections:**
   * Connect the Character Rig output to the **Yellow** (Input).
   * Connect FN\_Boil output to the **Green** (Foreground/Map) input.
3. **Refining the Look:**
   * **Type:** Set to **XY**. This allows the noise to push pixels horizontally and vertically.
   * **X/Y Refraction:** These sliders control the intensity. Start very low (0.005). Even a value of 0.02 can destroy the image. We want a subtle imperfection, not a funhouse mirror.
   * **Light Power/Shadow:** Uncheck these. We only want geometric distortion, not pseudo-3D lighting effects.7

### 5.4 Temporal Quantization (The "Anime" Framerate)

A "scuffed" look requires a low framerate (e.g., 12fps or 8fps) to emulate hand-drawn animation on "twos" or "threes." However, the timeline is likely 24fps or 30fps. We must artificially drop frames *only* for the character, while the camera moves might remain smooth.

**Option A: The Stop Motion Node (Studio/ResolveFX)**

If available (sometimes restricted to Studio), the Stop Motion node (ResolveFX Temporal) is the easiest solution.

* **Frame Repeat:** Set to 2 (effectively 12fps on a 24fps timeline).
* **Free Version Availability:** The availability of this specific OpenFX plugin in the Free version varies by version number. If locked (watermarked), use Option B.8

**Option B: The TimeStretcher Node (Universal/Free)**

This is the robust, math-based method available in all versions.

1. Add a TimeStretcher node (TS\_Stutter) at the end of the chain.
2. **Expression Setup:**
   * Right-click the Source Time parameter in the Inspector.
   * Select Expression.
   * Enter the formula: floor(time/2)\*2
3. **Explanation:** The time variable is the current frame number (e.g., 25). 25 / 2 = 12.5. floor(12.5) = 12. 12 \* 2 = 24.
   * Frame 24 -> Output 24.
   * Frame 25 -> Output 24.
   * Frame 26 -> Output 26. This effectively holds every frame twice, creating a 12fps look.5

## 6. Automated Audio-Reactive Lip Sync

This section addresses the user's critical need for "Simple lip sync synced to TTS." This is the most technically complex aspect because Fusion is primarily an image compositor, not an audio analyzer.

### 6.1 The Challenge: Fusion's Audio Limitations

Fusion does not natively "listen" to audio tracks on the timeline to drive animation parameters. Unlike After Effects, which has a "Convert Audio to Keyframes" function, Fusion requires the audio to be imported directly into the Node Graph or processed via a modifier.

### 6.2 Method A: The "Reactor" Workflow (Recommended)

The Fusion community has developed a standard solution: the **"Suck Less Audio"** modifier (often labeled **Audio (Wav)**), available via the **Reactor** package manager. This is widely considered the *only* viable workflow for automated daily production in Fusion without advanced coding.6

**Step 1: Installing Reactor (One-Time Setup)**

1. Download the Reactor-Installer.lua file from the WeSuckLess community.
2. Drag and drop this file into the Fusion Node Graph area.
3. An installation window appears. Click "Install."
4. Once installed, open the Reactor menu (Workspace > Scripts > Reactor > Open Reactor).
5. Search for "Audio" and install the Audio (Wav) or Suck Less Audio modifier. Restart Resolve.

**Step 2: Driving the Mouth**

1. Import your daily TTS audio file into Fusion using a Loader node. (Though the modifier reads from disk, having the loader helps visualize the waveform if using the AudioWaveform tool for debugging).
2. Select the Dissolve node (DX\_Mouth) created in Section 4.2.
3. Right-click the **Background/Foreground** slider.
4. Select **Modify With** > **Audio (Wav)**.
5. **Configuration:**
   * A "Modifiers" tab will appear at the top of the Inspector.
   * **Source:** Click "Browse" and select the *same* WAV file used for the podcast.
   * **Amplitude Offset:** 0.0.
   * **Amplitude Scale:** 2.0 to 4.0. TTS audio is often compressed/normalized. You need to boost the signal so that spoken words hit the value of 1.0.

**Step 3: Thresholding (The "Snap")**

The audio modifier outputs a continuous float value (e.g., 0.3, 0.7, 0.9). This makes the mouth fade in and out. For the "scuffed" look, we want a binary state: Open or Closed.

1. We need to apply a conditional logic. However, you cannot directly write an expression *on top* of a modifier in the same field.
2. **The Workaround:** Use a Custom Tool or intermediate control.
   * *Simpler Method:* Right-click the **Background/Foreground** slider *again* (on the modifier control itself, specifically the Level or Output if accessible, but usually we modify the curve).
   * *The Curve Method:* In the Audio Modifier settings, look for the **Response Curve** or mapping. Adjust the curve so it shoots straight up to 1.0 immediately after 0.0.
   * *The Expression Method (Robust):* Instead of modifying the Dissolve directly, modify a generic slider on a dummy node (e.g., BrightnessContrast). Then, on the Dissolve node, write an expression: iif(BrightnessContrast1.Gain > 0.1, 1, 0).
     + iif is "Inline If."
     + Check: Is the Audio-driven Gain > 0.1 (Noise floor)?
     + True: Return 1 (Mouth Open).
     + False: Return 0 (Mouth Closed).

### 6.3 Method B: The Native "Probe" Method (Free/No-Plugin)

If the user cannot install Reactor (due to security/IT restrictions or simplicity), there is a native method using the **Probe** modifier. This converts audio to visual brightness and "probes" it.

1. **Visualize Audio:** In the Edit Page, you cannot easily render a waveform to video in Free.
2. **The Workaround:** You must use a 3rd party tool (or an online visualizer) to convert the daily audio into a black-and-white video where loudness = brightness.
3. **Import to Fusion:** Load this video.
4. **Probe:** On the Dissolve slider, Right-click > **Modify With** > **Probe**.
5. **Input:** Drag the audio-video node into the "Image to Probe" input of the modifier.
6. **Logic:** The modifier reads the pixel brightness of the video and drives the slider.
   * *Verdict:* This is **too slow** for daily production. It requires an intermediate render step every day. Method A (Reactor) is strongly advised despite the installation hurdle.12

### 6.4 Handling Qwen3-TTS Audio

Since Qwen3-TTS output is likely clean and consistent, the thresholding (Step 3 in Method A) is reliable. Unlike a human recording with breath sounds and background noise, TTS drops to absolute silence (digital zero) between words, making the > 0.01 threshold extremely accurate for lip sync.

## 7. Python Scripting for Daily Automation

The user creates *daily* videos. Manually opening Fusion, browsing for the new audio file in the Audio Modifier, and re-rendering is a friction point that can be eliminated. We can use DaVinci Resolve's Python API to automate the asset swap.

### 7.1 Scripting Environment Setup

* **Resolve Studio vs Free:** External scripting (running a .py file from Terminal) is technically a Studio feature. However, **internal scripting** (Workspace > Console > Python) works in the Free version.
* **Preparation:** You must save your Fusion project as a template. Rename the specific nodes you need to target:
  + Rename the Loader for the audio file to AudioLoader.
  + Rename the Audio Modifier is trickier to target via script.
  + *Alternative Strategy:* Use a **Loader** node for the audio file (even if using Reactor, Reactor can sometimes reference a Loader's path). Or, use the script to update the file path in a Loader that is connected to the audio logic.

### 7.2 The Automation Logic

The script will:

1. Find the Fusion composition on the current timeline.
2. Locate the Loader node responsible for the audio.
3. Update its Clip property to the new daily file.
4. Trigger the render.

### 7.3 The Python Script

Copy this code into a text file or the Resolve Console.

Python

#!/usr/bin/env python  
import DaVinciResolveScript as dvr\_script  
import sys  
import os  
  
# 1. Initialize API Connection  
resolve = dvr\_script.scriptapp("Resolve")  
fusion = resolve.Fusion()  
projectManager = resolve.GetProjectManager()  
project = projectManager.GetCurrentProject()  
  
if not project:  
 print("Error: Please open the Ovi English School project.")  
 sys.exit()  
  
# 2. Target the Composition  
# We assume the Rig is the first clip on the current timeline  
timeline = project.GetCurrentTimeline()  
if not timeline:  
 print("Error: No timeline active.")  
 sys.exit()  
   
clip = timeline.GetCurrentVideoItem()  
comp = clip.GetFusionCompByIndex(1)  
  
if not comp:  
 print("Error: No Fusion composition found on the first clip.")  
 sys.exit()  
  
print(f"Targeted Composition: {comp.GetName()}")  
  
# 3. Define Daily Variables  
# In a real workflow, you might pass these as arguments or read from a config file  
import datetime  
today\_str = datetime.date.today().strftime("%Y-%m-%d")  
# CHANGE THIS PATH to match your actual file structure  
daily\_audio\_path = f"C:/Podcast\_Assets/Audio/ovi\_{today\_str}.wav"  
  
# Check if file exists to prevent errors  
if not os.path.exists(daily\_audio\_path):  
 print(f"Warning: Audio file not found at {daily\_audio\_path}")  
  
# 4. Update the Audio Loader  
# You MUST rename your Loader node in Fusion to 'AudioLdr' for this to work  
loader\_tool = comp.FindTool("AudioLdr")  
  
if loader\_tool:  
 # Fusion uses a dictionary-like access for some properties,   
 # but the 'Clip' input is often a direct property or input.  
 # The syntax to set a clip path in Python for Fusion:  
 loader\_tool.Clip = daily\_audio\_path  
 print(f"Updated Audio Loader to: {daily\_audio\_path}")  
else:  
 print("Error: Could not find node named 'AudioLdr'")  
  
# 5. Render Setup (Optional - if you want to auto-render)  
# This uses the 'Quick Export' or specific Render Settings  
project.SetRenderSettings({  
 "SelectAllFrames": 1,  
 "TargetDir": "C:/Podcast\_Assets/Renders/",  
 "CustomName": f"ovi\_daily\_{today\_str}",  
 "Format": "mp4",  
 "Codec": "H264"  
})  
  
project.AddRenderJob()  
# project.StartRendering() # Uncomment to auto-start  
  
print("Daily setup complete. Verify sync and render.")

**Note on Node Paths:** When using the **Reactor Audio Modifier**, the modifier stores the file path in its own internal settings, not on a standard Loader node. Scripting a modifier's internal inputs is significantly more complex than scripting a standard tool.

* **Workaround:** If scripting proves difficult for the modifier, it is faster to simply **expose the Modifier's file path** as a Macro control (see Section 8), which allows you to drag-and-drop the file in the Edit Page inspector, bypassing the need for Python for that specific step.14

## 8. Templating: The "Macro" Strategy

To satisfy the "Fast Production" requirement, the user should ideally **never** open the Fusion page during daily production. We will package the entire rig into a **Macro** (a reusable template) for the Edit Page.

### 8.1 Designing the Macro

1. **Selection:** In the Fusion flow, hold Ctrl and select every node **except** MediaOut.
2. **Creation:** Right-click one of the selected nodes > Macro > Create Macro.
3. **The Macro Editor:** A window appears listing every parameter of every node selected. This is where you choose what to expose to the user interface.
   * **Audio File:** Find your Audio (Wav) modifier (or the Loader if using one). Check the Sound File or Clip parameter. Rename it "Daily Audio."
   * **Head Bob:** Find the Transform node for the head. Check Center. Rename it "Head Position."
   * **Character Position:** Find the Transform node for the whole body (if exists). Check Center and Size.
   * **Boil Intensity:** Find the FastNoise. Check Seethe Rate. Rename it "Jitter Speed."
4. **Saving:**
   * Give the macro a name: KuChan\_Rig\_v1.
   * Save it as a .setting file.

### 8.2 Installing for the Edit Page

To make this appear in the Edit Page "Effects" library:

1. **Windows Path:** %AppData%\Blackmagic Design\DaVinci Resolve\Support\Fusion\Templates\Edit\Titles\
2. **Mac Path:** ~/Library/Application Support/Blackmagic Design/DaVinci Resolve/Fusion/Templates/Edit/Titles/
3. **Usage:** Restart Resolve. Go to the Edit Page. Open "Effects" > "Titles." You will see KuChan\_Rig\_v1. Drag it onto the timeline.
4. **Daily Workflow:** You now have a "Clip" on the timeline. Click it. In the Inspector, you will see the "Daily Audio" browse button. Select your new TTS file. The cat animates automatically. **No Fusion page required.**.15

## 9. Hardware Optimization (MacBook Pro M5)

The user is on an M5 MacBook Pro, which is a powerful machine, but Fusion can still be demanding.

### 9.1 Render Cache Strategies

The Displace node driven by FastNoise is computationally expensive because it calculates pixel-level distortion for every frame.

1. **Smart Cache:** In the Edit Page, go to Playback > Render Cache > Smart.
2. **Fusion Output Caching:** Right-click the Ku-chan clip on the timeline > Render Cache Fusion Output > On.
   * *Effect:* A red line will appear above the clip. Resolve will background-render the Fusion frames. When the line turns blue, playback is effortless.
   * *Storage:* Ensure your Cache location (Project Settings > Master Settings > Working Folders) is on a fast SSD (the internal Mac drive is perfect).

### 9.2 Proxy Mode

If the UI becomes sluggish:

* Go to Playback > Timeline Proxy Resolution > Half.
* This renders the viewport at lower resolution. Since the "scuffed" style is low-fi, this visual degradation is often unnoticeable during editing but drastically improves speed.18

## 10. Comparative Analysis: Why Fusion?

Comparing this workflow to industry alternatives highlights why Fusion is the correct choice for this specific user.

| **Feature** | **DaVinci Fusion** | **Adobe After Effects** | **Blender (Grease Pencil)** |
| --- | --- | --- | --- |
| **Cost** | Free (Included) | Subscription (~$22/mo) | Free |
| **Integration** | **Seamless** (Edit+Comp in one app) | **Disconnected** (Requires Dynamic Link or Export) | **Disconnected** (Requires Rendering) |
| **Audio Sync** | Difficult Native / Easy with Reactor | Native "Convert Audio to Keyframes" | Native "Bake Sound to F-Curve" |
| **Boiling Line** | Procedural (Nodes) | Expression wiggle() or Effects | Native Noise Modifier |
| **Daily Speed** | **Fastest** (Once templated as Macro) | Slow (Must open AE, render, import to Premiere) | Slow (Must render, import to NLE) |

**Conclusion on Alternatives:**

While After Effects offers easier *native* audio tools, the **Round-Trip Penalty** (exporting video from AE to import into an editor for the podcast) adds significant friction to a daily workflow. Fusion's ability to exist as a template on the editing timeline means the user creates the video and edits the audio in a single pass.

## 11. Conclusion

The "Ku-chan" pipeline transforms character animation from an artistic task into a procedural systems-engineering task. By investing upfront time in constructing the Node Graph—specifically the hierarchical merge structure, the noise-driven displacement for "scuffed" aesthetics, and the audio-driven thresholding for lip sync—the user creates a robust asset that requires zero daily keyframing.

For a beginner using the Free version of Resolve on Apple Silicon, this workflow is not only viable but optimal. It maximizes the hardware's GPU capabilities (Metal) while circumventing the software's native audio limitations via community tools (Reactor) or clever templating (Macros). The result is a high-velocity production line capable of generating stylized content at the speed of speech.

### Appendix: Node Tree Reference Summary

1. **FN\_Boil** (FastNoise) -> **Green Input** of **DSP\_Boil** (Displace).
2. **AudioLdr** (Loader) -> **Modifier** (Audio Wav) -> Controls **Mix** of **DX\_Mouth**.
3. **LD\_MouthClosed** + **LD\_MouthOpen** -> **DX\_Mouth** (Dissolve).
4. **LD\_Head** -> **XF\_Head** (Transform/Bob) -> **Mrg\_Face** (Merge).
5. **DX\_Mouth** -> **Green Input** of **Mrg\_Face**.
6. **LD\_Body** -> **Yellow Input** of **Mrg\_Body**.
7. **Mrg\_Face** -> **Green Input** of **Mrg\_Body**.
8. **Mrg\_Body** -> **Yellow Input** of **DSP\_Boil**.
9. **DSP\_Boil** -> **TS\_Stutter** (TimeStretcher 12fps) -> **MediaOut**.

#### Works cited

1. Animating Motion Graphics in Fusion - Resolve 18 Advanced Tutorial - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=yj11wSo0Zug>
2. Fusion - Blackmagic Design, accessed February 1, 2026, <https://documents.blackmagicdesign.com/UserManuals/Fusion17_Manual.pdf>
3. Animate ANY Character in Davinci Resolve! - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=rZRcc8lCFDE>
4. Fast Noise Fusion In DAVINCI RESOLVE TUTORIAL - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=hzXc48E8Ojo>
5. Davinci Resolve 15 (/Fusion) – Posterize Time Tutorial - A Blackbird Called Sue, accessed February 1, 2026, <https://ablackbirdcalledsue.com/2018/04/22/davinci-resolve-15-fusion-posterize-time-tutorial/>
6. Change Clip Transparency Using Audio Davinci Resolve - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=kCnIzcSJwno>
7. Displace and Fast Noise Nodes in Davinci Resolve Fusion - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=5CpWZxCC3dM>
8. DaVinci Resolve 20 Fusion Stop Motion Node - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=42KVihfrLMo>
9. How to get this effect with free Davinci? : r/davinciresolve - Reddit, accessed February 1, 2026, <https://www.reddit.com/r/davinciresolve/comments/16ijupc/how_to_get_this_effect_with_free_davinci/>
10. Davinci Resolve 15 (/Fusion) - Posterize Time (strobe/stop motion effect) tutorial - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=og0JpMSjp2c>
11. Playhead Scrubbing with Audio in Fusion for Animation Sync - Blackmagic Forum, accessed February 1, 2026, <https://forum.blackmagicdesign.com/viewtopic.php?uid=16&f=22&t=210276&start=0>
12. How to Use PROBE MODIFIER In Davinci Resolve - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=bXa6zrio6zo>
13. Fusion Animation Basics #5 | DaVinci Resolve | The Probe Modifier - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=RW1vlpThZ5U>
14. How to create a working Loader node in a fusion composition using scripting API - Reddit, accessed February 1, 2026, <https://www.reddit.com/r/davinciresolve/comments/1o6n0nq/how_to_create_a_working_loader_node_in_a_fusion/>
15. Making Fusion Templates in DaVinci Resolve - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=fLVIynCwQKM>
16. Build new Fusion tools with Macros and Templates - VFXstudy, accessed February 1, 2026, <https://vfxstudy.com/tutorials/macros-templates/>
17. Make Fusion Templates For The Edit Page - DaVinci Resolve 17 - YouTube, accessed February 1, 2026, <https://www.youtube.com/watch?v=IBv4wGVraBs>
18. Best Practices for Smooth DaVinci Resolve Performance on Mid - Blackmagic Forum, accessed February 1, 2026, <https://forum.blackmagicdesign.com/viewtopic.php?t=231540&p=1196091>