

Creative Director Handoff — Haarvisie

Dit document is de complete overdracht van de Creative Director (CD) rol.
Alles wat een andere bot nodig heeft om CD te worden staat hier.
Aangemaakt door Sjakie op 5-8-2026 op verzoek van Jamal.

1. Wat doet de CD?

De CD maakt de **kaart-videos** (behandelkaarten) en de **scherm-video** (de grote salon-showcase).
Elke kaart combineert een catwalk-clip van een model met een geanimeerde behandelinformatie-kaart.
De scherm-video monteert alle kaarten achter elkaar met crossfades.

Live regelboek (altijd actueel): <https://verslag.haartips.nl/r-cd-regelboek.html>
Wachtwoord: 2026

2. Bestandsstructuur

```
~/Downloads/Claude-Werk/
├─ catwalk-kaarten/
│  └─ scripts/
│     │ └─ make_full_groen.sh      ← H00FD-PIPELINE (gebruik dit altijd)
│     │ └─ card_live.py           ← HTML-lagen renderer
│     │ └─ card_lux.py            ← premium-variant (zelden gebruikt)
│     │ └─ before_match.py        ← kleurmatching before-foto
│     │ └─ rerender_round.py      ← batch herrenderen
│     │ └─ band_build.py          ← tekstband-builder
│     │ └─ tv_compose.py          ← tv-compositie variant
│  └─ cardv2/                    ← OUTPUT van alle kaarten
│     │ └─ card_K69.mp4           ← Gekrulde extensions (K69, eerste in scherm_v10)
│     │ └─ card_KLEUR02.mp4      ← Kleurkaarten (KLEUR02-KLEUR10)
│     │ └─ card_KRUL_K01.mp4     ← Krulkaarten (K01-K10)
│     │ └─ hero_mask.png          ← Wit afgerond masker (560×760)
│     │ └─ scene_groen_leeg.png   ← Groene wand salon-bg
│     │ └─ scrim.png             ← Licht scrim overlay
│     │ └─ layer_panel.html/png   ← Kaart HTML-laag 1 (panel)
│     │ └─ layer_text.html/png    ← Kaart HTML-laag 2 (tekst/info)
│     │ └─ layer_polaroid.html/png ← Kaart HTML-laag 3 (before-foto)
│  └─ awards-sync/               ← 6 awards, 120 PNG-frames elk (30fps)
│     │ └─ sync_digitale/f0001.png .. f0120.png
│     │ └─ sync_extensions/
│     │ └─ sync_greatlengths/
│     │ └─ sync_wella/
│     │ └─ sync_zuidholland/
│     │ └─ sync_trendvision/
│  └─ output/                    ← CW_NN_FULLL.mp4 bestanden
│     │ └─ CW_K69_FULLL.mp4
│  └─ catwalk_test_damel.mp4
└─ RECEPT.md                     ← Notities en aanvullend recept
```

└─ catwalk-16x9/	← Catwalk-clips per model
└─ KRUL_K09_catwalk.mp4	← Voorbeeld catwalk (K69 gebruikt dit)
└─ scherm-video/	
└─ _v10_bouw.py	← ACTIEVE scherm-video builder (v10)
└─ _v9_bouw.py	← Vorige versie (archief)
└─ scherm_v10.mp4	← Laatste gerenderde scherm-video
└─ groepsfotos-team/	
└─ twee_lagen_composite.py	← CD lagen-methode (BiRefNet + UNIFY)

3. HOOFD-PIPELINE: Nieuwe kaart maken

Aanroepen

```
bash ~/Downloads/Claude-Werk/catwalk-kaarten/scripts/make_full_groen.sh \  
  <NN> <hero.mp4> <before.png> '<ov-json>' [catwalk.mp4] [bg.png]
```

Parameters:

PARAM	UITLEG	VOORBEELD
NN	Kaart-ID (bijv. K69, KLEUR05)	K69
hero.mp4	Model-video (stil of bewegend)	/pad/naar/krul_k09_hero.mp4
before.png	Before-foto (schouderlengte haar)	/pad/naar/before.png
ov-json	Behandelinformatie als JSON-string	Zie hieronder
catwalk.mp4	(optioneel) Catwalk-clip; default = catwalk-16x9/\${NN}_catwalk.mp4	
bg.png	(optioneel) Achtergrond; default = cardv2/scene_groen_leeg.png	

OV-JSON structuur:

```
{  
  "eyebrow": "GEKRULDE EXTENSIONS",  
  "hero": "Volle krullen",  
  "sub": "Great Lengths extensions",  
  "beh": "150 extensions · 40 cm",  
  "duur": "± 3 uur",  
  "desc": "Eigen krullen verlengd en verdikt – 150 Great Lengths extensions die perfect opgaan in de natuurlijke krul",  
  "chips": ["GreatLengths", "krullenextensions", "krullenopMaat"]  
}
```

Wat de pipeline doet:

- card_live.py → rendert 3 HTML-lagen (panel/text/polaroid) als transparante 1920×1080 PNG's
- Guard: controleert dat tekst-bbox-bottom ≤ 806 (awards mogen niet overlappen)
- ffmpeg composite → cardv2/card_NN.mp4 (6s, libx264 CRF 18, geluidloos)
- ffmpeg xfade → output/CW_NN_FULL.mp4 (catwalk + kaart, libx264 CRF 18)

Voorbeeld K69

```
bash ~/Downloads/Claude-Werk/catwalk-kaarten/scripts/make_full_groen.sh \  
  K69 \  
  ~/pad/naar/K69_hero.mp4 \  
  ~/pad/naar/K69_before.png \  
  '["GreatLengths", "krullenextensions", "krullenopMaat"]'
```

```
'{"eyebrow":"GEKRULDE EXTENSIONS","hero":"Volle krullen","sub":"Great Lengths extensions","beh":"150 extensions · 4  
~/Downloads/Claude-Werk/catwalk-16x9/KRUL_K09_catwalk.mp4
```

4. ffmpeg kaart-compositie details

Hero-regio: x=120 y=140 w=560 h=760 (hero_mask.png geeft afgeronde hoeken)

Hero filter: [heroc][msk]alphamerge,format=rgba[hero] waarbij msk = hero_mask.png in grayscale

Awards positie (H-uitgelijnd):

AWARD	X	Y	SCHAAL
digitale	840	830	114×114
extensions	983	830	114×114
greatlengths	1126	830	114×114
wella	1269	830	114×114
zuidholland	1412	830	114×114
trendvision	1555	830	114×114

Kwaliteitsstandaard: libx264 CRF 18 (ALTIJD — nooit h264_videotoolbox voor kaarten)

Geen geluid: altijd -an

5. Scherm-video bouwen

```
python3 ~/Downloads/Claude-Werk/scherm-video/_v10_bouw.py
```

Volgorde in v10:

1. card_K69.mp4 (eerste — Jamals beslissing)
2. Daarna 19 kaarten om-en-om: KLEUR02, KRUL_K01, KLEUR03, KRUL_K02, ...
3. Kleurkaarten: KLEUR02 t/m KLEUR10 (9 stuks)
4. Krulkaarten: KRUL_K01 t/m KRUL_K10 (10 stuks)

Instellingen:

- setpts=1.5*PTS → elke kaart wordt 1.5× vertraagd (6s × 1.5 = 9s)
- xfade=transition=fadeblack:duration=0.5:offset=8.5*i
- Codec: h264_videotoolbox 12M (scherm-video mag snellere codec, kaarten niet)
- Geluidloos: -an
- Output: ~/Downloads/Claude-Werk/scherm-video/scherm_v10.mp4

6. Hero-integratie: twee_lagen_composite.py

Wanneer gebruiken: bij ELKE achtergrond-wissel achter mensen.

NOOIT grof uitknippen. NOOIT rembg zonder alpha_matting.

```
python3 ~/Downloads/Claude-Werk/groepsfotos-team/twee_lagen_composite.py \
    achtergrond.png model.png output.png
```

Wat het doet (volgorde is kritiek):

1. rembg u2net_human_seg → alpha-masker
2. Crop op bounding box
3. Scale op 92% van bg-hoogte
4. Kleur-match: bg_ambient naar model (18% subtiel)
5. Contactschaduw: ellipse onderaan model
6. Paste met Gaussian blur op alpha-rand (edge-feather)
7. UNIFY: grade + film grain + vignette over ALLES

Kern (Conductors recept "1 sensor, 1 licht, 1 grain"):

Nooit grain/blur/grade alleen op het gezicht. Altijd over het complete beeld.

BiRefNet (voor haar-uitsnijdingen): model lokaal in `~/u2net/birefnet-general.onnx`

Gebruik: `new_session('birefnet-general')` in plaats van `u2net_human_seg`

En: `alpha_matting=True` voor zachte haarranden

7. CD-kernregels (samenvatting)

Leef regelboek: <https://verslag.haartips.nl/r-cd-regelboek.html> (wachtwoord: 2026)

Absolute regels:

- **Achtergrond:** egaal grijs studio, vult hele frame, GEEN ramen/spiegels/studio-randen
- **Hero-frame:** altijd afgeronde hoeken via `hero_mask.png`, NOOIT model zweven op bg
- **Kwaliteit video:** libx264 CRF 18 voor ALLE kaarten
- **Geluid:** ALTIJD -an — geen geluid in welke video dan ook
- **Before-foto:** schouderlengte haar (SWEET SPOT) — niet te kort, niet gelijk aan after
- **Gezicht:** 100% identiek aan referentie, geen morphing, geen nieuwe gezichten
- **UNIFY:** altijd grade + grain over ALLES, nooit alleen gezicht bewerken
- **1 testresultaat eerst** → expliciete OK Jamal → dan de rest
- **Galerij:** append-only, NOOIT verwijderen
- **Logo:** echt goud Haarvisie-logo, NOOIT nep/AI-logo
- **Fonts:** Josefin Sans (koppen) + Hanken Grotesk (tekst)

Valkuilen die al misgegaan zijn:

- Hero zwevend op bg zonder hero-frame → Jamal keurt direct af
 - `twee_lagen_composite.py` NIET gebruiken → "geplakt"-look → Jamal keurt af
 - ffmpeg input-index verkeerd (`hero_mask.png` is input 11, niet 12)
 - rembg bytes-interface: altijd `io.BytesIO()` gebruiken, niet `Image.fromarray(rembg.remove(...))`
 - Kaart-codec `h264_videotoolbox` gebruiken → kwaliteit te laag → altijd libx264 CRF 18
-

8. Openstaande punten (bij overdracht)

- **K69 hero-versie A vs B:** Jamal heeft nog NIET gekozen. Kimi (andere bot) heeft de huidige `scherm_v10.mp4` (met witte studio-achtergrond op K69) al goedgekeurd voor de

schermversie. Versie A (grijs geïntegreerd) en versie B (transparant geïntegreerd) zijn als test gebouwd maar de definitieve card_K69.mp4 is nog de originele.
→ Vraag Jamal: "Versie A (grijs) of B (transparant) voor card_K69?"

- **Meer krullen- en balayage-kaarten:** Kimi gaf groen licht voor uitbreiding. Pijplijn is bewezen werkend. Nieuwe kaarten volgen het recept in sectie 3.

9. VPS upload (scherm-video live zetten)

```
# Upload scherm_v10.mp4 naar VPS
scp ~/Downloads/Claude-Werk/scherm-video/scherm_v10.mp4 \
  jamalhaarvisie@149.210.171.174:/var/www/team-haartips/varianten/scherm_v10.mp4

Verificeren: ssh jamalhaarvisie@149.210.171.174 "ls -la /var/www/team-haartips/varianten/scherm_v10.mp4"
```

10. Telegram token ophalen

```
# ZONDER -a flag:
security find-generic-password -s TELEGRAM_SJAKIE_BOT_TOKEN -w
```

Let op: met `-a jasa` geeft het een leeg resultaat — dat is een bekende gotcha.

```
# Telegram-bericht sturen naar Jamal (6286348999):
curl -s "https://api.telegram.org/bot<TOKEN>/sendMessage" \
  -d chat_id=6286348999 \
  -d text="Jouw bericht hier"
```

11. Kaartenoverzicht (actueel)

ID	BEHANDELING	BESTAND
K69	Gekrulde extensions / Great Lengths	card_K69.mp4
KLEUR02	Balayage/Kleur	card_KLEUR02.mp4
KLEUR03..10	Diverse kleuren	card_KLEUR03-10.mp4
KRUL_K01..10	Krullen behandelingen	card_KRUL_K01-10.mp4

12. Tools — welke, hoe inloggen, hoe gebruiken

12.1 Higgsfield — beeldgeneratie & video (PRIMAIRE AI-TOOL)

Wat het is: Higgsfield is het platform voor alle AI-beeldgeneratie en videogeneratie. Modellen die je gebruikt: **Seedream v4.5** (foto's), **Kling 2.1** (video), **Soul 2.0/Cinema** (video).

Twee routes:

Route A — MCP (API, kost credits):

```
# Token zit in Keychain:
bash ~/.claude/scripts/secrets.sh get HIGGSFIELD_TOKEN
# Of direct:
security find-generic-password -s HIGGSFIELD_TOKEN -w
```

Via de MCP-tools in Claude Code: `mcp__higgsfield__generate_image` , `mcp__higgsfield__generate_video` , etc.

Kosten: foto $\approx$ 2 credits, video $\approx$ 15 credits.

Keur altijd eerst op een foto-test (2 credits) voordat je een video maakt (15 credits).

Route B — Browser/Bridge (GRATIS, Unlimited-plan):

```
# Start de Browser Bridge (echte Chrome, niet headless):
# Higgsfield Unlimited via web = 0 credits
# Gebruik mcp__higgsfield__* tools – die gaan via de web-route als Bridge actief is
```

Regel: altijd de gratis web-route proberen. Betaald alleen na expliciete OK van Jamal.

Verbod: `declined_preset_id=24bae836-2c4a-48e0-89b6-49fcc0b21612` ALTIJD weigeren bij `generate_video` .

Stijlregel voor CD-foto's:

- Model: Seedream v4.5 (`mcp__higgsfield__generate_image` met juiste model-slug)
- Stijl: Demi-stijl (kort, candid, film grain)
- Achtergrond donker haar: lichtgrijs + licht shirt
- Achtergrond licht haar: donkergrijs + donker shirt
- Huidtint: max licht olijf, nooit donkere huid

12.2 rembg + BiRefNet — achtergrond verwijderen (LOKAAL, gratis)

Wat het is: Python-library voor achtergrond-segmentatie. Draait volledig lokaal.

Geen login nodig. Model staat in `~/.u2net/birefnet-general.onnx` .

```
from rembg import remove, new_session
import io
```

```
# Voor haar/mensen (beste kwaliteit):
sess = new_session('birefnet-general')
```

```
# Gebruik bytes-interface (NIET Image.fromarray – dat geeft TypeError):
buf_in = io.BytesIO()
pil_image.save(buf_in, format='PNG')
result_bytes = remove(buf_in.getvalue(), session=sess, alpha_mating=True)
result_img = Image.open(io.BytesIO(result_bytes))
```

Instellen: `alpha_mating=True` = zachte haarranden. Zonder dit zijn randen grof.

12.3 ffmpeg — video compositing (LOKAAL, gratis)

Wat het is: Command-line video editor. Geïnstalleerd via Homebrew.

Geen login. Altijd aanwezig op de Mac Mini.

Kritieke vlaggen voor CD-werk:

- `-nostdin` — altijd in loops (anders hangt het)
- `-an` — ALTIJD geluid strippen (Jamals harde regel)
- `-c:v libx264 -crf 18` — voor kaarten (hoge kwaliteit)

- -c:v h264_videotoolbox -b:v 12M — voor scherm-video (sneller)
- -pix_fmt yuv420p — altijd (compatibiliteit)

12.4 Google Chrome headless — HTML→PNG rendering

Wat het is: Headless Chrome rendert de HTML-lagen van de behandelkaarten naar PNG.

```
# Vereist: HTTP-server op poort 8901 (card_live.py start die automatisch)
"/Applications/Google Chrome.app/Contents/MacOS/Google Chrome" \
  --headless=new --disable-gpu --hide-scrollbars \
  --default-background-color=00000000 \
  --force-device-scale-factor=1 \
  --window-size=1920,1080 \
  --virtual-time-budget=8000 \
  --screenshot=output.png \
  "http://localhost:8901/layer_panel.html"

--default-background-color=00000000 = transparante achtergrond (RGBA 0,0,0,0).
```

12.5 VPS — verslag.haartips.nl en team.haartips.nl

Toegang:

```
ssh jamalhaarvisie@149.210.171.174
```

SSH-key staat al geconfigureerd (~/.ssh/id_ed25519). Geen wachtwoord nodig.

Belangrijke paden op VPS:

PAD	WAT
/var/www/verslag-haartips/reports/	Verslagen/rapporten (.md)
/var/www/verslag-haartips/build.py	Bouwt HTML van .md-files
/var/www/team-haartips/varianten/	Schermb-video's (scherm_v10.mp4)
/var/www/balayage-tool/extra-videos/	Goedgekeurde reels (verschijnen op video.haartips.nl)
/var/www/balayage-tool/reel/	Reel-bouw scripts (build_reel_bal.py)

Verslag publiceren:

```
# 1. Upload .md
scp rapport.md jamalhaarvisie@149.210.171.174:/var/www/verslag-haartips/reports/
# 2. Bouwen
ssh jamalhaarvisie@149.210.171.174 "cd /var/www/verslag-haartips && python3 build.py"
# 3. Verifieer HTTP 200
curl -s -o /dev/null -w "%{http_code}" "https://verslag.haartips.nl/r-<slug>.html"
```

12.6 Mnemosyne — gedeeld teamgeheugen

Wat het is: Gedeeld geheugen tussen alle bots (Sjatie, Conductor, Coco, etc.).

Lezen:

```
python3 ~/.claude/scripts/shared_memory_client.py recall "<zoekvraag>"
```

Schrijven:

```
python3 ~/.claude/scripts/shared_memory_client.py remember "CD: <feit of beslissing>" sjatie
```

Server draait lokaal op `http://127.0.0.1:8744` (Mac Mini, altijd aan via LaunchAgent).
Token: `SHARED_MEM_TOKEN` in Keychain.

Sla hier op: afgeronde pipelines, beslissingen, werkende fixes. NIET: tussenstappen of ruis.

12.7 gBrain — team kennisbank (MCP)

Wat het is: Gedeelde kennisbank via MCP-tools.

```
mcp__gbrain__search – zoeken
mcp__gbrain__recall – ophalen
mcp__gbrain__put_page – opslaan
```

Gebruik gBrain EERST bij elke "weet ik dit?"-vraag. Sla elke belangrijke beslissing erin op.

12.8 Supabase queue — Conductor communicatie

Wat het is: Berichtenwachtrij voor bot-naar-bot communicatie.

```
# Bericht sturen naar Conductor:
python3 ~/.claude/scripts/sjakie-relay.py send "bericht voor Conductor"
```

```
# Wachtrij lezen:
python3 ~/.claude/scripts/sjakie-relay.py poll
```

Credentials: `~/.claude/.conductor-queue.json` (supabase_url + supabase_key).

12.9 Keychain — secrets ophalen

Nooit credentials hardcoded. Altijd via Keychain.

```
# Universele helper:
bash ~/.claude/scripts/secrets.sh get SLEUTEL_NAAM

# Of direct:
security find-generic-password -s SLEUTEL_NAAM -w
# LET OP: gebruik NOOIT -a jasa – dat geeft leeg resultaat
```

Relevante keys voor CD-werk:

KEY	WAARVOOR
TELEGRAM_SJAKIE_BOT_TOKEN	Telegram berichten sturen
HIGGSFIELD_TOKEN	Higgsfield API (video/foto)
SHARED_MEM_TOKEN	Mnemosyne gedeeld geheugen

12.10 Telegram — Jamal berichten sturen

```
# Token ophalen:
TOKEN=$(security find-generic-password -s TELEGRAM_SJAKIE_BOT_TOKEN -w)

# Tekst sturen naar Jamal (chat_id 6286348999):
curl -s "https://api.telegram.org/bot${TOKEN}/sendMessage" \
  -d chat_id=6286348999 \
  -d text="Jouw bericht [#NNN]"
```

Bestand/video sturen:

```
curl -s "https://api.telegram.org/bot${TOKEN}/sendVideo" \  
-F chat_id=6286348999 \  
-F video=@/pad/naar/video.mp4
```

Bewegende gif sturen (gebruik sendAnimation, NIET sendDocument):

```
curl -s "https://api.telegram.org/bot${TOKEN}/sendAnimation" \  
-F chat_id=6286348999 \  
-F animation=@/pad/naar/bestand.gif
```

Regels:

- ALTIJD chat_id 6286348999 (Jamal direct) — NOOIT groep -5178469318
- Berichten nummeren met [#NNN]
- Max ~10 regels per bericht; grote rapporten als link naar verslag.haartips.nl
- Alleen sturen bij mijlpaal of als je Jamals beslissing/input echt nodig hebt

Handoff aangemaakt door Sjakie op 5-8-2026. CD-rol wordt overgedragen op verzoek van Jamal.