

Návrh platformy VQCC pro automatickou technickou kontrolu videí

1. Kompletní datový workflow od uploadu po QC report

Upload a ingest: Uživatel nahraje video soubor (do 4K rozlišení, délky max. 1 hod) přes webové rozhraní aplikace. Frontend (Next.js na Vercelu) může použít přímo nahrávání do cloudového úložiště (např. předpřipravený podpis URL pro S3/Bunny) pro efektivní přenos. Po úspěšném uploadu se vygeneruje událost (např. záznam v databázi nebo zpráva ve frontě), která spustí zpracování souboru.

Zpracování videa (QC pipeline): Backend aplikační vrstvy (např. serverless funkce či microservisa) založí úlohu kontroly kvality videa. Tato úloha může být zpracována asynchronně – uloží se do fronty (např. AWS SQS nebo jiná queue) a specializovaný worker (např. kontejner na GPU instanci) ji zpracuje. Worker stáhne vstupní video z úložiště a provede sadu automatických QC testů (viz další sekce) – typicky pomocí FFmpeg a dalších knihoven pro analýzu videa a audia. Zpracování probíhá off-line, paralelně s možností škálování (další videa lze kontrolovat souběžně na jiných workerech). Díky event-driven architektuře a frontám může být workflow škálovatelné a odolné [1](#) [2](#) – upload souboru vyvolá zpracování, které po dokončení vygeneruje výstup.

Uložení výsledků a report: Výstupy QC analýzy (nalezené chyby, metadate, statistiky) se uloží do datové vrstvy. Strukturovaná data mohou být v JSON formátu uložená v Neon PostgreSQL (NeonDB) pro další práci (umožní dotazování výsledků, sestavování statistik apod.). Pro uživatele se zároveň vygeneruje čitelný QC report – buď jako webová stránka v aplikaci (s možností stáhnout PDF) nebo jako soubor (PDF/HTML) uložený v úložišti. Report obsahuje souhrnné informace o videu (formát, délka, kodeky), a seznam zjištěných problémů s časovými kódy a popisem. Po uložení výsledků může backend například pomocí e-mailu nebo notifikace informovat uživatele, že analýza je hotova a report připraven.

Distribuce reportu: QC report je dostupný v uživatelském rozhraní – uživatel jej může zobrazit, stáhnout nebo sdílet (např. odkazem kolegům/klientovi). Využití cloudového úložiště a CDN zajistí rychlé dodání reportu globálně. Bunny.net například nabízí integrovaný CDN s **edge** úložištěm; S3 lze kombinovat s CloudFront CDN. (*Pozn.: Bunny.net Storage nemá nativní S3 API, ale je velmi levné \$0.01/GB měsíčně a soubory jsou geo-replikované* [3](#).)

Čištění a opakování běhy: Po nějaké době je možné vstupní videa či mezivýsledky smazat pro úsporu místa (podle nastavených pravidel retention). Aplikace by měla umožnit opakovat analýzu (např. po úpravách videa) a v budoucnu porovnat výsledky různých verzí téhož videa. Celý workflow od uploadu po report je plně automatický – minimalizuje ruční zásahy a umožňuje uživatelům rychle zjistit technickou kvalitu videa v clodu [4](#) [5](#).

Shrnutí: Uživatel nahraje video → triggne se asynchronní zpracování (na GPU workeru) → vygeneruje se záznam v DB a QC report → uživatel notifikován a může si report prohlédnout. Toto oddělení frontend uploadu a backend zpracování zajistí, že i velká videa (4K/1h) lze zpracovat spolehlivě, aniž by uživatel musel mít stále otevřený prohlížeč.

2. Návrh QC skriptů a testů

Automatické QC skripty v MVP budou detekovat nejčastější technické nedostatky videa a audia. K dispozici jsou osvědčené nástroje jako **FFmpeg** (případně s Python wrappery jako `ffmpeg-python`) či využití `ffprobe`, knihovna **scikit-video** pro pokročilejší analýzy videa v Pythonu nebo open-source AI modely (HuggingFace či jiné) pro specifické úlohy. Níže jsou navrženy hlavní testy:

- **Detekce black frames (černých snímků):** Skript hledá sekvence snímků, které jsou čistě nebo převážně černé, delší než určitý práh (např. >1 sekundy). FFmpeg nabízí filtr `blackdetect`, který vrací časové úseky s černou obrazovkou. Parametry zahrnují minimální délku tmy (`d`) a práh „černosti“ pixelů (`pix_th`). Např. `ffmpeg -vf "blackdetect=d=2:pix_th=0.1"` najde černé úseky delší než 2 s, přičemž `pix_th=0.1` nastavuje, že pixel je považován za černý pod určitý jas ⁶. Výstupem filtru jsou časové kódy začátku a konce černých úseků ⁷ ⁸. Tyto lze parsovat a uvést v reportu (např. "Černá pauza od 00:10:05 do 00:10:07 (2 sekundy)"). Alternativně lze využít i detekci „blank frames“ z API Encoding.com ⁹ ¹⁰.
- **Detekce freeze frames (zamrzlého obrazu):** Freeze nastává, když se po určitou dobu nemění obraz (identické snímky). FFmpeg má filtr `freezedetect` (dostupný v novějších verzích) pro odhalení dlouhých neměnných úseků ¹¹. Ten porovnává rozdíly mezi snímky; pokud rozdíl klesne pod zadaný šumový práh a trvá déle než nastavenou dobu (`duration`), označí to jako zamrzlý obraz ¹². Např. `freezedetect=n=-60dB:d=2` detekuje freeze delší než 2 s při toleranci šumu -60 dB ¹². Filtr vypíše metadata s časy začátku a konce zamrzlého úseku ¹³, podobně jako `blackdetect`. QC skript může tyto údaje extrahovat a upozornit např. "Zamrzlý obraz od 00:05:10, délka 3.1 s".
- **Detekce tichých úseků:** Pro audio kontrolu je klíčové najít pasáže bez zvuku (pod určitou úroveň hlasitosti). FFmpeg filtr `silencedetect` v audiu označí ticho na základě prahu hlasitosti (v dBFS) a délky ¹⁴. Skript spustí FFmpeg s filtrem na audio stopu (např. `-af silencedetect=noise=-50dB:duration=1`) a z konzolového výstupu získá časové intervaly ticha. Tímto lze zachytit neočekávané výpadky zvuku. (*FFmpeg audio filtry pro detekci ticha jsou analogické k video filtrov - existují tedy „podobné audio filtry pro detekci ticha“* ¹⁵ ¹⁶.)
- **Kontrola formátu a metadat:** Před zpracováním skript vytěží metadata souboru (pomocí `ffprobe` nebo knihovny jako `pymediainfo`). Zkontroluje se kontejner, kodek videa a audia, rozlišení, poměr stran, snímková frekvence, datový tok, počet kanálů atd. Tyto údaje se porovnají s očekávanými/standardy. Například ověříme, že video má zvukovou stopu, že snímková frekvence je konzistentní, že rozlišení není větší než deklarované, atd. Také lze detektovat pokud je video prokládané vs. progresivní. V rámci QC se typicky ověřuje i integrita kontejneru (správná struktura, žádné chybějící snímky) – to lze detektovat přes návratové kódy dekodéru nebo `ffmpeg log` (varování o poškozených framech).
- **Kontrola hlasitosti (loudness):** Důležitým testem je měření průměrné a špičkové hlasitosti podle broadcast standardů (např. EBU R128 / ITU BS.1770). QC skript může využít `ffmpeg` filtr `ebur128`, který při přehrání souboru vypočte integrovanou hlasitost (LUFS) a true peak. Alternativně existuje nástroj `ffmpeg-normalize` nebo knihovny v Pythonu pro BS.1770. Výsledkem by bylo zjištění, zda celková hlasitost videa odpovídá normě (např. -23 LUFS ± tolerance pro broadcast, -14 LUFS pro web, dle nastavení profilu). Pokud ne, report upozorní na odchylku. Rovněž je možné detektovat clipping (přetečení 0 dBFS) – `ffmpeg` v logu ohláší překročení hlasitosti podle CALM Act ¹⁷.

- **Další testy pro MVP:** Základní skripty pokryjí i další jednoduché kontroly:
 - **Barvy/gamut** – ověření, že luma a chroma jsou v legálních mezích (16–235 pro Y, 16–240 pro CrCb u SDI videa apod.). Lze detekovat pomocí histogramu nebo filtru ffmpeg `signalstats`.
 - **Rozlišení a poměr stran** – zda video neobsahuje černé pruhy (letterbox/pillarbox) pokud nemá ¹⁸. Jednoduše: detekce okrajových pruhů pomocí analýzy světlosti krajů snímku.
 - **Blokování a artefakty** – pro MVP spíše vizuální kontrola (ne automatická), ale do budoucna lze nasadit AI model, který vyhodnotí úroveň blokování/komprese (např. bezreferenční metriky nebo trénovaný model na odhad MOS/VMAF ¹⁹ ²⁰). V MVP aspoň uvést bitrate a zda je přiměřený k rozlišení.
 - **Dead pixels** – pro začátek nejspíš neimplementujeme plně (jde o velmi pokročilou detekci na úrovni jednotlivých pixelů). Ale např. QScan v updatech přidal „Advanced Dead Pixel Detection“ ²¹, což později můžeme dosáhnout s pomocí AI (model hledající drobné jasné body ve více snímcích).
 - **Kontrola délky** – ověřit, že deklarovaná délka videa odpovídá skutečnému obsahu (bez nečekaných extra snímků na konci apod.). Porovná se metadata duration vs. poslední frame timestamp.

Použité knihovny a technologie:

Základem je **FFmpeg**, které poskytuje bohatou sadu filtrů pro QC (viz blackdetect, silencedetect, freezedetect atd.) a dokáže zpracovat většinu formátů. FFmpeg lze volat z příkazové řádky skriptem, nebo použít např. wrapper `ffmpeg-python`. Pro získání metadat je užitečný `ffprobe` (výstup JSON) nebo knihovna **Pymediainfo**. Knihovna **scikit-video** může pomoci při implementaci vlastních algoritmů – např. má funkce na detekci scén (scenedet) ²², což by se dalo využít k odhalení tvrdých stříhů či analýze obrazu mezi stříhy. Případně lze využít **OpenCV** pro čtení snímků a pixelové operace (např. detekce černé = průměrná hodnota pixelů blízko 0).

AI a modely: Do budoucna (či pro některé advanced funkce) lze zapojit modely z HuggingFace hubu nebo trénované algoritmy:

- Pro **detekci rozmazání** či **blokování** by mohl posloužit model, který ohodnotí kvalitu snímku (existují modely pro no-reference video quality, často vyjadřují odhad MOS nebo indikátor ostrosti).
- **Detekce scénářů PSE (photosensitive epilepsy)** – hledání rychlých záblesků – by mohla být usnadněna AI (klasifikace záběrů jako nebezpečných). Batón např. umí PSE detektovat i korigovat ²³.
- **Klasifikace obsahu** (jazyk, detekce nevhodných scén) je spíše nad rámec technického QC, ale Interra Batón má třeba automatickou detekci jazyka zvuku ²³; to by šlo realizovat integrací speech-to-text modelu pro zjištění jazyka pro případnou budoucí kontrolu korektnosti titulků apod.

Pro MVP však **hlavní zaměření skriptů zůstává na technických kvalitativních kontrolách**: černota, zamrznutí, ticho, formátová správnost a hlasitost. Tyto testy jsou běžnou součástí profesionálních QC nástrojů – např. Telestream Vidchecker detekuje black/freeze frames, výpadky audio (silence), kontroluje gamut, hlasitost a mnoho dalších parametrů ²⁴ ²⁵. Stejně tak QScan identifikuje „black frames that shouldn't be there“, „frozen frames“, artefakty, atd. ²⁶ – námi navržené skripty pokrývají klíčovou podmnožinu těchto kontrol v rámci MVP.

3. Výběr výpočetního provozovatele (GPU, 4K/1h video)

Pro zpracování videí ve 4K rozlišení je vhodné nasadit GPU akceleraci – dekódování a analýza 60min 4K videa je výpočetně náročná. Zvažujeme několik variant cloud providerů s ohledem na **výkon, cenu a škálovatelnost**:

- **AWS (Amazon Web Services):** Poskytuje široký výběr GPU instance. Pro MVP by mohl stačit typ **g5.xlarge** (obsahuje GPU NVIDIA A10G s 24 GB VRAM) ²⁷. NVIDIA A10 nabízí solidní výkon pro

video dekódování i případné AI inference, při relativně dostupné ceně. On-demand cena A10 GPU se u hyperscalerů (AWS, Azure, GCP) pohybuje kolem \$1–2.5 za hodinu ²⁸, což pro občasné použití v MVP není prohibitivní. AWS výhodou je integrace s dalším stackem (S3, Lambda, atd.) a snadná budoucí škálovatelnost – lze přidávat další instance dle potřeby nebo využít autoscaling group. Do budoucna AWS umožní i přechod na výkonnější GPU (A100, H100) pokud by bylo třeba pro pokročilé AI analýzy, i když ty jsou mnohem dražší.

- **Google Cloud Platform (GCP):** Alternativně má GCP GPU ve službě Compute Engine. K dispozici jsou např. NVIDIA T4 (16 GB) nebo novější L4 (24 GB, Ada Lovelace architektura). T4 je cenově cca \$0.35/hod (jen GPU příplatek) ²⁹, celkově s VM kolem \$1+/hod, tedy srovnatelné s A10. GCP umožňuje snadné použití preemptible (Spot) instancí – podobně jako AWS Spot – což může zlevnit provoz až na ~25% ceny ²⁸, vhodné pro batch zpracování. Škálování na GCP je také bez problému (Managed Instance Groups, Kubernetes, etc.).
- **Microsoft Azure:** Nabízí GPU řady NC, ND atd. s NVIDIA (včetně A10, V100, A100). Cenově a koncepčně podobné AWS. Mohl by být zvažován, pokud by infrastrukturně dávalo smysl (např. využití Azure Functions + storage), ale v našem stacku preferujeme spíš AWS/GCP kvůli ostatním komponentám.
- **Specializovaní GPU provideři:** Pro úsporu nákladů lze zvážit služby jako **Lambda Labs Cloud**, **Vast.ai**, **RunPod**, apod. Ty pronajímají GPU (často GeForce/RTX karty nebo enterprise karty) výrazně levněji. Např. Lambda Cloud uvádí A10 (24GB) za cca \$0.75/h ²⁷. Tyto služby mohou nabídnout i lepší dostupnost více GPU při špičce a zúčtování po minutách. Nevýhodou je ale méně integrovaný ekosystém – museli bychom řešit networking a napojení na naše úložiště ručně. Pro MVP, kde běží jednotky úloh denně, by klidně stačila jediná on-demand instance u AWS; pokud ale budget tlačí, dalo by se experimentovat s preemptibleinstancemi (u AWS Spot nebo GCP Preemptible) – GPU za ~\$0.2–0.6/h ²⁸, s tím, že by úlohy musely umět obnovení při přerušení.

Doporučení pro MVP: Vzhledem k jednoduchosti integrace se nabízejí AWS služby. **Kombinace AWS S3 + EC2 GPU instance** (typ G5 s Nvidia A10G) dává smysl: video se nahraje do S3, poté se spustí EC2 instance (nebo použije již běžící) s workerem, který video stáhne a zkontroluje. Instance může běžet nepřetržitě (při nízkém zatížení by to byl určitý fixní náklad ~ \$1.5/h), nebo ji můžeme spouštět jen na úlohy (pomocí AWS Batch nebo orchestrací z AWS Lambda, případně využít kontejnery na Fargate s GPU podporou). Pro začátek lze minimalizovat náklady třeba využitím **spot instance** – A10G na spot vyjde kolem ~\$0.5/h ³⁰, a stačí implementovat, aby se úloha případně restartovala, pokud by spot byl zrušen.

Budoucí škálovatelnost: Jakmile vzroste počet uživatelů a požadavků, architektura umožní horizontální škálování – lze paralelně spouštět více GPU instancí/kontejnerů. Fronta úloh zajistí, že se úlohy zpracují, jakmile je volný výpočetní slot. Lze zavést **automatické škálování** – např. v klidu 0 instancí, při nové úloze se spustí instance, při návalu více front-end požadavků se rozjede více instancí (do maxima dle rozpočtu). Vzhledem k cílové skupině (freelanceři, malá studia) se neočekává masivní zatížení zpočátku, spíše jednotky paralelních úloh, což navržená GPU VM zvládne.

Pokud by do budoucna přibyly AI-náročné funkce (např. hluboká analýza obrazu), je možné nasadit výkonnější GPU (A100, H100) jen na ty konkrétní úlohy. Výhodou cloutu je, že nejsme omezeni hardwarem – můžeme si dynamičtěji volit instanci podle potřeby (např. 4K SDR video kontrola stačí A10G, ale 8K HDR s AI by vyžadovalo A100 – architektura to umožní za cenu vyšších nákladů).

Závěr: Zvolíme pravděpodobně AWS kvůli ekosystému a spustíme MVP na jedné GPU instanci (Nvidia A10 nebo ekvivalent). Cena v MVP bude relativně nízká (řádově jednotky dolarů za hodinu výpočtu ²⁸), což při několika videích denně je únosné). Důležité je, že řešení škáluje – v budoucnu lze více instancí a případně migrace mezi cloud providery (když využijeme kontejnerizaci a abstrakci fronty, můžeme worker nasadit i jinde).

4. Návrh struktury aplikace (frontend, backend, DB, storage)

Navrhujeme moderní cloudovou architekturu, která je modulární a škálovatelná:

- **Frontend:** Bude postaven na frameworku **Next.js** (React) a hostován na Vercel. Next.js umožní rychlý vývoj frontendu a Vercel zajistí globální nasazení s minimálními starostmi o infrastrukturu. Aplikace bude v angličtině (s ohledem na mezinárodní cílovku) a nabídne responsivní web UI pro nahrávání videa, sledování stavu analýzy a zobrazení výsledného reportu. Výhodou Next.js na Vercelu je i možnost využití ISR (incremental static regeneration) pro některé statické stránky (např. dokumentace, landing page) a velmi jednoduchá integrace s backend API.
- **Backend / API vrstva:** Pro orchestraci a lehkou logiku poslouží serverless funkce. Vercel nabízí **Serverless Functions** (Node.js) pro API routes – vhodné pro rychlé akce (auth, iniciace uploadu atd.). Náročnější operace však nemohou běžet přímo na Vercelu (limit ~10s běhu). Proto implementujeme hybridní přístup:
 - Krátké akce (získání upload URL, dotaz na stav jobu, autentizace uživatele) budou obsluženy serverless funkcemi na Vercelu (popř. lze využít Cloudflare Workers pro něco, ale není nutné).
 - Dlouho běžící akce (vlastní QC analýza videa) budou delegovány na **background worker** mimo Vercel. To zajistíme pomocí **fronty (queue)** – např. po uploadu souboru zavolá frontend API „prosím zpracovat video X“, tato API funkce zapíše záznam do databáze (stav „pending“) a vloží zprávu do fronty (např. Redis-based queue, nebo AWS SQS). Worker, který běží na GPU instanci (viz předchozí sekce), zprávu z fronty přijme a spustí analýzu. Po dokončení aktualizuje stav v DB a uloží výsledky.
 - Tím oddělíme webový frontend od výpočetní části – což zvyšuje spolehlivost. Kdybychom hostovali i backend na Vercelu, narazíme na limit doby běhu; takto využijeme Vercel jen na rychlou API orchestraci, a robustní batch job poběží jinde.
- **Databázová vrstva:** Zvolíme **NeonDB (Neon)** – cloudová PostgreSQL databáze. Neon je navržen pro serverless prostředí, podporuje okamžité škálování, pauzování (scale-to-zero) a funguje skvěle s platformami jako Vercel ³¹. Využijeme ji pro:
 - Ukládání uživatelských účtů, přihlašovacích údajů (hesla hashovaná, případně OAuth data).
 - Evidenci nahraných videí a jejich stavů (tabulka Jobs: ID videa, uživatel, stav = pending/running/done/failed, čas nahrání, čas dokončení apod.).
 - Uložení výsledků QC testů: buď normalizovaně (tabulky issues, např. typ='black_frame', start_time, duration atd.) nebo jednoduše JSON sloupec s celým reportem. Případně kombinace – pro rychlé filtry a přehledy můžeme ukládat některé parametry zvlášť (např. průměrná hlasitost, celkový bitrate).
 - Nastavení uživatelských profilů, tarify (kdo má Starter/Pro/Agency), logy o spotřebě (např. kolik videí zpracoval tento měsíc).

Neon DB výhodou je, že se umí uspávat, když se nepoužívá (šetří náklady), a probouzí se rychle na dotaz. Též podporuje **branching** – do budoucna by šlo třeba pro každý zákaznický tenant mít

samostatnou větev DB pro testování změn, ale to je spíše benefit pro vývoj/testy. Integrace s Vercel je zdokumentovaná ³¹, takže nastavení spojení je přímočaré.

- **Úložiště souborů (Storage):** Pro ukládání nahraných videí a generovaných reportů použijeme **S3-kompatibilní objektové úložiště**. Dvě varianty:
 - **AWS S3:** Klasické řešení, vysoká spolehlivost. Snadno se integruje s AWS službami (např. worker na EC2 si stáhne soubor přímo ze S3). Můžeme využít i **pre-signed URLs** pro přímý upload z frontendu – Vercel funkce vygeneruje URL, frontend tam PUTne soubor. S3 má ale poplatky za výstup (egress); pokud by uživatelé často stahovali reporty velké objemem dat, může to něco stát, ale QC reporty bývají malé. 1h video může mít stovky MB až jednotky GB, jejich egress ke zpracování je interní (v rámci AWS regionu zanedbatelné).
 - **Bunny.net Storage:** Moderní varianta – objektové úložiště s velmi nízkou cenou (\$0.01/GB) a vestavěnou CDN pro distribuci obsahu koncovým uživatelům ³. Hodí se, pokud bychom chtěli rychle servírovat náhledy videí nebo samotné video soubory uživatelům po analýze. Bunny nemá S3 API, ale poskytuje vlastní API a FTP – integrace by byla mírně složitější než S3. Nicméně pro uživatelské stahování reportů či náhledů by Bunny ušetřil náklady na datový přenos oproti AWS. Možné řešení je hybrid: S3 pro komunikaci mezi backend komponentami (upload & worker), a Bunny.net pro publikování výstupů ke klientům (mohli bychom z AWS po vygenerování reportu nahrát kopii na Bunny CDN, pokud by to dávalo smysl).

Pro MVP pravděpodobně zvolíme **S3 (nebo kompatibilní službu jako Wasabi)** kvůli jednoduchosti. Architekturu ale navrhнемe abstraktně: přístup k úložišti přes rozhraní, které lze zaměnit. Tím pádem do budoucna můžeme snadno přejít na jiného providera nebo doplnit CDN před S3.

- **Zpracovatelský worker:** Jak zmíněno, poběží na GPU instanci (např. AWS EC2). Ten bude implementován nejspíše jako Docker kontejner (obsahující potřebné nástroje: FFmpeg, Python skripty atd.). Tohoto workera můžeme spouštět buď ručně (cron/na vyžádání) nebo pod správou orchestrátoru:
- Jednoduchá varianta: trvale běžící EC2 instance, na ní Python skript, který periodicky kontroluje frontu (např. AWS SQS, nebo even. používá WebSocket k Neon pro notifikaci nového jobu). Když najde úlohu, zpracuje ji. Pokud není práce, idle (případně se může sám vypnout po nečinnosti – to by pak musela nějaká lambda budit novou instanci na nový job).
- Pokročilejší: využít **AWS Batch** nebo **AWS Step Functions** – nahrajeme definici jobu (docker image) a Batch sám alokuje kontejner s GPU, provede a vypne. Toto by ušetřilo nutnost stále běžící instance a umožní efektivní škálování na více paralelních jobů (Batch fronta). Z pohledu MVP však nasazení AWS Batch přináší jistou komplexitu navíc, takže je to spíše úvaha do budoucna.
- **Monitoring a logování:** Vercel poskytuje základní logging pro frontend a lambdy. Pro workera bychom nasadili třeba **CloudWatch** (na AWS) pro sběr logů z FFmpeg a skriptů, a monitor metrik (CPU, GPU util.). Neon DB má vlastní monitoring (počet spojení, dotazy). Důležité je mít přehled, kolik videí se zpracovalo, průměrné časy, případné chyby (např. formát videa nepodporován kodekem, spadne ffmpeg – nutno logovat a uvést do reportu jako „error“).

Shrnutí architektury:

- **Klientská část:** Next.js + Vercel (UI, formuláře, zobrazení výsledků).
- **API vrstva:** serverless funkce (autentizace, spouštění jobů, dotazy na výsledky).
- **Databáze:** Neon Postgres (uživatelé, metadata jobů, výsledky testů).
- **Úložiště:** S3/Bunny (videa + reporty).

- **Asynchronní backend:** GPU worker procesující frontu (na AWS EC2 nebo Batch).
- **Integrace:** Vercel s Neon (snadné škálování DB) ³¹, AWS komponenty se S3 a GPU.

Tato modulární struktura zajišťuje, že každý komponent může škálovat nezávisle – např. při náporu web návštěv se automaticky škálují Vercel instance frontendu, při náporu analýz se spustí více workerů, DB Neon se přizpůsobí zátěži atd. Navíc je to velmi flexibilní: v případě potřeby lze backend část (worker + queue) provozovat i u jiného poskytovatele nebo v on-premise prostředí pro enterprise klienty, zatímco frontend SaaS může běžet v cloudu.

(Pozn.: Alternativní zvažovaná tech: využití serverless workflow orchestration – např. Temporal.io nebo orchestrátor typu AWS Step Functions ³² – ale pro MVP by to bylo zbytečně robustní. Jednodušší je výše popsaný manuální model s frontou a workerem.)

5. Cenový model (tarify Starter / Pro / Agency)

Navrheme **víceúrovňové SaaS předplatné** s bezplatnou zkušební verzí. Inspirujeme se běžnými SaaS praktikami a konkurenčními nástroji:

- **Free Trial:** Noví uživatelé získají možnost vyzkoušet VQCC po omezenou dobu nebo s omezeným počtem analýz zdarma (např. 14 dní trial, nebo 5 videí zdarma). Cílem je odstranit bariéru vyzkoušení a ukázat hodnotu nástroje. Např. QScan také nabízí **Start Free Trial** ³³. Free trial nebude vyžadovat kartu, ale videa mohou mít omezení (např. max 10 minutové video, SD rozlišení) nebo limit funkcí.
- **Starter (např. \$29–\$49 měsíčně):** Tento tarif cílí na jednotlivé **freelancery a tvůrce** s menší potřebou. Omezení by mohlo být:
 - Zpracování **1 videa najednou** (žádný paralelní processing) ³⁴.
 - Omezená kapacita za měsíc, např. **5 hodin videa měsíčně** ke kontrole v ceně, s možností dokoupit extra.
 - Dostupné základní QC kontroly a šablony (např. přednastavení pro YouTube/Vimeo standardy) ³⁵.
- **Neomezený počet uživatelů** nemusí být potřeba (u jednotlivce irrelevantní). Přesto by jeden účet Starter mohl být omezen na 1 uživatele.
- Podpora komunitní / e-mail s odpovědí do 2 dnů.

Cenu ladíme podle trhu: QScan One stojí \$39/měsíc ³⁶ s podobným zaměřením (single file analysis, základní šablony). Tudiž Starter by mohl být kolem ~\$39 pro obdobný value.

- **Pro (např. ~\$99 měsíčně):** Pro malé a střední **produkční týmy** nebo studia. Obsahuje:
 - Možnost paralelně analyzovat **více videí současně** (např. 2–4 souběžné úlohy) ³⁷.
 - Vyšší měsíční limit zpracování, např. **20 hodin videa** nebo více.
 - Rozšířenou sadu kontrol, včetně pokročilejších testů: např. detekce PSE (blikání pro epilepsii), pokud ji integrujeme, a detailnější reporty.
 - **Více výstupních formátů reportu** (PDF, JSON, XML) pro snadné předání klientům ³⁸.
 - Možnost **vlastních šablon kontroly** – uživatel si může nastavit vlastní thresholdy a pravidla (např. jinou hlasitost normu, povolit/nepovolit prokládání apod.).
 - Zákaznická podpora s vyšší prioritou (odpověď do 1 pracovního dne).

QScan Pro je nabízen za \$99/měsíc ³⁷ a nabízí až 4 souběžné analýzy, PSE detekci a pokročilé reporty. Náš **Pro tarif** by se tedy pohyboval v této cenové relaci a funkcionality, aby byl konkurenčeschopný a atraktivní pro profesionály.

- **Agency (např. \$199+ měsíčně):** Pro větší firmy, postprodukční domy, **televize** atd. Obsahuje:
- **Neomezenou nebo velmi vysokou kvótu** analýz za měsíc (např. 100 hodin videa nebo neomezeně) ³⁹.
- **Více souběžných úloh** – např. 10 paralelních, neomezeně uživatelů v rámci jednoho týmu.
- Plnou sadu funkcí včetně všech pokročilých testů (HDR analýza, verifikace Dolby Vision metadata, porovnávání dvou verzí videa side-by-side apod. – viz budoucí funkce).
- Integrace do workflow: API přístup pro automatizaci (možnost napojit do CI/CD pipeline, media asset management systémů).
- **White-label reporty** – možnost vložit logo firmy do QC reportu, generovat vlastní závěry atd.
- **Prioritní podpora**, dedikovaný account manager, případně SLA na zpracování (garantovaná doba).
- Možná on-premise nebo privátní instance pro velké klienty (za příplatek).

Cena kolem \$199/měsíc vychází z QScan Max ³⁹, které za \$199 nabízí neomezené souběžné analýzy a plnou integraci. U velmi velkých zákazníků by mohl být i vyšší enterprise plán (např. **Agency+** s individuální cenou), ale pro začátek bychom komunikovali max. ~\$199–299 jako veřejnou cenu Agency plánu, s tím že vyšší objemy řešíme individuálně.

- **Extra poplatky:** Je vhodné nastavit politiky pro překročení limitů – např. **overage fee**: pokud zákazník na Starter plánu potřebuje jednorázově zpracovat více hodin videa, umožníme to za poplatek \$X za hodinu navíc. To pokryje variabilní náklady na GPU čas. Stejně tak API přístup by mohl být za počty volání atp., ale v první fázi se zaměříme na jednoduchý model „férove nastavené limity a možnost přejít na vyšší plán“.

• **Srovnání s konkurencí:**

- **QScan** (EditShare) cílí podobně široce: nabídka od \$39 do \$199 ⁴⁰ ³⁹. Náš model kopíruje tuto osvědčenou strukturu (nízkoprahový vstup pro freelancery, střední pro studia, vyšší pro enterprise).
- **Telestream Vidchecker** prodává licence jednorázově za tisíce \$ (8k–75k) ⁴¹, což je mimo dosah malých zákazníků – to potvrzuje, že náš SaaS model s měsíčními platbami v řádu desítek až stovek \$ cílí na neobsazený segment (dostupnost pro malé subjekty).
- **Interra Baton** má **spotřební model** (platba za zpracované minuty) ⁴². Ten nabízí větší flexibilitu pro obří objemy, ale pro menší uživatele je méně čitelný. My zvolíme raději pevné balíčky (snadno pochopitelné) s tím, že pro Agency zákazníky můžeme interně také přepočítávat, zda se jim nevyplatí spíše consumption model – případně nabídnout obě varianty později.

Z pohledu běžných SaaS praktik jsou 3 plány (Basic/Pro/Enterprise) standardem a cenové hladiny by měly odrážet vnímanou hodnotu. Důležité je jasné komunikovat rozdíly mezi plány (nejlépe tabulkou funkcí na webu). Například: - **Starter:** 1 současná analýza, základní testy, komunitní podpora. - **Pro:** 4 současná analýza, všechny testy + PSE, PDF reporty, standardní podpora. - **Agency:** neomezeně analýz, pokročilé funkce (HDR, API), priority podpora.

Takový model je validní i podle konkurenčních nabídek – QScan One vs Pro vs Max se liší v podobných parametrech (souběžnost, šablony, PSE, HDR atd.) ⁴³ ⁴⁴. Cenově tedy zapadneme do rozpětí, které již trh akceptuje, a můžeme se odlišit případně délkou trialu či drobnostmi (např. **Free tier** by mohla být

místo time-limited trialu – třeba 1 video do 2 minut zdarma měsíčně – ale to musíme opatrně, aby nekanibalizovalo placené plány).

Platební model: Měsíční předplatné (s možností slevy při roční platbě). Pro freelancery je důležité mít možnost kdykoli zrušit. Pro Agency klidně kontrakty. Budeme využívat standardní billing (Stripe integration například).

6. Analýza konkurence (video QC nástroje) a návrh konkurenčních výhod VQCC

Na trhu existují zavedené nástroje pro automatický video Quality Control, většinou zaměřené na broadcast a velká studia:

- **Telestream Vidchecker (a Telestream Aurora):** Tradiční on-premise software pro broadcasters. Má velmi širokou škálu testů: od formátové compliance (ATSC/DPP atd.) přes video parametry (bitrate, GOP struktura) až po obrazové a zvukové vady. Detekuje například:
 - **RGB gamut porušení, luma/chroma level mimo meze** 45 46 .
 - **Kvalitativní defekty:** blokovité artefakty, analogové dropouts, digitální výpadky, rušivé pruhy v obraze 47 .
 - **Black frames, freeze frames** – výslovně zmiňují detekci černých sekvencí a zamrznutí 18 .
 - **Color bars detekce** (zda video obsahuje barevné pruhy na začátku), **letterboxing/pillarboxing** detekce 47 .
 - **Dead pixel detection, PSE flashing** (nebezpečné záblesky pro epileptiky) 48 18 .
 - **Audio kontroly:** od počtu kanálů, pořadí kanálů, přes úroveň šumu, detekci fázových posunů mezi kanály, až po kompletní měření hlasitosti (Momentary, Short-term, Integrated Loudness, dialnorm) a identifikaci ticha, špiček, lupanců 25 .
 - **Compliance templates:** Obsahuje přednastavené profily pro Netflix, Amazon Prime, iTunes, EBU R128, DPP atd. 49 , které definují kritéria pro pass/fail.
 - Unikátní vlastností Vidcheckera je i možnost **automatické opravy** některých chyb 50 17 – umí např. korigovat barevný gamut, úroveň černé, opravit audio loudness na požadované LKFS, ba dokonce vložit/vymout color bar sekvenci či opravit časový kód. To je patentovaná funkce, která jde nad rámec čisté detekce.

Omezení: Vidchecker je drahý (min. jednotky tisíc \$ za licenci) 41 , cílí na enterprise. Rozhraní není cloudové SaaS, obvykle běží na Windows serveru. Pro freelancery není dostupný. Telestream ale nabízí i cloud QC službu **Telestream Cloud Qualify**, která funguje jako API v cloudu pro QC (použito v AWS demo 51), ale cenově a dostupností to míří taky spíš na B2B trh.

- **Interra Systems BATON:** Další enterprise QC systém, známý spolehlivostí. Nově se profiluje jako „AI-powered QC“ 52 :
 - Využívá strojové učení k „pochopení kontextu“ – např. obsahově chytřejší analýza, adaptivní učení se podle dat 53 .
 - Umožňuje hybridní workflow: kombinaci automatických a ručních kontrol (operátor validuje sporné věci).
 - **Funkce:** Podobné spektrum testů jako Vidchecker (compliance, audio, video). Navíc zdůrazňuje **blockiness detection, intelligent loudness monitoring, frame-accurate timecode checks, language detection** z audio 54 23 .
 - Mají i modul na kontrolu správnosti titulků, integritu IMF/DCP package atd.
 - Baton je dostupný on-prem i jako cloud (má licenční model i pay-per-use). Consumption pricing (platba za minuty videa) dává flexibilitu 42 , ale opět – typický klient je televize/OTT platforma.

- **Silné stránky konkurence:** vysoce škálovatelné (cluster deployment), osvědčené ve velkých provozech, široká integrace (podpora API, pluginy do MAM atd.).
- **Slabiny vzhledem k nám:** velmi vysoká cena pro malého uživatele, komplexnost (pro nastavení často třeba odborník), není to samoobslužné SaaS pro jednotlivce.
- **EditShare QScan:** Toto je významný konkurent pro segment, kam míříme. QScan je cloudová aplikace dostupná ve více edicích, včetně levného plánu pro content creatory [55](#).
- QScan provádí **125+ automatických kontrol** [55](#) [56](#). Kromě standardních (formát, metadata, black/freeze) zahrnují i **HDR analýzu** (podpora Dolby Vision, HDR10), **color space compliance** (CIE diagramy) [57](#), **dead pixel detection**, **blurriness** a další kvality [58](#).
- Nabízí moderní webové rozhraní s timeline náhledem, vizuálními grafy, možností porovnání dvou verzí videa **side-by-side** [57](#), a export výsledků do markerů pro Adobe Premiere či Davinci Resolve [21](#) (to je velká výhoda, šetří čas editorům).
- QScan má **GPU akceleraci** zpracování, podporu S3 úložiště, integrace s NLE, a tzv. **Visual QC** režim (QScan Verify aplikace) pro ruční kontrolu sporných míst s profi náhledem a waveform monitorem [59](#).
- **Cenově** QScan cílí i na jednotlivce: plán One za \$39 (single job) [36](#), Pro \$99 (4 jobs, PSE, PDF) [60](#), Max \$199 (unlimited, HDR) [39](#). Tedy pokrývá freelancy až enterprise lehčím stylem.
- **Silné stránky:** Dostupnost (cena), intuitivní UI, široká funkcionality, cloud based. V podstatě QScan je vzor úspěšného SaaS v této doméně.
- **Slabé stránky:** Jako produkt od EditShare (dříve Grass Valley) je zaměřen i na broadcast, ale možná nemá tak silnou AI komponentu jako Baton (neuváděj learning, spíše fixní algoritmy). Může postrádat některé extrémně specifické kontroly, co má Baton/Vidchecker (např. testy specifických MXF standardů nebo korekce chyb).
- Z pohledu nového konkurenta (VQCC) není mnoho slabin – budeme muset konkurovat hlavně **UX a inovacemi**, protože cenově i pokrytím je QScan dost napřed.
- **Další hráči:**
 - **Venera Pulsar:** Další populární automatizovaný QC systém. Známý rychlostí – umí analyzovat HD video až 6x rychleji než realtime [61](#). Cílí spíš opět na profi segment (OTT, broadcast). Funkcemi obdobný (pokrývá HDR, IMF, atd.).
 - **Drastic VideoQC:** Tento nástroj (od Drastic Technologies) je spíše **softwarový přehrávač s QC funkcemi**. Je to Tiered produkt od ~\$99 až po \$1495 [62](#) [63](#). Nabízí real-time monitoring videa až 8K, HDR, profi nástroje (vektorskop, waveform) [64](#) [65](#), ale zřejmě méně automatických výstupů – spíše pomůcka pro ruční QC. Může sloužit spíš jako doplněk pro ruční ověření výstupu, ne automatizovaná cloud služba.
 - **Open source QC Tools:** Existuje například projekt **QC Tools** (BAVC), volně dostupný nástroj hlavně pro archiváře na kontrolu video archivů (detekce dropoutů, overscan atd.). Není to ale uživatelsky přívětivý SaaS a zaměřuje se na analogové digitzované video. Pro naše zákazníky nepříliš relevantní.
 - **FFmpeg + vlastní skripty in-house:** Někteří freelanceri si dnes pomáhají ručně ffmpegem, ale to vyžaduje technickou znalost a je to pracné. Naše služba vlastně cílí i na tyto uživatele, kteří by radši klikací řešení než studovat ffmpeg parametry.

Konkurenční výhody VQCC:

Abychom uspěli, musíme nabídnout něco **navíc nebo lépe** oproti výše zmíněným: 1. **Zacílení na uživatelský komfort a jednoduchost:** VQCC bude navrženo tak, aby i netechnický uživatel (např. kameraman, stříhač) snadno porozuměl reportu. Přehledné rozhraní s vizualizací problémů na časové

ose, vysvětleními co která chyba znamená a jak ji případně opravit. Mnohé konkurenční reporty jsou textové nebo příliš technické. My můžeme přidat třeba náhledové obrázky z místa chyby (frame grab z místa kde obraz zčernal apod.), graf hlasitosti s vyznačením překročení limitů, atd. 2. **Cena a dostupnost:** Oproti Vidchecker/Baton budeme řádově levnější a dostupní online on-demand. Tady se vyrovnáme QScan – můžeme třeba nabídnout o něco delší trial, nebo free tier, což QScan explicitně nemá (má trial ale ne free ongoing). To by mohla být výhoda pro komunitu tvůrců obsahu s malým rozpočtem. 3. **Inovace s AI:** Zapojíme umělou inteligenci tam, kde to dává smysl pro zvýšení hodnoty. Např.: - *AI asistent v reportu:* Představme si, že kromě faktického výpisu chyb by report obsahoval sekci „Doporučení“, kde by AI model vygeneroval lidsky čitelné shrnutí: "Video má příliš nízkou průměrnou hlasitost (-30 LUFS), doporučujeme zvýšit úroveň zvuku o ~7 dB, nebo aplikovat broadcast limiter." nebo "Nalezené černé snímky v čase 00:10:05 pravděpodobně odpovídají střihu - ověřte, zda tam nemá být obsah.". Tato přidaná interpretace by klientům pomohla problém nejen odhalit, ale i pochopit a vyřešit. - *AI analýza obsahu:* Mohli bychom nabídnout volitelný modul, který využije třeba Computer Vision k detekci log nebo nevhodného obsahu (to už míří za rámec čistě technické QC do content QC, ale některé agentury to mohou chtít – např. zkontovalovat, že video neobsahuje konkurenční logo, atd.). - *Lepší detekce obrazu pomocí ML:* Jak zmíněno, AI může zlepšit detekci artefaktů (bloky, šum) spolehlivěji než statické prahy. Pokud VQCC v budoucnu integruje trénované modely, můžeme identifikovat problémy, které starší nástroje možná nedetektují nebo generují hodně false-positives. 4. **Integrace a automatizace pro malé týmy:** Enterprise nástroje se integrují do velkých MAM systémů, ale my můžeme nabídnout pluginy nebo jednoduché API klíče pro populární kreativní nástroje. Např. **Adobe Premiere Pro panel:** Uživatel by z Premiere mohl odeslat aktuální sekvenci k QC na VQCC a zpět by dostal markerový soubor s komentáři. QScan to dělá (Export to Premiere/Resolve)²¹, takže abychom konkurowali, měli bychom totéž v plánu. Dále třeba integrace na cloud storage typu Dropbox/Google Drive: automatický QC souboru po nahrání do určité složky. 5. **Funkce porovnání videí (version comparison):** Mnozí konkurenti to zmiňují jen okrajově – QScan umí side-by-side vizuální porovnání dvou souborů⁵⁷, což je skvělé pro kontrolu, zda po kompresi nedošlo ke zhoršení kvality. VQCC by mohlo mít specializovaný „diff report“: uživatel nahraje zdrojový master a komprimovaný export, a nástroj je nejen jednotlivě vyhodnotí, ale i srovná – např. spočítá PSNR, SSIM, případně VMAF⁶⁶ mezi nimi, a upozorní, které části se zhoršily nejvíce. Toto by byla výrazná konkurenční výhoda pro postprodukční studia, co potřebují ověřit kvalitu různých výstupů. 6. **Rychlosť a škálovatelnost v cloudu:** Pulsar vyzdvihuje rychlosť 6× realtime⁶¹ – pokud bychom dokázali s GPU dosáhnout např. 2–3× realtime na 4K (možné paralelním dekódováním a multi-thread analýzou), můžeme se také chlubit výkonem. Díky škálování v cloudu by VQCC nikdy nemělo fronty – pokud má agentura 10 videí najednou, na Agency plánu je zpracujeme souběžně (dostanou 10 GPU kontejnerů paralelně). Tohle enterprise nástroje on-prem limitují licencemi (např. Vidchecker Base max 4 jobs⁶⁷, více nutno dokoupit licencí). U nás by vyšší plán prostě pustil více instancí – flexibilnější. 7. **Uživatelská podpora a komunita:** Jako nový hráč můžeme získat komunitu freelancerů např. aktivním fórem, znalostní bází, tutoriály „why QC matters“ (eduкаční marketing). Konkurence jako Baton/Vidchecker tohle u malých klientů nedělá.

Resumé: Naše konkurenční výhody budou **dostupnost (cenová i snadnost použití)**, **moderní AI-enhanced funkce, pohodlné UI a specializace na potřeby menších produkcí** (např. integrace do jejich workflow s Adobe, možnost rychle sdílet online report klientovi – třeba přes tajný link, což nahradí posílání PDF e-mailem). Cílem je nabídnout 80% funkcí high-end QC za zlomek ceny a s lepší použitelností. Funkce jako automatické opravy (co má Vidchecker) zatím nemáme, ale případně v budoucnu můžeme nabídnout poloaautomatické: např. „odstraň černé snímky na začátku“ jako volbu generování nové verze videa – to by byla zajímavost do budoucna.

7. Postup vývoje (roadmap) včetně timeline, priorit a využití AI

Abychom efektivně dostali VQCC na trh, rozdělíme vývoj do fází:

Fáze 1: MVP (Minimum Viable Product)

Cíle MVP: Umožnit jednomu uživateli nahrát video (max 4K/1h) a získat automatický QC report s hlavními nálezy. Podporovat základní technické kontroly (černé snímky, zamrznutí, ticho, hlasitost, formát) a poskytnout jednoduché UI. Tato verze bude anglicky, a nabídne registraci uživatele, free trial a případně jeden placený plán (Starter) jako proof-of-concept pro platby.

Časový odhad: cca 3 měsíce vývoje malým týmem.

Konkrétní milníky MVP: - **M1: Základ backendu a QC skriptů (týden 1-4):** Nastavení FFmpeg a vývoj skriptů pro jednotlivé testy. Otestování offline na vzorcích videí (viz test data níže). Vybudování jednoduchého worker skriptu co vezme soubor a vrátí JSON s výsledky. - **M2: Cloud infrastruktura (týden 2-6):** Nastavení NeonDB, implementace datového modelu (uživatelé, jobs). Nasazení úložiště (S3 bucket), napsání Vercel API route pro generování upload URL. Zavedení fronty a rozběh první EC2 GPU instance s workerem; ověření, že z fronty umí vzít job a pošle zpět výsledek (možno zatím jen do logu/ DB). - **M3: Frontend aplikace (týden 4-8):** Vytvoření UI v Next.js – stránky: registrace/přihlášení, stránka pro upload videa + formulář (název jobu, soubor), stránka se seznamem uživatelských jobů a jejich stavem, stránka detailu reportu. Implementace volání API (upload, polling stavu). Zajištění, že UI reaguje na stav „Processing...“ a po dokončení zobrazí data. - **M4: Generování reportu (týden 6-10):** Navržení šablony QC reportu – HTML stránka, která z JSON výsledků vygeneruje seznam zjištěných chyb s časovými údaji, plus základní metadata videa. Stylování reportu (přehledné zvýraznění závažných chyb). Možnost exportu do PDF (např. pomocí serverless knihovny jako Puppeteer na Vercelu, nebo klientsky přes print to PDF). - **M5: Autentizace a účet (týden 8-10):** Zprovoznění uživatelských účtů (registrace s e-mailem, heslem). Možné použít hotové Auth lib (Auth0, NextAuth). Nastavení jednoduchého rate-limitu, aby trial uživatel nemohl zneužít (např. max 1 job v jeden okamžik). - **M6: Testování MVP (týden 10-12):** Otestování end-to-end s různými videi: krátké, dlouhé, s chybami. Ladění chybek, zajištění stability worker procesu (co když ffmpeg spadne na neznámém formátu – musí to zachytit a označit job jako failed s chybovou zprávou). - **Launch MVP (týden ~12):** Otevřená beta pro první uživatele (můžeme pozvat pár známých z oboru na test). Získání zpětné vazby.

Minimální testovací data pro vývoj: Připravíme sadu videí simulující různé situace: - Video se vloženým **černým intervalom** (např. 10 sekund černé na začátku nebo mezi scénami). - Video se segmentem, kde je statický obraz (**freeze**) – lze vyrobit tak, že část videa kopíruje stejný snímek (např. 5s). - Audio soubor s **tichem** – např. videoklip, kde na 3s ztišíme zvuk zcela. - Video s **překročenou hlasitostí** – např. normalizovaný na -10 LUFS, abychom viděli alarm na R128. - Video v netypickém formátu – např. 720x480 anamorf (abychom testli detekci aspect ratio), nebo video s prokládáním (interlaced) aby skript zjistil „interlaced = true“. - **Referenční bezchybný soubor** – abychom ověřili, že skripty nedávají false pozitivy (např. minutový klip bez problémů, očekáváme zelený report). - Pár reálných kratších videí od beta testerů (požádáme je, aby poskytli něco s typickými chybami, třeba špatně ořezané video s pillarboxy, apod.).

Tyto soubory budeme iterativně používat k ladění QC detekcí.

Fáze 2: Verze 1.0 (Production-ready)

Po MVP beta testu (řekněme měsíc zkoušení) přejdeme k verzi 1.0, která by byla připravena pro ostrý provoz a platící uživatele. Časově cca **+3 další měsíce** po MVP.

Prioritní vylepšení ve verzi 1.0:

- **Stabilizace a škálovatelnost:** Dopoladění auto-scaling mechanizmů. Nasazení více workerů (např. 2–3 GPU instance v poolu) – potřebné pro Pro/Agency plány. Zavedeme monitoring, aby se nové instance zapnuly při frontě čekajících úloh.
- **Plány a platby:** Implementace Stripe pro subscription management. Zavedení tarifů Starter/Pro/Agency v systému, omezení funkcí podle plánu (např. kontrola před spuštěním jobu: má user volnou kvótu?). Také portál pro uživatele – upgrade/downgrade plánu, vidí svou spotřebu.
- **Pokročilé QC testy:** Přidání některých funkcí, které se nestihly do MVP:
- Detekce **dead pixelů** (pokud najdeme vhodný algoritmus nebo open dataset na trénink AI, případně to necháme na později).
- **HDR metadata kontrola** – pokud zjistíme, že video je HDR (PQ křivka, Dolby Vision), ověřit přítomnost/platnost dynamických metadat, MaxFALL/MaxCLL hodnoty atd. (Telestream tohle dělá ⁶⁸ a QScan také – pro profi je to důležité).
- **Dropouty a šum v obraze** – třeba jednoduchá detekce prudkého lokálního ztmavnutí obrazu (analog drop) nebo datového výpadku (čtverečky). To jde hůře automatizovat, ale možná najdeme heuristiku.
- **Titulky** – pokud vstup obsahuje titulkovou stopu, ověřit, že není prázdná, nebo že sedí formát (to možná není v MVP vůbec, ale v 1.0 bychom mohli aspoň zobrazit „obsahuje/neobsahuje titulky“).
- **Report vylepšení:** Doplnit grafické prvky do reportu (např. malý waveform pro audio s vyznačením silent parts, tabulku s celkovými loudness hodnotami, histogram jasů pro black frames apod.). Umožnit **stažení reportu jako PDF** jedním klikem.
- **UI/UX vylepšení:** Na základě beta feedbacku upravit ergonomii. Třeba přidat drag&drop upload, indikátor průběhu uploadu, možnost pojmenovat job, filtr/sort v seznamu jobů (užitečné pro Pro/Agency s desítkami videí).
- **Mezinárodní aspekty:** I když app je anglicky, můžeme připravit lokalizační mechanismus (do budoucí čeština, španělština pokud bude požádáno). Zkontrolujeme podporu různých časových formátů v reportu, atd.
- **Zabezpečení a spolehlivost:** Projít aplikaci z hlediska bezpečnosti (úložiště – expirování URL, neoprávněný přístup; API rate limiting; zabezpečení dat v DB). Zálohování DB (Neon má vestavěné). Nastavení alarmů (pokud job fail rate > X, upozornit tým).
- **Využití AI (interně):** Zde můžeme nasadit prvky jako zmiňovaný AI asistent pro popis chyb v reportu. Mohli bychom využít model typu GPT (přes API OpenAI) – v čase 2025 by to šlo. Feeda by byl JSON s chybami a GPT by vygenerovalo odstavec shrnutí. To by ale mělo být volitelné (abychom neplýtvali tokeny pro každý job, možná jen pro placené plány). Každopádně integrace AI tady spíše v roli **kopiloty** při interpretaci dat.
- **Testy a QA:** Před 1.0 nasadíme sadu unit/integration testů (např. simulace uploadu, kontrola že job projde pipeline a do DB se zapíší správné hodnoty). Otestujeme škálování (spustit 5 paralelních jobů a sledovat, jestli to stihá).

Verze 1.0 by měla být připravena k produkčnímu nasazení pro první ostré zákazníky. Plánujeme nasazení ~6 měsíců od startu projektu, což zahrnuje MVP + doladění.

Fáze 3: Růst a další funkce (post-1.0 roadmap)

Jakmile je základ stabilní, budeme rozšiřovat funkčnost podle zpětné vazby a konkurenční nutnosti:

- **Porovnávání verzí videa:** Implementace funkce „Compare“ – uživatel nahraje dvě videa (např. před a po kompresi, nebo před a po barevné korekci) a nástroj je analyzuje a porovná. Výstupem by byl speciální report: vedle běžných QC pro každé video i sekce s metrikami kvality mezi nimi (PSNR, SSIM, VMAF graf po čase). To může přinést vizuální graf, kde jsou úseky s největší ztrátou

kvality. Tato funkce by mohla být součástí vyšších plánů (Pro/Agency). Časově to odhadujeme na +2 měsíce vývoje (FFmpeg umí SSIM/PSNR snadno, VMAF s knihovnou libvmaf také).

- **Real-time integrace do editace:** Vyvinout plugin/panel pro Adobe Premiere (a případně DaVinci Resolve). To si vyžádá použití jejich API – u Premiere lze vytvořit panel v HTML/JS, který bude komunikovat s naším backendem. Funkčně: editor klikne „Analyze Sequence“, plugin vyexportuje timeline do jednoho souboru (třeba ProRes) a zavolá náš API. Po dokončení stáhne EDL/XML s markery, které zobrazí v timeline. Tohle je spíše pro Agency klienty, timeframe další ~3 měsíce vývoje spolu s testováním v praxi.
- **Rozšíření AI analýz:** Zapojit pokročilejší modely:
- Detekce **obsahových anomalií** – např. použít ML na detekci rozostřených záběrů (out-of-focus), které technicky nemusí být „chyba“ ale klienti to chtějí vědět. Nebo detekce scén, kde je špatná expozice, atd. Tady lze experimentovat s dostupnými CV modely.
- **Automatický objektivní scoring kvality** – něco jako odhad MOS skóre bez reference, či zda video „vypadá profesionálně“ (to už hraničí se subjektivním hodnocením, nicméně některé nástroje marketingově zmiňují VMAF a MOS 69 70, tak bychom mohli taky ukázat nějaké skóre kvality).
- **Mobilní a další platformy:** Aplikace by se mohla dočkat zjednodušené mobilní verze (např. notifikace a náhled výsledků na mobilu), či dokonce jednoduché mobilní appky. Není priorita, spíš nice-to-have.
- **Více formátů vstupu:** Rozšířit podporu na živé streamy (kontrola streaming playlistů HLS/DASH), nebo analýza audio-only souborů (pro podcasty QC). To by otevřelo nové trhy, ale vyžaduje další specializaci (např. HLS segmety kontrolovat mezery, manifest etc., což Telestream dělá 71).
- **Scénář multi-uživatelských účtů:** Pro Agency nabídnete možnost, aby jedna firma měla více loginů pod jedním účtem, s rolemi (admin, viewer). To vyžaduje doplnit user management (další 1 měsíc práce).
- **On-premise deployment / self-hosted:** Velcí klienti můžou chtít běžet QC na vlastní infrastruktuře. VQCC by mohl později nabídnout variantu „appliance“ – tj. docker balíček, který si zákazník spustí u sebe (a my poskytneme licence). To by byl spíš dlouhodobý cíl (až budeme mít stabilní verzi a zájem z enterprise).
- **Certifikace standardů:** Pokud budeme chtít proniknout i do broadcast sféry, museli bychom implementovat specifické standardy (DPP, IMF, etc.) a získat certifikace, že je splňujeme. To je velký úkol, možná horizont 2+ let, a ne nutný pro primární trh freelancers.

Timeline zhruba:

- *MVP beta:* Q2 2025 (funkční základ).
- *V1.0 launch:* Q3 2025 (placené plány, spolehlivost).
- *Rozšíření (v1.1 – v1.5):* Q4 2025 až Q2 2026 – postupně přidávané funkce (porovnání videí, pluginy, AI vylepšení).
- *Dál:* podle poptávky a trendů (možná integrace s generativní AI pro automatické opravy menších vad, atd.).

Delegování na AI (v procesu vývoje): Kromě integrace AI do produktu můžeme AI využít i při samotném vývoji: - Použití GitHub Copilota při programování (urychlí psaní kódu). - Využití ChatGPT pro generování části dokumentace, testovacích scénářů, nebo i k vytvoření trénovacích datasetů (např. vygenerovat skriptem spoustu variací videosouborů s určitými chybami). - Pro QA by šlo nasadit AI testy – např. nechat ChatGPT analyzovat JSON výstupy a kontrolovat konzistenci, či navrhovat další edge-cases k testování. - V marketingu nasadíme AI pro tvorbu obsahu (blog post „Proč je automatický QC důležitý“ apod.).

Závěrem, vývoj budeme řídit agilně, s prioritou vždy dodat nejdřív to, co přináší největší hodnotu cílovým uživatelům (freelancerům a malým studiím). Budeme průběžně sbírat feedback od beta testerů

z řad stříhačů a produkčních, aby VQCC skutečně řešilo jejich problémy efektivně. S tímto plánem máme šanci vybudovat konkurenčeschopný SaaS nástroj pro video QC, který vyplní mezeru na trhu mezi drahými enterprise systémy a amatérským ručním kontrolováním.

Zdrojové odkazy:

- Telestream Vidchecker – seznam dostupných kontrol a automatických oprav [24](#) [45](#)
 - FFmpeg filtry pro detekci černých a zamrzlých snímků [6](#) [11](#)
 - QScan (EditShare) – funkce a pricing plán konkurence [26](#) [40](#)
 - Interra Baton – AI přístup a spotřební model [54](#) [42](#)
 - Ceny cloud GPU (Nvidia A10) – srovnání hyperscaler vs specializovaní provideři [28](#)
 - Bunny.net Storage – cenová výhoda oproti S3 [3](#)
-

1 2 4 5 32 51 Automatic media ingest and quality control with Telestream and Ateliere on AWS | AWS for M&E Blog

<https://aws.amazon.com/blogs/media/automatic-media-ingest-and-quality-control-with-telestream-and-ateliere-on-aws/>

3 Honest Bunny.net Review: Performance, Pricing & Features

<https://affinco.com/bunny-net-review/>

6 7 8 audio - Is there a way to detect black on FFMPEG video files - Stack Overflow

<https://stackoverflow.com/questions/58971875/is-there-a-way-to-detect-black-on-ffmpeg-video-files>

9 10 71 Black Frame Detection

<https://api.encoding.com/reference/features-black-detection>

11 12 13 Way to detect frozen video with ffmpeg? - Super User

<https://superuser.com/questions/1313070/way-to-detect-frozen-video-with-ffmpeg>

14 How to input noise values in dBFS for ffmpeg silence detect

<https://sound.stackexchange.com/questions/50418/how-to-input-noise-values-in-dbfs-for-ffmpeg-silence-detect>

15 16 hondrou thoughts: FFMPEG detecting black frames and generating keystamps

<http://hondrouthoughts.blogspot.com/2016/09/ffmpeg-detecting-black-frames-and.html>

17 18 24 25 45 46 47 48 49 50 68 Vidchecker automated quality control and correction | Product | Technical Specifications

<https://www.telestream.net/vidchecker/specs.htm>

19 20 23 26 34 35 36 37 38 39 40 41 42 43 44 52 53 54 55 56 57 60 61 62 63 64 65 66 67

69 70 5 Best Video QC Software to Ensure Top-Quality Video Content

<https://vidpros.com/video-qc-software/>

21 33 58 59 Automate Media QC Effortlessly | Streamline Workflows with QScan

<https://qscan.io/>

22 Measurement Tools — scikit-video 1.1.11 documentation

<https://www.scikit-video.org/stable/measure.html>

27 Instances | Lambda

<https://lambda.ai/instances>

28 30 gmicloud.ai

<https://www.gmicloud.ai/blog/how-much-do-gpu-cloud-platforms-cost-for-ai-startups-in-2025>

29 GPU pricing | Google Cloud

<https://cloud.google.com/compute/gpus-pricing>

31 Neon for Vercel

<https://vercel.com/marketplace/neon>