

Cory<>Viral Vacations

- [Product Vision](#)
- [Tech Implementation](#)
- [Proposal](#)
- [Pitch deck](#)

Product Vision

PRODUCT VISION: VIRAL VACATION

The Operating System for Autonomous Local Tourism Marketing

Version: 1.0 | **Strategic Lead:** Cory Peterson

1. EXECUTIVE SUMMARY

Viral Vacation is the first AI-driven ecosystem designed to solve the "Content Famine" in local tourism. While Real Estate Agents and Property Managers are cash-rich, they are time-poor and struggle to translate the "30A Lifestyle" into consistent social media marketing.

We are building the "**Nexus Engine**"—an autonomous system that separates creative thinking from technical doing. By leveraging a proprietary "Data Moat" of terabytes of hyper-local, 4K footage, we turn static data triggers (Reviews, Weather, Menus) into broadcast-quality commercials instantly. We do not just build a video tool; we are digitizing the local word-of-mouth economy.

2. THE CORE PHILOSOPHY: "AUTONOMOUS AUTHENTICITY"

The current AI market (Sora, Runway) is racing to generate synthetic video. This is the wrong approach for tourism. Travelers do not book a vacation based on a hallucinated beach; they book based on reality.

- **The Viral Vacation Way:** We use AI to **assemble** reality, not invent it.
- **The Mechanism:** Our system acts as a "Just-in-Time" film editor. It retrieves real, geotagged assets from our library and assembles them into a narrative structure based on real-time data inputs.
- **The Promise:** A business owner never edits a video. They only approve it.

3. THE "DATA MOAT" (UNFAIR ADVANTAGE)

Our defensibility lies in our proprietary asset library.

- **The Library:** Terabytes of 4K drone, lifestyle, and underwater footage of the 30A/Destin corridor, all legally cleared and geotagged.
- **The Intelligence:** We use Computer Vision to index every second of this footage. We don't just know a file is named Beach_01.mp4; the system knows it contains [Sunset, Couple, White Sand, Wine Glass, 7:00 PM].
- **Why It Wins:** A competitor can build the same software, but without our local data library, their output will look generic. We possess the "Digital DNA" of the destination.

4. THE ECOSYSTEM: "SYNDICATED CROSS-POLLINATION"

Viral Vacation breaks the marketing silos between local businesses. We view the destination as a living organism where every business supports the other.

- **The Logic:** A Vacation Rental company needs to sell the "experience," not just the bed.
- **The Workflow:** When a guest leaves a review for a Condo saying, "We walked to Austin's for oysters," the Nexus Engine detects the entity "Austin's."
- **The Output:** The generated video features the Condo's balcony view **AND** a high-energy clip of oysters from Austin's Sports Bar.
- **The Result:** The Condo gets better lifestyle content; Austin's gets free advertising. This creates a network effect where businesses are incentivized to join the platform to be part of the "Content Supply Chain."

5. THE "NEXUS ENGINE" ARCHITECTURE

To deliver TV-quality results at scale, we are moving away from prototype tools (N8N/Drive) to a **Hybrid Cloud Factory**.

A. The Ingestion Layer (The Eyes)

- **Technology:** AWS S3 + Computer Vision (Banana.dev/Gemini).
- **Function:** As content is uploaded, AI models analyze the footage frame-by-frame. It tags objects, emotions, lighting, and "Vibe." This turns a folder of dumb files into a searchable "Smart Database."

B. The Brain (The Logic)

- **Technology:** OpenAI (GPT-4o) + Vector Database (Pinecone).
- **Function:** The Brain holds the "Brand Voice" of every client.
 - Input: A 5-Star Google Review.
 - Process: The Brain writes a script using the client's specific tone (e.g., Luxury vs. Family Fun) and queries the database for the perfect matching shots.
 - Fallback: If a specific shot is missing (e.g., "Kid eating ice cream"), the Brain triggers a generative image model (Nano Banana Pro) to create the asset instantly, ensuring the video flow never breaks.

C. The Factory (The Assembly)

- **Technology:** Shotstack API + Python/Librosa.
- **Function:** This is the rendering engine.
 - It pulls the 4K assets from S3.
 - It analyzes the music track to identify "Downbeats."
 - It cuts the video **exactly on the beat**, applies transitions, and overlays HTML-based text (the Review) in the client's brand font.

D. The Distribution (The Growth)

- **Technology:** Ayrshare API + Google Business Profile API.
- **Function:**
 - **Social:** One-click publishing to Instagram Reels, TikTok, and YouTube Shorts.
 - **Search:** Auto-posting the video as an "Update" to the client's Google Maps listing, injecting SEO metadata to dominate local search results.

6. THE USER JOURNEY (A DAY IN THE LIFE)

1. **09:00 AM:** Property Manager receives a notification: "New 5-Star Review detected."
2. **09:01 AM:** They click the link. The Nexus Engine has already generated 3 video variations (Luxury, Fast-Paced, Story-Mode).
3. **09:02 AM:** They tap "Approve Option 1."
4. **09:03 AM:** The video is live on Instagram, TikTok, and Google Maps. The Manager goes back to work, having done 2 hours of marketing work in 3 minutes.

7. REVENUE STRATEGY & ROADMAP

- **Phase 1 (Summer 2026 - MVP):** Launch the "Review-to-Revenue" Engine. Secure high-ticket subscriptions (

1k-1k-1k-

3k/mo) from Real Estate/Property Managers.

-

Tech Implemenation

1. HIGH-LEVEL ARCHITECTURE: THE "HYBRID CLOUD"

We are utilizing a **Hybrid approach** where **N8N** acts as the API Gateway/Orchestrator, but all heavy computation and storage are offloaded to **AWS** and specialized APIs.

The 4-Layer Stack

Layer	Component	Technology Stack	Function
1. Interaction	Frontend / UI	React / Next.js	User Dashboard, Approval Interface, Upload Portal.
2. Orchestration	The Controller	N8N (Self-Hosted)	Traffic control, Webhook listening (Reviews), API routing.
3. Intelligence	The Brain (RAG)	OpenAI (GPT-4o) + Pinecone	Scriptwriting, Semantic Search, Brand Voice Memory.
4. Factory	The Engine	AWS Lambda + S3 + Shotstack	File storage, Computer Vision, Beat-Sync, Rendering.

2. CORE COMPONENT DEEP DIVE

A. The "Data Vault" (Storage & Ingestion)

Design Pattern: Asynchronous S3 Trigger

We do not pass video files through the application server. We use a **Direct-to-Cloud** pattern.

1. **Architecture:** Amazon S3 (Simple Storage Service) configured with Transfer Acceleration.
2. **Security:** All uploads use **Presigned URLs**. The frontend requests a temporary "Key" to upload directly to AWS. This keeps our servers lightweight.
3. **Event Loop:**
 - Event: File lands in s3://raw-uploads.
 - Trigger: AWS Lambda function fires automatically.
 - Action: Calls **Banana.dev / Google Gemini Vision** to analyze the footage.
 - Output: Metadata tags [Sunset, Oysters, Luxury, 4K] are sent to the Vector Database.

B. The "Nexus Brain" (RAG - Retrieval Augmented Generation)

Design Pattern: Vector Semantic Search

Standard AI (ChatGPT) hallucinates because it lacks context. We implement **RAG** to ground the AI in reality.

1. **Vector Database (Pinecone):** We convert every asset tag and every client brand document into "Embeddings" (numerical representations of meaning).
2. **The Query Logic:**
 - Input: "Make a video for a luxury foodie couple."

- **Vector Search:** The system queries Pinecone for vectors matching "Luxury" + "Food" + "Couples" + "High Resolution."
 - **Retrieval:** It pulls the exact File IDs for the best footage from the Data Vault.
3. **Script Generation:** GPT-4o receives the Context (the selected file descriptions) and generates the JSON script.

C. The "Video Factory" (Rendering Engine)

Design Pattern: Serverless Media Processing

This is the core engineering challenge. We use **Shotstack** driven by code.

1. **Audio Sync Engine (Python):**

- We run a Python microservice using the librosa library.
- It analyzes the selected music track to extract **Onsets** (Beats) and **Energy Levels**.
- It mathematically calculates the cut points (e.g., Cut at 3.4s, 5.8s, 9.2s) to ensure the video is rhythmically satisfying.

2. **The Assembly (JSON):**

- The Logic Layer compiles a JSON payload containing: S3 URLs for video, HTML/CSS for text overlays (Reviews), and the Cut Points.
- This is sent to the Shotstack API for cloud rendering.

D. The Fallback Protocol (Nano Banana / Gemini)

Design Pattern: Conditional Logic

1. **Confidence Check:** If the Vector Search returns a match score < 0.3 (Missing Footage), the system flags a "Gap."
2. **Generative Pipeline:**

- **Step 1:** Call **Nano Banana Pro (Gemini 3 Image)** to generate a photorealistic static image.
 - **Step 2:** Pipe the image to **Runway/Luma API** (Image-to-Video) to add 2 seconds of ambient motion (steam, light leaks, water movement).
 - **Step 3:** Inject the resulting URL into the Shotstack timeline.
-

3. DATA FLOW DIAGRAMS

Pipeline 1: The "Smart Ingestion" (Building the Moat)

“ Goal: Turn raw files into searchable data without human tagging.
User Upload -> AWS S3 Bucket -> (Trigger) -> AWS Lambda (Computer Vision) -
> Extract Tags -> Store in Pinecone (Vector DB)

Pipeline 2: The "Review-to-Revenue" Engine

“ Goal: Autonomous Video Creation.

1. **Trigger:** Google Maps Webhook (New 5-Star Review).
2. **Analysis:** N8N parses text -> GPT-4o extracts Keywords & Sentiment.
3. **Retrieval:** Pinecone searches for matching Video Assets + Cross-Pollination Partner Assets.

4. **Fallback:** If asset missing -> Trigger Generative AI Pipeline.
5. **Logic:** Python calculates Music Beat Sync.
6. **Assembly:** N8N constructs Shotstack JSON.
7. **Render:** Shotstack processes 4K video -> Saves to s3://rendered-output.
8. **Notify:** User receives SMS/Email with Approval Link.

Pipeline 3: Cross-Pollination Logic (Relational Filtering)

“ Goal: Promoting the right partner.
We query the **PostgreSQL** relational database before the Vector search.
Query: Partner Business
WHERE distance < 2 miles
AND category = 'Restaurant'
AND tier != 'Budget' (If Client is Luxury)
RETURN Top 3 Partner IDs.

4. DATABASE SCHEMA DESIGN

We utilize a **Dual-Database Strategy**:

A. Relational DB (PostgreSQL) - Structured Data

- **Table: Users** (Auth, Credits, Role)
- **Table: Businesses** (Name, Location, Brand_Tier, Google_Place_ID)
- **Table: Assets** (S3_URL, File_Type, Owner_ID, License_Status)
- **Table: Relations** (Which businesses are allowed to cross-pollinate)

B. Vector DB (Pinecone) - Unstructured Intelligence

- **Index: Brand_Voice** (Embeddings of previous successful captions/scripts)
 - **Index: Asset_Library** (Embeddings of visual content descriptions)
-

5. SECURITY & INFRASTRUCTURE

Multi-Tenancy (Data Isolation)

To address your concern about confidentiality, we implement **Row Level Security (RLS)** in the database.

- **Rule:** A client can only query assets where Owner_ID == Self OR Asset_Type == Public_B_Roll.
- **Logic:** The API enforces this filter on every request. Even if the AI wants to use a competitor's clip, the database layer will block access before it happens.

Scalability

- **Statelessness:** Our N8N and Python workers are stateless.
 - **Load Handling:** If 500 reviews come in at once, they are added to a **Message Queue (AWS SQS)**. The rendering engine processes them in order (FIFO), ensuring the server never crashes, no matter the volume.
-

Here is the architecture diagram: [Tech architecture](#)

Summary

This architecture moves Viral Vacation from a "Tool" to a "**Platform.**" By leveraging **S3 for heavy storage**, **Vector DBs for intelligence**, and **Shotstack for rendering**, we create a system that is robust, secure, and infinitely scalable.

Chandan & Tushar

Proposal

1. The Vision: "Pure Authenticity" at Scale

Most AI video tools (like Sora or Runway) try to invent reality. They create fake people and fake beaches.

Viral Vacation is different.

Your competitive advantage—your "Moat"—is the terabytes of **Real, Geotagged, 4K Local Footage** you possess. The vision is to build an autonomous engine that:

1. **Ingests** the real world (Reviews, Weather, Menus).
2. **Matches** it with authentic local footage (The "Data Moat").
3. **Assembles** cinematic commercials in seconds.
4. **Cross-Pollinates** the local economy, breaking the silos between a Condo, a Restaurant, and an Activity Provider.

The Goal: To separate the "Creative Thinking" from the "Technical Doing," allowing a property manager to scale their marketing infinitely without adding a single human to their payroll.

2. Destination & ICP: Who are we building for?

To secure Angel Investment, the MVP cannot just be "cool tech." It must be a revenue engine. We are optimizing the build for two specific user types:

A. The High-Ticket Payer (The "Hearth & Home" Group)

- **Who:** Vacation Property Managers & Luxury Real Estate Agents.
- **Why they buy:** They are not selling four walls; they are selling the 30A Lifestyle. They have the budget (

1k-1k-1k-

3k/mo) but zero time to edit reels.

- **The MVP Promise:** "A 5-star review comes in; a marketing video goes out. You don't lift a finger."

B. The Content Feeder (The "Flavor & Flair" Group)

- **Who:** Restaurants, Retail, Activity Providers.
- **Why they join:** Visibility. They provide the "Oyster Shots" and "Sunset Cruises" that the Real Estate agents need to sell the lifestyle.
- **The MVP Promise:** "Free exposure. Your brand gets featured in the marketing of the most expensive homes in the area."

3. Current State of Thought (The Baseline)

Based on the workflows and spreadsheets you provided, the initial technical assumptions were:

- **Orchestration:** Using **N8N** for all logic and file handling.
 - **Storage:** Using **Google Drive** to host the video assets.
 - **Logic:** Relying on **Direct Text Mentions** in reviews (e.g., "We went to Austin's") to trigger cross-pollination.
 - **Generative AI:** Using generic tools to "add lifestyle elements" (like fake kids playing) into photos.
-

4. Missing Gaps & Critical Risks

To protect your investment, we need to be honest about the risks in the current plan. If we built it exactly as originally described, here is where it would break:

1. The "Heavy File" Crash (The N8N Limit)

N8N is incredible for logic, but it is not designed to handle 4K video files (which can be 500MB+). If we try to pass a video file through an N8N node, the automation will time out and crash. We need a "Pass-Through" architecture.

2. The "Google Drive" Latency

Google Drive is a personal storage tool, not a cloud video server. It has strict rate limits. If 50 clients try to render a video at once, Google will block the connection. We must move to professional infrastructure.

3. The "Missing Shot" Problem

What happens if the AI script calls for "A close-up of a kid eating chocolate ice cream" but your library doesn't have that specific shot? Currently, the automation would fail. We need a fallback protocol.

4. The "Logic" Trap

Statistically, less than 5% of reviews mention a partner business by name. If we rely solely on "Text Mentions," the cross-pollination feature—your biggest selling point—will rarely fire. We need a smarter "Persona-Based" logic.

5. Phasing Misalignment

The original plan listed "Social Publishing" in **Phase 4**. You cannot launch a product called "Viral Vacation" in Phase 1 if the user has to download the video and post it manually. The loop must be closed immediately.

5. Our Improved Planning: The "Hybrid Cloud Factory"

We are shifting the architecture from a "Low-Code Prototype" to a "**Professional Cloud Application**."

A. The New Architecture (The Workflow)

We will use N8N as the "Manager" (traffic control), but we will use Python code and specialized APIs for the "Heavy Lifting."

- **The Vault (AWS S3):** We will use Amazon S3 instead of Google Drive. It is infinite, secure, and instant. We will use "**Presigned URLs**" so that N8N only handles lightweight text links, never the heavy video files.
- **The Brain (RAG):** We will build your "Poppi.ai" replacement using **OpenAI (GPT-4o)** connected to a **Pinecone Vector Database**. This database will store your "Brand Voice" and "Tourism Trends" so the AI writes scripts that sound like a local expert.
- **The Factory (Shotstack + Python):** We will use **Shotstack** (as you suggested) for the rendering. But we will add a **Python/Librosa** layer to analyze the music beats. This ensures every video cut happens exactly on the beat, which is the difference between a "Slideshow" and a "Viral Reel."

B. The "Nano Banana" Fallback Protocol

You identified **Nano Banana Pro** (Google Gemini 3 Image Model). This is a brilliant find. We will use this to solve the "Missing Shot" problem.

- **Scenario:** Script calls for "Oysters," but no footage exists.
- **Action:** System generates a photorealistic image via Nano Banana Pro.
- **The Magic:** We pass this image through a "Motion Layer" (using Shotstack's Ken Burns effect or a tool like Runway API) to make the steam rise or the light move.
- **Result:** The video flow never breaks, and it looks cinematic.

C. The "Google Search" Domination

Instead of building a complex Google Ads Manager (which is high-risk for an MVP), we will automate **Google Business Profile Updates**.

- **The Strategy:** Every video we generate is auto-posted to the client's Google Maps listing.
 - **The SEO Hack:** We will inject **Geotags** and **Keyword Metadata** into the file headers before uploading. This signals to Google's algorithm that your content is hyper-relevant, boosting your clients' search rankings organically.
-

6. The Phased Roadmap

We have restructured your list to prioritize **Revenue** and **Technical Stability**. We pulled "Phase 2" items into "Phase 1" because they are mission-critical.

Phase 1: The "Investor-Ready" MVP (Weeks 1-8)

Goal: A working "Review-to-Video" engine that posts to social media.

- **1. The Data Vault & Smart Ingestion:**
 - Setup AWS S3 Storage.
 - **Computer Vision Layer:** As you upload content, our AI (Banana.dev) auto-tags it: [Sunset, Balcony, Wine, Oysters]. This eliminates the need for manual tagging.
- **2. The Nexus Engine (The Logic):**
 - **Input:** System listens for a Google Review.
 - **Scripting:** GPT-4o generates the JSON Shot List based on the review sentiment.
- **3. The Factory (Video Generation):**
 - **Shotstack Integration:** We connect the JSON output to the Shotstack rendering engine.
 - **Fallback Logic:** Implement the Nano Banana image generator for missing assets.
- **4. Publishing (Ayrshare Integration):**
 - Strategic Shift: We are pulling "Social Publishing" from Phase 4 to Phase 1.
 - **Outcome:** The dashboard will allow "One-Click Scheduling" to IG, TikTok, and YouTube Shorts immediately.

Phase 2: The Ecosystem Activation (Weeks 9-14)

Goal: Activating Cross-Pollination & Growth.

- **1. Smart Cross-Pollination Logic:**
 - Move beyond "Text Matches." Implement "**Persona Matching.**" (e.g., A "Luxury" Condo review automatically pulls content from "Fine Dining" partners, even if they weren't mentioned).
 - **Brand Tiers:** Logic to ensure Luxury brands don't accidentally promote budget activities.
- **2. The "Scraper" Suite:**
 - Deploy the N8N workflows to scrape local events and prices to feed the "Brain" with real-time context.
- **3. Google Business Profile SEO:**
 - Automate the "Update" posts to Google Maps with injected SEO metadata.

Phase 3: Scale & Ads (Month 4+)

Goal: High-Volume Growth.

- **1. Contributor App:** A portal for influencers to upload B-Roll and get credited (Faceless content only for legal safety).
- **2. Ad-Boosting Bridge:** Allow users to "Boost" their best organic videos via Meta Ads API.

7. Investment & Operational Estimates

We believe in radical transparency. Below is the breakdown of the **One-Time Development Investment** to build the platform, and the **Estimated Monthly Infrastructure Costs** (OpEx) that you will pay directly to providers to keep the lights on.

A. Development Investment (Fixed Cost)

We have estimated the effort at approximately 700-900 **engineering hours** to deliver Phase 1 (The MVP) and Phase 2 (Ecosystem Activation).

Instead of an hourly billing model which carries risk for you, we propose a **Fixed Project Fee** to guarantee delivery within budget.

Phase	Deliverables	Investment
Phase 1: The MVP	The Video Factory: AWS S3 Data Vault, Smart Ingestion (Auto-Tagging), Nexus Engine (Scripting), Shotstack Integration, and Social Publishing (Ayrshare). Includes QA & Testing.	\$8,500

Phase 2: Ecosystem	The Growth Engine: Smart Cross-Pollination Logic, "Scraper" Suite for local data, and Google Business Profile SEO Automation.	\$9,500
Total Build Cost	End-to-End Product Launch	\$18,000

Payment Schedule (Milestone-Based):

- **30% Deposit:** To kick off architecture & setup
- **40% Milestone:** Upon completion of the Phase
- **30% Final:** Upon Completion

B. Estimated Monthly Operational Costs (OpEx)

These are the rough costs for the 3rd-party tools powering the engine. You will pay these directly (your credit card on the accounts). These costs scale with your usage—meaning you only pay more when you have more paying customers.

Estimates based on ~50 Active Clients generating ~500 Videos/Month:

Provider	Purpose	Estimated Cost
AWS (S3 + Lambda)	Hosting the app, database, and storing terabytes of video.	~\$80 - \$120 / mo
Shotstack	The Video Rendering API. Charges per minute of video rendered.	~\$150 - \$250 / mo (Scales with revenue)
OpenAI (GPT-4o)	The "Brain" writing scripts & analyzing data.	~\$50 - \$100 / mo
Banana.dev / Replicate	Computer Vision for Auto-tagging & Nano Banana Image Gen.	~\$50 / mo
Ayrshare	The Social Media API (Posts to IG, TikTok, YouTube).	~\$150 / mo (Premium Plan)

Pinecone	Vector Database for the "Brand Voice" memory.	~\$70 / mo
TOTAL	Rough Monthly "Burn Rate"	~\$550 - \$750 / mo

Why this is a healthy model:
If you charge your 50 clients an average of

500/month**, your revenue is **500/month**, your revenue is **500/month**, your revenue is **

25,000/month.
Your tech cost is only ~\$750/month. This is an extremely high-margin SaaS model (97% Gross Margin).

8. The "No-Bug" Guarantee

Cory, you mentioned trauma from previous developers leaving you with a buggy product. Because we are building this on **Shotstack and AWS** (Enterprise Grade) rather than cobbling it together with N8N (Prototype Grade), we are confident in the stability.

Our Commitment:
We include a **30-Day Hyper-Care Period** after launch. Any bugs, glitches, or logic errors found in the first month are fixed at **\$0 cost** to you. We don't walk away until the engine is humming.

Pitch deck

[Pitch deck](#)

<https://cory-pitch-deck.edgeone.dev/>