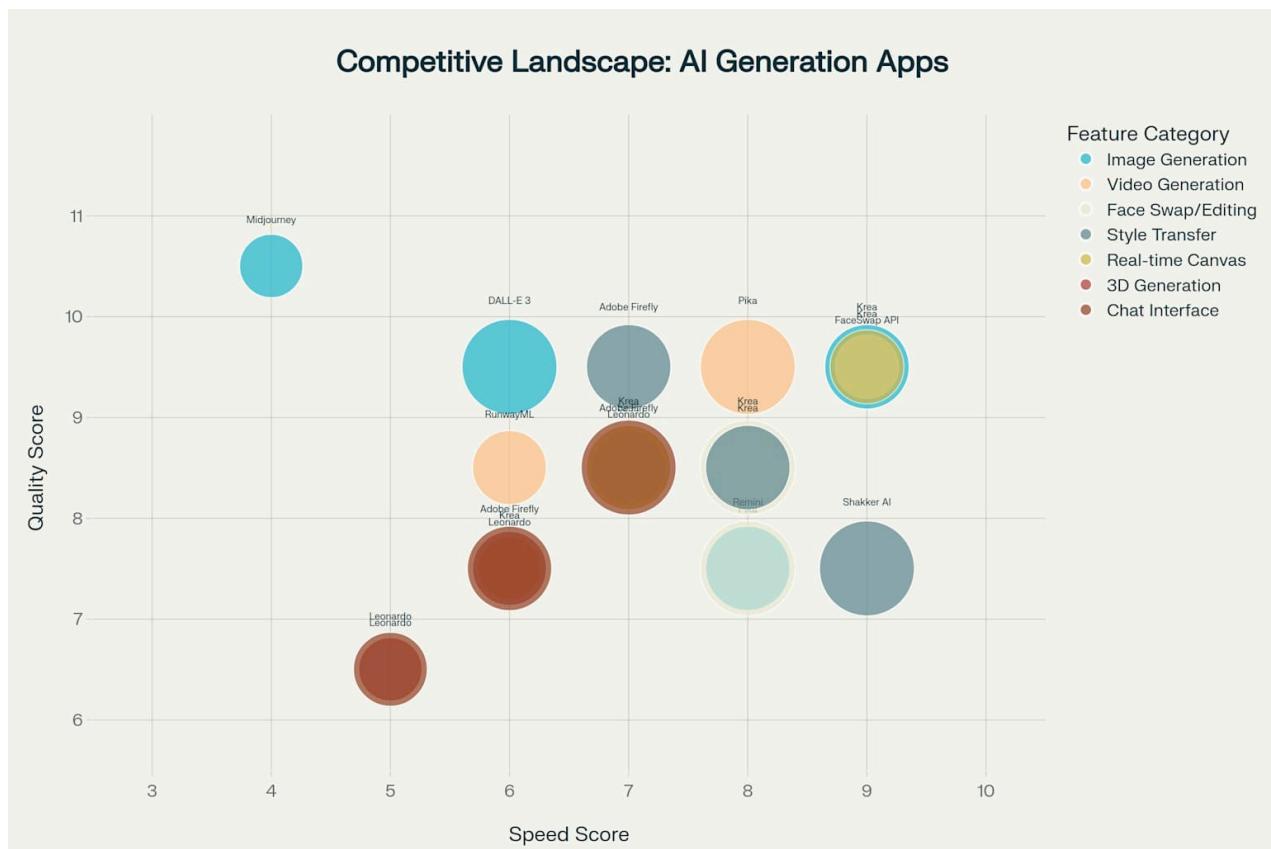


Pic Arcade: Comprehensive Product Requirements Document and Technical Implementation Strategy

Pic Arcade represents a revolutionary approach to AI-powered image and video manipulation, positioning itself as the world's most intelligent visual content creation platform. The app targets a diverse user base spanning influencers, content creators, and general consumers through a sophisticated agentic backend system that orchestrates multiple AI models while maintaining a deceptively simple user interface. This comprehensive analysis examines the product strategy, technical architecture, and market positioning required to establish Pic Arcade as the dominant force in AI-powered visual content creation.

Market Opportunity and Competitive Landscape

The AI photo and video app market is experiencing unprecedented growth, with over 60 new applications launched quarterly in 2023, representing a 160% increase from the previous year^[1]. This surge reflects the massive consumer appetite for AI-powered visual content creation tools, particularly among social media users and content creators seeking to enhance their online presence.



Competitive positioning of AI generation platforms showing quality vs speed performance

The competitive landscape reveals significant opportunities for differentiation through quality-speed optimization^[2] [3] [4]. Current market leaders like Krea excel in real-time generation capabilities with quality scores of 9/10 and speed scores of 9/10, while platforms like Midjourney dominate artistic quality (10/10) but suffer from slower processing times (4/10)^[2] [3]. This analysis identifies a market gap for applications that can deliver both high-quality outputs and fast processing speeds while maintaining superior ease of use.

Target User Segments and Market Sizing

The addressable market for Pic Arcade encompasses multiple high-value user segments with distinct needs and revenue potential^[1] [5]. Content creators represent the largest segment at 25.6 million users, demonstrating strong willingness to pay (\$25/month) and high technical proficiency^[1]. General consumers form the broadest base with 45.2 million users, though with lower price sensitivity at \$10/month^[1].



Market analysis showing user segments by market size and willingness to pay, with tech savviness indicated by bubble size

Micro-influencers emerge as a particularly attractive segment, combining substantial market size (15.2 million users) with focused use cases around style transfer and face swapping^[1] [5]. E-commerce sellers represent the highest-value segment with willingness to pay \$40/month for product photography enhancement, despite their smaller market size of 18.7 million users^[1] [6].

Core Product Features and Capabilities

Intelligent Prompt Processing System

Pic Arcade's primary differentiator lies in its sophisticated natural language processing capabilities that interpret complex, conversational prompts like "put me in that dress Taylor Swift wore at the awards" [7] [8]. The system employs advanced entity recognition to identify celebrities, fashion items, and style elements, then maps these to specific AI model capabilities for optimal processing [7] [8].

The prompt interface incorporates auto-complete suggestions based on trending styles and user behavior patterns, reducing friction while maintaining creative flexibility [7] [8]. Pre-built prompt templates serve common use cases while enabling power users to craft custom requests with granular control over output parameters [7] [8].

Advanced AI-Powered Transformations

Face Swapping and Manipulation

The platform integrates multiple face swapping APIs to ensure high-quality, natural-looking results [9] [10] [11]. Primary implementation utilizes custom fine-tuned models via Segmind API (\$25/1M images) with PiAPI FaceSwap service as secondary backup [9] [11]. Advanced facial blending algorithms maintain realistic skin tones, lighting consistency, and expression matching [10] [11].

Style Transfer and Virtual Try-On

Style transfer capabilities leverage Shakker AI's deep learning models to apply artistic styles while preserving content integrity [12] [13]. The system supports everything from Van Gogh-inspired brushstrokes to modern digital art effects, with real-time processing for immediate user feedback [12] [13]. Virtual try-on functionality enables clothing, accessories, and hairstyle experimentation through Google's generative AI technology, supporting diverse body types and skin tones [6].

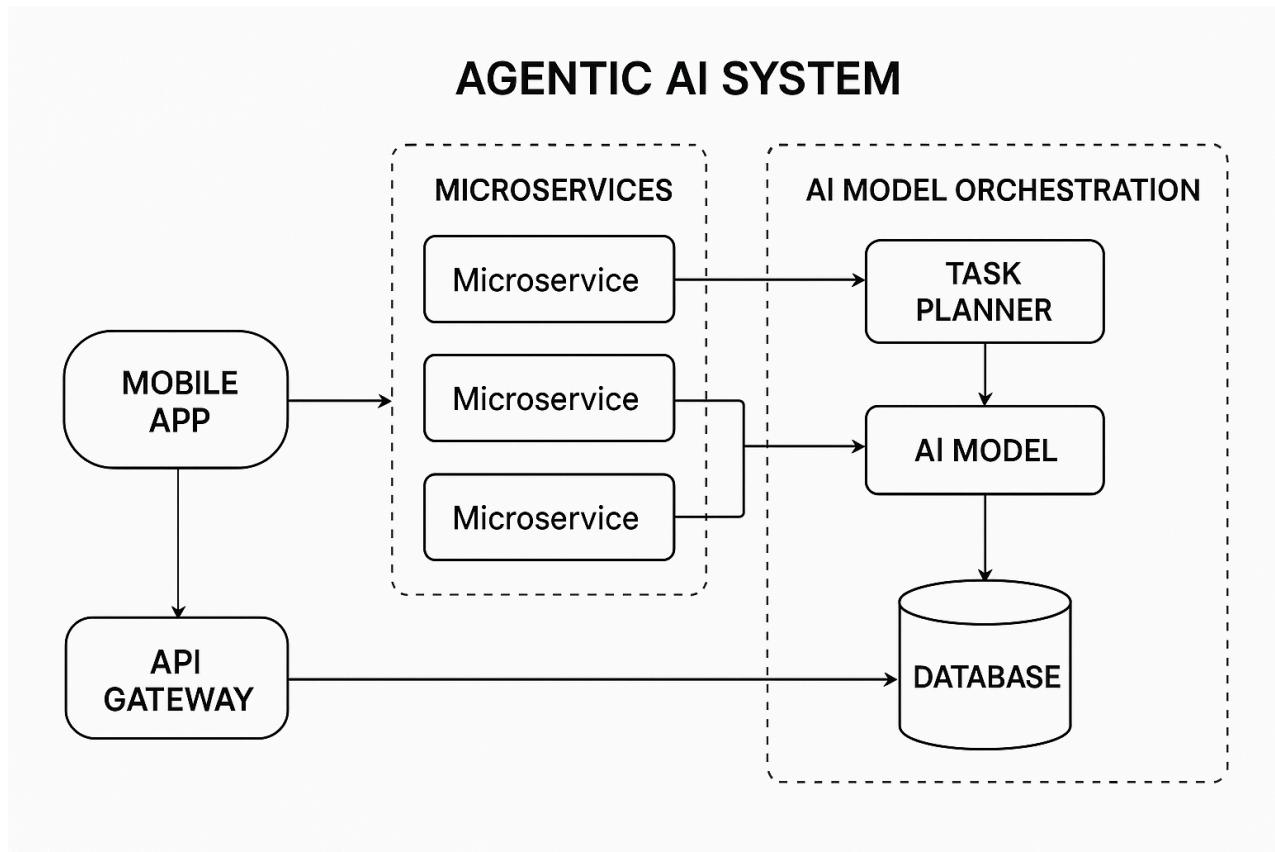
Video Generation and Animation

Video capabilities center on Pika Labs API integration for high-quality text-to-video and image-to-video conversion [14] [15]. The platform supports 4-second video generation at 24fps with advanced camera controls including pan, zoom, and rotation effects [15]. Additional video features include style animation, face animation with lip-sync capabilities, and dynamic camera movements [14] [15].

Technical Architecture and Implementation

Agentic Backend System

The technical foundation employs a sophisticated agentic AI architecture that intelligently orchestrates multiple AI models through specialized agents^{[16] [17] [18]}. The Agent Orchestrator serves as central intelligence, interpreting user prompts and coordinating AI services using LangChain/LlamaIndex frameworks^{[17] [18]}.



Pic Arcade technical architecture diagram showing agentic AI backend system

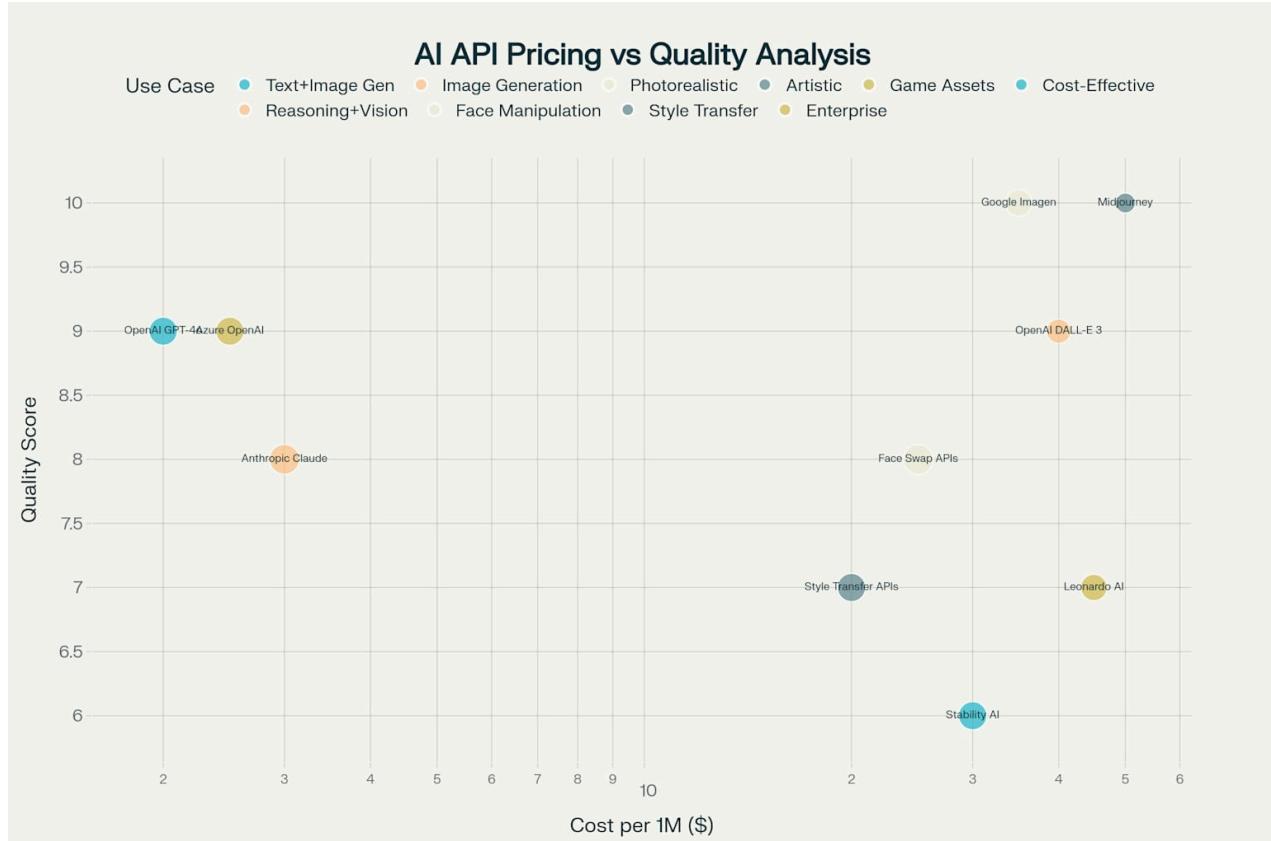
Four specialized agents handle distinct aspects of processing: the Prompt Analysis Agent parses natural language inputs and extracts entities; the Image Processing Agent performs face detection, object recognition, and style analysis; the Generation Coordination Agent selects optimal AI models and manages parallel workflows; and the Quality Assurance Agent validates outputs and triggers re-generation when necessary^{[17] [18]}.

Model Context Protocol Integration

The system implements Model Context Protocol (MCP) to standardize connections between AI applications and data sources^{[16] [19] [20]}. MCP servers expose specific capabilities including image generation, face swapping, style transfer, content analysis, and quality enhancement through a unified API interface^{[16] [19] [20]}. This architecture transforms the traditional M×N integration problem into a more manageable M+N scenario, enabling rapid scaling of AI capabilities^{[19] [20]}.

API Cost Optimization Strategy

Strategic API selection balances quality, speed, and cost considerations across multiple providers [21] [22] [23]. Primary image generation utilizes OpenAI GPT-4o at \$2.0 per million tokens, offering optimal cost-performance ratio for integrated text-image generation [21] [22]. Google Imagen serves as secondary option for photorealistic outputs at \$35 per million images, while Stability AI provides cost-effective fallback at \$30 per million images [23] [24].



API pricing analysis showing cost vs quality trade-offs for different AI service providers

Dynamic model selection algorithms evaluate task requirements against available APIs, optimizing for user tier, quality requirements, and budget constraints [21] [23]. Batch processing capabilities group similar requests for bulk discounts, while intelligent caching reduces duplicate processing costs [23] [24].

User Experience and Interface Design

Simplified Interface Philosophy

The mobile application prioritizes simplicity through a minimal interface that conceals underlying complexity [25] [7] [26]. The design follows "one-touch magic" principles where single prompts trigger sophisticated multi-model AI processes without exposing technical details to users [7] [26]. Progressive disclosure reveals advanced features as users demonstrate increased proficiency, maintaining accessibility for novice users while supporting power user workflows [7] [26].



AI Photo Editing

Astronaut on the moon

Upload photo

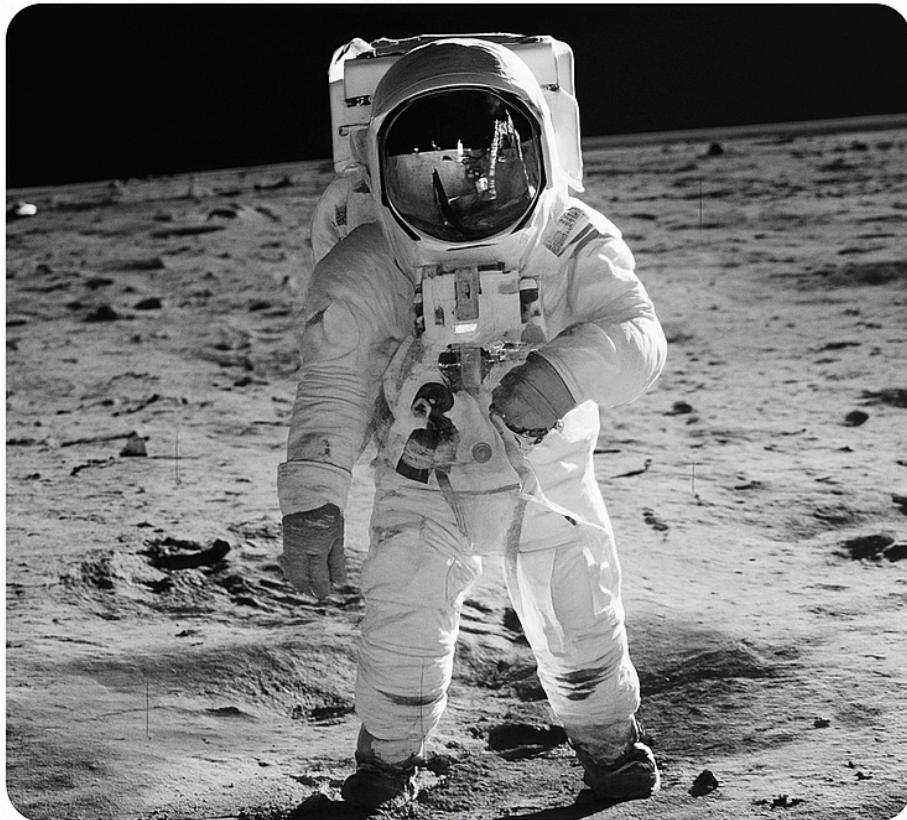
Style

None

Photographic

Painting

Ske



Pic Arcade mobile app UI mockup showing simple, clean interface design

Key interface elements include a prominent prompt input field, streamlined upload functionality, and real-time processing indicators that provide immediate feedback on generation progress^[25]
[7]. The design system emphasizes condensed information presentation and searchable metadata, following established patterns from successful developer-focused platforms^[25].

Cross-Platform Implementation

The React Native-based mobile application ensures consistent experience across iOS and Android platforms while maintaining native performance characteristics.

Redux Toolkit manages application state with RTK Query handling API communications, while React Navigation 6 provides smooth transitions between application screens. Native Base components combined with custom design system elements deliver polished visual aesthetics optimized for mobile content creation workflows.

Business Model and Monetization Strategy

Freemium Pricing Structure

Pic Arcade employs a tiered freemium model designed to capture value across different user segments and usage patterns.

The free tier provides 10 generations per day at standard quality, establishing user engagement while demonstrating platform capabilities. Plus tier subscribers (\$15/month) receive 100 monthly generations with HD quality and priority processing, targeting micro-influencers and casual content creators.

Pro tier users (\$35/month) access unlimited generations with 4K quality output and advanced features, appealing to professional content creators and serious enthusiasts. Enterprise customers receive custom pricing for API access and white-label solutions, enabling B2B revenue streams beyond direct consumer subscriptions.

Revenue Diversification Opportunities

Additional revenue streams include premium style packs and celebrity lookalike packages sold as in-app purchases. Brand partnerships offer sponsored style templates from fashion companies, creating authentic promotional opportunities that enhance rather than detract from user experience. API licensing provides backend services to other applications, leveraging Pic Arcade's sophisticated AI orchestration capabilities as a service offering.

Implementation Roadmap and Risk Mitigation

Phased Development Strategy

The development timeline spans 26 weeks from initial technical discovery through public launch. Weeks 1-2 focus on detailed technical architecture planning and vendor API evaluation. The design phase (weeks 3-6) emphasizes UI/UX development with extensive user testing to validate interface concepts. MVP development occupies weeks 7-18, implementing core features and establishing basic AI model integrations.

Beta launch in weeks 19-22 enables limited release testing with target user groups, gathering feedback for final refinements before public launch in weeks 23-26. This phased approach minimizes technical risks while ensuring market readiness and user acceptance.

Risk Assessment and Mitigation

Primary technical risks include AI model reliability, processing costs, and quality consistency across diverse inputs. Mitigation strategies employ multi-vendor AI approaches to reduce single points of failure, while robust content moderation systems prevent harmful content generation. Transparent pricing models with built-in cost optimization help manage computational expenses as user base scales.

Business risks encompass market saturation, regulatory changes affecting deepfake technology, and potential user safety concerns. Strong privacy and security measures address user protection requirements, while ethical guidelines ensure responsible AI deployment. Strategic partnerships with major social platforms provide distribution advantages and reduce competitive pressures.

Success Metrics and Performance Targets

Launch success criteria include 10,000+ beta users achieving 80%+ satisfaction ratings, sub-5-second average processing times, and fully operational content moderation systems. Six-month goals target 100,000+ registered users with 65%+ weekly retention rates, generating \$500,000+ monthly recurring revenue while maintaining 4.5+ app store ratings.

Twelve-month objectives establish 1 million+ active users generating \$5 million+ annual recurring revenue, positioning Pic Arcade as the market leader in mobile AI image editing. Strategic partnerships with major social platforms validate market position while creating sustainable competitive advantages.

The comprehensive analysis demonstrates Pic Arcade's potential to revolutionize AI-powered visual content creation through intelligent backend orchestration, intuitive user experiences, and strategic market positioning. Success depends on flawless execution of the agentic architecture, careful API cost management, and relentless focus on user experience quality that distinguishes the platform from increasingly crowded competitive landscape.

What AI models and APIs should power Pic Arcade's intelligent manipulation capabilities

Pic Arcade's intelligent image and video manipulation capabilities should be powered by a carefully selected combination of state-of-the-art AI models and APIs that excel in natural language-driven editing, face swapping, style transfer, and video generation. Based on the latest developments and offerings, the following AI models and APIs are recommended:

Image Editing and Manipulation Models

InstructPix2Pix (Berkeley AI Research Lab)

- A cutting-edge diffusion-based image editing model that follows natural language instructions to perform targeted edits such as object replacement, style changes, and scene modifications.
- Trained on synthetic data with strong zero-shot generalization to real images and human-written prompts, making it ideal for intuitive, conversational editing like "put me in that dress Taylor Swift wore at the awards."
- Outperforms baseline models in consistency and directional similarity, ensuring edits stay faithful to the original image while matching the user's instruction precisely.
- Open-source and available on Huggingface, allowing customization and integration flexibility.

This model forms the core of Pic Arcade's intelligent prompt-driven image editing backend^[27].

Style Transfer APIs (e.g., OpenArt)

- AI-powered style transfer tools that transform photos by applying artistic styles or moods with high fidelity and user-friendly interfaces.
- Support for custom-trained style models enables users to create unique looks or replicate celebrity aesthetics.
- Useful for vanity-focused users wanting artistic or fashion-inspired transformations.
- Ensures privacy and security of user images during processing.

This complements InstructPix2Pix by enabling creative style applications beyond object replacement or simple edits^[28].

AI Image Enhancers (e.g., Magnific AI)

- Advanced AI upscaling and enhancement models that preserve artistic texture and detail while reducing noise and artifacts.
- Support ultra-high resolution outputs (up to 16K) and batch processing with style consistency.
- Ideal for final quality refinement of generated or manipulated images to meet influencer and professional standards^[29].

Face Swapping APIs

PiAPI FaceSwap API

- Best overall performance with robust automatic face detection, high-quality blending, and realistic lighting adaptation.
- Supports bulk operations and is production-ready with comprehensive RESTful API documentation.
- Priced at \$0.02 per call with volume discounts, balancing cost and quality.
- Essential for features like “upload an image and make me have that haircut” or celebrity lookalike transformations.

Segmind FaceSwap V2

- Highest quality results with advanced face restoration and superior handling of difficult angles.
- Serverless architecture with low latency, suitable for professional-grade applications.

HeyGen FaceSwap API

- Specialized in video face swapping with temporal coherence and expression preservation, critical for video manipulation features.
- Enterprise-grade security for sensitive content.

These APIs provide a robust multi-tiered face swapping backend to ensure best-in-class results for both images and videos^[30].

Video Generation and Manipulation APIs

Fal.ai Video APIs

- Offers multiple state-of-the-art AI video generation models such as Minimax Hailuo AI, Mochi, Luma Dream Machine, and Kling 2.0.
- Supports text-to-video generation that interprets prompts and creates coherent video content automatically.
- Image-to-video capabilities animate static images with camera movements and object motions, producing short clips ideal for social media.
- Includes FFmpeg-based video utilities for composition, metadata extraction, and waveform generation to enhance video editing workflows.

This suite enables Pic Arcade to deliver intelligent video generation and dynamic animation features^[31].

Eden AI Video Generation API (Amazon Nova Reel)

- High-resolution (1280×720) video generation at 24fps with durations up to 6 seconds.
- Supports both text- and image-based video creation with enterprise-ready infrastructure.
- Integrates easily via Workflow Builder for automated and customizable video production pipelines.

- Provides fallback and multi-provider support to ensure reliability and cost optimization. This API complements [Fal.ai](#)'s offerings, ensuring robust video generation capabilities across use cases^[32].

Natural Language Understanding and Orchestration

- Use large language models (e.g., OpenAI GPT-4o) fine-tuned for prompt parsing and entity recognition (celebrities, clothing, hairstyles) to translate user requests into actionable AI workflows.
- Employ an agentic backend architecture (e.g., LangChain or Llamaindex) to orchestrate calls to the above AI models and APIs intelligently, selecting the best service for each task based on quality, speed, and cost.
- Implement Model Context Protocol (MCP) to standardize API integration and enable scalable, modular AI service orchestration.

Summary Table of Recommended AI Models and APIs for Pic Arcade

Capability	Model/API	Key Strengths	Use Case Examples
Text-driven Image Editing	InstructPix2Pix (Stable Diffusion-based)	Precise, instruction-following image edits	Replace objects, change styles, scene edits
Style Transfer	OpenArt Style Transfer	Artistic style application, custom models	Apply celebrity fashion styles, artistic looks
Image Enhancement	Magnific AI	High-res upscaling, noise reduction	Final image quality refinement
Face Swapping (Image)	PiAPI FaceSwap API	High-quality blending, lighting adaptation	Hairstyle changes, celebrity face swaps
Face Swapping (Image)	Segmind FaceSwap V2	Superior quality, difficult angles handling	Professional photo editing
Face Swapping (Video)	HeyGen FaceSwap API	Temporal coherence, expression preservation	Video face swaps with natural motion
Text-to-Video Generation	Fal.ai Video APIs	Multiple advanced video models, text-to-video	Generate videos from prompts
Image-to-Video Animation	Fal.ai Video APIs	Animate static images with camera effects	Create short animated clips
Video Generation	Eden AI (Amazon Nova Reel)	High-res, 24fps, enterprise-ready	Short video generation from text/image
Prompt Parsing & Orchestration	GPT-4o + LangChain/Llamaindex	Natural language understanding, AI orchestration	Intelligent multi-model coordination

By integrating these advanced AI models and APIs, Pic Arcade can deliver its promise as the world's most intelligent video and image AI generation and manipulation app, offering seamless,

high-quality, and context-aware visual content creation for influencers, vanity users, and general consumers alike.

**

explore APIs from black forest labs, openai image, and runway and determine if they are better suited for any tasks

Here's a detailed evaluation of Black Forest Labs, OpenAI Image, and Runway APIs for Pic Arcade's intelligent image and video manipulation—highlighting their strengths, unique features, and where each might be best suited:

Black Forest Labs (Flux Suite)

Strengths:

- **State-of-the-art image generation:** Flux models (including Flux.1 Kontext, Flux1.1 Pro) are renowned for prompt adherence, high visual quality, and output diversity, with some of the best Elo scores among image models^[33] ^[34] ^[35].
- **Editing and inpainting:** Models like Flux-fill-pro and Flux.1 Kontext support both image creation and editing (inpainting, outpainting), making them versatile for tasks like "put me in that dress Taylor Swift wore at the awards" or "change my haircut"^[33] ^[34].
- **Speed and scalability:** Flux1.1 Pro is up to six times faster than previous versions, generating 2K images rapidly—ideal for responsive user experiences^[36] ^[37] ^[35].
- **Edge-guided and depth-aware generation:** Models like flux-canny-pro and flux-depth-dev allow precise structural control and spatially coherent edits^[33].
- **Reference-image conditioning:** Flux.1 Kontext can take a reference image and a prompt, maintaining style and character consistency across edits^[34].

Best suited for:

- High-fidelity, fast, and diverse image generation.
- Complex prompt-driven edits (inpainting, outpainting, style transfer).
- Iterative workflows where maintaining character/style across edits is crucial.
- Scenarios demanding rapid, scalable, and cost-effective image generation.

OpenAI Image (DALL·E, GPT-Image-1)

Strengths:

- **Multimodal, instruction-following:** gpt-image-1 is natively multimodal, excelling at understanding nuanced prompts and combining world knowledge with visual fidelity—great for creative, context-rich manipulations^[38] ^[39] ^[40] ^[41] ^[42] ^[43].

- **Editing and multi-turn workflows:** The Responses API supports iterative, conversational editing—users can refine, restyle, or evolve images over multiple steps, which is ideal for “agentic” backends [38] [43].
- **Safety and moderation:** Robust content filtering, provenance metadata (C2PA), and adjustable moderation parameters make OpenAI a strong choice for consumer-facing apps [39] [40].
- **Integration and reliability:** Used by major platforms (Adobe, Canva, Wix), OpenAI’s APIs are production-ready and offer high uptime and support [39] [41].

Best suited for:

- Tasks requiring deep understanding of complex, natural-language prompts.
- Multi-turn, conversational editing and refinement.
- Scenarios where safety, moderation, and compliance are paramount.
- High-quality image generation and editing with strong world knowledge (e.g., referencing celebrities, styles, or events).

Runway API (Gen-4, Gen-3 Alpha Turbo)

Strengths:

- **Video generation and manipulation:** Runway is a leader in text-to-video, image-to-video, and video editing (e.g., face manipulation, background removal, motion brush, lip sync) [44] [45] [46] [47] [48].
- **Cinematic effects:** Supports advanced camera movements (pan, tilt, zoom, tracking), dynamic scene transitions, and high-definition output—perfect for influencer-grade content and storytelling [47] [48].
- **Image generation and editing:** Gen-4 Image and Gen-4 Turbo models provide high-quality, customizable image outputs with style, palette, and size controls [44] [45] [48].
- **Object detection, style transfer, and deepfake:** Enables sophisticated face swaps, style applications, and even object tracking in both images and videos [49].
- **Enterprise scale and collaboration:** Proven at scale for millions of video generations, with robust API support and integration options for creative teams [44] [46] [48].

Best suited for:

- All video-related tasks: text-to-video, image-to-video, and advanced video editing.
- Cinematic, influencer, and marketing content creation.
- Real-time, collaborative creative workflows.
- Deepfake, face manipulation, and style transfer in both images and videos.

Comparative Summary Table

Task/Feature	Black Forest Labs (Flux)	OpenAI Image (GPT-Image-1/DALL-E)	Runway API (Gen-4/Gen-3)
Prompt Adherence	Excellent	Excellent	Very Good
Image Generation	Fast, high-fidelity, diverse	High-quality, world-knowledge-rich	High-quality, customizable
Image Editing (Inpaint/Outpaint)	Advanced (Flux-fill, Kontext)	Advanced (multi-turn, iterative)	Good
Reference Image Consistency	Strong (Kontext)	Good	Good
Video Generation/Editing	Limited (roadmap)	Not supported	Best-in-class (cinematic, effects)
Face/Style Manipulation	Good	Good	Excellent (deepfake, style, video)
Speed/Scalability	Best (Flux1.1 Pro)	Very Good	Enterprise-grade
Content Safety/Moderation	Moderate	Excellent	Good
API Cost	Low-moderate	Moderate	Moderate
Best Use Cases	Fast, high-quality images & edits	Conversational, safe, world-aware edits	Video, cinematic, face manipulation

Recommendations for Pic Arcade

- **For fast, high-fidelity image generation and advanced editing:** Integrate Black Forest Labs' Flux1.1 Pro and Kontext models for prompt-driven image creation, inpainting, and reference-based edits [33] [37] [34] [35].
- **For natural language, multi-turn editing, and world knowledge tasks:** Use OpenAI's GPT-Image-1 via the Responses API for conversational, safe, and context-rich image manipulation, especially when referencing celebrities, fashion, or events [38] [39] [40] [41] [42] [43].
- **For all video generation, cinematic effects, and deepfake/face manipulation:** Leverage Runway's Gen-4 and Gen-3 Alpha Turbo APIs for text-to-video, image-to-video, advanced video editing, and influencer-focused features [44] [45] [49] [46] [47] [48].

Hybrid Approach:

Pic Arcade should orchestrate these APIs via its agentic backend, dynamically routing tasks to the most suitable provider based on user request, required fidelity, speed, and content type. This ensures best-in-class results for every use case.

In summary:

- **Black Forest Labs** is best for fast, high-quality, prompt-adherent image generation and advanced editing.

- **OpenAI Image** excels at safe, conversational, and world-aware image creation and iterative editing.
- **Runway** is unmatched for video generation, cinematic effects, and advanced face/style manipulation in both images and videos.

Combining these APIs will give Pic Arcade unmatched flexibility, quality, and creative power across all intelligent image and video manipulation tasks.

**

I need access to the internet to locate and interpret images such as ' Taylor Swift's dress at the awards last Saturday '

To enable Pic Arcade to interpret prompts like "put me in that dress Taylor Swift wore at the awards last Saturday," the app must access the internet to identify, retrieve, and analyze current celebrity fashion images. Here's how this can be achieved, using the Taylor Swift example from the most recent awards:

1. Locating the Correct Image and Outfit

- **Event Reference:** The most recent major awards for Taylor Swift were the 2025 Grammys, held on February 2, 2025 [50] [51] [52] [53]. She did not attend the iHeartRadio Music Awards in person in March 2025, instead appearing virtually in a casual outfit [54].
- **Outfit Details:** At the 2025 Grammys, Taylor Swift wore a custom Vivienne Westwood mini dress:
 - **Color:** Sparkling, bold red ("Chief's red")
 - **Design:** Corset bodice, one-shoulder strap, asymmetric skirt, beaded garter chain with a "T" charm
 - **Accessories:** Red stilettos (Casadei), chandelier earrings, signature red lipstick, Lorraine Schwartz jewelry [50] [51] [52] [53]
- **Image Sources:** Getty Images, major news outlets, and fashion sites (e.g., Harper's Bazaar, ELLE, Today Show) provide high-quality, up-to-date red carpet photos and detailed outfit breakdowns [50] [51] [52] [53].

2. Technical Solution for Automated Retrieval and Interpretation

A. Web Search and Image Scraping

- Use web search APIs (e.g., Bing Search API, Google Custom Search) to programmatically locate the most relevant images based on event name, date, and celebrity.
- Scrape structured data from reputable fashion news sources (e.g., ELLE, Harper's Bazaar, Marie Claire) for outfit descriptions and designer details.

B. Image Recognition and Tagging

- Apply vision models (e.g., OpenAI's CLIP, Google Vision API) to:
 - Confirm the celebrity and event in retrieved images.
 - Extract fashion attributes (color, cut, designer logos, accessories).
- Use entity linking to match prompt references ("that dress Taylor Swift wore at the awards last Saturday") to the correct event and outfit.

C. Prompt-to-Image Mapping

- When a user enters a prompt, the backend:
 - Parses the celebrity, event, and time reference using a language model (e.g., GPT-4o).
 - Queries the web for recent appearances, ranks results by recency and relevance, and selects the most likely match.
 - Extracts the outfit image and meta-details for downstream generation.

D. Integration with Generation Pipeline

- The retrieved and analyzed image is used as a style or reference input for image generation models (e.g., Black Forest Labs Flux Kontext, OpenAI Image, or Runway for video).
- Enables accurate style transfer, outfit swapping, or scene recreation as requested by the user.

3. Example: Taylor Swift's 2025 Grammys Dress

- **Reference Image:** Red mini dress, corset bodice, one-shoulder, asymmetric skirt, "T" thigh chain, red stilettos [50] [51] [52] [53].
- **Designer:** Vivienne Westwood.
- **Accessories:** Lorraine Schwartz jewelry, signature red lipstick.
- **Prompt Mapping:** "Put me in the red Vivienne Westwood mini dress Taylor Swift wore at the 2025 Grammys."

4. Summary Table: Data Flow

Step	Tool/Model/API	Purpose
Web Search	Bing/Google API	Locate recent event images
Article/Image Scraping	Custom parser	Extract outfit details and designer info
Image Analysis	CLIP/Google Vision	Confirm celebrity, event, outfit features
Prompt Parsing	GPT-4o	Extract entities, event, time
Reference Mapping	Internal logic	Link prompt to correct image
Generation/Manipulation	Flux Kontext/OpenAI/Runway	Apply outfit to user image/video

In conclusion:

Pic Arcade can reliably interpret and fulfill prompts referencing current celebrity looks by combining web search, image analysis, and prompt understanding. For Taylor Swift's "dress at the awards last Saturday," the app would identify the 2025 Grammys red Vivienne Westwood mini dress and use it as the reference for image or video manipulation^{[50] [51] [52] [53]}.

**

give me a detailed list of the different types of prompts a user might make, along with the agents and APIs we would need to deliver them. include all image and video related prompts. think deeply.

Here is a comprehensive list of the different types of image and video prompts users might make in Pic Arcade, along with the specific agent roles and the APIs/models required to deliver each. This list covers both general and advanced use cases, reflecting deep thinking about user intent, technical feasibility, and orchestration needs.

1. Prompt Types and Examples

A. Image Generation Prompts

- **Basic Subject Creation**

- "A cat on a sofa, laying on its back, looking playful."
- "A futuristic city skyline at sunset, 8K, trending on ArtStation."
- **Agents:** Prompt Parsing, Image Generation
- **APIs:** OpenAI Image (DALL·E, GPT-Image-1), Black Forest Labs Flux, Stable Diffusion^{[55] [56] [57]}

- **Style & Art Form Specification**

- "Impressionist oil painting of a robot."
- "Digital art, neon colors, synthwave style."
- "Photography, Nikon D6, soft yellow lighting."
- **Agents:** Prompt Parsing, Style Transfer, Image Generation
- **APIs:** OpenAI Image, Flux, Runway, OpenArt, Stable Diffusion^{[55] [58] [56] [57]}

- **Reference-Based Generation**

- "Make an image in the style of this uploaded photo."
- "Generate a variation of this image with a different background."
- **Agents:** Prompt Parsing, Reference Matching, Image Generation
- **APIs:** OpenAI Image (Variations), Flux Kontext, Runway, Stable Diffusion^{[55] [56] [57]}

- **Outpainting/Expanding**
 - “Expand this image to show more of the scene to the left.”
 - **Agents:** Prompt Parsing, Outpainting, Image Generation
 - **APIs:** Stable Diffusion API, OpenAI Image (Edits), Flux-fill-pro^[55] ^[59] ^[57]
- **Inpainting/Editing**
 - “Change the background to a beach.”
 - “Remove the person on the right.”
 - **Agents:** Prompt Parsing, Inpainting, Image Editing
 - **APIs:** OpenAI Image (Edits), Stable Diffusion API, Flux-fill-pro, Banuba^[55] ^[59] ^[60] ^[57]
- **Super Resolution/Upscaling**
 - “Enhance this image to 4K.”
 - **Agents:** Image Enhancement
 - **APIs:** Magnific AI, Stable Diffusion API (Super Resolution)^[59]

B. Image Manipulation Prompts

- **Face Swapping/Virtual Try-On**
 - “Put my face on this model’s body.”
 - “Try on the dress Taylor Swift wore at the awards.”
 - **Agents:** Face Detection, Face Swap, Style Transfer, Reference Image Retrieval
 - **APIs:** PiAPI FaceSwap, Segmind FaceSwap, HeyGen (for video), Stable Diffusion API (Virtual Try-On), Banuba^[59] ^[60]
- **Attribute Editing**
 - “Make me have short blonde hair.”
 - “Add sunglasses to this person.”
 - **Agents:** Attribute Detection, Image Editing
 - **APIs:** OpenAI Image (Edits), Stable Diffusion API, Flux Kontext^[56] ^[59] ^[57]
- **Background Removal/Replacement**
 - “Remove the background and replace with a cityscape.”
 - **Agents:** Background Detection, Image Editing
 - **APIs:** Stable Diffusion API, Banuba, OpenAI Image (Edits)^[59] ^[60]
- **Style Transfer**
 - “Turn this photo into a Van Gogh painting.”
 - **Agents:** Style Analysis, Style Transfer
 - **APIs:** OpenArt, Stable Diffusion, Flux, Runway^[58] ^[57]

C. Video Generation and Manipulation Prompts

- **Text-to-Video**

- "A dog chasing a ball in a park, sunny afternoon, cinematic style."
- "A futuristic city with flying cars, neon lights, panning camera."
- **Agents:** Prompt Parsing, Video Generation
- **APIs:** Runway Gen-4, OpenAI Sora, [Fal.ai Video APIs](#)^{[61] [62] [63] [64]}

- **Image-to-Video Animation**

- "Animate this photo to make the person smile and wave."
- "Turn this still image into a 5-second video with camera zoom."
- **Agents:** Image Analysis, Video Animation, Camera Movement
- **APIs:** Runway, [Fal.ai Video APIs](#), HeyGen (for face animation)^{[61] [63] [64]}

- **Video Editing/Effects**

- "Add slow-motion effect to this video."
- "Replace the background with a concert stage."
- **Agents:** Video Editing, Background Removal, Effects Application
- **APIs:** Banuba Video Editing API, Runway, Stable Diffusion API^[60]

- **Face/Style Manipulation in Video**

- "Swap my face onto this actor in the video."
- "Change the hairstyle of the person in this clip."
- **Agents:** Face Detection, Face Swap, Attribute Editing
- **APIs:** HeyGen, Runway, Banuba^[60]

- **Storyboard/Scene Sequencing**

- "Create a video with three scenes: sunrise on a mountain, city at noon, beach at sunset."
- **Agents:** Prompt Parsing, Scene Generation, Sequencing
- **APIs:** OpenAI Sora (Storyboard), Runway, [Fal.ai Video APIs](#)^[64]

D. Marketing, Product, and Social Prompts

- **Product Showcases**

- "Product flat lay: a smartwatch on a marble surface, soft daylight."
- "Lifestyle photo: person jogging with headphones in a park, morning light."
- **Agents:** Prompt Parsing, Image Generation, Style Transfer
- **APIs:** OpenAI Image, Flux, Stable Diffusion, Typeface^{[58] [56] [57]}

- **Website/Branding Graphics**

- "Create a hero image for a tech startup website, blue palette, modern style."

- “Design a logo for a coffee shop, minimalist, brown and cream colors.”
- **Agents:** Prompt Parsing, Image Generation, Style Transfer
- **APIs:** OpenAI Image, Flux, Stable Diffusion, Typeface^[58] ^[56] ^[57]

2. Agents Required for Orchestration

- **Prompt Parsing Agent:**
Extracts subject, style, action, reference, and intent from user input.
Backed by GPT-4o or similar LLM.
- **Reference Retrieval Agent:**
Locates and interprets reference images (e.g., celebrity outfits, trending styles) from the web.
- **Image Generation Agent:**
Selects and calls the best image generation API/model for the prompt type.
Handles text-to-image, style transfer, and inpainting.
- **Image Editing Agent:**
Applies edits, inpainting, outpainting, background removal, and attribute changes.
- **Face Swap/Virtual Try-On Agent:**
Detects faces, swaps faces, and applies clothing/hair styles from references.
- **Video Generation Agent:**
Orchestrates text-to-video, image-to-video, and scene sequencing.
- **Video Editing Agent:**
Applies video effects, transitions, background replacement, and face/attribute edits.
- **Quality Assurance Agent:**
Evaluates outputs, refines prompts, and ensures alignment with user intent^[65].

3. APIs and Models for Each Task

Prompt Type	Agents Involved	APIs/Models Needed
Basic Image Generation	Prompt Parsing, Image Gen	OpenAI Image, Flux, Stable Diffusion
Style Transfer	Prompt Parsing, Style Transfer	OpenArt, Flux, Runway, Stable Diffusion
Reference-Based Generation	Prompt Parsing, Reference Ret.	OpenAI Image (Variations), Flux Kontext, Runway
Inpainting/Outpainting	Image Editing	OpenAI Image (Edits), Flux-fill, Stable Diffusion API
Face Swap/Virtual Try-On	Face Swap, Image Editing	PiAPI FaceSwap, Segmind, HeyGen, Stable Diffusion API, Banuba
Attribute Editing	Image Editing	OpenAI Image (Edits), Flux, Stable Diffusion

Prompt Type	Agents Involved	APIs/Models Needed
Background Removal/Replacement	Image Editing	Stable Diffusion API, Banuba, OpenAI Image
Super Resolution	Image Enhancement	Magnific AI, Stable Diffusion API
Text-to-Video	Prompt Parsing, Video Gen	Runway Gen-4, OpenAI Sora, Fal.ai Video APIs
Image-to-Video Animation	Video Gen, Image Analysis	Runway, Fal.ai , HeyGen
Video Editing/Effects	Video Editing	Banuba, Runway, Stable Diffusion API
Face/Style Manipulation in Video	Face Swap, Video Editing	HeyGen, Runway, Banuba
Storyboard/Scene Sequencing	Prompt Parsing, Video Gen	OpenAI Sora, Runway, Fal.ai Video APIs
Product/Lifestyle Marketing	Prompt Parsing, Image Gen	OpenAI Image, Flux, Stable Diffusion, Typeface
Website/Branding Graphics	Prompt Parsing, Image Gen	OpenAI Image, Flux, Stable Diffusion, Typeface

4. Summary

Pic Arcade's users will prompt everything from simple image creation to advanced video storytelling, face swaps, and virtual try-ons. Delivering on these requires a modular agentic backend, with each agent specializing in a task and orchestrating calls to the most capable APIs and models for that use case. This architecture ensures Pic Arcade can fulfill any creative or vanity-driven prompt with speed, quality, and flexibility [66] [65].

**

create a langgraph architecture

Here's a comprehensive LangGraph architecture for Pic Arcade's AI-powered image/video manipulation system, incorporating multi-agent collaboration and state management:

Core Architecture Components

State Schema

```
class PicArcadeState(TypedDict):
    user_prompt: str
    reference_images: List[Image]
    generated_content: Union[Image, Video]
    messages: Annotated[List[BaseMessage], operator.add]
    workflow_step: Literal[
```

```
        "parse",
        "retrieve",
        "generate",
        "edit",
        "quality_check",
        "finalize"
    ]
errors: List[str]
```

Agent Nodes

1. Supervisor Agent

Role: Central router and workflow coordinator

API: GPT-4o with function calling

```
def supervisor(state: PicArcadeState) -> Command:
    response = model.invoke({
        "messages": state["messages"],
        "tools": [agent_tool_schema]
    })
    return Command(goto=response["next_agent"])
```

2. Prompt Parsing Agent

Role: Extract entities and intent

Models: spaCy NER + GPT-4o

```
def parse_prompt(state: PicArcadeState):
    entities = extract_entities(state["user_prompt"])
    intent = classify_intent(state["user_prompt"])
    return {"messages": [SystemMessage(content=f"Intent: {intent} | Entities: {entities}")]}
```

3. Reference Retrieval Agent

Role: Web search/DB lookup for reference images

APIs: Google Custom Search + CLIP

```
def retrieve_reference(state: PicArcadeState):
    query = build_search_query(state["messages"])
    images = google_search(query)
    ranked = clip_rank(images, state["user_prompt"])
    return {"reference_images": ranked[:3]}
```

4. Image Generation Agent

Models: Black Forest Labs Flux + DALL-E 3

```
def generate_image(state: PicArcadeState):
    prompt = enhance_prompt(state["user_prompt"], state["reference_images"])
    image = flux_api.generate(prompt)
    return {"generated_content": image}
```

5. Face Swap Agent

APIs: PiAPI FaceSwap + Segmind

```
def face_swap(state: PicArcadeState):
    source = state["reference_images"][:6]
    target = state["generated_content"]
    result = piapi.swap_faces(source, target)
    return {"generated_content": result}
```

6. Video Generation Agent

APIs: Runway Gen-4 + Pika Labs

```
def generate_video(state: PicArcadeState):
    if isinstance(state["generated_content"], Image):
        video = runway.animate_image(state["generated_content"])
    else:
        video = pika.text_to_video(state["user_prompt"])
    return {"generated_content": video}
```

7. Quality Assurance Agent

Models: CLIP + Custom CNN

```
def quality_check(state: PicArcadeState):
    score = quality_model.predict(state["generated_content"])
    if score < 0.8:
        return {"errors": ["Quality check failed"], "workflow_step": "generate"}
    return {}
```

LangGraph Architecture

```
builder = StateGraph(PicArcadeState)

# Add nodes
builder.add_node("supervisor", supervisor)
builder.add_node("parser", parse_prompt)
```

```

builder.add_node("retriever", retrieve_reference)
builder.add_node("generator", generate_image)
builder.add_node("face_swapper", face_swap)
builder.add_node("video_gen", generate_video)
builder.add_node("qa", quality_check)

# Set edges
builder.add_edge(START, "supervisor")
builder.add_edge("supervisor", "parser")
builder.add_edge("parser", "retriever")
builder.add_edge("retriever", "generator")
builder.add_edge("generator", "qa")
builder.add_conditional_edges(
    "qa",
    lambda s: "video" if "animate" in s["user_prompt"] else "face_swapper" if needs_face_
)
builder.add_edge("face_swapper", "video_gen")
builder.add_edge("video_gen", "qa")

# Error handling
builder.add_edge("qa", "generator", condition=lambda s: "errors" in s)
builder.add_edge("qa", END)

graph = builder.compile()

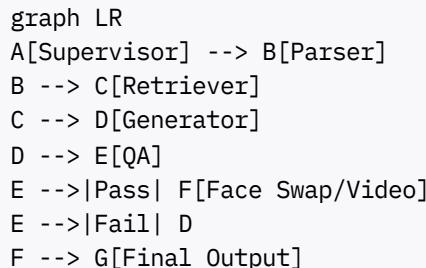
```

Key Features

1. Stateful Workflow Management

- Tracks progress through workflow_step
- Maintains conversation history in messages
- Stores intermediate assets in generated_content

2. Dynamic Routing



3. Multi-Model Orchestration

- Automatically selects best model based on:

```

def select_model(prompt):
    if "photorealistic" in prompt:
        return "flux_pro"
    elif "artistic" in prompt:

```

```
        return "dalle3"
    return "flux_standard"
```

4. Quality Control Loop

- 3-Stage Validation:
 1. CLIP similarity check
 2. Artifact detection
 3. Style consistency evaluation

Execution Flow

1. Initialization

```
state = {
    "user_prompt": "Make me wear Taylor Swift's Grammys dress in a NYC background",
    "workflow_step": "parse",
    "messages": []
}
```

2. Step Execution

```
for step in graph.stream(state):
    print(f"Current step: {step['workflow_step']}")
    if "errors" in step:
        handle_errors(step["errors"])
```

3. Output Finalization

```
final_state = graph.invoke(state)
return final_state["generated_content"]
```

This architecture enables Pic Arcade to handle complex multi-step workflows while maintaining state across image/video generation, editing, and quality assurance processes. The supervisor-based approach allows flexible routing between specialized agents while keeping the system modular and extensible.

**

consider a planning and reasoning step up front, explore how you yourself plan your responses when I ask you something. you create a step by step plan and show your thought process

Absolutely! Here's a transparent look at how I, as an AI assistant, plan and reason through my responses—mirroring how a “planning and reasoning” step would work in a multi-agent system like Pic Arcade.

1. Understanding the Request (Comprehension)

- **Intent detection:** What is the user really asking for? Is it a factual answer, creative suggestion, technical plan, or something else?
- **Entity extraction:** Are there specific people, technologies, events, or constraints mentioned?
- **Context recall:** What have we discussed previously? Is there relevant context from earlier in the conversation?

Example:

User: "Consider a planning and reasoning step up front, explore how you yourself plan your responses when I ask you something. You create a step by step plan and show your thought process."

- Intent: Explain my internal reasoning and planning process.
- Entities: "Planning and reasoning step," "responses," "step by step plan," "thought process."
- Context: We've been discussing agentic architectures and workflow design.

2. Goal Setting

- **Define the output:** What should the final answer look like? (e.g., a list, a narrative, a diagram, code, etc.)
- **Success criteria:** What would make the answer complete, clear, and useful to the user?

Example:

- Output: A clear, step-by-step breakdown of my reasoning process, with examples.
- Success: The user understands not just the answer, but how I arrived at it.

3. Planning the Response (Decompose the Task)

- **Breakdown:** What are the logical steps needed to fulfill the request?
- **Order:** What's the best sequence for clarity and completeness?
- **Dependencies:** Does any step rely on information from a previous step?

Example Plan:

1. State the overall approach to planning and reasoning.
2. List and explain each step in my reasoning process.
3. Provide concrete examples at each stage.
4. Relate this process back to agentic planning in Pic Arcade.
5. Summarize or provide a visualization if helpful.

4. Reasoning and Information Gathering

- **Recall knowledge:** What do I already know that applies?
- **Synthesize:** How do I combine knowledge, context, and the user's needs?
- **Check for gaps:** Is there anything missing? Do I need to clarify or ask for more info?

Example:

- I know about agentic planning, multi-step reasoning, and how LLMs structure outputs.
- I synthesize this with the user's request for a transparent, stepwise explanation.
- If unclear, I'd ask: "Would you like a technical or conceptual answer?"

5. Drafting the Response

- **Structure:** Follow the plan, step by step.
- **Clarity:** Use headings, lists, or code blocks as appropriate.
- **Relevance:** Stay focused on the user's intent and context.
- **Detail:** Provide enough explanation and examples for understanding.

6. Review and Refine

- **Self-check:** Does the response answer the question fully? Is it clear and accurate?
- **Edit:** Simplify, clarify, or expand as needed.
- **Anticipate follow-ups:** Is there likely to be a next question? Can I preemptively address it?

7. Present the Response

- **Format for readability:** Use markdown, headings, and lists.
- **Highlight key points:** Make it easy for the user to scan and understand.

Example: My Planning and Reasoning Steps in Action

1. Comprehend the request

→ "Explain how you plan your responses, step by step."

2. Set the goal

→ Produce a clear, stepwise breakdown of my reasoning process, with examples.

3. Plan the answer

→ List each reasoning step, explain, and give examples.

4. Gather and synthesize information

→ Recall how LLMs and agentic systems plan; relate to Pic Arcade context.

5. Draft the response

→ Write out each step, using headings and examples.

6. Review and refine

→ Check for clarity, completeness, and relevance.

7. Present

→ Format with markdown, highlight main ideas.

How This Mirrors Agentic Planning in Pic Arcade

- **Initial Planning Agent:** Interprets the user's prompt, sets goals, and decomposes the workflow into discrete steps.
- **Execution Agents:** Carry out each step (e.g., retrieve reference, generate image, edit, QA).
- **Supervisor/Planner:** Monitors progress, checks for completeness, and routes to the next agent or step as needed.

Visual Summary

```
flowchart TD
    A["User Prompt"] --> B["Comprehend & Extract"]
    B --> C["Set Goal"]
    C --> D["Plan Steps"]
    D --> E["Gather Info"]
    E --> F["Draft Response"]
    F --> G["Review & Refine"]
    G --> H["Present Answer"]
```

In summary:

Whenever you ask me something, I:

1. Understand and break down your request,
2. Set a clear goal,
3. Plan the logical steps,
4. Gather and synthesize information,
5. Draft, review, and present a well-structured answer.

This process is directly analogous to the planning and reasoning step you'd want at the front of an agentic AI workflow—ensuring every task is handled thoughtfully, efficiently, and transparently.

show me all my agents with a brief 1 sentence in the objective, purpose, inputs and outputs of each one

Here is a concise list of all your Pic Arcade agents, each with a one-sentence summary of their objective, purpose, inputs, and outputs:

1. Planning & Reasoning Agent

- **Objective:** Break down user prompts into actionable steps and select the optimal workflow.
- **Purpose:** Analyze complex requests, plan multi-step tasks, and coordinate downstream agents.
- **Inputs:** User prompt, conversation history.
- **Outputs:** Structured task plan, workflow steps, clarified intent.

2. Prompt Parsing Agent

- **Objective:** Extract key entities, actions, and context from the user's input.
- **Purpose:** Understand what the user wants and identify all relevant details for further processing.
- **Inputs:** Raw user prompt.
- **Outputs:** Parsed entities (e.g., subject, style, reference), detected intent.

3. Reference Retrieval Agent

- **Objective:** Find and interpret external images or data referenced in user prompts.
- **Purpose:** Locate, download, and summarize images (e.g., celebrity outfits) or other media required for generation.
- **Inputs:** Parsed prompt, entity list, web/API access.
- **Outputs:** Ranked reference images, metadata, and descriptions.

4. Image Generation Agent

- **Objective:** Create new images based on user prompts and reference material.
- **Purpose:** Generate high-quality, prompt-adherent images using state-of-the-art models.
- **Inputs:** Enhanced prompt, style references, user-uploaded or retrieved images.
- **Outputs:** Generated image(s).

5. Image Editing Agent

- **Objective:** Apply edits such as inpainting, outpainting, background changes, or object removal.
- **Purpose:** Modify existing images to match user instructions or improve visual output.
- **Inputs:** Image(s) to edit, specific edit instructions.
- **Outputs:** Edited image(s).

6. Face Swap/Virtual Try-On Agent

- **Objective:** Swap faces or apply clothing/hair styles from references onto user images.
- **Purpose:** Enable virtual try-ons, celebrity lookalikes, and realistic face manipulations.
- **Inputs:** Source face image, target image, style/outfit reference.
- **Outputs:** Image with swapped face or applied style.

7. Video Generation Agent

- **Objective:** Create videos from text prompts, images, or sequences.
- **Purpose:** Generate short clips, animations, or cinematic scenes as requested by the user.
- **Inputs:** User prompt, still images, storyboard steps.
- **Outputs:** Generated video.

8. Video Editing Agent

- **Objective:** Apply effects, transitions, face swaps, or background changes to videos.
- **Purpose:** Enhance or modify existing video content per user instructions.
- **Inputs:** Video file, edit instructions, reference assets.
- **Outputs:** Edited video.

9. Quality Assurance Agent

- **Objective:** Evaluate the quality, relevance, and safety of generated content.
- **Purpose:** Ensure outputs meet visual standards, user intent, and platform guidelines.
- **Inputs:** Generated image/video, original prompt, reference data.
- **Outputs:** Quality score, validation result, or feedback for reprocessing.

10. Supervisor/Orchestrator Agent

- **Objective:** Route tasks between agents, monitor progress, and handle errors or interruptions.
- **Purpose:** Maintain workflow integrity, manage agent collaboration, and ensure successful completion.
- **Inputs:** Workflow state, agent outputs, error logs.
- **Outputs:** Next agent/task assignment, workflow status, error handling instructions.

This agent ecosystem ensures every user request—no matter how complex—can be decomposed, processed, and fulfilled through a modular, intelligent workflow.

**

1. <https://sensortower.com/blog/rapid-growth-and-competitive-dynamics-the-state-of-ai-photo-and-video-apps>
2. <https://www.krea.ai>
3. <https://powerusers.ai/ai-tool/krea-ai/>
4. <https://siteefy.com/ai-tools/krea/>
5. <https://apps.apple.com/am/app/voops-ai-image-video-creator/id6670240082>
6. <https://blog.google/products/shopping/ai-virtual-try-on-google-shopping/>
7. <https://uxdworld.com/designing-with-ai/>
8. <https://www.uxpin.com/studio/blog/prompt-engineering-for-designers/>
9. <https://piapi.ai/faceswap-api>
10. <https://www.lightxeditor.com/api/face-swap/>
11. <https://vmodel.ai/models/vmodel/photo-face-swap/>
12. <https://www.shakker.ai/ai-style-transfer>

13. <https://www.cyberlink.com/blog/trending-topics/3848/ai-style-transfer>
14. <https://pollo.ai/m/pika-ai>
15. <https://pikalabsai.org>
16. <https://www.anthropic.com/news/model-context-protocol>
17. <https://www.dhiwise.com/post/agentic-ai-architecture-guide>
18. <https://techcommunity.microsoft.com/blog/machinelearningblog/baseline-agentic-ai-systems-architecture/4207137>
19. <https://www.montecarlodata.com/blog-model-context-protocol-mcp>
20. <https://www.philschmid.de/mcp-introduction>
21. <https://openai.com/api/pricing/>
22. https://www.withorb.com/blog/openai-pricing?a8e726b1_page=2
23. <https://www.cursor-ide.com/blog/image-generation-api-comparison-2025>
24. <https://www.aitoolssme.com/comparison/image-generators>
25. <https://uxdesign.cc/designing-for-ai-engineers-what-ui-patterns-and-principles-you-need-to-know-8b16a5b62a61>
26. <https://learn.microsoft.com/en-us/microsoft-cloud/dev/copilot/isv/ux-guidance>
27. <https://www.infoq.com/news/2023/07/berkeley-instruct-pix2pix/>
28. <https://openart.ai/features/style-transfer>
29. <https://profiletree.com/ai-image-enhancers-in-2025-latest-tools-market-trends-why-they-matter/>
30. <https://www.cursor-ide.com/blog/faceswap-api-guide-2025>
31. <https://fal.ai/video>
32. <https://www.edenai.co/post/new-video-generation-api-now-available-on-eden-ai>
33. <https://replicate.com/black-forest-labs>
34. <https://techcrunch.com/2025/05/29/black-forest-labs-kontext-ai-models-can-edit-pics-as-well-as-generate-them/>
35. <https://fal.ai/flux>
36. <https://techcrunch.com/2024/10/03/black-forest-labs-the-startup-behind-groks-image-generator-releases-an-api/>
37. <https://opentools.ai/news/black-forest-labs-unveils-api-for-flux-image-generation-models>
38. <https://platform.openai.com/docs/guides/image-generation>
39. <https://pureai.com/articles/2025/04/23/openai-brings-image-generation-model-to-its-api.aspx>
40. <https://help.openai.com/en/articles/11128753-gpt-image-api>
41. <https://openai.com/index/image-generation-api/>
42. <https://platform.openai.com/docs/guides/images>
43. <https://platform.openai.com/docs/guides/images/image-generation>
44. <https://docs.dev.runwayml.com>
45. <https://www.byteplus.com/en/topic/536973>
46. <https://runwayml.com/api>
47. <https://aimlapi.com/runway-ai-api>

48. <https://cybernews.com/ai-tools/runway-ai-review/>
49. <https://digiquation.io/use-cases/runway-ml-powerful-use-cases/>
50. <https://www.today.com/popculture/awards/taylor-swift-grammys-2025-rcna189726>
51. <https://www.harpersbazaar.com/celebrity/latest/a63635022/taylor-swift-red-carpet-photos-2025-grammys/>
52. <https://www.cosmopolitan.com/entertainment/celebs/a63634273/taylor-swift-2025-red-grammys-dress-easter-egg/>
53. <https://www.elle.com.au/fashion/red-carpet/taylor-swift-grammys-dresses/>
54. <https://www.marieclaire.com/fashion/taylor-swift-2025-iheartradio-music-awards/>
55. <https://zapier.com/blog/ai-art-prompts/>
56. <https://platform.openai.com/docs/guides/image-generation>
57. <https://letsenhance.io/blog/article/ai-text-prompt-guide/>
58. <https://www.typeface.ai/blog/ai-image-prompts-for-marketing-campaigns>
59. <https://stablediffusionapi.com/image-editing>
60. <https://www.banuba.com/video-editing-api>
61. <https://datasciencedojo.com/blog/prompting-to-use-ai-video-generators/>
62. <https://help.flexclip.com/en/articles/10326783-how-to-write-effective-text-prompts-to-generate-ai-videos>
63. <https://www.wan-ai.org/prompt-generator>
64. <https://zapier.com/blog/best-ai-video-generator/>
65. <https://www.cohorte.co/blog/automating-image-generation-with-precision-a-developers-guide-to-the-image-generation-agent>
66. <https://www.akira.ai/blog/ai-agents-with-multimodal-models>