ChatGPT

BirthdayGen Process & Architecture Document (PFD)

1. System Architecture Overview

BirthdayGen is built as a client-heavy Single Page Application (SPA) using **React + Vite** and the **shadcn/ui** component library. There is no traditional server; instead, the application relies on **Supabase** for authentication, database, storage and serverless functions. The key architectural components are:

1. **Frontend (React SPA)** – Presents the user interface, manages state, communicates with Supabase via its JavaScript client and handles real-time previews. It loads card templates, runs the aura quiz and interacts with AI functions.
2. **Supabase Services** – Provides:
3. **Auth**: email/password login and (future) social OAuth.
4. **Database**: Postgres tables for users, contacts, cards, aura quiz responses, gift recommendations and (future) organizations.
5. **Row-Level Security (RLS)**: Ensures that each user only accesses their own data.
6. **Edge Functions**: Serverless functions for card CRUD operations, email sending, AI image generation, agent orchestration and Stripe checkout.
7. **Storage**: Stores card images, AI-generated images and user-uploaded assets.
8. **AI Services** – Integrations with AI image models (OpenAI DALL-E, local Stable Diffusion via AUTOMATIC1111/SDXL) and text models for personalization and prompts. These are invoked from edge functions or the client.
9. **External APIs & Integrations** – Google People API for contact import, email delivery provider (Resend or SendGrid) for sending cards, Stripe for payment and subscription management, Twilio for future SMS delivery, Slack/Microsoft Teams webhooks for B2B notifications.
10. **Future Agent Platform** – A multi-agent orchestration layer built with CrewAI; each agent (Creative Director, Content Writer, Image Artist, Relationship Analyst, Delivery Coordinator, Quality Controller) is responsible for a specific part of the personalization workflow.

2. Major Process Flows

2.1 Onboarding & Setup Flow

The onboarding flow aims to get new users from sign-up to first automation within five minutes:

1. **Sign Up:** User registers via email/password (future: social login). Supabase creates a user record.
2. **Aura Quiz:** Immediately after sign-up, the app launches a short personality quiz; the resulting aura type is stored in the `users` table.
3. **Contact Import:** User imports contacts via CSV or Google People API. The system parses the data and inserts rows into `contacts`.

4. **Sample Card Generation:** AI generates a few sample cards for the first few contacts (optional, uses the user's aura and contact info). The user can preview and tweak these samples.
5. **Automation Setup:** A simple toggle enables auto-sending of cards for upcoming birthdays; the user can adjust tiers (message, card, card + gift) and scheduling preferences.
6. **Confirmation:** The app confirms upcoming birthdays and shows the user a dashboard with upcoming events.

2.2 Card Creation & Customization Flow

1. **Template Selection:** User chooses a base template from the library (cosmic, minimal, playful, elegant etc.).
2. **Message Editing:** User writes or edits the greeting; optionally uses AI to suggest personalized text based on aura and relationship.
3. **Artwork Generation:** If the user requests AI artwork, the frontend calls a Supabase edge function (`generate-ai-image`) which invokes the selected model (OpenAI DALL-E or local SDXL). If the primary service fails, fallback images are used.
4. **Real-Time Preview:** The UI updates the card preview live as the user changes text, colors, fonts and images.
5. **Save or Schedule:** The user can save the card as a draft, schedule it for a specific date/time or send it immediately. The card metadata is stored in the `cards` table with `status` (draft/scheduled/sent) and `scheduled_at` fields.
6. **Send:** For an immediate send, the frontend calls `send-card` edge function, which retrieves card content, contacts the email service (e.g., Resend) and logs the delivery in a `deliveries` table.

2.3 Contact Import Flow

1. **Initiate Import:** User clicks "Import Contacts." If CSV, they upload a file; if Google, they trigger OAuth (requires Google OAuth credentials). The UI calls `/api/import/google/start` to obtain a URL and redirects.
2. **OAuth Callback:** After authentication, Google redirects to `/api/import/callback/google`, passing an authorization code. The API exchanges the code for an access token and calls People API to fetch contacts.
3. **Data Parsing & Storage:** Extract names and birthdays, map them to our contact schema and insert into `contacts` table. Duplicate checks are performed to avoid duplicate entries.
4. **User Feedback:** Provide progress and success messages; show imported contacts on the dashboard.

2.4 Automation & Delivery Flow

1. **Configure Automation Rules:** Each contact row has a toggle for "Auto-send on birthday" and a selection of the tier (message only, card, card + gift). Settings are saved in the `contacts` table or an `automations` table with the user ID, contact ID and tier.
2. **Scheduled Job:** A Supabase scheduled function (`send-due-cards`) runs daily at midnight. It queries contacts whose birthday matches today and whose `auto_send_enabled` is true, then loops through them.
3. **Prepare Content:** For each contact, a card draft is generated or retrieved. An AI agent could personalize the message and artwork (future). If a fallback is needed, the default template and message are used.

4. **Delivery:** The `send-card` edge function sends the card via email, logs the result and returns success/failure to the scheduler. Additional channels (SMS via Twilio, Slack/Teams) can be supported by specialized functions.
5. **Notifications:** The recipient receives the card; the sender gets a confirmation. Failures are logged and surfaced to the user via notifications.

2.5 AI Personalization & Agentic Orchestration (Future)

When multi-agent orchestration is introduced, personalization becomes a collaborative process:

1. **Creative Brief Generation:** A Creative Director agent examines occasion and aura, producing a brief that guides style, tone and imagery.
2. **Content Writing:** A Content Writer agent uses the brief, contact context and relationship data to draft the card message.
3. **Image Creation:** An Image Artist agent generates artwork matching the brief; fallback images are selected if AI fails.
4. **Relationship Analysis:** A Relationship Analyst agent assesses the relationship to fine-tune tone and gift recommendations.
5. **Delivery Coordination:** A Delivery Coordinator agent determines optimal send time and channel based on preferences and analytics.
6. **Quality Control:** A Quality Controller agent reviews content for appropriateness and alignment with branding.
7. **Orchestration Loop:** The Orchestrator coordinates these agents, handling retries, fallbacks and final assembly before sending.

2.6 Payment & Subscription Flow (Future)

1. **Plan Selection:** User navigates to pricing page and chooses a premium plan or one-time arcade purchase.
2. **Checkout Creation:** The frontend calls `create-checkout` edge function, passing the user ID and selected plan. The function creates a Stripe Checkout Session and returns the URL.
3. **Redirect & Payment:** User is redirected to Stripe's hosted checkout. After successful payment, Stripe redirects to `/payment-success` with a session ID.
4. **Webhook Handling:** A Stripe webhook triggers an edge function that verifies the event and updates the user's subscription status in the database.
5. **Access Management:** Premium features become available to the user; the UI reflects the upgraded plan.

2.7 B2B Organization Flow (Future)

1. **Organization Creation:** A user upgrades to a business plan and creates an organization; the `organizations` table stores org data and `organization_members` links users to orgs.
2. **Team Management:** Admins invite team members and manage roles. Contacts can now belong to the organization rather than individuals; RLS policies allow org admins to access relevant data.
3. **Branded Card Design:** Admins set up default templates and corporate branding guidelines; Creative Director agent ensures brand consistency across all cards.
4. **Employee Celebrations:** Automation runs across the entire organization, sending cards/gifts to employees via email or Slack/Teams. Analytics dashboards track engagement and participation.

3. Data Model Overview

The following tables constitute the core data model (fields simplified):

Table	Key Fields	Purpose
`users`	`id (UUID)`, `email`, `name`, `aura_type`, `subscription_status`	Stores user credentials, profile and subscription info
`contacts`	`id (UUID)`, `user_id`, `name`, `email`, `birthday`, `relationship`, `aura_type`, `auto_send_enabled`, `tier`	Holds contacts imported by each user
`cards`	`id`, `user_id`, `contact_id`, `template_id`, `message`, `image_url`, `status`, `scheduled_at`, `sent_at`	Stores card drafts, scheduled and sent cards
`aura_quiz_responses`	`id`, `user_id`, `responses`, `aura_type`, `created_at`	Records quiz answers and results
`gift_recommendations`	`id`, `aura_type`, `title`, `description`, `image_url`, `link`	Static list of gift suggestions
`deliveries`	`id`, `card_id`, `channel`, `sent_at`, `status`	Logs deliveries of cards via email/ SMS/etc.
Future: `organizations`, `organization_members`, `automations`, `agent_logs`	For B2B features, scheduling rules and agent orchestration logs.	

4. Integration Points & External Services

- **Google People API** – Fetches user contacts after OAuth; requires client ID/secret and People API enabled.
- **AI Image Services** – DALL-E API, Stable Diffusion (AUTOMATIC1111 or SDXL) or other third-party image models. Must support fallback to placeholder images.
- **Email Service** – Resend or SendGrid used by `send-card` edge function to deliver cards. Requires domain configuration for deliverability.
- **Stripe** – Handles payments and subscription billing; integrated via `create-checkout` function and webhooks.
- **Twilio** (Future) – For SMS delivery; integrate using Supabase functions.

- **Slack/Microsoft Teams** (Future) – Use webhooks or OAuth to post birthday messages in company channels.

5. Security Considerations

- **Row-Level Security (RLS).** Every table must implement RLS to ensure users only access their own data. For organizations, policies must allow org members to access their org data.
- **OAuth Security.** Google import flows must protect client secrets; tokens should be stored securely and revoked when user revokes access.
- **API Keys.** AI and email service keys must be stored as environment variables and never exposed to the frontend. Edge functions should handle secrets.
- **Data Privacy.** Provide transparency on how personal data (contacts, birthdays, aura types) is used. Offer data export and deletion on demand.

6. Summary

This PFD outlines how BirthdayGen's client-first architecture interacts with Supabase, AI services and external APIs to deliver a seamless celebration experience. It maps the critical flows from onboarding through card creation, automation and future agentic orchestration. These processes and structures will guide engineering efforts and ensure the platform scales toward the ambitious future vision.
