Alright. I took everything **from the "Source: DCVaaS Platform: Go-to-Market Strategic Plan.pdf"** section (only that source), **combined + deduped it**, and rebuilt it into a clean taxonomy you can actually implement.

Legend:

- **[SHARED]** = must stay consistent across all 3 brands (platform truth)
- **[HYBRID]** = facts are shared, but the *angle/copy/ordering* changes per brand
- **[PER-BRAND]** = belongs inside each brand wrapper (marketing voice, emphasis, plan naming, etc.)

---

# 1) Market Opportunity and Urgency

**What the source says (deduped):**

- HTTPS is universal now, and certificate lifecycles are becoming **high-frequency operations**, not a periodic chore.
- The industry mandate for **90-day SSL certificates** (with proposals to shorten further to **47 days**) makes manual renewal workflows **untenable**.
- This creates a new category opportunity for **automated, secure infrastructure tooling** (DCVaaS).

**Where it goes: [HYBRID]**

- **All brands** should carry the "why now" urgency.
- The *translation* changes by audience.

**Implementation placement:**

- **Marketing site:** "Why now" block near the hero.
- **In-app:** lightweight reassurance copy ("renewals handled automatically") on domain status views.
- **Sales deck / one-pager:** top slide framing.

**Brand angle mapping:**

- **keylessssl.dev:** "renewal velocity is exploding (90→47); manual is dead; automation must be secure."
- **delegatedssl.com:** "portfolio ops risk explodes; agencies need centralized control."
- **autocertify.net:** translate to outcomes: "expired cert = broken trust / lost conversions; we prevent that automatically."

---

# 2) The Core Problem Being Solved

**What the source says (deduped):**

- Custom/vanity domains are required for trust + branding, but HTTPS forces SaaS providers to own the **entire cert lifecycle** (validation, issuance, renewal) for **customer domains they don't own**.
- With 90-day (and potentially 47-day) lifetimes, "manual renewal" is not a strategy—automation is mandatory.
- DIY solutions introduce **operational risk** and **security vulnerabilities**.

**Where it goes: [HYBRID]**

- The *facts* are platform-shared. The *wording* should match each brand's persona.

**Implementation placement:**

- **Marketing site:** "Problem" section (immediately after hero).
- **Docs:** "What this solves" page + onboarding explanation.
- **App onboarding:** first-screen explanation before asking user to add DNS.

---

# 3) The Solution Mechanism: Delegated DCV via CNAME

**What the source says (deduped):**

- Standard automation uses **ACME DNS-01** (CA requires a TXT record in DNS).
- Requiring customers to update TXT records every renewal cycle is a non-starter.
- The core innovation is **Delegated Domain Control Validation (DCV)** using a **one-time CNAME** delegation of the `_acme-challenge` subdomain to the platform.

**Workflow (canonical, deduped):**

1. Customer adds **one CNAME** delegating `_acme-challenge.<domain>` to the platform
2. CA issues a DNS-01 challenge when cert is needed
3. Platform (controlling delegated subdomain) creates required TXT response
4. **Zero-touch renewals** forever after

**Where it goes: [SHARED]**

- This is the product's "physics." Don't let brands drift into contradictory explanations.

**Implementation placement:**

- **App onboarding wizard:** this is the step-by-step core.
- **Docs:** canonical “How it works” page (shared content module).
- **Marketing site:** simplified “How it Works” block (brand-specific copy, same diagram).

**Implementation notes:**

- Build one shared **CNAME Delegation Wizard component** used across all brands.
- Present **manual setup** always; optionally add “Single-click setup” when OAuth integrations exist (see category #9).

---

# 4) Primary Security Narrative

**What the source says (deduped):**

- The agentless CNAME delegation model is **fundamentally more secure** than storing root DNS credentials on servers.
- It eliminates the catastrophic blast-radius of holding **high-privilege DNS API keys** in production systems.

**Where it goes: [HYBRID]**

- Security is shared as a pillar, but intensity varies by audience.

**Implementation placement:**

- **Marketing:** “Security model” section + comparison snippets vs DIY.
- **Docs:** security explainer + threat model page.
- **In-app:** tooltip/help copy near DNS setup and verification screens.

**Brand angle mapping:**

- **keylessssl.dev:** lead with this (it’s the spear).
- **delegatedssl.com:** frame as risk reduction for agency operations (“no client DNS keys in your tooling”).
- **autocertify.net:** keep it non-technical (“you never share sensitive DNS access”).

---

# 5) Market Sizing and Opportunity Numbers

**What the source says (deduped):**

- **TAM:** ~200M active websites × 90% SSL penetration = **180M websites**

- **Top-down SAM:** 15% wildcard-eligible estimate ⇒ **27M domains**
- **Bottom-up SAM: ~6.2M domains**, estimated annual value **>$104M**

**Where it goes: [SHARED]**

- These numbers should not vary brand-to-brand (or you'll look sloppy).

**Implementation placement:**

- **Internal strategy / investor deck / sales enablement** (not necessarily homepage).
- Optional: long-form blog post or "market context" page for dev brand.

---

# 6) Target Customers: ICPs

**What the source says (deduped):**

**ICP 1: Lean SaaS Provider / Startup**

- 1–10 dev team; hybrid/multi-cloud (example: frontend on Vercel, backend on AWS)
- Pain: wildcard automation across heterogeneous infra; DIY requires storing high-privilege DNS keys
- Willingness to pay: **High**

**ICP 2: Modern Digital Agency**

- Manages 20–200 client websites across many hosts; wildcard certs common for staging/subdomains
- Pain: operational inefficiency + reputational risk (one expired cert destroys credibility)
- Willingness to pay: **High** (absorbed into retainers)

**ICP 3: Power Developer / Homelab Enthusiast**

- Technical individual, many projects/self-hosted services; heavy wildcard usage
- Pain: technical friction; aware of security risks; price sensitive
- Willingness to pay: **Low** (best served with generous free tier)

**Where it goes: [PER-BRAND] (as messaging), [SHARED] (as segmentation truth)**

- The ICP definitions are shared truth.
- Which ICP you *lead with* is brand-dependent.

**Implementation placement:**

- **Marketing positioning:** homepage + "Who it's for" block

- **SEO pages:** create ICP-specific pages (Agency page, SaaS page, Dev page)
- **Onboarding:** ask 1 question ("SaaS / Agency / Personal") to tailor onboarding copy only (not backend logic)

**Brand mapping:**

- **keylessssl.dev:** ICP1 + ICP3 (developer-led)
- **delegatedssl.com:** ICP2 primary (agency + teams)
- **autocertify.net:** translate ICP2/ICP1 into "site owners / marketers" language (same product, different wrapper)

---

# 7) Beachhead Market Strategy

**What the source says (deduped):**

- Initial beachhead should be **ICP 1 (Lean SaaS Provider/Startup)** because:
    - Highest urgency + understands the security value proposition
    - High willingness to pay
    - Strong influence inside developer communities → creates social proof for later segments

**Where it goes: [SHARED]**

- This is operational truth for GTM sequencing, even if a brand's homepage targets another ICP.

**Implementation placement:**

- **Internal GTM plan** + launch sequencing.
- **Content strategy:** early content should skew dev-heavy.

---

# 8) Competitive Landscape and Positioning

**What the source says (deduped):**

**Tier 1 Incumbent: Cloudflare for SaaS**

- Strength: scale + free tier (100 hostnames)
- Weaknesses:
    - **Platform lock-in**: certs only usable for traffic proxied through Cloudflare
    - **Gated features**: wildcard + apex behind expensive enterprise

    - **“Success tax”**: usage overage **$0.10/hostname/month** becomes unpredictable at scale

**Tier 2 Niche Challengers: BrandSSL & SaaSCustomDomains**

- BrandSSL: white-label but **no team management**
- SaaSCustomDomains: feature set but **no white-label** and **no team management**; has usage-based overage scaling anxiety
- Both have base plans starting **$20–$29/mo**, blocking low-end experimentation

**Feature gap matrix conclusion:**

- Nobody combines **CNAME delegation + white label + team management** in a platform-agnostic product the way “Our Platform” does.

**Where it goes: [HYBRID]**

- Competitive facts are shared.
- Which competitor you attack first depends on brand + channel.

**Implementation placement:**

- **Marketing SEO:** “alternative to” and “limitations” pages (explicitly called out in the source)
- **Docs:** “migration from Cloudflare for SaaS” guide
- **Sales enablement:** battlecard + comparison table

**Brand mapping:**

- **keylessssl.dev:** “secure alternative to certbot DNS plugins” + “Cloudflare for SaaS limitations”
- **delegatedssl.com:** emphasize “overage success tax” + “agency-grade missing features”
- **autocertify.net:** keep comparisons minimal; focus on results unless you’re running search ads

---

# 9) Core Differentiators

**What the source says (deduped):**

1. **Superior security model**: agentless delegation, no root DNS credentials stored
2. **Platform agnosticism**: works across hybrid/multi-cloud (not a walled garden like Cloudflare/Vercel/AWS)

3. **Pro/Agency-grade feature set**: **Team Management (RBAC)** + **White-label Control Plane**

**Where it goes: [HYBRID]**

- Same 3 pillars everywhere; different emphasis + ordering.

**Implementation placement:**

- **Marketing:** 3-pillar block
- **In-app:** feature surfaces aligned to pillars (RBAC screens, white-label settings, audit trail)
- **Sales deck:** “why us” slide

---

# 10) Core Platform Feature Set

**What the source says (deduped):**

- Automated certificate lifecycle management (set-it-and-forget-it; one-time CNAME for zero-touch issuance + renewals)
- Wildcard cert support included in **all plans**
- Single-click CNAME setup via OAuth with DNS providers (Cloudflare, GoDaddy cited)
- Resilient backend:
  - durable queues
  - **Dead-Letter Queue**
  - idempotent jobs
  - automated CA failover (**Let’s Encrypt + ZeroSSL**)
- Developer-first tooling:
  - REST API
  - webhooks (events like `domain.active`, `domain.error`)
  - immutable audit trail
- Agency & team features:
  - white-label control panel
  - multi-user + **RBAC**
- Testing & staging sandbox for safe integration / avoiding CA rate limits

**Where it goes: [SHARED]**

- These are product requirements. Don’t let brand wrappers “invent” new features or omit real ones.

**Implementation placement:**

- **App:** features and navigation
- **Pricing page:** entitlements table (shared)
- **Docs:** API/webhooks/audit trail + sandbox docs

---

# 11) High-Level Technical Architecture

**What the source says (deduped):**

- Modern serverless architecture on Cloudflare:
  - **Cloudflare Workers** (API + app logic at edge)
  - **Cloudflare D1** (primary SQL database)
  - **Cloudflare Queues** (durable background jobs)
- Rationale: scalability, reliability, cost efficiency → enables aggressive pricing strategy

**Where it goes: [HYBRID]**

- This is shared truth, but only dev-focused audiences care deeply.

**Implementation placement:**

- **Docs:** architecture overview
- **Marketing:** only for keyless/dev brand as a credibility asset
- **Status/Reliability page:** optional, but aligns with “trusted infrastructure tool” positioning

---

# 12) Pricing and Packaging Model (Product-Led Growth)

**What the source says (deduped):**

- Designed to avoid a big jump from free → expensive; gradual upgrade ladder.

**Tiers (from the source):**

- **Free (Acquisition):** Free, **5 domains**
  Purpose: destroy competitor $20–$29/mo entry barrier; own low-end funnel
- **Startup (Activation): $19/mo**, **100 domains + API access**
  Purpose: smooth upgrade from free; undercuts BrandSSL ($29 for 100 domains)
- **Growth (Monetization): $49/mo**, **500 domains + team management**
  Purpose: undercuts BrandSSL $99 plan; predictable off-ramp from Cloudflare overages
- **Business (Scaling): $99/mo**, **1,000 domains + full white-label control plane**
  Purpose: primary value capture; combines what competitors lack; predictable scaling

**Where it goes: [HYBRID]**

- **Shared core:** entitlement ladder (what features unlock at which tier).
- **Per brand:** plan naming, price points, bundle emphasis *if you choose to customize.*

**How to implement cleanly (so brands can differ without breaking the product):**

- Create a **single entitlements matrix** (feature flags) that is consistent:
  - `domains_limit`, `api_access`, `team_rbac`, `white_label`, `sandbox`, etc.
- Each brand gets a **Pricing Presentation Layer** that maps:
  - brand plan name + price + marketing copy → to the same entitlement IDs
    This prevents three brands from becoming three backends.

---

# 13) GTM Strategy: Phase 1 and Phase 2

**What the source says (deduped):**

**Phase 1 (Launch → 6 months): Developer-Led Growth**

- Content marketing + SEO:
  - deep tutorials
  - “alternative to” comparison pages
  - target searches like “secure alternative to certbot DNS plugins” and “Cloudflare for SaaS limitations”
- Community engagement:
  - “Show HN”
  - relevant subreddits (r/devops, r/saas, r/selfhosted)
- Message pivot: not just automation → **secure automation**

**Phase 2 (6 → 18 months): Scale into Agencies + SaaS**

- Targeted outreach: agencies + growing SaaS using testimonials/case studies
- Partnerships: developer-first hosts (DigitalOcean, Linode) and BaaS platforms

**Where it goes: [SHARED]**

- These are GTM motions you execute. Brand wrappers just express them differently.

**Brand mapping (execution):**

- **keylessssl.dev:** spearhead Phase 1 content + community credibility
- **delegatedssl.com:** spearhead Phase 2 outreach + agency partnerships
- **autocertify.net:** can still benefit from Phase 1 SEO, but keep copy non-technical

---

# 14) Launch Plan

**What the source says (deduped):**

1. **Private beta:** small, high-touch cohort of Lean SaaS developers; gather feedback + testimonials
2. **Coordinated public launch:** Product Hunt + Show HN simultaneously
3. **First 48 hours:** founders actively respond in discussions to build trust + convert curiosity into usage

**Where it goes: [SHARED]**

- One launch plan. Don't fragment it by brand.

**Implementation placement:**

- Internal runbook + checklist
- Public launch asset pack: PH page, HN post, demo video, docs ready, onboarding polished

---

# 15) Post-Launch Roadmap Priorities

**What the source says (deduped):**

- Expanded OAuth integrations for one-click CNAME setup (more DNS providers)
- Advanced analytics/reporting (status, trends, renewal history)
- Enterprise-grade features:
    - SSO
    - support for custom domains for the white-label control plane
- Ecosystem integrations:
    - official Terraform provider (infrastructure-as-code)
- Embeddable onboarding components:
    - white-label JS component customers embed to guide their users through CNAME setup

**Where it goes: [SHARED]**

- This is your product roadmap, not marketing fluff.

**Implementation placement:**

- Product board + quarterly roadmap
- Docs roadmap page (optional)
- App placeholders only if you're confident you'll ship soon (don't tease vapor)

---

# The Clean Implementation Rule Set

If you want this multi-brand system to stay sane:

1. **One shared "Platform Truth Layer"**
    - All [SHARED] categories become a single canonical module used by all brands (docs snippets, diagrams, onboarding flows).
2. **Three "Brand Wrappers"**
    - Only [PER-BRAND]/[HYBRID presentation] content varies: voice, CTA, page ordering, ICP emphasis, plan names/prices.
3. **Single Entitlements Matrix**
    - Pricing can be presented differently per brand, but the **feature gating must map to one shared entitlement schema**.