## KeylessSSL Landing Page + Product UI Spec

Brand: security-first, API-first, developer-to-developer.

Vibe: “this is a real infra tool,” not marketing fluff.

Primary conversion: generate API key (free for 3 domains) → add CNAME → green “validated” → first cert issued.

---

# 1) One-page site structure (single scroll, anchored sections)

### Sticky Header (always visible)

- Left: KeylessSSL wordmark + small glyph (a key with a slash through it inside a shield).
- Center (desktop only): anchor links

  #Architecture #Integration #Pricing #Docs
- Right:
  - secondary: Status (links to status page)
  - secondary: Sign in
  - primary CTA button (persistent): Get API Key — Free for 3 Domains
- Behavior: on scroll, header condenses; CTA remains full-size.

Microcopy (tooltip on CTA):

No credit card. Key shown once. Tokens are hashed server-side.

---

## Section A — Hero (above the fold)

Headline (bold, hard-edged):

Stop leaking your DNS root keys to production servers.

Subheading (technical, succinct):

KeylessSSL automates wildcard SSL via Delegated DCV (DNS-01 / ACME challenge) so your root DNS credentials never leave your vault. Add one CNAME. Ship renewals forever.

Primary CTA: Get API Key — Free for 3 Domains

Secondary CTA: View integration (opens snippet modal)

Tertiary link: Read: Why certbot is technical debt

Hero "proof strip" (3 compact chips):

- No DNS API keys in CI/CD
- Built for 47-day renewal cycles
- Cloudflare edge execution

Hero visual (right side, schematic not fluffy):

A minimal architecture diagram showing:

- Your DNS Provider (keys stay in vault)
- CNAME delegation: _acme-challenge.<domain> → <tenant>.dcv.keylessssl.dev
- KeylessSSL edge (performs ACME DNS-01 proof)
- CA (ACME) → certificate issued

> The hero should feel like a clean RFC diagram, not a "startup illustration."

# Section B — Problem Statement (fear, but technical)

Title: Root Key Vulnerability

Body (direct, developer voice):

To automate wildcard certs today, teams drop high-privilege DNS API credentials onto build agents, app servers, or Kubernetes secrets. That's not "DevOps." That's a zone takeover waiting to happen.

Callout box (dark red border, monospace emphasis):

- If one node is compromised → your entire DNS zone is compromised.
- Wildcard automation should not require global DNS write access.

Bullet list (pain points):

- Certbot + DNS plugins = credentials sprawl (machines, repos, secret stores).
- Stateful ACME clients don't love stateless infra (containers, autoscaling).
- Manual cron renewals won't survive shorter cert lifetimes (47-day reality).

Small "Threat model" accordion (collapsed by default):

- What an attacker can do with root DNS keys
  - add malicious A/AAAA records
  - hijack MX, DKIM, SPF
  - intercept ACME challenges
  - persistent domain compromise

## Section C — Solution / Architecture (how it works)

Title: Delegated DCV, not delegated trust.

Three feature pillars (icon + heading + 2 lines max):

1. Air-Gapped Validation (shield icon)

   Only _acme-challenge is delegated. Root keys never touch KeylessSSL, your servers, or your CI runners.

2. 47-Day Renewal Readiness (refresh/calendar icon)

   Designed for high-frequency renewals without fragile cron glue. Treat certificates as ephemeral artifacts.

3. Cloudflare-Powered Reliability (cloud/network icon)

   Validation runs at the edge for low latency and predictable execution. “Enterprise-grade uptime without the enterprise price tag.”

Under the pillars: a compact “Flow” with 4 steps (numbered):

1. Add CNAME delegation
2. Verify domain
3. Issue/renew wildcard certs via API
4. Monitor status + rotate scoped tokens

## Section D — Integration (code-first, modal + inline)

Title: Add CNAME and go.

## Inline quick steps (visible in page)

Step 1 — Delegate validation

```
_acme-challenge.example.com  CNAME  example-com.<tenant>.dcv.keylessssl.dev
```

Step 2 — Issue via API

```
curl -s https://api.keylessssl.dev/v1/certs \
  -H "Authorization: Bearer $KEYLESSSSL_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "domain": "example.com",
    "wildcard": true,
    "provider": "letsencrypt"
  }'
```

Step 3 — Install (your way)

- Nginx / Caddy / Traefik / ALB / Cloudflare — you decide.
- KeylessSSL provides cert + chain (or push to your secret store via webhook).

## “View integration” Modal (high conversion element)

Modal tabs (top): cURL Terraform GitHub Actions Kubernetes

Each tab shows:

- a copy-to-clipboard code block
- minimal required env vars
- expected JSON response shape (status, next_renewal_at, cert_url)

Example: GitHub Actions tab

```
- name: Issue wildcard cert
  env:
    KEYLESSSSL_API_KEY: ${{ secrets.KEYLESSSSL_API_KEY }}
  run: |
    curl -s https://api.keylessssl.dev/v1/certs \
      -H "Authorization: Bearer $KEYLESSSSL_API_KEY" \
      -H "Content-Type: application/json" \
      -d '{"domain":"example.com","wildcard":true}' \
    | jq .
```

Modal footer microcopy:

Tokens are scoped. Keys are shown once. Rotate any time.

---

# Section E — Pricing (simple, blunt, undercut)

Title: Pricing that doesn’t punish automation.

## Two plan cards side-by-side (with “Most Popular” on Pro)

Hacker (Free)

- $0 / mo
- Up to 3 domains
- Wildcards included
- Delegated DCV
- Community queue
- Standard rate limits

CTA: Get API Key — Free

Pro

- $15 / mo
- Up to 50 domains

- Wildcards included
- Priority queue
- Higher rate limits
- Team access + audit events

CTA: Start Pro

Small note under pricing (edgy but factual to the positioning you provided):

Cheaper than BrandSSL’s $29/mo starter — without shipping your root keys.

FAQ (tight, technical):

- “Do you store my DNS provider credentials?” → No. We never ask for them.
- “What do I delegate?” → Only _acme-challenge via CNAME.
- “Can I use multiple DNS providers?” → Yes. Delegation is DNS-agnostic.
- “Do you support multi-tenant SaaS?” → Yes. Domain objects + scoped keys + automation hooks.

---

## Section F — Docs / Blog Teaser (thought leadership)

Title: Built by people who’ve been burned by cert automation.

Two link cards:

1. Docs: ACME + DNS-01 via Delegated DCV
   - “API reference, domain lifecycle, renewal semantics, rate limits.”
2. Blog: Why certbot is technical debt

    - “Stateful clients, credential sprawl, 47-day renewals, and why delegation wins.”

Optional small row of “developer trust signals”:

- Public status page
- Changelog
- Security policy
- Bug bounty (only if real; otherwise keep as “Security policy”)

---

## Section G — Final CTA + Footer

Final CTA block (high contrast):

Get an API key. Delegate once. Never ship DNS root keys again.

Button: Get API Key — Free for 3 Domains

Footer links:

Docs API Reference Security Status Contact Terms Privacy

Footer microcopy:

KeylessSSL is an ACME DCV automation layer. Your DNS root keys stay yours.

---

# 2) Visual system (developer aesthetic, dark-first)

### Theme

- Default: Dark mode (no toggle needed initially; optional later).
- Background: near-black with slight blue cast (security + depth).
- Surfaces: subtle elevation; no glassmorphism.
- Borders: thin, low-contrast; “infra UI” feel.

## Typography

- UI font: Inter / system sans for readability
- Code font: JetBrains Mono (or SF Mono fallback)
- Headings: tight tracking, strong weight, no playful curves.

## Color tokens (example)

- --bg: #070B12
- --surface: #0B1220
- --surface2: #0F1A2E
- --text: #E6EDF7
- --muted: #9FB0C7
- --border: rgba(255,255,255,0.08)
- Accent (secure green): --accent: #22C55E
- Warning (risk): --warn: #F97316
- Danger (root key vuln): --danger: #EF4444

## Iconography

- Line icons, consistent stroke, no gradients:
  - shield
  - key with slash
  - refresh/cycle
  - cloud/network nodes
  - terminal prompt

### Motion

- Minimal: hover + focus states, gentle section reveal only.
- Respect prefers-reduced-motion.

---

# 3) UX elements that matter to developers (and convert)

### Persistent conversion mechanics

- Sticky header CTA always present.
- Second CTA repeated after Problem, after Pricing, and in footer.

### Code blocks done right

- Copy button (with “Copied” toast).

- Language label (dns, bash, yaml).
- No giant code walls; show the smallest viable example.

## “Integration confidence” features

- “Expected response” snippet (status codes, JSON shape).
- Rate-limit info is visible (devs hate mystery throttling).
- Link to “Domain lifecycle” docs from the Integration section.

## Signup / API key flow (landing → product)

CTA opens a modal (or takes to /signup, but modal reduces drop-off).

Modal steps:

1. Create account (email + password) + optional GitHub OAuth
2. Generate API key
   - show key once
   - force copy action (“I’ve stored it”) checkbox to continue
   - microcopy: Stored as a hash. You can rotate anytime.
3. Add first domain
   - domain input
   - auto-generated CNAME record + copy
   - “Verify DNS” button with live polling + timeout messaging

# 4) Post-signup dashboard UI (minimal but “real infra”)

### Dashboard layout

Left nav

- Domains
- Certificates
- API Keys
- Webhooks
- Audit Log
- Usage / Rate Limits

### Domains table (default landing)

Columns:

- Domain
- Validation status (Pending CNAME / Verified)
- Cert status (Active / Renewing / Error)
- Next renewal
- Last issuance
- Actions (View / Rotate token / Re-check DNS)

Status chips should look like ops tooling:

- Green: Active
- Amber: Pending
- Red: Error
- Gray: Disabled

### API Keys screen

- Create scoped key:
  - scope by domain(s)
  - scope by actions (issue, renew, read-only)
  - optional expiration
- Keys shown once; after that: sk_live_•••••••••••
- “Rotate key” flow is 2 clicks.

### Audit Log (trust builder)

- key created / rotated
- domain added
- DNS verified
- cert issued / renewed
- webhook delivered / failed

You don’t need a massive dashboard. You need the 4 screens developers expect from a security automation tool: domains, keys, logs, limits.

---

# 5) The landing page copy (production-ready snippets)

### Hero CTAs

- Primary: Get API Key — Free for 3 Domains
- Secondary: View integration (CNAME → API → cert)
- Under CTA microcopy: No credit card. Keys shown once. Rotate anytime.

### Problem section line (anchor quote)

> "If one server is compromised, your entire DNS zone is gone."

### Architecture bullets (tight)

- Delegated DCV keeps root DNS keys off your infrastructure.
- Automation that survives 47-day certificate lifetimes.
- Edge execution for predictable validation latency.

### Pricing tagline

Undercut pricing. Overbuilt security.

---

# Vance Mercer's take (ruthless review + next moves)

Here’s what you’ve got: a landing page that actually talks like an infra product and doesn’t insult the reader. Good. But “good” doesn’t win. We win by making the risk feel non-optional and the integration feel inevitable.

My verdict: This structure is strong. The hero headline is a weapon. The integration modal is the conversion engine. Pricing is clean and punches incumbents in the mouth.

What I’d tighten immediately (no debate):

1. Make the threat unavoidable in 6 seconds.

   Add a 1-line red banner under the hero:

   Root DNS keys on servers = total zone compromise blast radius.

   Developers move when the blast radius is explicit.

2. Add a “Proof of minimal delegation” mini-diagram right in the Problem section.

   Show two boxes:

   - “Root keys (zone-wide write)” → Blocked / Never requested
   - “_acme-challenge delegation” → Allowed / required

     That’s the mental unlock.

3. Pricing should appear as a single line in the hero.

   Right under the CTA:

   Free: 3 domains • Pro: $15/mo for 50

   Don’t make them scroll to understand the deal.

4. Ship a public “Security” page fast.

   Not marketing. A real one: token handling, key storage semantics, scope model, audit logging, incident response contact. That page closes enterprise-minded CTOs without sales calls.

5. Make “Get API Key” frictionless and “Add Domain” addictive.

   The first domain should go green in under a minute, or we lose the moment. DNS polling, clear error states, and copy that tells them exactly what record is wrong.

Recommendation:

Launch exactly this one-pager + a dead-simple dashboard (Domains, Keys, Audit Log, Limits). No extra fluff. Then hammer content that attacks the status quo: certbot is technical debt, root key sprawl is negligence, delegation is the only sane model. That’s how we dominate the dev mindshare and convert on fear + precision.