High■Performance, Secure Web App Architecture — PokeAPI Integration

Overview

This document unifies modern best practices for a high■traffic web/mobile app that connects to an API, tailored to the

System Architecture

- Client: Web (SSR/ISR/CSR) and Mobile (offline■capable caches)
- Edge: CDN + WAF, TLS 1.3, HTTP/3, bot mgmt, rate limits, image optimization
- BFF: Backend■for■Frontend (same eTLD) with HttpOnly cookies (web) / short■lived tokens (mobile)
- API Gateway: AuthN/Z, schema validation, quotas, circuit breakers, idempotency keys
- Services: Modular monolith or microservices; mTLS; async workers for heavy jobs
- Data: Postgres (RLS), Redis (cache/queue), Object storage, search index
- Observability: OpenTelemetry traces/metrics/logs; dashboards with p50/p95/p99 latency

Security

- HttpOnly, Secure, SameSite cookies for web; WebAuthn/passkeys for login
- CSRF tokens; strict CSP with nonces; Trusted Types
- Input validation with JSON schemas; output encoding by context
- mTLS service■to■service; least privilege IAM; secrets in vaults
- RBAC/ABAC; DB RLS; transport/storage encryption
- Supply chain: SBOM, dep scan, container signing, provenance

API Performance

- Edge caching (Cache■Control, ETag, stale■while■revalidate)
- SSR/ISR streaming; code splitting, tree shaking
- Redis L2 caching (TTL tiers)
- Async jobs with queues; prefetch hot data
- Idempotency keys for writes; retries with backoff+jitter
- Read replicas; connection pooling; request hedging

PokeAPI Primer

• Base URL (REST v2): https://pokeapi.co/api/v2/
• Lists paginated (default 20 items) with limit & offset
• GraphQL beta: https://graphql.pokeapi.co/v1beta2 (POST only)
• No strict server■side rate limits, but caching encouraged
• Key resources: /pokemon/{id}, /pokemon-species/{id}, /evolution-chain/{id}, /type/{id}, /move/{id}

Integration Design

- BFF proxies all calls; never expose PokeAPI directly
- Redis caches hot endpoints (species, types, moves)
- TTL: 12–24h static; 1–6h lists; 5–30m search results
- Token bucket limiter per tenant + global; backoff on 429/5xx
- Normalize filters to PokeAPI params (limit/offset)

UX Wireframes & Interfaces

Discover (Desktop):

```
+-----------------------------------------+
| Search [____] [Type ■] [Gen ■] [Sort ■]  |
| Chips: Fire ✕ Water ✕                    |
+-----------------------------------------+
| Grid: Pikachu  | Charmander | Bulbasaur  |
|      ★ HP:35  |    HP:39   |   HP:45    |
|      ATK:55   |    ATK:52  |   ATK:49   |
+-----------------------------------------+
```

Footer: Data timestamp; attribution

Mobile (Bottom Sheet):

```
+--------------+
| Search bar   |
+--------------+
| Bottom sheet |
|  Pikachu     |
|  Charmander  |
+--------------+
```

Detail View:

```
+-----------------------------------------+
| Pikachu #25 [Electric] ■ [Share]         |
|  [Artwork]                               |
| Tabs: Stats | Abilities | Moves | Evol.  |
| Stats: radar chart, base values          |
| Evolutions: Pichu → Pikachu → Raichu     |
+-----------------------------------------+
```

Data & Caching Strategy

- Long■lived cache (12–24h): species, types, moves
- Medium (1–6h): lists, evolution chains
- Short (5–30m): search results
- Redis keys include tenant/user and filters
- Tag■based invalidation for projections
- Precompute hot projections (e.g., type matchups)

Deployment & Ops

- Stateless services on Kubernetes/ECS; autoscaling; multi■AZ
- Blue/green & canary deploys with feature flags
- DR tested; defined RPO/RTO; run game■days
- CI/CD: tests, scans, signed artifacts, SBOM
- Observability: OTel end■to■end; dashboards for latency/error/saturation
- SIEM for security monitoring

Checklist

✓ TLS 1.3 + HSTS; CDN/WAF + bot rules
✓ BFF with secure cookies / short■lived tokens
✓ API gateway with validation & circuit breakers
✓ Redis caching with TTLs; background prefetchers
✓ Observability wired end■to■end
✓ CI/CD with tests, SAST/DAST, signed images
✓ Accessibility & i18n verified
✓ DR plan validated

References

• PokeAPI REST docs: https://pokeapi.co/docs/v2/

• PokeAPI GraphQL beta: https://pokeapi.co/docs/graphql

• Fair use / caching: https://docs.airbyte.com/integrations/sources/pokeapi