API Best Practices

Managing APIs holistically across the enterprise
APIs are critical to evolving application architectures
Companies need to address these new challenges …

- **Insecure**: Inability to secure critical assets exposed beyond trust boundaries.
- **Silos**: Disparate implementations are creating new silos, limiting reuse.
- **Flying Blind**: Lack of insight into usage, performance, users, products, and anomalies.

Public Clouds

- Distributed apps
- Microservices

Private Cloud

- APIs
- App workloads
- Microservices

Platform-as-a-Service
... by managing APIs holistically across the enterprise
Design and build your APIs focusing on ease of use for app developers
Adopt an API-first, layered strategy for agility

- EXPERIENCE APIs
  - Highly consumable APIs, specific for set of devices or apps
    - /android/customer

- BUSINESS APIs
  - Consumable and reusable APIs, the core set of developer-friendly APIs
    - /customer

- EXPOSURE APIs
  - Highly reusable APIs, mimic underlying data models & resources
    - /crm/account
    - /support/customer

SYSTEMS OF RECORD
- CRM
- Financials
- Inventory
- HR
- Supply chain

©CC BY SA
Design APIs that are easy to consume for developers

- Easy-to-consume APIs
- Open API specifications
- Readable, intuitive URLs
- Good error handling
- Non-hierarchical URLs
- Data-centric modeling
- Relationships as links
- Simple JSON

Get detailed API design best practices in the Apigee Web API Design ebook.
Handle multiple northbound API versions effectively

**Moderate business logic in API tier**

- **MEDIATION**
  - Requests & responses processed to deliver as expected to both backend and clients

- **ROUTING**
  - Backend versions are transparent. Requests routed based on header, payload, user, etc.

- **ERROR HANDLING**
  - Returns errors gracefully for old API version requests

- **PASS THRU**
  - Forwards all incoming requests to the right backend version

**Backend target supports only one API version**

**Backend target supports multiple API versions**

©CC BY SA
Enforce a consistent set of security policies across all of your APIs
Protect against cyber threats with API management

**DATA SECURITY**
- Two-way TLS
- API key
- OAuth2

**THREAT PROTECTION**
- SQL threat protection
- JSON threat protection
- Regular expression protection
- IP-based restrictions
- Bot detection

**ACCESS CONTROL**
- OAuth2
- SAML
- API key verification
- IP access control
- Logging & auditing

**API MANAGEMENT**

---

Sensitive data exposure
Injection threats
Broken authentication & session mgmt attacks
Cross-site request forgery
Insecure direct object reference
Missing function-level access control
Cross-site scripting

Apps

Backend

©CC BY-SA
Handle volumetric attacks and business spikes

Percentage of companies using capability in the API tier*

- **78%** Spike arrest
- **68%** Rate limiting
- **74%** Caching

**OUT-OF-THE-BOX TRAFFIC MANAGEMENT IN API PLATFORMS**

- Spike arrest
  - Concurrent rate limit
  - Quota
- Response cache
  - Lookup cache
- Populate cache
  - Invalidate cache
  - Reset quota

* See "About this report"
Use AI-based solutions to secure against bots

API TRAFFIC DATA
Continuously monitor billions of API calls to identify anomalies

MITIGATION ACTIONS
Block, throttle, or honeypot bot traffic depending on your needs

AI MODELS & RULES
Continuously recognize bot patterns and create new rules

06
Don’t just rely on WAFs for API security

Advanced API platforms cover all API security needs. Eliminating WAF lowers latencies and provides consistent app security policies.

If your existing WAF is built into your CDN, this option might be the right approach (API platform sits behind the WAF/CDN).

In situations where applications can only be accessed through a WAF gateway, this might be the right approach.
Sync API lifecycle with your SDLC and automate testing and deployment of APIs
Align the API lifecycle with your SDLC and automate

Create similar number of API environments as your SDLC stages to simplify testing and automation.

To ensure separation of concerns between production and non-production APIs, use the concept of organization in API management to separate users, APIs, and API traffic.

Use mock API targets early in the API lifecycle, in line with API-first and test-driven development (TDD) principles.

With API management plugins to DevOps tools like Apache Grunt or Maven, automate your API lifecycle and fit into your SDLC.
Deploy APIs to the cloud based on workload

**Target app – microservices, REST APIs**

- **CLOUD**
  For internal use cases and modern apps, deploy APIs to the cloud (public or private cloud)

- **PUBLIC CLOUD**
  As microservices are in the cloud and given external use, deploy your APIs in the public cloud for scale and cost

**Target app – legacy interfaces, monolith**

- **COLLOCATE**
  For legacy target apps focused on internal use cases, collocate your APIs with the application

- **CENTRALIZED**
  Given external use, deploy APIs to the centralized API management for ease of operations

For external use cases (partner, consumer-focused):
- Deploy APIs to the centralized API management for ease of operations.
- Given external use, deploy APIs in the public cloud for scale and cost.

For internal use cases and modern apps, deploy APIs to the cloud (public or private cloud).
Publish easy-to-use APIs with interactive documentation and self-service capabilities
Package APIs for easy developer consumption

- API /store
  - Location
  - traffic
  - inventory

- API /catalog
  - Item id
  - Promotions
  - ratings

- Product 1
  - Price: $0.0005 per call
  - Quota: 10 requests per sec

- Product 2
  - Price: FREE
  - Quota: 100 requests per sec

- Provide differentiated access to APIs and resources to various user groups
- Quickly try out new API-based business models (e.g., revenue share, API pricing)
Publish automated, interactive documentation

Sandbox environment to test APIs

Interact with the API and see the request and response formats

Create rich documentation directly from your Open API specifications
Automate onboarding for scalable developer adoption

**Broad adoption**

To drive broader adoption of public APIs and for internal API usage, set up your developer portal with automated onboarding.

In this mode, developers sign up, register their app, get app keys, and get started, without any portal admin approvals.

**Managed adoption**

To engage strategic partners with a set of private APIs, you need to simplify partner developer onboarding, while retaining access control.

In this mode, app developers register on their own, but you have an admin approval step. Upon approval, the developer can register apps and get keys and access to docs.

**Controlled access**

This is uncommon, but in certain, highly sensitive cases, organizations use this mode to restrict access to APIs.

In these use cases, the portal administrators sign up app developers. Although you have control, you lose scalability due to manual onboarding.
Use analytics to gain better insights into your API usage and performance.
Enable API developers to optimize API functionality

Run step by step through an API request to get timing information for each stage as it flows through the API proxy.
Equip ops teams to monitor performance and availability

**TRAFFIC**
Operations teams need to monitor API traffic volumes, understand traffic patterns over time and identify anomalies.

**AVAILABILITY**
Be the first to know if your API error rates increase beyond a particular threshold—and then rectify any issues.

**LATENCY**
Ensure your APIs are meeting target response time SLAs that have been set with your API consumers.
Measure API program success with the right metrics

**Product manager**

**USAGE**
As an API product owner, always know the most popular APIs, API traffic volumes, and usage patterns.

**ADOPTION**
Identify the most active developers and apps consuming your APIs, from traffic volume and breadth of APIs.

**BUSINESS METRICS**
Track business metrics associated with your APIs (API-related fees, revenue share etc.).
Empower app developers with data

App developers have visibility into availability of APIs they're consuming.

Identify if the errors are due to API policies, API infrastructure, or target backends.

App developers can track the error rates of APIs they consume over time.
Automate API operations by integrating API management into your enterprise infrastructure
Deploy your API platform in cloud, based on needs

<table>
<thead>
<tr>
<th>API OPERATIONS</th>
<th>PUBLIC CLOUD</th>
<th>HYBRID</th>
<th>PRIVATE CLOUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to success</td>
<td>★★★</td>
<td>★★</td>
<td>★</td>
</tr>
<tr>
<td>Total cost of ownership</td>
<td>★★★</td>
<td>★★</td>
<td>★★★</td>
</tr>
<tr>
<td>Security / compliance</td>
<td>★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>Performance</td>
<td>★★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>Scale / reliability</td>
<td>★★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
</tbody>
</table>

- Public cloud satisfies most customers’ needs, except for specific situations.
- For apples-to-apples comparison, use a three-year period and include all operations costs.
- Public cloud satisfies most use cases (exception: purely internal use cases).
- Leading API cloud providers can provide significant scale with auto scaling & reliability.
- With skilled people and infrastructure ready to go, private cloud is still a viable option.
Integrate API platform with existing monitoring infrastructure

Log monitoring
Use built-in message logging policies in API platform to generate logs and use logging tools like Splunk

Runtime data
Collect runtime statistics like response time and error rates using JMX MBeans and access them using any JMX-compliant APM tool

Analytics
Gain visibility into a variety of usage (developer usage, API traffic) and performance data with out-of-the-box API analytics provided by API platform

Component monitoring
Monitor availability of infrastructure components (CPU, memory, thread statistics) with management APIs and existing monitoring tools

API monitoring
Monitor performance and conduct stress testing of APIs and target systems with health check capability of API management platform
Automate scaling your API platform infrastructure

Actual API traffic

Auto scale runtime and database Infrastructure

Infrastructure manually provisioned for peak
SPEED MATTERS - API best practices are only one part

- **Adopt modern software practices**
  Agile, scrum, test-driven development

- **Execute First, Align Later**
  Focus on business impact with your initial API projects and address broader alignment after initial successes

- **Fail-fast approach**
  Get the set of APIs out FAST—you never know if and how they will be consumed by developers
About Apigee

Apigee® powers the APIs that make every business a digital business. Apigee provides a leading API platform that helps companies—from disruptive start-ups to the Fortune 100—rapidly adapt to the business and technology requirements of the connected, digital world.

Many of the world’s largest organizations select Apigee to enable their digital business, including over 30 percent of the Fortune 100, four of the top five Global 2000 retail companies, and five of the top 10 global telecommunications companies.

For more information, visit apigee.com.

About this report

As a leader in API management, Apigee is privileged to work with hundreds of customers to develop and manage a large number of APIs. By reflecting on our experiences and those of our customers and the industry at large, we have gained some insights into which API design innovations are bringing real benefits and becoming notable trends. This report aims to capture some of the best practices in API design that we have seen emerge in the past couple of years.

Data in this report are based on hundreds of billions of API calls made through the Apigee cloud and distributed across Apigee’s global network of datacenters from customers spanning 20 industry sectors. None of these data are from companies that deploy our technology in their private cloud.

If you have questions regarding the report, please email info@apigee.com or tweet @apigee.