Twilio Integration for Multi-Tenant Applications

Separate (Individual) Accounts vs Subaccounts: Procedures, Pros and Cons, and Business Impact

Prepared for: Product, Engineering and Finance stakeholders. Note: Twilio limits, fees and program names change over time. Verify figures marked "(verify)" against Twilio's current documentation and pricing page before committing.

1. Purpose and Summary

We are building a tenant-based (multi-tenant) application where each customer (tenant) needs messaging, voice, verification or other Twilio capabilities. Twilio offers two main ways to isolate tenants:

1. **Individual (separate) accounts**: every tenant owns a fully independent Twilio account, with its own login, billing and compliance profile.
2. **Subaccounts**: the main company holds one parent Twilio account and creates one subaccount per tenant under it.

Recommendation in one line: For most SaaS products where the main company owns the customer relationship and bills tenants, **subaccounts under one parent account** are the better fit technically and the more profitable model. Individual accounts make sense only when a tenant must own its Twilio relationship, data and billing directly (enterprise, regulated or white-label-with-ownership cases).

2. Key Concepts

- **Parent (main) account:** The master account owned by the main company. It can create, suspend, close and report on subaccounts.
- **Subaccount:** A child account with its own Account SID, Auth Token, phone numbers, API keys, usage and logs. Resources in one subaccount are isolated from others.
- **Consolidated billing:** Subaccount usage rolls up into the parent account's invoice. Tenants do not receive Twilio invoices.
- **API keys:** Scoped credentials that can be created per subaccount so that application servers never need to use the main Auth Token.

- **Usage Records / Usage Triggers:** APIs to read per-subaccount consumption and fire alerts at thresholds. These are the basis for metering and tenant billing.
- **Tenant:** One customer organization of our application.

3. Option A: Subaccount per Tenant

3.1 How it works

The application keeps a mapping of `tenant_id` to the tenant's subaccount SID. All Twilio calls for that tenant are made in the context of its subaccount.

3.2 Procedure (step by step)

1. **Set up the parent account.** Create the main Twilio account, upgrade it from trial, enable billing, set up auto-recharge and enable two-factor authentication for all console users.
2. **Plan the structure.** Decide naming, for example `tenant-<tenant_id>-<slug>`, and which products each tenant will use (SMS, voice, WhatsApp, Verify, etc.).
3. **Automate subaccount creation at tenant onboarding.** Call the Accounts API (`POST /2010-04-01/Accounts.json` with a `FriendlyName`) from the backend when a tenant signs up.
4. **Create scoped API keys for the subaccount.** Generate an API Key and Secret for that subaccount and store them encrypted (secrets manager or KMS). Do not store or use the Auth Token in application code.
5. **Provision resources.** Buy or port phone numbers, create Messaging Services, set up Verify services and configure webhooks into the subaccount.
6. **Configure webhooks with tenant routing.** Point each subaccount's numbers to your webhook URL and include the tenant identifier in the URL path or in a parameter, then **validate the** `X-Twilio-Signature` on every request using the correct subaccount credentials.
7. **Handle compliance registration.** Complete the required regulatory registrations for each tenant where applicable (for example A2P 10DLC brand and campaign registration for US SMS, toll-free verification, and local regulatory bundles for numbers in some countries). Registration is usually tied to the tenant's business identity, so collect the tenant's business details at onboarding.

8. **Meter usage.** Pull per-subaccount usage through the Usage Records API on a schedule, add your margin, and invoice the tenant through your own billing system. Set Usage Triggers to alert on abnormal spending.
9. **Set guardrails.** Apply per-tenant sending limits in your own application, monitor for spam or fraud, and be ready to suspend a subaccount quickly.
10. **Offboard.** Suspend, then close the subaccount, release numbers, and archive the usage data needed for audits.

3.3 Pros

- **One contract, one invoice, one set of credentials to manage at the top level.** Finance reconciles a single Twilio bill.
- **Fast, fully automated onboarding.** Creating a subaccount is an API call, so tenants are live in minutes.
- **Resource and data isolation.** Numbers, logs, API keys and usage are separated per tenant.
- **Accurate per-tenant cost tracking.** Usage APIs report cost per subaccount, which makes margin calculation exact.
- **Central control.** The main company can suspend or close any tenant instantly, which is valuable for abuse control.
- **Volume pricing leverage.** All tenant traffic aggregates into one account, so committed-use or volume discounts are negotiated once and apply to total usage (verify with Twilio sales).
- **Strong pricing power.** The main company sets the tenant price, so the spread between Twilio cost and tenant price is retained as revenue.
- **Simple operations.** One integration pattern, one monitoring setup, one place for support to look up a tenant's traffic.

3.4 Cons

- **Shared risk at the parent level.** If one tenant sends spam or violates policy, Twilio may take action on the whole account, which can affect all tenants. Strong abuse controls are required.
- **The main company carries the financial exposure.** Tenant non-payment, fraud-driven charges and carrier fines are billed to the parent account.
- **Compliance burden sits with us.** We must collect, verify and submit tenant business information for regulatory registration, and delays affect tenant go-live.

- **Limits apply.** There is a cap on subaccounts per account (default is commonly cited at 1,000, verify) and shared rate or throughput limits can affect noisy-neighbor scenarios. Raising limits requires contacting Twilio.
- **Tenant has no direct Twilio relationship.** Enterprise tenants who want their own contract, data residency choices or direct support may object.
- **Credential concentration.** The parent Auth Token is powerful. A leak exposes every tenant, so key hygiene is critical.

4. Option B: Individual (Separate) Twilio Account per Tenant

4.1 How it works

Each tenant creates (or is helped to create) its own Twilio account. The tenant grants our application access, typically by sharing API credentials, or through an authorization or partner-style connection if available for your program (verify current options such as Twilio's ISV or partner programs).

4.2 Procedure (step by step)

1. **Guide the tenant to create an account.** Provide onboarding instructions or a deep link to Twilio sign-up. The tenant upgrades from trial and adds a payment method.
2. **Tenant completes verification and compliance.** The tenant registers its own business profile, brand and campaigns, as it is the direct Twilio customer.
3. **Credential or authorization exchange.** The tenant creates an API Key (or authorizes our app) and provides the Account SID, API Key SID and Secret through a secure form in our app. Never ask for the primary Auth Token.
4. **Encrypt and store credentials per tenant** with strong access controls and rotation reminders.
5. **Provision resources inside the tenant's account**, or have the tenant buy numbers and configure Messaging Services, then point webhooks to our endpoints with the tenant identifier included.
6. **Validate webhooks** using the signature and the tenant's own credentials.
7. **Monitor credential health.** Detect revoked keys, suspended accounts or exhausted balances, and notify the tenant.
8. **Billing.** Twilio bills the tenant directly. Our company bills only for its own software subscription.

9. **Offboard.** The tenant revokes our keys. Our system deletes stored credentials and webhooks.

4.3 Pros

- **Maximum isolation.** An abusive tenant can only harm its own account. There is no risk to other tenants or the parent.
- **No financial exposure** for Twilio usage, fraud or carrier fines. The tenant pays Twilio directly.
- **Compliance ownership sits with the tenant,** who is the legal sender. This reduces our regulatory liability.
- **Suits enterprise and regulated customers** who require their own contract, audit trail, region or data controls.
- **No account-count ceiling to manage** on our side.
- **Tenants can use Twilio features and negotiated rates** independently.

4.4 Cons

- **Slow, friction-heavy onboarding.** Tenants must sign up, verify and register, which causes drop-off and support load.
- **Credential management risk.** We store many third-party credentials that tenants can revoke or rotate without notice.
- **Fragmented billing and visibility.** We cannot easily see cost, balance or usage across tenants, so margin on messaging is not available.
- **Support complexity.** Troubleshooting requires tenant cooperation and access to their console.
- **No central control.** We cannot suspend a tenant's messaging at the Twilio level, only stop sending from our app.
- **No aggregated volume discount** for the main company, since traffic is split across many accounts.
- **Inconsistent tenant setups** (different configurations, plans or mistakes) increase engineering and support effort.

5. Side-by-Side Comparison

Criteria	Subaccounts (Option A)	Separate Accounts (Option B)
Onboarding speed	Instant, API-driven	Slow, tenant-driven
Billing	Single consolidated invoice to us	Each tenant billed by Twilio
Revenue on usage	Yes, we set tenant pricing and keep the margin	No usage margin
Cost visibility	Full, per tenant	Limited
Isolation	Resource-level isolation, shared account risk	Full isolation
Abuse risk to other tenants	Present, needs controls	None
Financial exposure to us	High (unpaid or fraudulent usage)	None
Compliance responsibility	Mostly ours	Tenant's
Central control and kill-switch	Yes	No
Volume discount potential	High	Low
Credential management	Our own keys, simple	Many external credentials, complex
Enterprise and white-label ownership	Weaker	Stronger
Scale limits	Subaccount caps (raisable)	None on our side
Support effort	Lower	Higher

6. Why Subaccounts Are Usually Suitable for Tenant-Based Applications

- **Fits the multi-tenant model.** A tenant is a logical unit, and a subaccount is a native Twilio unit that maps one-to-one with it.
- **Self-service at scale.** Automated provisioning removes manual steps from the signup funnel.
- **Built for isolation plus consolidation.** Each tenant gets separate numbers, logs and keys, yet the company keeps one billing and governance point.
- **Operational control.** Usage, spend and traffic per tenant can be monitored and limited in one place.
- **Better tenant experience.** Tenants never need to understand Twilio, create accounts or manage compliance paperwork alone.

7. Which Option Profits the Main Company More?

Subaccounts are more profitable in most cases. Reasons:

1. **Usage margin.** The company buys messages, minutes and numbers at its Twilio rate and resells them to tenants at a higher rate (or bundles credits into plans). With separate accounts, this revenue line does not exist at all.
2. **Volume discounts.** Aggregated traffic can qualify for committed-use or volume pricing, increasing the margin on every unit.
3. **Higher conversion and retention.** Instant onboarding raises signup completion, and tenants are less likely to leave a product that bundles communications.
4. **Upsell opportunities.** Tiered plans, number rentals, premium sender registration help, and priority support can be priced on top of usage.
5. **Lower support and engineering cost.** One standard setup means fewer tickets and less custom work.
6. **Data for pricing decisions.** Accurate per-tenant cost data lets Finance set prices that protect margin.

Illustrative example (hypothetical numbers): If Twilio charges us $0.008 per SMS segment and we charge tenants $0.012, our gross margin is $0.004 per segment. At 5 million segments per month, that is about $20,000 per month in margin, before fees and registration charges. Under separate accounts, the same traffic generates $0 usage margin.

Cost side to account for: carrier surcharges, number rental, registration fees, bad debt from unpaid tenants, fraud losses, and the engineering cost of metering and billing. Profit stays positive only with prepaid credits or deposits, spending limits and fraud controls.

When separate accounts win instead: if the company's strategy is to sell software only (no communications revenue), serves highly regulated or enterprise tenants who must own their sender identity, or wants to avoid all Twilio-related liability.

8. Recommended Hybrid Approach

- **Default:** Subaccount per tenant for standard and self-service customers.
- **Exception path:** Allow "bring your own Twilio account" for enterprise tenants. The app stores that tenant's credentials and uses them instead of a subaccount, behind the same internal interface.
- **Design principle:** Build a `TwilioClientProvider` abstraction that returns the right credentials for a `tenant_id`, so the rest of the application does not care which model the tenant uses.

9. Risk Controls and Best Practices

- **Prepaid wallet or credit model** for tenants, with auto-pause at zero balance.
- **Per-tenant rate limits and daily caps** enforced in our application.
- **Content and opt-out handling:** honor STOP keywords, store consent records, and screen message content for abuse.
- **Credential security:** secrets manager, rotation schedule, least-privilege API keys, no Auth Tokens in code or logs.
- **Webhook security:** signature validation, HTTPS only, idempotent handlers.
- **Monitoring and alerts:** Usage Triggers, spend anomaly detection, error-rate dashboards per tenant.
- **Compliance data:** keep tenant business details, brand and campaign IDs, and consent logs auditable.
- **Contract terms:** include an acceptable-use policy in tenant agreements and the right to suspend immediately.
- **Capacity planning:** track subaccount count against limits and talk to Twilio early about increases or enterprise features.

10. Implementation Checklist

#	Task	Owner
1	Decide default model and exception policy	Product, Finance
2	Confirm Twilio limits, pricing and discounts with Twilio	Finance, Engineering
3	Build tenant-to-subaccount mapping and provisioning service	Engineering
4	Implement secure credential storage and rotation	Engineering, Security
5	Build usage metering and tenant billing integration	Engineering, Finance
6	Create compliance onboarding flow (brand, campaign, bundles)	Product, Legal
7	Add abuse controls, rate limits and suspend workflow	Engineering, Trust and Safety
8	Write acceptable-use terms and tenant pricing	Legal, Finance
9	Pilot with a few tenants, then roll out	Product

11. Conclusion

For a tenant-based application, **subaccounts** give the best balance of automation, isolation, central control and revenue. They let the main company earn margin on communications usage, negotiate better volume pricing and deliver a smoother tenant experience, at the cost of carrying financial and compliance risk that must be managed with prepaid billing and abuse controls. **Separate accounts** are the safer but lower-revenue and higher-friction choice, best reserved for enterprise or regulated tenants through a "bring your own account" option.

12. Price per Message for Each Case

Pricing basis: public Twilio list prices and fee schedules reviewed in November 2026. Several figures come from third-party summaries of Twilio's pricing page, and Twilio states that prices may change without notice. Before sending this to the client, confirm each

figure against Twilio's official SMS pricing page and its downloadable pricing CSV (verify).

12.1 Twilio list cost per message (the same under both options)

Twilio's per-message price does not change because of the account structure. What changes is **who pays Twilio** and **who earns the margin**.

Message type	Twilio price per segment	Notes
US long-code SMS, outbound	About $0.0083	Billed per segment (160 characters for plain GSM-7 text), not per message
US long-code SMS, inbound	About $0.0083	Inbound is also charged
US carrier fee (example: AT&T)	About +$0.0035	Fee varies by carrier and destination. An AT&T outbound message is therefore about $0.0118 per segment all-in
US MMS, outbound	About $0.022	Carrier fees are higher for MMS (for example about $0.009 on AT&T)
US MMS, inbound	About $0.0165	
US toll-free SMS	About $0.0083	Plus toll-free carrier surcharges
Failed message processing fee	$0.001	Charged only on messages that end in "Failed" status
WhatsApp (Twilio fee)	About $0.005	Added on top of Meta's per-template fee, for example about $0.025 for US marketing templates and about $0.0118 for India marketing templates
India SMS	Verify on Twilio's India pricing page	Third-party sources show conflicting figures. India traffic also needs DLT registration under TRAI rules

Volume discounts apply automatically after roughly 150,000 messages per month for a given number type, and larger committed-use discounts can be negotiated with Twilio (verify).

12.2 Case A: Subaccounts under the main company's account

Item	Per message (illustrative, US SMS via AT&T)
Main company's cost from Twilio	$0.0118 (list, before any discount)
Suggested resale price to tenant	$0.0160 (assumption, set by Finance)
Gross margin to main company	$0.0042 per segment (about 26% of the resale price)
Who pays Twilio	Main company (consolidated invoice)
Who pays the main company	Tenant, through prepaid credits or monthly invoice

Example at 100,000 segments per month per tenant: cost $1,180, tenant charge $1,600, margin about $420. Across 100 such tenants this is about $42,000 per month of gross margin before number rental, registration costs, bad debt and operating costs.

12.3 Case B: Separate Twilio account per tenant

Item	Per message (illustrative, US SMS via AT&T)
Tenant's cost from Twilio	$0.0118 (the tenant's own rate, which may include its own discounts)
Main company's message revenue	$0.0000
Optional platform fee charged by the main company	For example $0.0010 to $0.0020 per message, as a software usage fee (assumption)
Who pays Twilio	Tenant, directly

Example at 100,000 segments per month per tenant: message margin $0. With an optional $0.002 platform fee the main company earns about $200 per tenant per month, and tenants may resist a fee on top of their own Twilio bill.

12.4 Registration and compliance fees (US A2P 10DLC)

Fee	Amount	Who pays in Case A	Who pays in Case B
Brand registration, sole proprietor or low-volume standard	$4.50 one-time	Main company, usually passed to tenant	Tenant
Brand registration, standard (includes secondary vetting)	$46 one-time	Main company, usually passed to tenant	Tenant
Campaign vetting	$15 one-time per campaign	Main company, usually passed to tenant	Tenant
Campaign monthly fee	$1.50 to $10 per campaign, by use case (sole proprietor $2, low-volume mixed $1.50, standard $10, charity $3, emergency services $5)	Main company, usually passed to tenant	Tenant
Phone number rental	Varies by country and number type (verify)	Main company, passed to tenant with margin	Tenant

In Case A these fees are a recurring cost that the main company either absorbs or includes in the tenant's plan price, so they should be built into the pricing model.

12.5 Which case earns more per message

Measure	Case A (Subaccounts)	Case B (Separate accounts)
Margin per US SMS segment	About $0.0042 (illustrative)	$0 (or a small optional platform fee)
Margin at 100,000 segments per month per tenant	About $420	$0 to about $200
Exposure to unpaid usage	Yes	No

Case A earns more per message, and the margin grows further if Twilio volume discounts lower the main company's cost while tenant prices stay fixed.

13. Missing Information Required Before Submission to the Main Company Client

The items below are not yet in this document and are normally needed for the client to approve the integration approach. Items marked *Input needed* require information from the client or internal teams.

#	Missing item	Why it is needed	Source
1	**Scope of channels** (SMS, MMS, voice, WhatsApp, Verify/OTP, email)	Pricing and compliance differ for each channel	Input needed from client
2	**Target countries**, especially India and the US	India requires DLT/TRAI registration (entity, header and template registration). The US requires A2P 10DLC. Each country changes cost and timeline	Input needed from client
3	**Expected volumes**: number of tenants, messages per tenant per month, peak messages per second	Needed to calculate cost, margin, throughput and whether a discount or higher limits are needed	Input needed from client

#	Missing item	Why it is needed	Source
4	**Message types per tenant** (transactional, OTP, marketing)	Determines campaign type, monthly campaign fee and opt-in rules	Input needed from client
5	**Tenant pricing and billing model**: resale price per message, prepaid wallet or postpaid, minimum top-up, invoice currency and taxes (for example GST in India)	Required to confirm that margin is real after fees and tax	Finance
6	**Who bears registration and number fees**: client or tenant	Changes the cost model in Section 12.4	Finance, client
7	**Twilio commercial terms**: negotiated rates, committed-use discounts, subaccount limit, throughput limits, support plan cost	Official quote to replace the public list prices used here	Twilio sales
8	**Twilio partner or ISV program eligibility**	May affect how brand registration, billing and tenant ownership work in Case B and hybrid setups	Twilio sales
9	**Tenant ownership requirement**: do any tenants need their own Twilio account, contract or data location	Decides whether the "bring your own account" path is needed	Client
10	**Legal and policy documents**: terms of service, acceptable use policy, consent and opt-out policy, data processing agreement	Needed because the main company is liable for tenant content under Case A	Legal
11	**Data protection and retention**: personal data handling, message log retention, regional requirements (for example India's DPDP Act, GDPR if EU tenants)	Needed for compliance sign-off	Legal, Security

#	Missing item	Why it is needed	Source
12	**Security design**: secret storage, key rotation, access control, audit logging, webhook validation	Required by most client security reviews	Engineering, Security
13	**Architecture diagram** showing tenant mapping, provisioning flow, webhooks and billing data flow	Clients usually require a visual before approval	Engineering
14	**Implementation plan**: effort estimate, team, timeline, milestones and dependencies such as registration approval time	Required for budget and schedule approval	Engineering, Product
15	**Total cost of ownership**: development cost, Twilio costs, registration fees, support and monitoring	Gives the client a single investment figure	Finance, Engineering
16	**Risk register**: abuse, account suspension, carrier filtering, bad debt, outage, with owners and mitigations	Shows risks are understood and controlled	Product, Engineering
17	**Service levels and support model**: uptime expectations, incident handling, who contacts Twilio	Sets expectations for operations	Client, Engineering
18	**Success metrics**: delivery rate, onboarding time, margin per message, cost per tenant	Allows the client to judge the result after launch	Product
19	**Assumptions and exclusions**: which prices are list prices, which items are estimates, what is out of scope	Prevents disputes later	Product
20	**Approvals**: named decision makers and sign-off section	Needed to close the submission	Client

Information that would immediately strengthen the submission

If only a few items can be collected quickly, the most valuable are **items 2, 3, 5 and 7**: countries, volumes, the tenant pricing model, and an official Twilio quote. With these four the cost and profit figures in Section 12 can be replaced with client-specific numbers instead of list-price illustrations.