

Goal:

Build agent marketplace from where anyone from anywhere in the world can integrate agents in their web applications or websites.

Agent interpretation is Logic + Memory + Tools + Reasoning.

Integrations options:

1. Widget integration (UI+API) → available in **v0**
2. Public API integration (Only API) → available in **v1**
3. Plugin for various platforms → Later
4. SDK like ShadCN's components for React/[Next.js](#) ecosystem → Later
5. Mobile SDKs → Later

Widget integration is implemented using steps mentioned below:

1. Users write script tags in their website code.

```
<script
  src="https://cdn.agentengine.com/widget.js"
  agentId="A456"
  domainKey="sk_domain..."
  ...
>
</script>
```

2. Users' browser sends requests to Agent Engine's CDN and downloads the JavaScript.
3. Based on the domain-key, it authenticates the request if it is a valid request then it allows access to an agent otherwise throws 403.
4. Widget script hosting
 - a. Vercel (in public/ directory in [Next.js](#) project) → For **v0**
 - b. Cloudflare or CloudFront → For **v1**
5. The length of the domain-key is 2048 bits/256 bytes.
6. Notes:
 - a. User is able to regenerate the domain-key from the Agent Engine web application.

- b. For session and conversation management, **visitor_id** and **conversation_id** are sent through payload. When a visitor visits a website then 32 lengths of **uuids** are generated on the **client side** and **server side** respectively and stored into **localstorage** of the visitor's browser.
- c. **Context injection** is the core design pattern of this architecture.
- d. **Prompt structure**: Summary of old messages + Recent conversation window + Current input + System instructions
- e. Billing limitations
 - i. Messages per month
 - ii. Tokens per month
 - iii. Max concurrent conversations
 - iv. **After hitting plan limit, return 429 or 402**
- f. Technical burst limits (to stop spam/abuse/bugs)
 - i. # requests per second/time-duration of seconds
 - ii. # requests per minute per IP.
- g. **Origin** and **Referrer** header will be used to know the domain and the page respectively from which request is coming.
- h. Each user is able to connect each project to all listed agents.
 - i. The domain-key is generated per project per user.
- j. A single user may have more than one project in the agent engine.
- k. Subscription plan is applied to **per project; NOT PER USER**.

Flow:

- 1. For conversational agent (RAG)
 - a. Browser → Widget script → [Next.js](#) API router → Python worker → LLM provider
- 2. For lead qualification agent
 - a. Browser webhook → Widget script → [Next.js](#) API router (**Conversation as context if it is available for same user**) → Queue → Python worker → LLM provider
- 3. For instant-follow up agent
 - a. Lead qualification worker scores lead "HOT" → [Next.js](#) API router (**Conversation as context if it is available for same user**) → Queue → Python worker → LLM provider

Tech-stack:

Component	Tool	Remark
Frontend	Next.js	–
Backend	Next.js API routes + Supabase	–
Database	Supabase PostgreSQL	–
Auth	Supabase Auth	–
Vector store (RAG)	Supabase pgvector	For storing vectors of uploaded docs
Storage	Supabase Storage	–
Sync & Async workers	Python	For all agent; microservice architecture
Event queue	Upstash Redis	For lead-scoring and follow-up agent
Rate limiting	Upstash Redis	–
Frontend deployment	Vercel	–
CDN for Widget	Vercel	Serves widget.js to all customer sites.
SMS	Twilio	–
WhatsApp	Twilio	–
Email	Twilio SendGrid	–
LLM Provider	OpenRouter	–

Costing:

Tool	Pricing	Resources
Supabase	\$30 per month	1. 100k monthly active user 2. 8 GB disk size per project 3. 250 GB egress 4. 250 GB cached egress 5. 100 GB file storage 6. Daily backups stored for 7 days 7. 7-day log retention 8. CPU: 2-core ARM 9. Memory: 2 GB 10. \$10 of included usage credit 11. All types of Auth, SQL Database, Vector Database, Storage, Edge functions; everything is available on Supabase
Upstash Redis	Pay As You Go	1. Request price: \$0.20 per 100k requests 2. Bandwidth price: First 200 GB free, then \$0.03 per GB 3. Storage price: \$0.25 per GB
Vercel	\$20 per month + additional usage	1. \$20 of included usage credit 2. Cold start prevention
Twilio SMS	\$20 per month	1. \$0.0832 per SMS 2. 240 SMS per month 3. Note: Charges are for from International number to Indian number
Twilio WhatsApp	\$20 per month	1. \$0.005 per message 2. 4000 messages per month
Twilio SendGrid	\$20 per month	1. 50k emails per month
OpenRouter	Pay As You Go	1. Platform fees: 5.5% 2. 300+ models

Agent Engine system design

		3. 60+ providers 4. Prompt caching 5. Data policy-based routing
Dev-tools	\$60 per month for Cursor	1. Additional usage will be charged accordingly.
Total	\$170 per month + Pay As You Go charges	