LetsBe Biz — Foundation Document

Master Strategy, Architecture and Decisions Log

Version:	v1.1
Date:	February 26, 2026
Company:	LetsBe Solutions LLC
Contact:	matt@letsbe.solutions

221 North Broad Street, Suite 3A, Middletown, DE 19709

Confidential — For authorized recipients only

Contents

1. LetsBe Biz — Foundation Document

Date: February 25, 2026 **Authors:** Matt (Founder), Claude (Architecture & Strategy) **Status:** Version 1.1 **Companion Documents:** Technical Architecture v1.1, Product Vision v1.0, Pricing Model v2.2

1.1 1. Who We Are

LetsBe Biz is a privacy-first AI workforce platform. We give every small business their own private team of AI employees that run the business while the owner focuses on what they're actually good at.

Not AI-assisted tools. Not a chatbot. Not a workflow builder. A team of AI agents that operate 28+ business tools autonomously — reading data, sending campaigns, managing customer conversations, scheduling meetings, processing invoices, publishing content, handling IT operations — all on a server the business owns, with infrastructure secrets that never leave the machine.

Our background is enterprise infrastructure. We're channeling that into a productized offering where the AI does the work and humans steer the direction.

Tagline: "Where power meets privacy." **Logo:** Current branding (logo_long.png, logo_square.jpg) — no redesign planned.

1.2 2. The Problem We Solve

Small business owners are drowning. They're running their entire operation across 10-30 SaaS tools — each with its own subscription, login, data silo, and terms of service. They pay €500-2,000/month in SaaS subscriptions, often can't afford a VA or IT person, and spend 60% of their time on admin instead of revenue-generating work.

Even those who've consolidated onto self-hosted tools still face a core problem: someone has to operate them. Configure the CRM. Send the newsletter. Manage the calendar. Process the invoices. Handle the IT issues. That requires either expensive human labor (€1,500-3,000/mo for a part-time VA) or deep technical knowledge most owners don't have.

The current landscape forces an impossible choice: powerful but fragmented (SaaS), private but complex (self-hosted), or AI-powered but not private (cloud AI). No one offers all three.

What we replace: - 10-30 SaaS subscriptions (€500-2,000/mo) - A part-time virtual assistant (€1,500-3,000/mo) - Occasional IT contractor help (€100-200/hr) - All with better privacy, better consistency, and 24/7 availability

---

1.3 3. What We're Building

1.3.1 3.1 Infrastructure Layer

Every customer gets their own isolated VPS with a full suite of open-source business tools deployed via Docker Compose, fronted by Nginx, and provisioned automatically.

Current tool stack (28 applications):

Category	Tools
Cloud & Files	Nextcloud, MinIO
Communication	Stalwart Mail (email), Chatwoot (customer chat), Listmonk (newsletters)
Project Management	Cal.com, NocoDB
Development	Gitea, Drone CI, Portainer
Automation	Activepieces, n8n
CMS & Marketing	Ghost, WordPress, Squidex
Business & ERP	Odoo
Analytics	Umami, Redash
Design	Penpot
Security	Keycloak (SSO/IAM), VaultWarden (passwords)
Monitoring	Uptime Kuma, GlitchTip, Diun
Documents	Documenso (e-signatures)
Chat & AI	LibreChat

See Tool Catalog v2.2 for full details, licensing, and 27 expansion candidates. Removed since v1.0: Poste.io (→ Stalwart Mail), Windmill (managed-service license prohibition), Typebot (retained for internal use only), Twenty CRM, Akaunting, Budibase, Invoice Ninja.

3.1.1 Tool Selection Model Users don't pick from a raw list of 30. The flow is:

1. **Business type selection** → Pre-configured default bundle (e.g., "Freelancer," "Agency," "E-commerce," "Consulting")
2. **Customization screen** → Full tool catalog with defaults pre-checked. Toggle switches to add/remove.
3. **Live resource calculator** → As tools change, required CPU/RAM/storage updates in real-time.
4. **Server tier auto-selection** → Only tiers that meet or exceed the resource requirement are shown. Users cannot select an underpowered server. The cheapest visible option IS the right option.

Example bundles (defaults, all customizable):

Bundle	Default Tools	~Resource Estimate
Freelancer	Nextcloud, Stalwart Mail, Cal.com, Ghost, Odoo-lite, Keycloak, VaultWarden	4 vCPU, 8 GB RAM
Agency	Above + Chatwoot, Listmonk, NocoDB, Penpot	8 vCPU, 16 GB RAM
E-commerce	Freelancer + Odoo-full, Chatwoot, Umami, Documenso, Redash	8 vCPU, 16 GB RAM
Power User	Full 28-tool stack	16 vCPU, 32 GB RAM

3.1.2 Server Sizing Server tiers are dynamically gated by tool selection. Each tool has a known resource footprint. The live calculator shows minimum tier requirements and users cannot downgrade below calculated minimums.

Server tiers (Netcup RS G12, primary):

Tier	Specs	Netcup Plan	Cost to LetsBe	Use Case
Lite (hidden)	4 cores, 8 GB DDR5, 256 GB NVMe	RS 1000 G12	~€8.74/mo	Price-sensitive, 5-8 tools
Build (default)	8 cores, 16 GB DDR5, 512 GB NVMe	RS 2000 G12	~€14.58/mo	Small business, 10-15 tools
Scale	12 cores, 32 GB DDR5, 1 TB NVMe	RS 4000 G12	~€27.08/mo	Agency/e-commerce, 15-30 tools
Enterprise	16 cores, 64 GB DDR5, 2 TB NVMe	RS 8000 G12	~€58.00/mo	Full 28-tool stack

Hetzner Cloud CCX (backup/overflow):

Tier	Specs	Hetzner Plan	Cost to LetsBe
Starter	2 vCPU, 8 GB RAM, 80 GB NVMe	CCX13	~€12.49/mo
Growth	4 vCPU, 16 GB RAM, 160 GB NVMe	CCX23	~€24.49/mo
Scale	8 vCPU, 32 GB RAM, 240 GB NVMe	CCX33	~€48.49/mo

3.1.3 VPS Provider Strategy **Primary: Netcup RS G12.** Best price-to-performance in Europe. AMD EPYC 9645 (Zen 5), DDR5 ECC, NVMe, 2.5 Gbps networking. Provisioning automated via Ansible. Additional advantage: **domain reselling** through Netcup's reseller program (Level A free, 450+ TLDs) — customers can buy domains directly through us.

Dual-region support: Netcup offers data centers in both **Nuremberg, Germany** (EU) and **Manassas, Virginia** (US). Customers choose their region at signup. EU customers default to Germany; North American customers default to Virginia for lower latency. Pricing varies by approximately ±€1-2/mo depending on tier and region. Same RS G12 hardware in both locations.

Provisioning model: Maintain a pre-provisioned pool of servers at 12-month contract rates in both regions. When a customer signs up → assign from pool (matching their region) → provision tools → ready in minutes. Pool size managed dynamically based on signup velocity per region.

Backup/Overflow: Hetzner Cloud. Full Cloud API enables instant on-demand provisioning when Netcup pool is full. Hourly billing means no waste on overflow capacity. Acts as safety valve for signup surges.

Note: Hetzner prices rising 30-37% April 1, 2026. Netcup 12-month contracts are locked and significantly cheaper per core. Architecture is provider-agnostic — Ansible works on any Debian VPS regardless of host.

1.3.2 3.2 The AI Workforce

This is the product. Not "AI-managed infrastructure" — an **AI workforce** that operates the business tools on behalf of the user.

The AI runtime (**OpenClaw**) runs on the customer's VPS and connects to LLMs via OpenRouter. All config and history lives locally. We do not fork or modify upstream — OpenClaw is treated as a dependency. All LetsBe-specific logic (secrets redaction, command gating, Hub communication, tool adapters) lives in a separate **Safety Wrapper** layer that can pull upstream updates without conflicts.

3.2.1 What the AI Workforce Does Every deployed tool exposes APIs. The AI agents have full access to those APIs and execute autonomously within guardrails:

- **Marketing Agent** pulls last month's top blog posts from Ghost, composes a newsletter, and sends it through Listmonk — without being asked. Or on command: "Send the monthly newsletter with our best content."
- **Secretary Agent** receives a meeting request via Stalwart Mail, checks availability on Cal.com, sends a confirmation with a Zoom link, and adds a reminder. Handles all administrative correspondence.
- **Sales Agent** monitors Chatwoot for new leads, qualifies them based on conversation patterns, creates a quote in Odoo, and routes hot leads to the human for approval.

- **IT Agent** detects that Nextcloud storage hit 80%, identifies and removes old temp files, resizes if needed, and reports what it did. Reads Nginx configs, checks cert validity, restarts failed containers. Full sysadmin capability.
- **Custom agents** — Users create their own agents for domain-specific workflows: content planning, analytics reporting, customer segmentation, inventory management, compliance checking, anything the team needs.

The platform is deliberately open-ended. Users discover new capabilities organically — "can you do this?" → "oh shit, it can." Each user builds a unique, personalized workflow system they become attached to. That's the moat: not the tools (anyone can install Nextcloud), not the AI (anyone can use ChatGPT), but the *configured, trained, personalized AI team* that knows how *their* business works.

3.2.2 Agent Architecture Default agents pre-configured per business type:

Agent	Role	Tool Access	Example Workflows
Dispatcher	Message router & coordinator	Inter-agent messaging	Routes user requests, breaks complex tasks into ordered steps, morning briefing
IT Admin	Infrastructure & security	Portainer, Uptime Kuma, Shell, Docker	Auto-fix container crashes, rotate certs, resize storage, security checks
Marketing	Content & campaigns	Ghost, Listmonk, Umami, Penpot, WordPress	Draft & send newsletters, schedule posts, analyze performance
Secretary	Communications & scheduling	Cal.com, Stalwart Mail, Nextcloud, NocoDB	Manage calendar, handle email, organize files, send confirmations
Sales	Leads & revenue	Chatwoot, Odoo, Documenso	Qualify leads, create quotes, send contracts

Users can add/remove/customize agents anytime. Unlimited agents — no hardcoded limits. Each agent is configured via: - **SOUL.md** — Personality, domain knowledge, behavioral rules, brand voice - **Tool permissions** — Which APIs this agent can access, what operations it can perform - **Model selection** — (Advanced mode) Choose different LLMs per agent

3.2.3 Tool API Adapter Strategy This is the critical engineering investment that creates the competitive barrier.

Each business tool has APIs. Tool API adapters turn those APIs into tools that OpenClaw agents can invoke. 24+ adapters covering the tool stack:

Tool	API Type	Key Operations
Nextcloud	WebDAV + OCS REST	Files, shares, users, calendar, contacts
Chatwoot	REST	Conversations, contacts, labels, assignments
Odoo	XML-RPC + JSON-RPC	Invoices, quotes, contacts, inventory
Ghost	Content + Admin REST	Posts, pages, tags, members, newsletters
Cal.com	REST	Events, bookings, availability, teams
Stalwart Mail	REST + SMTP/IMAP/JMAP	Send/receive email, manage accounts
Portainer	REST	Containers, stacks, volumes, networks
Umami	REST	Page views, events, referrers, reports
Listmonk	REST	Campaigns, subscribers, templates
NocoDB	REST	Tables, records, views, webhooks
…	…	(18 additional adapters)

These adapters are built via a common framework (auth handling, error patterns, rate limiting, response formatting) then parallelized. Each adapter is isolated — Nextcloud's doesn't depend on Chatwoot's. Integration depth is the deepest moat: months of compounding engineering work per tool, tested against real tool versions with real edge cases.

1.3.3 3.3 Secrets Firewall & Safety Architecture

Highest security priority. Enforced at four independent layers:

3.3.1 How It Works The AI sees *everything* on the server — full configs, compose files, error logs, cert expiry — **except literal secret values**. Passwords, API keys, SSL private keys, and tokens are replaced with placeholders before reaching the LLM.

Layer 1 — Secrets Registry: All generated credentials (50+ per tenant) are logged in an encrypted local registry (SQLite) with key names, patterns, and locations. When credentials rotate, the registry updates.

Layer 2 — Outbound Redaction: Before any text leaves the VPS to the LLM, a middleware layer checks all outbound text against the registry. Known secrets are replaced with deterministic placeholders: `[REDACTED:postgres_password]`, `[REDACTED:nextcloud_admin_ke`
The AI can reason about which credentials are relevant without seeing values.

Layer 3 — Pattern Safety Net: Regex patterns catch secrets the registry might have missed: private key blocks, JWT tokens, bcrypt hashes, connection strings with credentials, high-entropy base64 blobs, common env var patterns.

Layer 4 — Function-Call Proxy: When the AI needs to *use* a secret (e.g., restart a service that needs a DB password), it doesn't receive the credential:

```
execute_with_credential("restart_postgres", credential_id="pg_main")
```

The Safety Wrapper injects the real value locally and executes. The AI gets the outcome but never sees the credential value.

Privacy messaging: “Your infrastructure credentials and secrets never leave your server. Your AI team manages tools through secure local commands — it never receives your passwords, keys, or certificates.”

3.3.2 Command Gating (Five-Tier) Every tool call is classified and gated based on autonomy level:

Green — Non-destructive (auto-execute at all levels): - `file_read`, `env_read` — Read files, logs, configs (output redacted) - `container_stats` — List/inspect containers - `query_select` — Database SELECT queries only - `check_status`, `dns_lookup`, `cert_check` — Infrastructure health checks

Yellow — Modifying (auto-execute at Level 2+, gated at Level 1): - `container_restart` — Restart services - `file_write`, `env_update` — Modify files and configs - `nginx_reload` — Reload web server - `chatwoot_assign`, `calcom_create` — Internal business operations

Yellow+External — External-facing (gated by default at all levels until user unlocks per agent/tool): - `ghost_publish` — Publish blog content visible to public - `listmonk_send` — Send email campaigns to subscribers - `poste_send` — Send emails to external recipients - `chatwoot_reply_external` — Reply to customer conversations - `social_post` — Post to social media - `documenso_send` — Send documents for external signature

External communications are gated independently of autonomy levels. A misworded email to a client or a prematurely published blog post damages the business’s reputation. Users must build trust with their AI team before allowing autonomous external-facing actions.

Red — Destructive (auto-execute at Level 3, gated at Level 1-2): - `file_delete`, `container_remove`, `volume_delete` — Delete resources - `user_revoke`, `db_drop_table`, `backup_delete` — Revoke access, drop data

Critical Red — Irreversible (always gated at all levels): - `db_drop_database`, `firewall_modify`, `ssh_config_modify` — Infrastructure-critical - `backup_wipe_all`, `user_delete_account`, `ssl_revoke` — Unrecoverable

---

1.3.4 3.4 AI Autonomy Levels

Customers control how much the AI can do without approval:

Level	Name	Auto-Execute	Requires Approval	Use Case
1	Training Wheels	Green only	Yellow + Red + Critical Red	New customers, cautious users, onboarding

Level	Name	Auto-Execute	Requires Approval	Use Case
2	Trusted Assistant (default)	Green + Yellow	Red + Critical Red	Established trust, daily operations
3	Full Autonomy	Green + Yellow + Red	Critical Red only	Power users, experienced teams

Invariants (all levels): - Secrets always redacted - Audit trail always logged - AI never sees raw credentials - External comms gated until explicitly unlocked per agent/tool - Destructive actions always gated

Each agent can have its own autonomy level independent of the tenant default (e.g., IT Admin at Level 3, Secretary at Level 1).

1.3.5 3.5 Dynamic Tool Installation

One of the most powerful capabilities in the platform.

A user says “I need a wiki” and the IT Agent can deploy BookStack or WikiJS from a curated catalog, configure it behind nginx with SSL, seed credentials in the secrets registry, and report back — all gated behind user approval.

How it works:

1. User requests a tool: “Can you set up a wiki for my team?”
2. IT Agent consults the **Tool Catalog** — a curated registry of pre-tested open-source tools with Docker Compose templates, nginx configs, and resource requirements
3. IT Agent presents: “I recommend BookStack — 256MB RAM required, you have 4GB free. Want me to install it?”
4. **User approves** (Red-tier operation — always gated)
5. IT Agent executes: deploys stack, configures nginx, generates credentials, stores in secrets registry, runs health check
6. IT Agent reports: “BookStack is live at wiki.yourdomain.com. [credentials via app]”

Tools are deployed only from the curated catalog — the IT Agent cannot deploy arbitrary Docker images. Resource checks prevent server overload. All deployments are audited and reversible.

1.3.6 3.6 Mobile App

React Native (iOS + Android). Primary client interface.

Core features: - Chat with agent selection ("Talk to your Marketing Agent") - Morning briefing from Dispatcher Agent (what happened overnight, what needs attention) - Team management (agent config, model selection, autonomy levels) - Command gating approvals (push notifications with one-tap approve/deny) - Server health overview (storage, uptime, active tools) - Usage dashboard (token consumption, activity) - External comms gate management (unlock sending per agent/tool)

Access channels: App-only at launch. WhatsApp/Telegram as fallback channels ready at launch with security disclaimer. Hub acts as relay/proxy (JWT auth, WebSocket) — no exposed VPS ports.

1.3.7 3.7 Hub (Central Platform)

Next.js admin dashboard and API. Production-ready infrastructure for customer management, billing, provisioning, monitoring.

Current capabilities: Customer/order management, Netcup SCP integration, Stripe billing, 2FA, DNS verification, Docker provisioning, enterprise monitoring.

New capabilities (this version): - Customer portal API (agent config, usage tracking, command approvals) - Token metering and overage billing - Agent management API (SOUL.md, TOOLS.md, permissions) - Safety Wrapper communication endpoints (heartbeat, registration, config sync) - Command approval queue (Yellow/Red commands surface here) - Token usage analytics dashboard - Founding member program tracking

All tenant servers communicate with Hub via the Safety Wrapper. The Safety Wrapper handles registration, heartbeat, telemetry, config sync, and approval request routing.

1.3.8 3.8 Website (letsbe.biz)

Separate from Hub. AI-powered onboarding flow:

Step	Details
1	Landing page with chat input: "Describe your business."
2	AI conversation (Gemini Flash) — 1-2 messages, constrained to business type classification
3	Tool recommendation — Pre-selected bundle for detected business type, full catalog visible

Step	Details
4	Customization — Add/remove tools, live resource calculator
5	Server selection — Only tiers meeting minimum requirement shown
6	Domain setup — User brings domain or buys one (Netcup reselling)
7	Subagent config — Optional. Template-based per business type
8	Payment — Stripe. Pay first, then provision
9	Provisioning status — Real-time progress. Email with credentials. App download links

Interactive demo (held loosely): Single shared VPS with fake business data ("Bella's Bakery"). Prospects chat with AI, watch it operate tools in real-time. Not a video — hands-on experience. One VPS (~€25/mo), session timeouts, rate limiting.

1.4 4. Business Model

1.4.1 4.1 Pricing Structure

Single subscription that scales with server resources. All 28 tools included. Unlimited agents.

Tier	Price	VPS Tier	Target	Monthly Cost to LetsBe
Lite (hidden)	€29/mo	4c/8GB	Price-sensitive, 5-8 tools	~€12.51
Build (default)	€45/mo	8c/16GB	Small business, 10-15 tools	~€22.36
Scale	€75/mo	12c/32GB	Agencies, power users	~€37.96
Enterprise	€109/mo	16c/64GB	Full 28-tool stack	~€60.05

Gross margins: 45-57% depending on tier (at full token pool consumption; actual margins higher as most users won't exhaust pools). Lite hidden to avoid anchoring; Build/Scale/Enterprise marketed.

Annual discount: 15% for upfront annual commitment. Aligns with 12-month Netcup contract pricing.

AI model tiers:

- **Included (base subscription):** 5-6 cost-efficient models (DeepSeek V3.2, GPT 5 Nano, GPT 5.2 Mini, GLM 5, MiniMax M2.5, Gemini Flash — final selection pending) with generous monthly token pools. Cover 90%+ of daily usage. No credit card needed beyond subscription.

- **Premium (credit card required):** Top-tier models (Gemini 3.1 Pro, GPT 5.2, Claude Sonnet, Claude Opus) available at per-usage metered rates with sliding markup: 25% on cheap models → 8% on expensive models. Lower markup on expensive models encourages adoption.

- **Founding members:** 2× included token allotment for 12 months ("Double the AI"). First 50-100 customers. Extra cost ~€3-25/mo depending on tier. All tiers remain margin-positive at 2× (Lite 47%, Build 35%, Scale 31%, Enterprise 22%). ~€134/user/year effective CAC.

Model selection UX: - **Basic Settings (no credit card):** Three presets — "Basic Tasks," "Balanced," "Complex Tasks." Non-technical users never see model names. - **Advanced Settings (credit card required):** Full model catalog. Per-agent model selection. Premium models metered to card.

Token pool: Included tokens are monthly budget across all agents. Pool only covers the 5 included models. Premium models always metered separately — never draw from pool. When pool runs out, usage pauses or user opts into overage billing at tiered markup.

Prompt caching: SOUL.md and TOOLS.md structured as cacheable prompt prefixes. Cache read prices are 80-99% cheaper than standard input — direct margin multiplier.

> See companion document: **LetsBe_Biz_Pricing_Model.md** (v2.2) for full cost analysis, revenue projections, and unit economics.

1.4.2 4.2 Target Customer

Horizontal with vertical templates. Not building "LetsBe for restaurants" — building "LetsBe for businesses" with a restaurant template that pre-selects the right tools.

Lead persona: Solo founders and freelancers (Sarah). Solo founder who is drowning in 60-hour weeks doing admin, can't afford staff for marketing/IT/scheduling/invoicing, uses 10-12 SaaS tools costing €800/mo. Sees the demo, realizes LetsBe replaces €800/mo in SaaS + 20 hours/week of admin. *"It runs my business."*

Secondary: Small agency owners (David). Managing client work across 15 different tools. Needs operational leverage. Each client gets their own LetsBe instance.

Tertiary: Privacy-conscious businesses (Dr. Weber). Healthcare, legal, finance in regulated markets. Data sovereignty is non-negotiable. GDPR-compliant on infrastructure they control.

1.4.3 4.3 The Moat

The competitive moat builds in layers:

Layer 1 — Integration depth: 24+ tool API adapters with cross-tool workflows, error recovery, edge-case handling. Months of compounding engineering work. Each adapter tested against real tool versions with real data — not something you can shortcut.

Layer 2 — Speed to market: Being first with a working product. Every week in market is a week of real user feedback, bug fixes, refinement that a competitor starting from zero doesn't have.

Layer 3 — User accumulated context: Each user's SOUL.md configurations, agent memories, workflow patterns, brand voice training, client knowledge, operational preferences make their instance uniquely valuable. This isn't data you can export. It's months of accumulated learning the AI team has absorbed through daily use. Switching costs are enormous.

Integration depth creates the initial barrier. Speed to market exploits it. User accumulated context makes it permanent.

1.5 5. Competitive Landscape

1.5.1 5.1 Market Position

Nobody combines privacy-first infrastructure + pre-deployed business tools + autonomous AI agents + secrets firewall + cross-tool workflows.

The market breaks into quadrants:

	SaaS (cloud)	**Self-hosted (private)**
Workflow automation	n8n Cloud, Make, Zapier	n8n, Dify, Flowise
AI workforce (operates tools)	OpenAI Frontier, YC startups	**LetsBe (alone here)**

LetsBe occupies an empty quadrant.

1.5.2 5.2 Competitor Analysis

Competitor	What They Do	How We Differ
Cloudron / YunoHost	Self-hosted app management	No AI, no cross-tool workflows
Coolify	Self-hosted PaaS	Developer tool, not business operations
Traditional SaaS	Cloud business tools	No privacy, no AI workforce, fragmented
OpenClaw hosting	Managed AI agent hosting	Commodity hosting. No business tools, no secrets firewall
Virtual assistants (human)	Manual business operations	Expensive, limited hours, inconsistent, doesn't scale

Competitor	What They Do	How We Differ
n8n / Make	Workflow automation	Users build flows manually. No pre-deployed tools, no AI execution
OpenAI Frontier	Enterprise AI agents	Enterprise-only, SaaS, expensive, not privacy-first

---

1.6 6. Where We Are Today

1.6.1 6.1 Architectural Foundation

The Technical Architecture v1.1 document defines the complete system:

- **OpenClaw** (upstream dependency, not fork) runs the AI agents on each customer's VPS
- **Safety Wrapper** (Node.js) provides secrets redaction, command gating, Hub communication
- **Tool adapters** (24+ adapters) expose business tool APIs to agents
- **Local storage** (SQLite) for all on-server state — no per-tenant database
- **Total LetsBe overhead:** ~640MB RAM per tenant (down from ~1.5GB+ in earlier designs)

Key architectural decisions: - OpenClaw treated as upstream dependency, not a fork (AD #1) - Safety Wrapper is Node.js, not Python (AD #11) - Orchestrator and Sysadmin Agent deprecated — capabilities absorbed into OpenClaw + Safety Wrapper (AD #3, #4) - MCP Browser deprecated — replaced by OpenClaw native browser tool (AD #14) - Five-tier command gating (Green/Yellow/Yellow+External/Red/Critical Red) with external comms gate independent of autonomy levels (AD #30) - Dispatcher Agent is first-class default component (AD #31) - Dynamic tool installation from curated catalog (AD #32) - Two-tier model strategy: 3 included presets + premium pay-as-you-go (AD #33) - Threshold-based sliding markup (AD #35) - Founding member 2× token program (AD #34)

1.6.2 6.2 Component Status

Component	Status	Role
Hub (Next.js)	Functional	Central control plane, customer portal, billing, provisioning, monitoring

Component	Status	Role
Provisioner (Bash)	Functional	One-shot server provisioning via Ansible
OpenClaw	Ready (upstream)	On-server AI agent runtime
Safety Wrapper	Ready to build	Secrets redaction, command gating, Hub communication
Tool adapters	Ready to parallelize	24+ tool API adapters
Mobile app (React Native)	Ready to build	Primary client interface
Secrets registry	Ready to build	Encrypted SQLite vault for credentials
Autonomy level system	Ready to build	Per-agent, per-tenant gating configuration
External comms gate	Ready to build	Independent unlock mechanism per agent/tool

1.6.3 6.3 What Needs Build (Critical Path)

Tier 1 — Must build before launch:

1. Safety Wrapper (secrets redaction, command classification, Hub communication)
2. Tool API adapters (24+ adapters, parallelizable)
3. Mobile app (React Native, iOS + Android)
4. Secrets registry (SQLite, encrypted)
5. Autonomy level system (per-agent gating, approval queue)
6. External comms gate (unlock mechanism)
7. Hub updates (customer portal API, token metering, agent management)
8. Command approval queue (Yellow/Red commands surface via app)
9. New letsbe.biz website + onboarding flow
10. Prompt caching architecture (SOUL.md + TOOLS.md as cacheable prefixes)

Tier 2 — Important but can follow launch:

1. Analytics dashboard (agent activity, token usage, cost tracking)
2. Telegram/WhatsApp fallback channel adapters
3. Direct API fallback (Anthropic, Google, DeepSeek) for OpenRouter outages
4. Interactive demo sandbox (“Bella’s Bakery”)

Tier 3 — Roadmap (v2+):

1. Data migration from Google Workspace / M365 (IMAP, CalDAV, WebDAV)
2. Workflow template marketplace
3. White-label / agency multi-tenant mode
4. User-created custom tool adapters
5. Community skills marketplace

1.7 7. Go-to-Market Strategy

1.7.1 7.1 Launch: Founding Member Program

Target 50-100 founding members in first 6 months.

Why founding members: - Direct feedback on core product from real users - Real usage data to optimize infrastructure, AI behavior, pricing - Community evangelists who become reference customers - Exclusive positioning creates urgency and prestige

Founding member benefits: - 2× included AI token allotment for 12 months ("Double the AI") - Direct access to founder for product feedback - Early influence on product direction - Public acknowledgment as early adopter (if desired) - Lifetime 10% discount on future upgrades (optional)

1.7.2 7.2 Channels

- **Content marketing:** Blog posts on privacy-first AI, comparisons with SaaS stacks, tutorials on autonomous AI for SMBs. SEO play for long-term organic discovery.
- **Self-hosted communities:** Reddit (r/selfhosted, r/homelab, r/smallbusiness), Hacker News, privacy forums. These audiences already value self-hosting.
- **Social media:** Short videos showing "oh shit" moments — AI sending a newsletter, scheduling a meeting, fixing a server issue in 60 seconds. Target self-hosted communities, solo founder forums.
- **Google Ads:** Targeted keywords — "self-hosted business tools," "AI business assistant," "private business software," "alternative to [SaaS tools]." Low volume but high intent.
- **Interactive demo:** Hands-on sandbox (Bella's Bakery) where prospects chat with AI, watch it operate real tools in real-time.
- **Network:** Early introductions via advisory network, founder communities, privacy advocates.

1.7.3 7.3 Growth Strategy (Year 2+)

Horizontal with vertical depth. Build best-in-class experience for solo founders first. Secondary verticals emerge naturally from usage patterns. Expand upmarket to 50-200 person teams with multi-department AI workforces, advanced RBAC, dedicated support.

1.8 8. Three-Year Vision

1.8.1 8.1 Year 1: Prove the Model

Launch with founding member program. 50-100 customers using the full product. Validate the core value proposition: that an AI workforce on private infrastructure genuinely saves time, unlocks capabilities, and replaces costs.

Success metrics: - Founding members measurably get 10+ hours/week back - Multiple SaaS subscriptions cancelled per customer - Retention rate above 90% after 3 months - NPS > 50

1.8.2 8.2 Year 2: Scale and Deepen

Hundreds of customers. Self-service signup to AI team ready in under 30 minutes. Deep vertical templates for top-performing business types. Mobile app polished and feature-complete.

New capabilities: Data migration tools, more messaging channels, community-contributed agent skills, white-label option for agencies.

Success metrics: Self-serve pipeline working, month-over-month growth, positive unit economics including AI token costs.

1.8.3 8.3 Year 3: Platform

LetsBe becomes the operating system for small businesses. Marketplace of tools, skills, templates creates network effects. Supports third-party tool integrations (user-created adapters), opening ecosystem beyond core 28 tools.

Expansion paths (choose based on traction): - **Vertical depth:** Specialized compliance and tooling for regulated industries (healthcare, legal, finance) - **Upmarket:** Larger teams (50-200 employees) with multi-department AI workforces - **Geographic:** Multi-region infrastructure (beyond EU) - **Partner channel:** MSPs and IT consultancies reselling LetsBe to their client base

Success metrics: Platform effects visible, community contributing templates and skills, third-party integrations being adopted.

1.9 9. Technical Architecture Summary

For complete technical details, see LetsBe_Biz_Technical_Architecture.md v1.1.

1.9.1 9.1 System Overview

```
[Mobile App] ↔ [Hub (Relay/Proxy)] ↔ [Tenant VPS]
                                          ↓
                                     [OpenClaw
                                       Gateway]
                                          ↓
```

```
[Safety Wrapper]
  Secrets Redaction
  Command Gating
  Tool Adapters
  Hub Communication
      ↓
[Business Tools
 (30 Docker
 services)]
```

1.9.2 9.2 Key Principles

1. **Secrets never leave the server.** All credential redaction happens locally before any data reaches an LLM. Enforced at transport layer, not by trusting the AI.
2. **The AI acts autonomously within guardrails.** Non-destructive operations execute immediately. Destructive operations require human approval. The boundary is enforced by code the AI cannot modify.
3. **OpenClaw stays vanilla.** No fork. All LetsBe-specific logic lives in the Safety Wrapper. Clean separation enables pulling upstream updates without conflicts.
4. **Four layers of defense.** Security is not one wall — it's four independent layers (Sandbox → Tool Policy → Command Gating → Secrets Redaction), each enforced separately, each unable to be bypassed.

1.9.3 9.3 Per-Tenant Infrastructure

Each customer VPS runs:

Component	Language	RAM	Role
OpenClaw Gateway	Node.js 22+	~512MB	AI agent runtime, conversation management
Safety Wrapper	Node.js	~64MB	Secrets redaction, command gating, Hub comms
28+ tool containers	Various	Varies	Nextcloud, Chatwoot, Ghost, Odoo, etc.
Nginx reverse proxy	-	~64MB	HTTPS, routing, rate limiting

Component	Language	RAM	Role
Total LetsBe overhead	-	**~640MB**	(Down from ~1.5GB+ in earlier designs)

1.9.4 9.4 API Layer

Tool adapters expose 200+ operations across 24+ business tools. OpenClaw agents invoke adapters via standardized function-call interface. Secrets Wrapper injects credentials and enforces gating. Results returned sanitized and redacted.

1.10 10. Decisions Log (Critical Decisions)

#	Decision	Rationale
1	OpenClaw as upstream dependency, not fork	MIT license allows divergence; clean separation enables pulling updates
3	Orchestrator deprecated	Capabilities absorbed by OpenClaw + Safety Wrapper
4	Sysadmin Agent deprecated	Capabilities ported as Safety Wrapper tools; process separation doesn't add meaningful security
5	No per-tenant PostgreSQL — SQLite for all on-server state	Saves ~256MB RAM per tenant
11	Safety Wrapper is Node.js, not Python	Consistency with OpenClaw (Node.js), lighter footprint
12	Hybrid plugin model for tool adapters	Adapters registered as OpenClaw plugins; secrets redaction in separate process
13	OpenClaw hooks for security integration	`message:received`, `tool_result_persist`, `gateway:startup`, `agent:bootstrap`
14	MCP Browser deprecated — replaced by OpenClaw native browser	Native browser saves ~256MB RAM, more capable (CDP + Playwright)
17	Three autonomy levels (Training Wheels, Trusted Assistant, Full Autonomy)	Pragmatic balance between safety and user autonomy
18	One customer = one VPS, permanently	Simplifies infrastructure, security isolation, customization
22	Four-layer access control (Sandbox, Tool Policy, Command Gating, Secrets Redaction)	Defense in depth; each layer independent

#	Decision	Rationale
25	OpenAI-compatible API locked down, not exposed	Internal only; prevents external API access
26	Web search/fetch via OpenClaw native tools	Simpler than sidecar service
29	User customization enabled (SOUL.md modification, custom agents, custom skills)	Users are experts in their domain; enable depth
30	External Communications Gate independent of autonomy levels	Product principle: misworded email worse than delayed newsletter
31	Dispatcher Agent is first-class default component	Primary user contact point; routes intent; coordinates workflows
32	Dynamic tool installation from curated catalog	Powerful user capability; safety maintained via catalog, resource checks, gating
33	Two-tier model strategy: 3 included presets + premium pay-as-you-go	Simple UX (Basic mode) for non-technical users; power users unlock Advanced
34	Founding member 2× token program (“Double the AI”)	Marketing lever; 12 months for first 50-100 customers; all tiers margin-positive
35	Threshold-based sliding markup (25% cheap → 8% expensive)	Fairness; don’t penalize power users; still profitable on cheap models
38	Netcup primary, Hetzner overflow	Best price-per-core in Europe; domain reselling; provider-agnostic architecture
39	Infrastructure provider positioning — LetsBe hosts, customer owns license	LetsBe is an infrastructure and AI orchestration provider, not a software vendor. Every open-source tool runs under its upstream license on the customer’s dedicated server. Customers have full SSH access and all credentials. If a customer wants enterprise features (e.g., Stalwart Enterprise, Rocket.Chat Enterprise), they purchase the license directly from the vendor — we help deploy it. This framing is legally protective (AGPL/open-core compliant), competitively differentiating (transparency as trust), and reinforces the “your server, your data” message.

#	Decision	Rationale
40	Public open-source tools page on website	Dedicated page listing every deployed tool with name, role, upstream link, and license type. Reinforces infrastructure-provider positioning, earns open-source community goodwill, generates backlink SEO value, and differentiates against opaque SaaS bundlers. Tool Catalog v2.2 is the source of truth.
41	BYOK (Bring Your Own API Key) deferred to post-launch	Architect the AI orchestration layer for provider-agnostic key injection from day one, but don't ship BYOK at launch. Rationale: early-stage support burden (misconfigured keys, rate limits, model compatibility issues) outweighs community goodwill gains. Launch BYOK as a Pro/Developer tier feature once the managed experience is stable and support load is predictable. BYOK users pay the same platform fee (margin is higher since we don't eat API costs). Protects core margins while keeping the door open for power users and self-hosting community.

See Technical Architecture v1.1 for complete decision log (45+ decisions on technical specifics).

1.11 11. Open Questions (Product/UX Only)

Technical questions (Architecture Design Decisions) are resolved in Technical Architecture v1.1. Remaining product questions:

1. **Interactive demo UX spec** — Fake business data set, session management, rate limiting, abuse prevention. Decision held loosely — implementation can proceed without this.
2. **Agent personality customization depth** — How much guidance vs. freeform SOUL.md editing in Basic mode? User research needed post-launch.

1.12 12. Document Lineage

Version	Date	Key Changes
0.1–0.7	Feb 24-25, 2026	Foundation Document evolution: problem definition, tool stack, AI workforce vision, subagent architecture, tool API layer, secrets firewall, command gating, pricing, competitive landscape. 63 cumulative decisions.
1.0	Feb 25, 2026	**Complete rewrite.** Synthesized from Technical Architecture v1.1 (45+ technical decisions) and Product Vision v1.0 (customer journey, principles, vision validation). Removed outdated references (Python/FastAPI/Orchestrator/Sysadmin/MCP Browser). Updated to reflect: OpenClaw as upstream dependency, Node.js Safety Wrapper, five-tier command gating, external comms gate, Dispatcher Agent, dynamic tool installation, two-tier model strategy, sliding markup, founding member program. Preserved all business decisions, pricing model, VPS strategy, tool selection, competitive analysis, moat definition. Restructured for clarity: Who We Are, Problem, What We're Building (6 subsections), Business Model, Competitive Landscape, Where We Are Today, Technical Summary, Go-to-Market, Three-Year Vision, Decisions Log.
1.1	Feb 26, 2026	**Tool stack + strategic updates.** Updated tool stack from 30 → 28 tools (Stalwart Mail replaces Poste.io; Windmill, Typebot, Twenty, Akaunting, Budibase, Invoice Ninja removed — see Tool Catalog v2.2). Added decisions #39-41: infrastructure provider positioning (hosting model + customer license ownership), public open-source tools page, BYOK deferred to post-launch. Updated bundle examples (Poste → Stalwart Mail, 30 → 28).

1.13 13. Companion Documents

This Foundation Document references three companion documents for deeper dives:

Document	Version	Purpose
LetsBe_Biz_Technical_Architecture.md	1.1	Complete technical specification: system overview, component details, architectural decisions, access control model, autonomy levels, tool adapters, skills system, memory architecture, inter-agent communication, provisioning pipeline. The “how it works” document.
LetsBe_Biz_Product_Vision.md	1.0	North star document: one-liner, customer personas, product principles, customer journey (discovery through 3 months), business strategy, competitive position, moat analysis, three-year vision, vision validation checklist. The “why and what experience” document.
LetsBe_Biz_Pricing_Model.md	2.2	Detailed cost analysis and revenue modeling: per-tier cost breakdown, AI token cost modeling, founding member program impact, server pool economics, unit economics, sensitivity analysis, cash flow projections. The “financial details” document.

1.14 14. Next Steps (Immediate)

1.14.1 14.1 Technical Builds (Critical Path)

1. **Safety Wrapper** — Secrets redaction (4 layers), command classification (5 tiers), Hub communication endpoints

2. **Tool adapters (24+)** — Base framework, then parallelize all adapters
3. **Mobile app** — React Native, iOS + Android, chat interface, agent team management, approval queue
4. **Secrets registry** — SQLite with encryption, rotation support, audit logging
5. **Autonomy level system** — Per-agent, per-tenant gating configuration, approval request routing

1.14.2 14.2 Hub & Platform

6. **Hub updates** — Customer portal API, token metering, agent management, command approval endpoints
7. **Website redesign** — AI-powered onboarding (business type classification, tool recommendation, server selection, payment)
8. **Provisioner updates** — Deploy OpenClaw + Safety Wrapper instead of deprecated components, migrate Playwright scenarios

1.14.3 14.3 Go-to-Market

9. **Founding member recruiting** — Target: 50-100 customers in first 6 months
10. **Interactive demo** — Bella's Bakery sandbox (single VPS, fake data, session management)
11. **Content marketing** — Blog, videos, SEO strategy
12. **Community presence** — r/selfhosted, r/homelab, Hacker News, privacy forums

---

1.15 15. Success Criteria

For v1 launch: - Founding members can sign up, provision server, deploy all 28 tools, configure agents, send first command via app within 30 minutes - AI workforce operates without human intervention for non-destructive tasks (Green tier) - All external communications gated by default — user approves first email, then can unlock per agent/tool - Secrets never visible in logs, transcripts, or LLM requests - Mobile app supports chat, approvals, team management, usage dashboard - No critical security breaches in founding member usage

By month 3: - Founding members report 10+ hours/week time savings - 80%+ retention rate - Multiple SaaS subscriptions cancelled per customer - NPS > 50 - Product feedback informs v2 roadmap (vertical templates, advanced customization, mobile polish)

This document represents the current state of LetsBe Biz strategy and architecture as of February 25, 2026. It is a living document — updates will be reflected in versioning and document lineage.

For questions or clarifications, refer to companion documents or consult the architecture and product teams.