

AI Center of Excellence — Tools Update

Provided for: All SBA teams **Purpose:** the working tool set behind the AI Center of Excellence launch — what we use every day, why we chose it, and what the team should learn first. **Revised:** July 2026

AI Center key tools — hit the ground running

1. **Ghostty — the terminal.** Our daily terminal on the Spark and the Mac — fast, native, and built for long agent sessions. ghostty.org
 2. **Herd — the agent control room.** Runs inside Ghostty and keeps every agent visible — one terminal for the whole herd. herd.dev
 3. **Matt Pocock Skills — the skill library.** Reusable agent skills for real engineering — the collection many of our installed skills come from. github.com/mattpocock/skills
 4. **GPT-5.6 Sol + Opus 4.8 — the frontier pair.** The hosted heavy lifters we relied on before Kimi — with Fable in limited use for complicated tasks.
-

Terminal tools: Ghostty + Herd

Why Ghostty instead of the stock terminal on the Spark

- **It stays smooth under agent load.** GPU-accelerated rendering keeps long agent logs, diffs, and test output scrolling cleanly — the stock terminal on the Spark stutters and lags on exactly the output we produce all day.
- **It feels instant.** Native, low-latency input — when you live in the terminal for hours, the difference is not subtle.
- **Splits and tabs are built in,** with sane defaults and excellent font rendering — no extra tooling needed before you are productive.
- **One config, every machine.** The same dotfiles drive Ghostty on the Mac and on the Spark, so the whole team shares one terminal setup.
- **It is Herd's home.** Herd builds on the Ghostty VT engine and runs best inside Ghostty — choosing the terminal chooses the control room.

Herd — one terminal for the whole herd

Herd is a terminal-native agent multiplexer: workspaces, tabs, and panes with tmux-style persistence, plus something tmux never had — semantic agent state. Every pane shows whether its agent is *working*, *blocked*, or *done*, so running several agents in parallel stops being guesswork. It stays in the terminal we already use; no Electron dashboard, no separate environment.

Matt Pocock Skills

Matt Pocock — the TypeScript educator behind Total TypeScript — open-sourced his agent skills collection: small, readable skills for **real engineering, not vibe coding**. Test-driven development, bug diagnosis, domain modeling, interface design, code review — each skill encodes a disciplined workflow any agent can follow. Many of the skills installed in our own environment trace directly back to this collection, and it is the fastest way for a CoE team to raise its agent game.

- **GitHub:** github.com/mattpocock/skills
- **YouTube:** Matt Pocock's channel

The model journey — and why Kimi now

Period	Models	What happened
The last several months	OpenAI GPT-5.5 + Anthropic Claude Opus 4.5	The daily pair that carried the early work — the benchmark everything since is measured against.
Then	GPT-5.6 Sol (Medium) + Claude Opus 4.8	We relied heavily on this pair, with Fable in limited use for the most complicated tasks.
The last two weeks	Kimi K2.7 + K3	Some of the best results since beginning this journey — now the daily drivers.

Why the switch

Kimi is roughly **half the price of Opus** and, in our day-to-day work, performs better — though the two are close, and Opus 4.8 with GPT-5.6 Sol remains a strong pair. To be clear about the institutional reality: Anthropic or OpenAI will likely remain the gold standard at the SBA — procurement, support, and governance all point that way, and for obvious reasons. But the Chinese open-weight models are amazing tools, and the Center of Excellence should know how to use them well.

AI Center of Excellence · Tools Update · July 2026 · Online version: <http://10.0.0.123:4116/tools-update>