OpenClaw Investment Intelligence — Google Sheets Integration Update

What Changed and Why

The original plan used a static `portfolio.json` file that required manual updates every time you bought or sold a position. This revision eliminates that file entirely and replaces it with live reads from your Google Sheet via the `gog` CLI tool.

Single source of truth: Your Google Sheet (`11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0`)

What stays local on the VPS: themes.json, watchlist.json, extraction-log.json (these are agent-generated analysis data — the sheet doesn't track theme scores)

What's deleted: portfolio.json — gone entirely

NEW PHASE 1.5: Install and Configure gog (Google Workspace CLI)

Insert this between the original Phase 1 (directory setup) and Phase 2 (blogwatcher).

Step 1.5.1 — Install gog

```bash
# If you already have gog installed from Google Workspace setup, skip to 1.5.2

# Option A: Install via Homebrew (if available on your VPS)
brew install steipete/tap/gogcli

# Option B: Install from source
git clone https://github.com/steipete/gogcli.git
cd gogcli && make build
sudo cp bin/gog /usr/local/bin/
cd ~
```

Step 1.5.2 — Configure Google OAuth (if not already done)

```bash
```

```
# Set up credentials (you need a Google Cloud OAuth client_secret.json)
gog auth credentials ~/path/to/client_secret.json

# Add your Google account with Sheets access
gog auth add your-email@gmail.com --services sheets --manual

# The --manual flag gives you a URL to paste in your browser
# Authorize, copy the code back to the terminal
```

Step 1.5.3 — Set environment variables for the OpenClaw gateway

```
bash

# Edit your systemd service file
sudo nano /etc/systemd/system/openclaw-gateway.service

# Add/update these Environment lines:
# Environment=GOG_KEYRING_PASSWORD=<your-keyring-password>
# Environment=GOG_ACCOUNT=<your-email@gmail.com>
# Environment=PATH=/home/clawd/go/bin:/home/clawd/.npm-global/bin:/usr/local/bin:/usr/bin:/bin

sudo systemctl daemon-reload
sudo systemctl restart openclaw-gateway
```

Step 1.5.4 — Test Google Sheets access

```
bash

# Test reading the Master Summary tab
gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 "Master Summary!A1:F6" --json
```

You should see your portfolio summary data (AI Infrastructure, Block Chain, China totals).

Step 1.5.5 — Discovery: Map the dashboard column structure

This is critical. Run these to see exactly what columns exist in each dashboard tab:

```
bash

# Read the first 3 rows of each dashboard to see headers + sample data
gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 "AI Infra Dashboard!A1:Z3" --json
gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 "Block Chain Dashboard!A1:Z3" --json
gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 "China Dashboard!A1:Z3" --json
```

Save the output. You'll need the exact column letters for the SKILL.md ranges. For example, if column A is Ticker, B is Company Name, C is Shares, D is Cost Basis, E is Current Price, F is Current Value, G is Return % — then the skill should reference those specific ranges.

Send via Telegram after setup:

> "Read my portfolio Google Sheet. Pull data from all 3 dashboard tabs using gog sheets get.
> Tell me: (1) what columns each dashboard has, (2) how many holdings per sleeve,
> (3) the complete list of tickers across all 3 portfolios. This is your baseline."

UPDATED PHASE 3: Investment Radar Skill

Replace the SKILL.md from the original guide with the new version that references
Google Sheets instead of portfolio.json.

New file location (same path): `/home/clawd/.openclaw/workspace-ceo/skills/investment-radar/SKILL.md`

Use the updated SKILL.md provided in the companion file. Key differences:

Original	Updated
"Read portfolio.json"	"Run `gog sheets get` on dashboard tabs"
Static ticker list	Live pull every time
Manual update after trades	Zero maintenance
Single flat file	10-tab sheet with tx history, dashboards, allocations
Theme-to-holding mapping hardcoded	Discovered dynamically from sheet

Step 3.2 — Create data files (UPDATED — no portfolio.json)

Only create these files:

```bash
```

```
# Theme database (same as before)
nano /home/clawd/.openclaw/workspace-ceo/data/investment/themes.json
# (paste the themes.json template from the original guide)

# Watchlist (same as before)
nano /home/clawd/.openclaw/workspace-ceo/data/investment/watchlist.json
# (paste the watchlist.json template from the original guide)

# Empty extraction log
echo '{"extractions": []}' > /home/clawd/.openclaw/workspace-ceo/data/investment/extraction-log.json
```

DO NOT create portfolio.json. It no longer exists in this system.

UPDATED PHASE 5: Cron Jobs

The cron job prompts need to reference `gog sheets get` instead of `portfolio.json`.

5.1 — Daily Source Scan (unchanged)

No changes needed — the daily scan only reads blogwatcher and updates themes.json.
It doesn't touch the portfolio.

5.2 — Weekly Briefing (UPDATED)

```bash
```

```
openclaw cron add \
  --name "Weekly Investment Briefing" \
  --cron "0 13 * * 1" \
  --tz "America/New_York" \
  --session isolated \
  --message "Generate the weekly investment briefing using the investment-radar skill.

1. Read data/investment/themes.json for all theme scores
2. Pull LIVE portfolio data from the Google Sheet using these commands:
   - gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 'Master Summary!A:F' --json
   - gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 'AI Infra Dashboard!A:Z' --json
   - gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 'Block Chain Dashboard!A:Z' --json
   - gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 'China Dashboard!A:Z' --json
3. Read data/investment/watchlist.json for pipeline status
4. Apply score decay: for any theme with no mention in the past 30 days, subtract 1 point (minimum 0)
5. Check for convergence: any theme mentioned by BOTH Innermost Loop AND Moonshots within the past 30 days gets +3
6. Generate the full weekly briefing:

PORTFOLIO SNAPSHOT (from Google Sheet):
- Total invested, current value, overall return
- Per-sleeve: AI Infra, Blockchain, China — invested vs current

THEME SCORECARD (sorted by score, highest first):
- Theme name | Score | Trend arrow | Last mention date | Top companies

CONVERGENCE ALERTS:
- Themes validated by both sources within 30 days

SILENCE ALERTS:
- Any holdings from the Google Sheet whose associated themes have ZERO mentions in the past 60 days

PIPELINE MOVEMENT:
- Any themes that crossed tier thresholds this week

NEW WATCHLIST CANDIDATES:
- Companies mentioned for the first time this week

PORTFOLIO GAPS:
- High-scoring themes (7+) where the Google Sheet shows ZERO exposure

7. Save the updated themes.json with new scores
```

```
8. Send me the full briefing" \
  --agent ceo --announce --channel telegram
```

5.3 — Moonshots Check (unchanged)

No changes needed — doesn't reference portfolio data.

5.4 — Monthly Deep Research (UPDATED)

```
bash

openclaw cron add \
  --name "Monthly Deep Research" \
  --cron "0 14 1 * *" \
  --tz "America/New_York" \
  --session isolated \
  --message "Run the monthly deep research analysis using the investment-radar skill.

1. Read data/investment/themes.json
2. Pull LIVE portfolio from Google Sheet:
   - gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 'AI Infra Dashboard!A:Z' --json
   - gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 'Block Chain Dashboard!A:Z' --json
   - gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 'China Dashboard!A:Z' --json
3. Identify any theme that crossed from Watch to Research tier (score 5+) in the past 30 days
4. For each newly Research-tier theme:
   a. Search the web for the top 3 public companies in that theme
   b. For each company, search for: recent earnings results, insider buying/selling, analyst targets
   c. Write a 3-paragraph investment thesis: bull case, bear case, timing assessment
5. Cross-reference holdings from the Google Sheet against theme scores:
   - TRIM CANDIDATES: holdings whose themes have 60+ days of silence
   - OPPORTUNITY GAPS: high-scoring themes (8+) with zero exposure in any sleeve
6. Check actual vs target allocations using the Allocation tabs
7. Generate a full monthly research report and send it to me
8. Update watchlist.json with any pipeline changes" \
  --agent ceo --announce --channel telegram
```

UPDATED Telegram Cheat Sheet

Portfolio Check (UPDATED)

> "Pull my current portfolio from the Google Sheet and check it against theme scores.
> Any silence alerts, gaps, or allocation drift?"

Add a New Holding (UPDATED)

> "I just bought [TICKER] in my [AI Infra/Blockchain/China] portfolio. I've already updated the Google Sheet. Pull the latest data and confirm it's mapped to the right theme in your analysis."

Remove a Holding (UPDATED)

> "I sold [TICKER]. I've updated the sheet. Pull the latest and remove it from any watchlist or theme mapping."

Full Portfolio Snapshot

> "Pull all 3 dashboard tabs from my Google Sheet and give me a complete portfolio snapshot: every ticker, cost basis, current value, return %, and which investment themes each one maps to."

UPDATED File Locations

File	Path	Purpose
OpenClaw config	`~/.openclaw/openclaw.json`	Main config
Investment Radar skill	`workspace-ceo/skills/investment-radar/SKILL.md`	Core skill (updated for Sheets)
Moonshots Monitor skill	`workspace-ceo/skills/moonshots-monitor/SKILL.md`	Podcast monitor (unchanged)
Theme database	`workspace-ceo/data/investment/themes.json`	Theme scores (local)
Watchlist	`workspace-ceo/data/investment/watchlist.json`	Pipeline tracker (local)
Extraction log	`workspace-ceo/data/investment/extraction-log.json`	Audit trail (local)
~~Portfolio~~	~~portfolio.json~~	**DELETED — lives in Google Sheet**
Portfolio (LIVE)	**Google Sheet `11kk1...`**	**Single source of truth**
Cron jobs	`~/.openclaw/cron/jobs.json`	Scheduled tasks
Blogwatcher data	`~/.config/blogwatcher/`	RSS feed state
gog tokens	`~/.gog/`	Google OAuth tokens

Implementation Order

1. **Phase 1** — Directory setup (same as original)
2. **Phase 1.5** — Install gog + configure Google OAuth + test sheet access ← NEW
3. **Phase 2** — Install blogwatcher (same as original)
4. **Phase 3** — Create Investment Radar skill (use UPDATED SKILL.md)
5. **Phase 3.2** — Create data files (themes.json + watchlist.json ONLY, no portfolio.json)
6. **Phase 4** — Moonshots Monitor skill (same as original)
7. **Phase 5** — Cron jobs (use UPDATED prompts for weekly/monthly)
8. **Phase 6** — Seed historical data (same, but verification step pulls from Sheet)
9. **Phase 7** — Test (same, but portfolio queries now hit the Sheet)

First-Run Validation Checklist

After everything is installed, run through these in order:

```bash
# 1. Verify gog can read the sheet
gog sheets get 11kk1KNHHxNNLgglS52mZt7-P5b-cH9ZPtv3FJHt8Dq0 "Master Summary!A1:F6" --json

# 2. Verify blogwatcher is working
blogwatcher blogs
blogwatcher scan

# 3. Verify OpenClaw sees both skills
# Send via Telegram: "List my available skills"

# 4. Verify OpenClaw can read the sheet through gog
# Send via Telegram: "Use gog to read my portfolio Google Sheet Master Summary tab. What do you see?"

# 5. Verify the full pipeline
openclaw cron run "Daily Source Scan"
openclaw cron run "Weekly Investment Briefing"
```

If step 4 fails, check:

- GOG_KEYRING_PASSWORD and GOG_ACCOUNT are in the systemd service Environment

- gog binary is in the PATH specified in the systemd service
- OAuth tokens haven't expired (`gog auth list` to check)