01 — Workspace Layout

> *Think of it like a filing cabinet with labeled drawers. Everything has a home.*

The Big Picture

All business files live under **one local Git repo**:

```
C:\GitHub Local\Business Ideas\
└── Setup Concierge\
    ├── Advisory Panel\    — AI panel members, consultations, action items
    ├── Black Nexus\       — Creative agency work (shared with JB)
    ├── Dashboard\         — Legacy (now in command-center repo)
    ├── Finances\          — Invoices (sent/received), expenses
    ├── Playbook\          — This manual you're reading
    ├── Resources\         — Walkthroughs, daily summaries, templates
    └── WNQ\               — Warriors Never Quit client work
```

Other Key Locations

What	Where
Dashboards (source of truth)	`C:\GitHub Local\command-center\`
Career Hub	`command-center\career\index.html`
Business Dashboard	`command-center\business\index.html`
Agent Workflows	`C:\GitHub Local\.agent\workflows\`
Web Dev Projects	`Business Ideas\Web Development\`
Snapshots (backups)	`Business Ideas\.snapshots\`

Rules

- **No OneDrive.** Everything is local Git. Sync happens via Gist or GitHub Pages.
- **Markdown first.** Write in `.md`, preview with your reader. Export PDF only for formal submissions or signatures.
- **Flat is better than nested.** Don't bury files 5 folders deep.

02 — Dashboards

> *Your cockpit. Two screens, one glance, everything that matters.*

The Hub

URL: thesetupconcierge.github.io/command-center

A landing page with two cards that link to:

Dashboard	Purpose (Unified CDA Governance)
Setup Concierge (Business)	Clinical consulting, revenue, client ops, Captain's Log
Career Hub	Professional Development — Scorecards, PPCL/Siemens prep

Clinical Context Isolation

The system operates in two distinct lanes to preserve mental focus. Business-specific metrics (Revenue, badges) are **hidden** in Career Hub mode to create a clean, diagnostic environment for high-stakes technical preparation.

Key Features (Both Dashboards)

- **Scorecard Circles** — visual progress rings per work area
- **Captain's Log** — timestamped notes with tags (#win, #concern, #note)
- **Task Checklists** — click to cycle: ⬜ → 🟡 → ✅
- **Hamburger Nav** — mobile slide-out drawer with links + actions
- **Mobile Stack Mastery** — Robust 1-column layout for iPhone usage.
- **Deep Link Mastery** — Section-level URLs (anchor links) with automatic highlight logic.
- **Gist Sync** — cloud-syncs state across devices via a private GitHub Gist

How Sync Works

1. Both dashboards share **one private Gist** with two separate files:
    - `sc_dashboard_state.json` (business)
    - `ch_dashboard_state.json` (career)
2. Your PAT + Gist ID are stored in `localStorage` under `sc_gist_config`
3. Set it once per browser → both dashboards auto-connect
4. Changes sync on every save with a 2-second debounce

Source of Truth

Edit dashboards directly in `C:\GitHub Local\command-center\` . Deploy with `/deploy-dashboard` .

03 — Workflows

> *Slash commands that run your operation. Type the name, follow the steps.*

Where They Live

All workflows: `C:\GitHub Local\.agent\workflows\*.md`

The Core Three

`/morning` — Open the Shop

Run at the start of each session. It:

1. Reads Captain's Log entries from Gist (what you noted between sessions)
2. Checks all 3 repos for uncommitted work
3. Verifies the revenue section is current month

4. Prints your Next Actions queue (NOW / SOON / LATER)
5. Scans for stale action item files (7+ days untouched)
6. Validates all agent file paths are correct
7. Outputs a formatted **Morning Brief**

`/consolidate` — Close the Shop

Run at the end of a session. It:

1. **Action Item Sweep** — catches commitments you made in the conversation
2. **Completion Sweep** — checks if existing items were finished
3. **Captain's Log Export** — archives today's log entries to markdown
4. **Git Checkpoint** — surgical sync strategy (Private + Shared repos)
5. **Daily Summary** — writes a session summary to Resources
6. **Playbook Staleness Check** — flags pages not updated in 7+ days
7. **Dashboard Link Refresh** — suggests new docs to link
8. **Security Sweep** — scans shared repos for strategic leaks before pushing
9. **Coherence Audit** — 360° doc/registry sync & path audit

`/deploy-dashboard` — Ship It

One command: stages, commits, and pushes `command-center` to GitHub Pages.

Other Workflows

Command	What it does
`/new-client`	Scaffold a client folder with standard subfolders
`/new-session`	Create a session folder with pre-filled docs
`/new-opportunity`	Scaffold a career opportunity folder
`/build-site`	Generate a website from brief + brand kit + playbook
`/snapshot`	Timestamped backup of critical files
`/cleanup`	Remove empty folders and temp files
`/search`	Full-text search across all business files
`/panel-review`	Run advisory panel review of the whole system
`/coherence`	360-degree doc/registry sync & path audit
`/organize-files`	Batch move/rename files into folder structures

Adding a New Workflow

1. Create `C:\GitHub Local\.agent\workflows\your-workflow.md`
2. Add YAML frontmatter with a `description`
3. Write numbered steps — the agent follows them literally
4. Add `// turbo-all` if all steps are safe to auto-run

04 — Git Strategy

> *Three repos, each with its own job. Local backups, cloud deploys.*

The Repos

Repo	Location	Remote?	Purpose
Business Ideas	`C:\GitHub Local\Business Ideas\`	❌ Local only	All business files — sensitive, never pushed
Command Center	`C:\GitHub Local\command-center\`	✅ GitHub Pages	Dashboards — pushed to deploy
Career Hub	`C:\GitHub Local\Career Hub\`	❌ Local only	Career-specific files

Why Local?

Business files contain client info, financials, strategy docs. They stay on your machine, versioned by Git for history, but never pushed remotely. The `.gitignore` excludes media, generated PDFs, and large binaries — source of truth is always the markdown.

What Gets Pushed?

Only `command-center` → triggers GitHub Pages rebuild at:

- `thesetupconcierge.github.io/command-center/`
- `.../command-center/business/`
- `.../command-center/career/`

Branching

Keep it simple: **everything on `main`**. No feature branches needed for a solo operation. Commit messages follow patterns:

- `checkpoint: morning Feb 11` — daily save from `/morning`
- `consolidate: end-of-day Feb 11` — wrap from `/consolidate`
- `[deploy] dashboard update 2026-02-11 15:57` — from `/deploy-dashboard`

Snapshots

For before-big-changes safety, run `/snapshot` to create a timestamped copy in `.snapshots/`.

05 — Advisory Panel

> *A boardroom full of experts — who happen to be AI personas. They challenge your thinking.*

What It Is

A simulated advisory board of AI personas, each with a distinct specialty. When you run `/panel-review`, they analyze your system, strategy, or deliverable and give feedback from their unique angle.

Build Your Own Panel

The panel is **fully customizable** — pick roles that match your business needs. Here's a template to get started:

Role	Focus Area	Example Questions They'd Ask
Strategist	Positioning, market timing	"Who's the ideal customer?"
Operations	Process, SOPs, efficiency	"Can this scale with 10 clients?"
Finance	Revenue, pricing, cash flow	"What's the unit economics here?"
Legal/Risk	Compliance, contracts	"What liability are you carrying?"
Marketing	Brand, outreach, content	"How does this sound to a cold lead?"
Tech	Architecture, tooling	"Will this break when X changes?"
UX/Design	User experience, accessibility	"Can someone use this in 5 seconds?"
Sales	Pipeline, acquisition	"What's the close rate look like?"
Client Success	Retention, delivery quality	"Would the client renew?"
Innovation	Future-proofing, emerging tech	"What's the 3-year version?"

> ***Tip:*** *You don't need all 10+. Start with 3–5 that cover your blind spots. Add more as your operation grows.*

Where It Lives

```
Setup Concierge\Advisory Panel\
├── Members\           — One folder per panelist (bios, perspectives)
├── Consultations\     — Full panel reviews with assets
├── Meeting Notes\     — Individual session notes
└── Action Items\      — Tasks generated from reviews
```

How to Use

- **Full review:** `/panel-review` — all panelists weigh in on a topic
- **Targeted:** Ask a specific panelist by name for focused advice
- **Consultation outputs** go into `Consultations\` as markdown (PDF only for formal deliverables)

Making It Yours

Each panelist folder in `Members\` contains their persona definition. To add or customize a member:

1. Create a folder with their name
2. Define their expertise, communication style, and the lens they evaluate through
3. The more specific you make them, the sharper their feedback gets

06 — Action Items

> *You say "I need to send that tomorrow." The system catches it so you don't have to.*

Two Modes, Always Listening

Mode 1: Capture — Detect New Items

The AI watches for commitment signals during conversation:

- "I will...", "I need to...", "let me...", "don't forget to..."
- "send to...", "upload to...", "follow up with..."

Noise filters — these are NOT captured:

- Vague: "I might...", "maybe I'll..."
- Questions: "should I...?"
- Instructions to the AI: "can you...", "build me..."

When items are detected, they're presented for approval before adding to the dashboard.

Mode 2: Completion — Detect Done Items

When you say things like "I already sent it" or "that's done," the AI:

1. Cross-references against your **Next Actions** on the dashboard
2. If a match is found, asks to mark it complete
3. Never marks anything done without your confirmation

Where Items Live

Next Actions section on the Business Dashboard (`NEXT_ACTIONS` array in `index.html`), sorted by urgency:

Urgency	When
🔴 **NOW**	Do today
🟡 **SOON**	This week
🟣 **LATER**	Backlog

Safety Rails

- **Capture is aggressive, insertion is conservative** — better to catch too many and dismiss
- **Never auto-adds** — always asks first
- **Cross-references before adding** — no duplicates
- **One item per deliverable** — doesn't over-split tasks

🛡️ RayRay's Lean-Out Protocol

QA & Compactor Logic: Ensuring the system remains surgical and navigable.

The 10-Second Rule

If you can't find a critical strategic document or task in under 10 seconds on mobile, the system has "Bloat Debt."

- **Surgical Cleanup**: After every major sprint, run `/cleanup` to purge legacy fragments and orphaned files.
- **Link Integrity**: All links must be relative to the repository. Never leave `file:///` or `../../` paths in the Master Hub.
- **Mobile Density**: Prioritize accessibility. Hide verbose labels and use icons (📝, 📕, 🔗) to save space.

End-of-Day Sweep

`/consolidate` runs a final sweep to catch anything the real-time detection missed, ensuring the Captain's Log remains a lean, surgical stream.

07 — Client Ops

> *Every client gets the same structure, every session gets documented.*

Client Folder Structure

When you run `/new-client`, this scaffold is created:

```
Setup Concierge\[Client Name]\
├── Action Items\       — Tracked tasks for this client
├── Media\              — Photos, videos, brand assets
├── Meeting Notes\      — Session-by-session notes
├── Sessions\           — Individual session folders (via /new-session)
└── Website & Strategy\ — Briefs, audits, strategy docs
```

Current Clients

Client	Folder	Status
Warriors Never Quit (WNQ)	`Setup Concierge\WNQ\`	Active — website revamp

Session Workflow

1. **Before:** Run `/new-session` to create a dated session folder with pre-filled templates
2. **During:** Take notes, capture decisions, track deliverables
3. **After:** Run `/consolidate` to sweep action items and archive the session

Invoicing

Invoices live in:

```
Setup Concierge\Finances\
├── Invoices Sent\     — What you've billed clients
├── Invoices Received\ — What vendors bill you
└── Expenses\          — Operational costs
```

Invoices are created as markdown and sent through J4AFoundation LLC (the legal entity behind Setup Concierge). Export to PDF on-demand when ready to send (mobile MD-to-PDF tools or `npx md-to-pdf`).

Website Delivery

For clients who need websites:

1. Create a **Site Brief** (goals, branding, content)
2. Run `/build-site` which combines the brief + brand kit + industry playbook
3. Output goes to `Web Development\` for staging
4. Deploy to client hosting (GitHub Pages, Wix, WordPress, etc.)