Tree Blueprint — Where to Put Things

⚠ READ FIRST when deciding any new path

> **Purpose:** Single answer to "where does this go?" — read FIRST before any new path creation, repo clone, or directory move.

> **One rule:** every file has exactly one canonical home determined by *what it is* + *whose it is*, not by *who created it* or *when*.

1. Top-level decision tree

For any new content, answer 4 questions in order:

```
Q1: Is it CODE / file content / data?  No → it's a process artefact, not a tree concern.
Q2: Whose data is it? (CEO personal / ALAI Holding AS / ALAI Tech DOO / a specific client / regenerable)
Q3: What kind? (product, client deliverable, internal tool, scholarly, financial record, brand asset, …)
Q4: What stage? (active draft, in-flight, completed deliverable, archive)
```

2. Routing matrix

Whose?	What kind?	Canonical home
CEO personal	CV, NAV records, ID docs	`~/personal/{cv,nav,legal-personal}/`
CEO personal	Scholarly research (Quran-19, etc.)	`~/personal/scholarly/<topic>/` (or own repo + own domain — see ucenje precedent)

Whose?	What kind?	Canonical home
CEO personal	Personal coding (games, hobby projects)	`~/personal/code/<project>/`
CEO personal	Real estate, family property	`~/personal/real-estate/<property>/`
CEO personal	Personal accounting (NOT ALAI Holding)	`~/personal/finance-personal/`
ALAI Holding AS	A product (own SaaS, own brand within ALAI)	`~/business/ALAI-Holding-AS/products/<product-name>/`
ALAI Holding AS	Client deliverable / engagement	`~/business/ALAI-Holding-AS/clients/<CLIENT-NAME>/`
ALAI Holding AS	Brand surface (alai.no, basicconsulting.no, forms.alai.no)	`~/business/ALAI-Holding-AS/brand-surfaces/<domain>/` OR `~/business/ALAI-Holding-AS/web/` if alai.no main
ALAI Holding AS	Brand assets (logo, palette, typography)	`~/business/ALAI-Holding-AS/brand/` (single source of truth)
ALAI Holding AS	Finance (accounting, grants, invoices, timesheets)	`~/business/ALAI-Holding-AS/finance/`
ALAI Holding AS	Legal (contracts, ROPA, vedtekter, certificates)	`~/business/ALAI-Holding-AS/legal/`
ALAI Holding AS	Internal libraries / shared packages	`~/business/ALAI-Holding-AS/internal/packages/`
ALAI Holding AS	Sales pipeline, partners, comms, marketing	`~/business/ALAI-Holding-AS/{sales,partners,comms,content}/`
ALAI Holding AS	Internal product architecture decisions	`~/business/ALAI-Holding-AS/product-architecture/` (NOT system architecture)
ALAI Holding AS	Service catalog / commercial offerings	`~/business/ALAI-Holding-AS/service-catalog/` (NOT system services)
ALAI Tech DOO	RS subsidiary content (Bilko/Tok/Drop distribution)	`~/business/ALAI-Tech-DOO/{products,deliverables,legal,ops}/`
External client	Client-owned repo (we work IN their repo)	`~/clients-external/<client-name>/` (matches their org name when possible)
External client	Engagement records, contracts with that client	`~/business/ALAI-Holding-AS/clients/<CLIENT>/engagement-docs/`
External client	Client's brand assets we use	`~/clients-external/<client-name>/branding/`
System runtime	Daemons, schedulers, hooks	`~/system/{daemons,tools,hooks}/`
System runtime	Architecture decisions (system-level)	`~/system/architecture/decisions/ADR-XXX.md`

Whose?	What kind?	Canonical home
System runtime	Operational services (authentik, planka, vault)	`~/system/services/<service>/`
System runtime	Agent persona definitions	`~/system/agents/personas/<PersonaName>/`
System runtime	Specs, plans, runbooks	`~/system/specs/`, `~/system/runbooks/`
System runtime	Active databases (Mission Control, HiveMind, ...)	`~/system/databases/` (symlinked) or `~/Library/Application Support/ALAI/db/`
ALAI engineering tools	Internal CLI / SDK / library used across ALAI	`~/projects/<repo-name>/` (must have GitHub remote, must be tracked, must NOT mix client work)
ALAI infra deploy workspace	CF/DNS/Tailscale/Vault/BookStack systemic configs	`~/aisystem/` (Mehanik gate reads BUILD-BLUEPRINT.md here)
Regenerable	node_modules, .gradle, .next, target, venv, build outputs	INSIDE the repo, gitignored, never outside
Backups	Tar archives, snapshot dumps	`~/backups/_archive/<sweep-name-date>/`
OS-managed	Library data (apps), Caches	`~/Library/` (DO NOT TOUCH)

3. Anti-patterns — explicitly forbidden

Anti-pattern	Correct alternative
Putting personal scholarly content under commercial brand site	Own repo + own domain (see ucenje → `johnatbasicas/ucenje` + `ucenje.alai.no`)
Client deliverable in `~/projects/`	`~/clients-external/<client>/`
Agent persona in `~/companies/<Name>/`	`~/system/agents/personas/<Name>/`
Pretending a domain is a separate firma (vedtekter for it, etc.)	Domain → `brand-surfaces/<domain>/` under owning entity
Mixing two clients in one tree	One tenant per top-level subdir under `~/clients-external/`
Creating new top-level home dir without entry in `~/system/specs/canonical-registry.md`	Update registry first, then create dir
Writing CEO PII (CV, NAV, ID) anywhere outside `~/personal/`	`~/personal/legal-personal/`, `~/personal/cv/`, `~/personal/nav/`
Auto-generated `_pii-staging-*` quarantine pattern (one-time fix)	Process docs/forms BEFORE landing them in tree
`~/Public` for sensitive content (it is local-network readable)	Never. Move to private tree.
Build artefacts (node_modules, target) outside their repo	Always inside repo with .gitignore entry

Anti-pattern	Correct alternative
`.env` files committed to git	`.env*` in .gitignore; secrets in Bitwarden / Keychain
Terraform state on local disk	Remote state (S3/Azure/CF KV); `*.tfstate*` in .gitignore

4. References

- ADR-023: `~/system/architecture/decisions/ADR-023-anvil-tenant-restructure-2026-05-07.md`
- Canonical registry: `~/system/specs/canonical-registry.md`
- Git structure rules: `~/system/specs/anvil-git-structure-2026-05-07.md`

Revision #2
Created 2026-05-07 20:30:59 UTC by John
Updated 2026-06-14 20:02:33 UTC by John