OSIX Tech - Naming Conventions

1. Repositories & Projects

- **Language:** English.
- **Format:** kebab-case (`nessie-ingestor` , `nessie-frontend`).
- **Structure:** `<project>-<module>` , always descriptive.
- **Published packages:** follow ecosystem convention
    - npm → kebab-case (`my-package`)
    - PyPI → snake_case or kebab-case (`my_package` or `my-package`)

2. Variables, Classes & Packages

- **Variables & functions (JS, TS, Python):** snake_case.
- **Classes:** PascalCase.
- **Constants:** ALL_CAPS.
- **Published modules/packages:** follow the ecosystem's naming standard.

3. Commits

- **Format:** vX.Y.Z type[scope]: message in English, imperative form
  Example: v1.4.2 fix[auth]: reject expired refresh tokens
- **SemVer rules:**
- MAJOR → breaking changes
- MINOR → new features without breaking
- PATCH → fixes and internal improvements
- **Automation:** use tools like `npm version` , `bump2version` , `standard-version` or `changesets` to avoid manual versioning.
- **Scope:** optional but recommended.

4. Branching

- **Format:** kebab-case (`feature/auth-token-rotation`).
- **Prefixes:**
- `feature/`
- `fix/`
- `hotfix/`
- `release/`
- `chore/`
- Ticket ID optional at the end.

5. Folders & Files

- **General rule:** kebab-case.
- **Exception:** React components use PascalCase (`UserCard.jsx`).
- **React Frontend specifics:**
- Folders: kebab-case (`user-profile`).
- Hooks: `useCamelCase` (`useAuth.js`).
- Utils & services: kebab-case (`api-client.js`).
- JSX props: camelCase.

6. Database & API

- **Database:**
- Tables & columns: snake_case, plural (`users` , `invoices`).
- Foreign keys: `<singular_table>_id` (`user_id`).
- **API:**
- Routes: kebab-case, plural, versioned (`/api/v1/user-accounts`).

7. Environment Variables

- Format: `ALL_CAPS_SNAKE_CASE` (`DATABASE_URL` , `API_KEY`).

8. Containers & Services

- **Docker images:** kebab-case (`nessie-backend` , `opills-dashboard`).
- **Cloud services:** `<project>-<module>-<env>` (`nessie-backend-prod`).
- **Environments:** `dev` , `staging` , `prod` .

Quick Reference Table

Category	Format / Rule	Example(s)
Repo Names	kebab-case, `<project>-<module>`	`nessie-ingestor` , `opills-dashboard`
Vars/Funcs	snake_case	`user_name` , `get_user_data()`
Classes	PascalCase	`UserProfile` , `AuthService`
Constants	ALL_CAPS	`MAX_USERS` , `API_KEY`
Commits	`vX.Y.Z type[scope]: message` (SemVer)	`v1.0.0 feat[auth]: add token refresh`
Branches	kebab-case with prefix	`feature/user-auth` , `fix/payment-bug`
Folders	kebab-case	`user-profile` , `api-client`
React Components	PascalCase for files, kebab-case for folders	`UserCard.jsx` in `user-card/`
DB Tables	snake_case plural	`users` , `invoices`
API Routes	kebab-case plural, versioned	`/api/v1/user-accounts`
Env Vars	ALL_CAPS_SNAKE_CASE	`DATABASE_URL`
Docker Images	kebab-case	`nessie-backend`

Category	Format / Rule	Example(s)
Cloud Services	<project>–<module>–<env>	opills–dashboard–prod