На чому стоїть кожен модуль

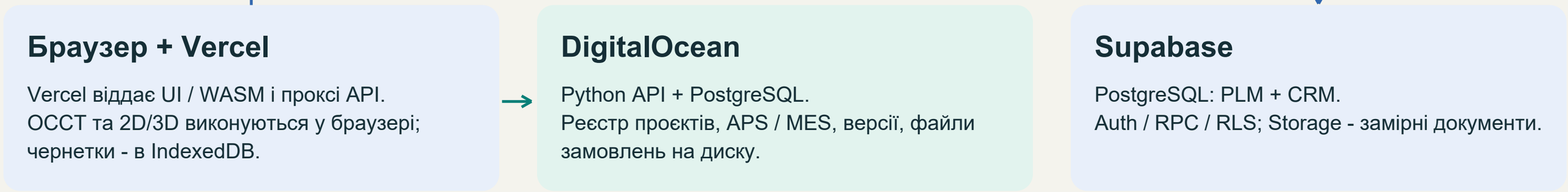
OCCT рахує геометрію; React будує інтерфейс; PostgreSQL зберігає дані; Vercel і Ocean виконують різні серверні ролі.

МОДУЛЬ	ІНТЕРФЕЙС / ЛОГІКА	ДАНІ / СЕРВЕР	РОЗМІЩЕННЯ
CAD + Studio	React + TypeScript + Vite. OCCT / Open CASCADE через opencascade.js / WebAssembly у браузері. Three.js / React Three Fiber - 3D; Konva - 2D.	Проєкт JSON; локальна чернетка IndexedDB. Серверні проєкти, налаштування й версії - через Python API у PostgreSQL.	UI: Vercel API: Ocean
PLM	React + TypeScript + Vite. Планування, статуси, ролі, історія. Серверні переходи через SQL RPC; карти MapLibre.	Supabase: PostgreSQL, Auth, PostgREST / RPC, приватне Storage для замірних файлів.	UI: Vercel Дані: Supabase
CRM	React + TypeScript + Vite. Клієнти, замовлення, комунікації. Тимчасовий спільний доступ через MES.	Приватна схема crm у Supabase PostgreSQL. Vercel Function → Supabase Edge Function → RPC.	UI: Vercel Дані: Supabase
MES / цифровий цех	Next.js + React + TypeScript, статична збірка UI. Three.js / React Three Fiber - 3D. Python - APS, операції, буфери й імпорт.	PostgreSQL на Ocean через psycopg. Python HTTP API, Nginx, systemd. SQLite - локальний режим і тести.	UI: Vercel API + БД: Ocean
BrainStone	Модуль React у PLM. Збирає факти й пояснює готовність замовлення. Голосовий інтерфейс - API браузера.	Читає PLM + MES + серверний проєкт. Окремої БД BrainStone немає; це шар над даними систем.	PLM + API Ocean / Supabase
Стоуник · ШІ	Помічник Studio/CAD. Запит, вибрані фото та огляд проєкту надсилаються явно користувачем.	Node.js Function на Vercel → Google Gemini API. Ключ залишається на сервері.	Vercel + Google

OCCT - геометричне ядро, а не готовий CAM. Траєкторії CombiCut, постпроцесор, розкладка й виробничі правила - окремий прикладний код.

Де живуть база, файли та налаштування

Постійні бізнес-дані розділені між PostgreSQL Ocean і PostgreSQL Supabase. Vercel віддає інтерфейси та проксі / функції.



Два контури HTTPS API: MES / проєкти / файли → Ocean; PLM / CRM → Supabase. Між ними - integration API, не реплікація баз.

Папки замовлень на Ocean

Налаштований корінь:
/var/lib/viyar-orders/files/<номер>/

1.Замір · 2.Чертеж · 3.Screen погодження
4.Combicut · 5.NC300 · 6.Монтаж
7.Доп. информация · 8.Раскрой · 9.Оцінка якості

Фізичний шлях - з конфігурації інсталятора; доступність файлів через API перевірена сьогодні.

Файл і запис про нього - окремо

Диск: ZIP, DXF, XML, TXT, документи та версійні набори.
PostgreSQL: каталог order_file_sets, manifest, номер замовлення, ревізія, контрольні суми.

Браузер отримує список і завантажує файли через захищений API. Це не відкрита мережева папка SMB.

Послуги: важливе уточнення щодо SQLite

Серверні quote / production передбачені у PostgreSQL Ocean (shared_settings + історія версій). На 12.10 обидва ключі повернули not_found, історія порожня: спільні набори ще не збережені. Studio працює з вбудованими / локальними значеннями. SQLite не є чинною production-БД MES.

Резервування: pg_dump для PostgreSQL і окрема копія папок. Сам дамп БД не замінює резервну копію файлів. Копія Studio на Ocean і Vercel-бандл оновлюються окремо.

Як зараз відбувається обмін

Основний транспорт - HTTPS: JSON для даних, ZIP для пакетів, окремі завантаження файлів. Єдиного загального event bus не підтверджено.

Studio → сервер POST /api/registry/orders	Проєкт JSON + номер ERP + очікувана ревізія. Сервер зберігає проєкт / версію; це не автоматичне створення PLM.
PLM → Studio / CAD /api/integration/handoffs → #handoff=…	Тимчасовий токен зв’язує конкретне PLM-замовлення з серверним проєктом. Проєкт читається через integration API.
CAD → MES POST /api/integration/cad-release	ZIP vs3d-pack-1: manifest, деталі, слеби, розкладка, операції. Пакет потрапляє в inbox; імпорт у маршрут підтверджується окремо.
CombiCut → оператор /api/registry/orders?files=1&…&kind;=combicut	Набір mes-combicut-2: manifest + DXF + пара SLAB<ключ>.xml / .TXT. Версійна папка 4.Combicut; оператор читає список і скачує.
MES → PLM / BrainStone GET /api/integration/progress?order=<UUID>	PLM UUID зіставляється з зовнішнім номером замовлення. Повертаються реальні операції та прогрес; UI періодично опитує API.
CRM → Supabase POST /api/registry/crm → Edge → RPC	Спільний MES-доступ перевіряє gateway. Команди пишуться у приватну схему crm. Прив’язка CRM → Studio зараз призупинена.
Послуги / ціни → Studio /api/registry/orders?settings=quote|production	GET читає версію; POST записує з перевіркою конфлікту. Оновлення при старті, поверненні у вкладку та раз на 10 хв; нині серверні набори порожні.

Ідентичність: UUID PLM / CRM і projectId / slabId різні; зовнішній номер ERP зв’язує системи. Ревізії, хеші та повтори без дублів захищають передачу. WMS Саші, подія оплати ERP та пряме керування станком ще не підключені.

Джерела: package.json модулів; settingsTransport.ts; mes_storage.py; shared_settings.py; order_files.py; plm_bridge.py; crm/server; чинний Vercel config. API MES підтвердив backend=postgres 12.10.2026.