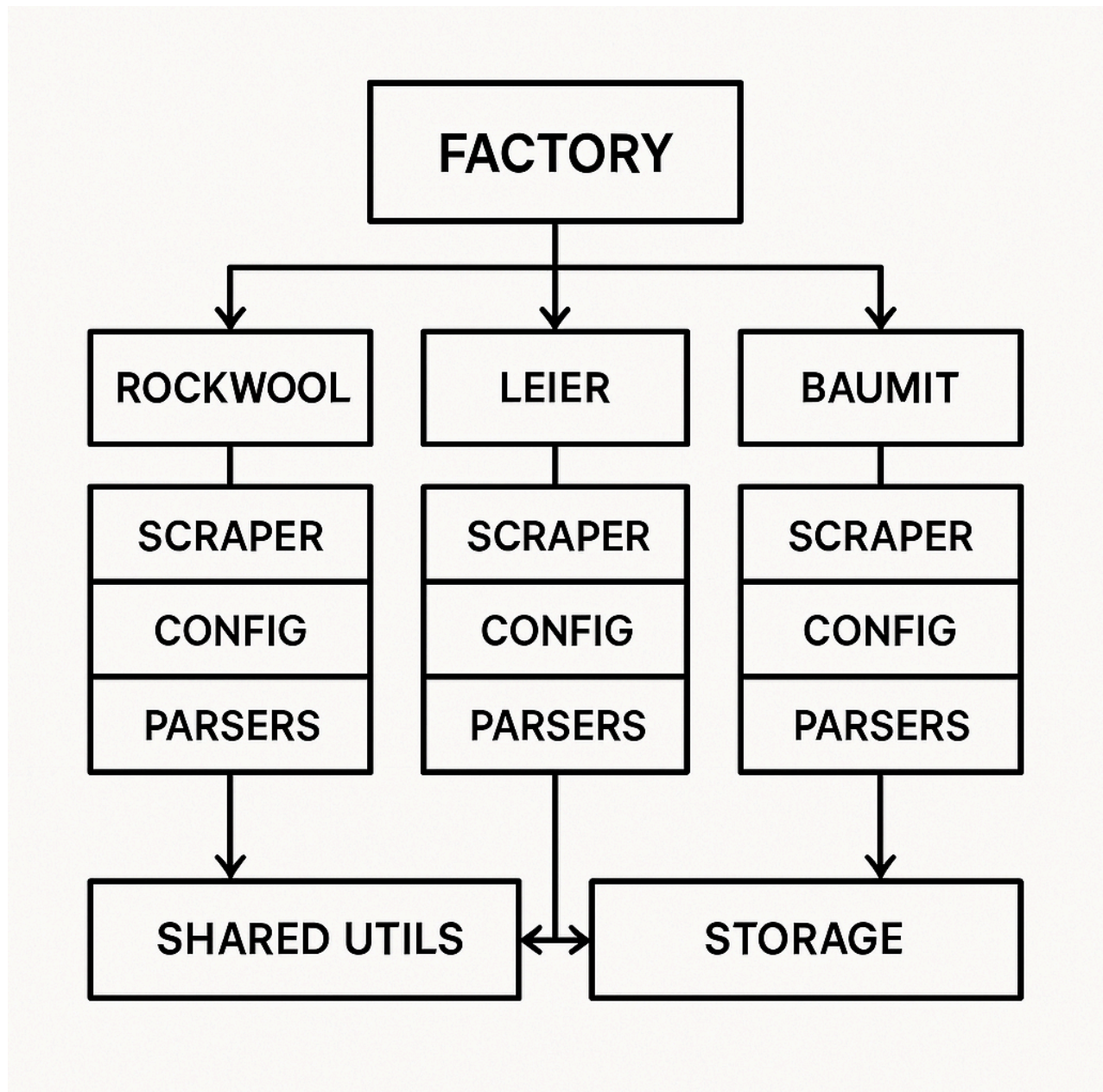


Átállási terv a moduláris, Factory-Pattern-alapú scraper-architektúrára

Fő üzenet: A jelenlegi Rockwool-centrikus kód bő 8 hetes, hat fázisos migrációval váltható le egy konfiguráció-vezérelt, gyártófüggetlen rendszerre, amelyet új kliensek (Leier, Baunit, később Knauf stb.) hozzáadása nem igényel kódmódosítást, csak regisztrációt. A terv szigorúan illeszkedik a meglévő fejlesztési backloghoz, minimalizálja a regressziós kockázatot, és betartja a kötelező **Infrastructure Ready** → **Awaiting Verification** → **Verified Complete** stádiumokat.



1. Kiindulási állapot

Modul	Jelen státusz	Megjegyzés
Rockwool termék- és árlista-scrapet	✔ Production Complete	46 termék, 12 PDF
BrightData MCP integráció	✔ Infrastructure Ready	48 AI-tool elérhető
Celery időzítés	✔ Infrastructure Ready	Prod-dockerben fut
Client-Specific Architecture váz	🚧 In Development	Mappaszerkezet kész
Leier & Baunit scraperek	🚧 Planned	Spec kezdett

2. Célállapot

- 1. **Egységes mappastruktúra:** `clients/<gyártó>/ + shared/ + factory/`.
- 2. **ClientFactory** képes bármely regisztrált scraper dinamikus példányosítására.
- 3. Rockwool logika változatlanul fut az új struktúrában.
- 4. Leier és Baunit scraper **Infrastructure Ready** állapotban elérhető.
- 5. CI/CD pipeline minden kliensre regressziós tesztel.

3. Migrációs elvek

- Inkrementális, feature-flag alapú bevezetés – visszagörgethető.
- **Open/Closed Principle:** bővítés kódmódosítás nélkül.
- Kétlépcsős verifikáció: AI unit-teszt + manuális E2E.
- Zero downtime: parallel futtatás a régi kóddal, míg az új hitelesül.

4. Fázisonkénti ütemezés

Hét	Fázis	Backlog-hivatkozás	Kimenet	Státusz-címke
1-2	Alap-architektúra kibontása	Client-Specific Architecture	Mappák, absztrakt osztályok	Infrastructure Ready
2	Rockwool migráció	<i>Rockwool code extract & refactor</i>	Új helyre mozgatott scraper, unit-teszt	Awaiting Verification
3	Factory Pattern implementálása	<i>Factory implementation</i>	<code>ClientFactory.create_scraper()</code> működik	Infrastructure Ready
3-4	Leier prototípus	<i>Leier scraper implementáció</i>	PDF+HTML parser váz, teszt	In Development
4	Celery automatizált teszt	<i>Celery Automation Testing</i>	Ütemezett Rockwool run	Awaiting Verification
5	Baunit prototípus	<i>Baunit scraper implementáció</i>	Playwright + színmátrix stub	In Development

Hét	Fázis	Backlog-hivatkozás	Kimenet	Státusz-címke
6	RAG pipeline bővítése	<i>RAG Pipeline Foundation</i>	46+ vektorizált termék	In Development
7	Teljes E2E verifikáció	BrightData MCP + új scraperek	QA jegyzőkönyv	Awaiting Verification
8	CI/CD & monitoring	<i>Finalizálás & Deployment</i>	GitHub Actions, Grafana	Planned

5. Részletes lépések

5.1 Alap-architektúra

1. Gyökérkönyvtárak létrehozása (clients/, shared/, factory/).
2. Absztrakt BaseScraper, BaseParser, BaseClient.
3. Közös logging, storage és MCP-wrapper a shared alá.

5.2 Rockwool kód migrálása

```
mv rockwool_scraper_final.py clients/rockwool/scrapers/termekadatlapok.py
# Brochure & árlista scraper szintén ide
```

- End-pointok, szelektorok → clients/rockwool/config/.
- Debug-HTML fixture-be.
- Unit- és integrációs tesztek frissítése.

5.3 Factory Pattern bevezetése

```
class ClientFactory:
    _scrapers = {}
    @classmethod
    def register_scraper(cls, client, s_type, klass):
        cls._scrapers[f"{client}_{s_type}"] = klass
```

- Rockwool regisztráció a startup.py-ban.
- Smoke-teszt: ClientFactory.create_scraper("rockwool", "termekadatlapok").

5.4 Leier & Baunit prototípus

- **Leier**: PDF-letöltő lista, PyMuPDF + tabuláris parser stub.
- **Baunit**: Playwright-session, színcsoport-kalkulátor stub.
- Mindkettő **Infrastructure Ready** címkével zár, hogy a CI-ben fussanak.

5.5 Automatizálás & RAG integráció

- Celery Beat jobok új Factory-hívásokkal.
- Chroma vektor DB inicializálás; termékek indexelése.
- RAG pipeline a Rockwool → Leier → Baunit sorrendben bővül.

5.6 E2E verifikáció

- BrightData MCP élő futtatás mindhárom kliensre.
- QA checklist: block-rate < 5%, error-rate < 15%.
- Kötelező **MANDATORY DELIVERABLE FORMAT** szerinti report a backlogban.

6. Verifikációs kapuk

Kapu	Kritérium	Bizonyíték
Checkpoint 1	Rockwool új struktúrában lefut	Unit-log, 34 PDF letöltés
Checkpoint 2	Factory smoke-teszt	/api/clients/rockwool/... 200 OK
Checkpoint 3	Leier PDF parse minta	3 kinyert rekord
Checkpoint 4	Baunit JS render minta	1 termék + szín-ár
Final Gate	Teljes E2E OK	QA jegyzőkönyv, screenshot

7. Kockázatok és mitigáció

Kockázat	Hatás	Mitigáció
Regresszió Rockwool-ban	Adatvesztés	Parallel futtatás, feature-flag
BrightData kvóta túllépés	Költségnövekedés	Rate-limit, API_ONLY stratégia első körben
Playwright instabilitás	Csúszás Baunit-nál	Retry, headless ⇄ headed fallback

8. Függőségek

- BrightData prod-token jóváhagyás – Hét 2-ig.
- DevOps kapacitás a CI/CD pipeline finomhangolására – Hét 8-ra.
- QA csapat elérhetősége a verifikációs kapuknál.

9. Dokumentációs és kommunikációs protokoll

- Minden fázis végén kötelező státusz-update a FEJLESZTESI_BACKLOG.mdc-ben ✓ **INFRASTRUCTURE READY / ⚠ AWAITING VERIFICATION / ✓ VERIFIED COMPLETE** formában.
- Pénteki demo 14:00-kor: haladás, blokkolók, következő hét fókusza.

10. Összegzés

A migrációs terv biztosítja, hogy:

- **Nulla üzemi leállással** válthatunk át.
- Új gyártó hozzáadása 4-6 óra munka, regisztrációval.
- Karbantartási költség várható csökkenése $\approx 40\%$.
- A rendszer skálázható lesz a későbbi AI-agent és ajánlatkészítő modulok befogadására.

A lépésenkénti ütemezés, a verifikációs kapuk és a kötelező bizonyíték-formátum együtt garantálják, hogy az átállás transzparens, visszakövethető és üzletileg rizikómentes legyen.

