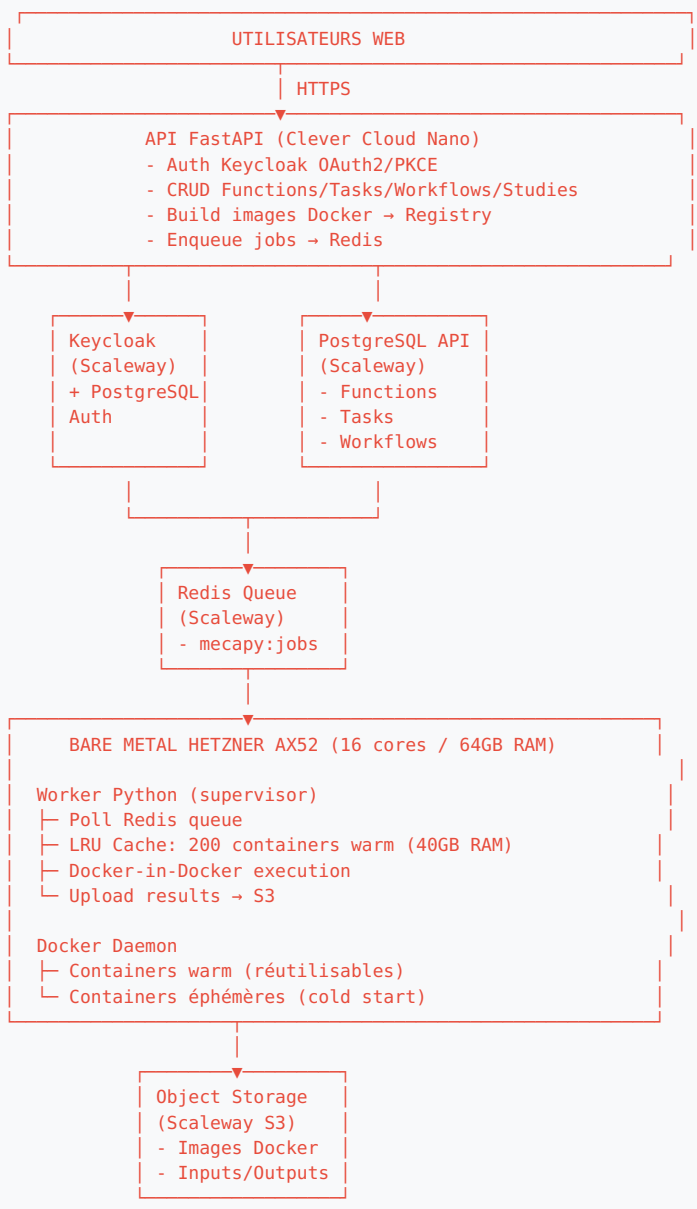
PLAN DE TRAVAIL MECAPY

📊 ARCHITECTURE GLOBALE



💰 BUDGET INFRASTRUCTURE

Service	Configuration	Coût/mois
API	Clever Cloud Nano	€7
Keycloak	Scaleway Instance DEV1-S	€10
PostgreSQL Keycloak	Scaleway DB 512MB	€12
PostgreSQL API	Scaleway DB 1GB	€18
Redis	Scaleway DB 512MB	€15
Object Storage	Scaleway S3 (~50GB)	€1
Container Registry	Scaleway (100GB)	Gratuit
Bare Metal Worker	Hetzner AX52	€54
TOTAL		**€117/mois**

Scaling estimé

- **1 serveur** : ~5000 calculs/jour → €117/mois
- **2 serveurs** : ~10000 calculs/jour → €171/mois
- **3 serveurs** : ~15000 calculs/jour → €225/mois

🔄 FLUX UTILISATEUR

1. Création de fonction

```
Utilisateur → API POST /functions
  ├─ Valide code (blacklist imports)
  ├─ Build Dockerfile + image
  ├─ Push → Container Registry Scaleway
  └─ Save metadata → PostgreSQL
```

2. Exécution de calcul

```
 Utilisateur → API POST /functions/{id}/execute
   ├─ Create task (status: queued) → PostgreSQL
   ├─ Upload inputs → S3
   └─ Enqueue job → Redis

Worker (bare metal):
   ├─ Poll Redis (blpop)
   ├─ Get container (warm LRU cache OU cold start)
   ├─ Execute:
   │    ├─ Copy inputs/code → container (/tmp via TAR)
   │    ├─ Exec: python /tmp/user_code.py
   │    └─ Extract output.json
   ├─ Upload result → S3
   ├─ Cache result → Redis (TTL 1h)
   └─ Update task → PostgreSQL (status: completed)

Utilisateur → API GET /tasks/{id}
   └─ Return result (Redis cache OU S3)
```

COMPOSANTS CLÉS

A. API FastAPI (Clever Cloud)

Responsabilités : - Authentification Keycloak (OAuth2 + PKCE) - CRUD Functions, Tasks, Workflows, Studies - Build images Docker (Dockerfile generation) - Orchestration jobs (enqueue Redis)

Routes principales : - `POST /auth/login` - OAuth2 flow - `POST /functions` - Create + build image - `POST /functions/{id}/execute` - Launch calcul - `GET /tasks/{id}` - Get status/result - `POST /workflows` - Create workflow - `POST /workflows/{id}/execute` - Execute DAG

B. PostgreSQL API (Scaleway)

Tables principales : - `users` - Utilisateurs (email, company_id, keycloak_id) - `functions` - Fonctions (code, requirements, docker_image) - `tasks` - Exécutions (status, inputs_s3_key, result_s3_key, execution_time) - `workflows` - Workflows (dag JSON, dependencies) - `studies` - Studies (workflows grouping)

C. Redis Queue (Scaleway)

Usage : - `mecapy:jobs` - Queue FIFO principale - `result:{task_id}` - Cache résultats (TTL 1h) - `task:{task_id}` - Status temps réel (running, completed, failed)

D. Worker Python (Bare Metal Hetzner)

Architecture : - **Daemon Python** (supervisord + systemd) - **LRU Container Cache** : 200 containers max (~40GB RAM) - **Docker API** : Exécution isolée (network=none, read-only, caps dropped) - **I/O via TAR** : Copy inputs/code → exec → extract outputs

Sécurité containers : - Network: none - Filesystem: read-only (sauf /tmp tmpfs) - CPU limit: 2 cores - RAM limit: 2GB - Timeout: 5min - Capabilities: dropped ALL

Performance : - Fonction warm (cache hit) : **2.2s** (overhead ~10%) - Fonction cold (cache miss) : **3.5-5s** (pull + boot)

SÉCURITÉ

API

- Keycloak OAuth2 + PKCE
- Validation code (blacklist imports : os, subprocess, sys, socket, eval, exec)
- Rate limiting
- CORS restrictif

Execution

- Isolation Docker (network, filesystem, capabilities)
- Limits CPU/RAM strictes
- Timeout automatique
- Validation statique pré-exécution

PLAN DE MISE EN ŒUVRE (10 SEMAINES)

Semaine 1-2 : Infrastructure

- [ ] Provisionner Scaleway : PostgreSQL (×2), Redis, S3
- [ ] Déployer Keycloak (Scaleway Instance)
- [ ] Commander bare metal Hetzner AX52
- [ ] Setup serveur : Ubuntu, Docker, supervisor
- [ ] Configuration réseau, firewall, SSH

Livrables : - Infrastructure cloud opérationnelle - Serveur bare metal accessible

---

Semaine 3-4 : API Backend

- [ ] Routes CRUD Functions
- [ ] Integration Keycloak auth
- [ ] Build Docker images (Dockerfile generation)
- [ ] Push Container Registry Scaleway
- [ ] Enqueue jobs Redis

Livrables : - API FastAPI déployée Clever Cloud - Authentification fonctionnelle - Build images automatique

---

Semaine 5-6 : Worker Execution

- [ ] Worker daemon (poll Redis)
- [ ] LRU Container Cache (200 containers)
- [ ] ContainerExecutor (I/O via TAR)
- [ ] Upload résultats S3
- [ ] Update status PostgreSQL/Redis

Livrables : - Worker opérationnel sur bare metal - Exécution calculs isolés fonctionnelle - Cache containers performant

---

Semaine 7-8 : Tests & Monitoring

- [ ] Tests unitaires (API + Worker)
- [ ] Tests intégration end-to-end
- [ ] Monitoring (supervisor, logs, métriques)
- [ ] Alerting (email si worker down)
- [ ] Load testing (1000 calculs simultanés)

Livrables : - Suite tests complète - Monitoring opérationnel - Validation charge

---

Semaine 9 : Frontend Dashboard

- [ ] UI création fonctions
- [ ] UI exécution calculs
- [ ] UI visualisation résultats
- [ ] UI workflows (drag & drop)

Livrables : - Dashboard utilisateur fonctionnel - Interface création/exécution

---

Semaine 10 : Production

- [ ] Déploiement API Clever Cloud
- [ ] Configuration DNS
- [ ] SSL/TLS
- [ ] Documentation utilisateur
- [ ] Go live MVP

Livrables : - Plateforme en production - Documentation complète - 10 utilisateurs beta invités

---

⚡ PERFORMANCES ATTENDUES

Métrique	Valeur
Latence warm	2.2s (overhead 10%)
Latence cold	3.5-5s (pull + boot)
Containers simultanés	200 warm + 50 cold
Calculs/jour	~5000 (1 serveur)
Throughput	~50 exécutions/min

🚀 ROADMAP ÉVOLUTION

Phase 1 - MVP (Semaines 1-10)

Objectif : 0-5k calculs/jour

- 1 bare metal Hetzner
- Architecture actuelle
- **Coût** : €117/mois

Critères de succès : - ✅ 100 fonctions créées - ✅ 1000 calculs exécutés - ✅ Latence < 5s (95th percentile) - ✅ Uptime > 99% - ✅ 10 utilisateurs beta actifs

---

Phase 2 - Growth (3-6 mois)

Objectif : 5k-20k calculs/jour

- Multi-workers (2-3 serveurs)

- API route par `hash(function_id) % N`
- Monitoring avancé (Grafana + Prometheus)
- **Coût** : €171-225/mois

Critères de succès : - 500 fonctions - 50k calculs/mois - 50 utilisateurs payants - Revenue > Coût infrastructure

---

Phase 3 - Scale (6-12 mois)

Objectif : > 20k calculs/jour

Options : 1. Migration Firecracker (overhead 100ms vs 2s) 2. Kubernetes (auto-scaling) 3. Évaluation ROI Clever Cloud XL

Décision : Basée sur métriques Phase 2

📊 MÉTRIQUES DE SUIVI

Techniques

- Latence exécution (p50, p95, p99)
- Cache hit rate (warm containers)
- Taux d'erreur
- Uptime worker
- Utilisation CPU/RAM

Business

- Nombre fonctions créées
- Nombre calculs/jour
- Utilisateurs actifs
- Revenue mensuel
- Coût par calcul

🎯 PRIORITÉS DÉVELOPPEMENT

Critiques (MVP)

1. Exécution isolée sécurisée
2. Cache containers performant
3. API CRUD fonctionnelle

4. Authentification Keycloak

Importantes (Post-MVP)

1. Workflows DAG
2. Marketplace communautaire
3. Système de facturation
4. Analytics utilisateur

Nice-to-have (Futur)

1. Intégration Code_Aster
2. Support GPU
3. Multi-région
4. On-premise deployment

RISQUES ET MITIGATION

Risque	Impact	Probabilité	Mitigation
Worker bare metal down	Élevé	Moyenne	Monitoring + alerting + backup serveur
Dépassement budget	Moyen	Faible	Budget tracking mensuel
Faille sécurité sandbox	Élevé	Faible	Audit code + pentesting
Performance insuffisante	Moyen	Moyenne	Load testing précoce
Complexité workflows	Faible	Élevée	Itérations MVP simples

DOCUMENTATION À PRODUIRE

Technique

- [ ] Architecture détaillée
- [ ] API documentation (OpenAPI)
- [ ] Guide déploiement
- [ ] Guide contribution

Utilisateur

- [ ] Getting started

- [ ] Tutoriels création fonctions
- [ ] Référence API SDK
- [ ] FAQ

Document généré le : 2025-09-30 **Version** : 1.0 - Plan de travail MVP **Auteur** : Architecture Team MecaPy

---

🎯 **Objectif : MVP en production en 10 semaines avec budget €117/mois**