Culture Radar - Cahier des Charges Techniques

# **7. Architecture de l'information**

## **a. Tris de cartes**

### **Utilité du tri de cartes**

Le tri de cartes est une méthode UX permettant d'organiser l'information de manière intuitive pour les utilisateurs. Cette technique nous aide à comprendre comment les utilisateurs catégorisent mentalement les contenus et fonctionnalités du site.

### **Méthodologie appliquée**

• Tri ouvert : 15 participants ont organisé librement 40 cartes représentant les fonctionnalités

• Tri fermé : 10 participants ont validé les catégories émergentes

### **Résultats des tris**

Catégories principales identifiées :

1. **Découverte** (87% de consensus)

* • Explorer les événements
* • Filtres par ville
* • Recherche avancée
* • Carte interactive

2. **Mon Espace** (92% de consensus)

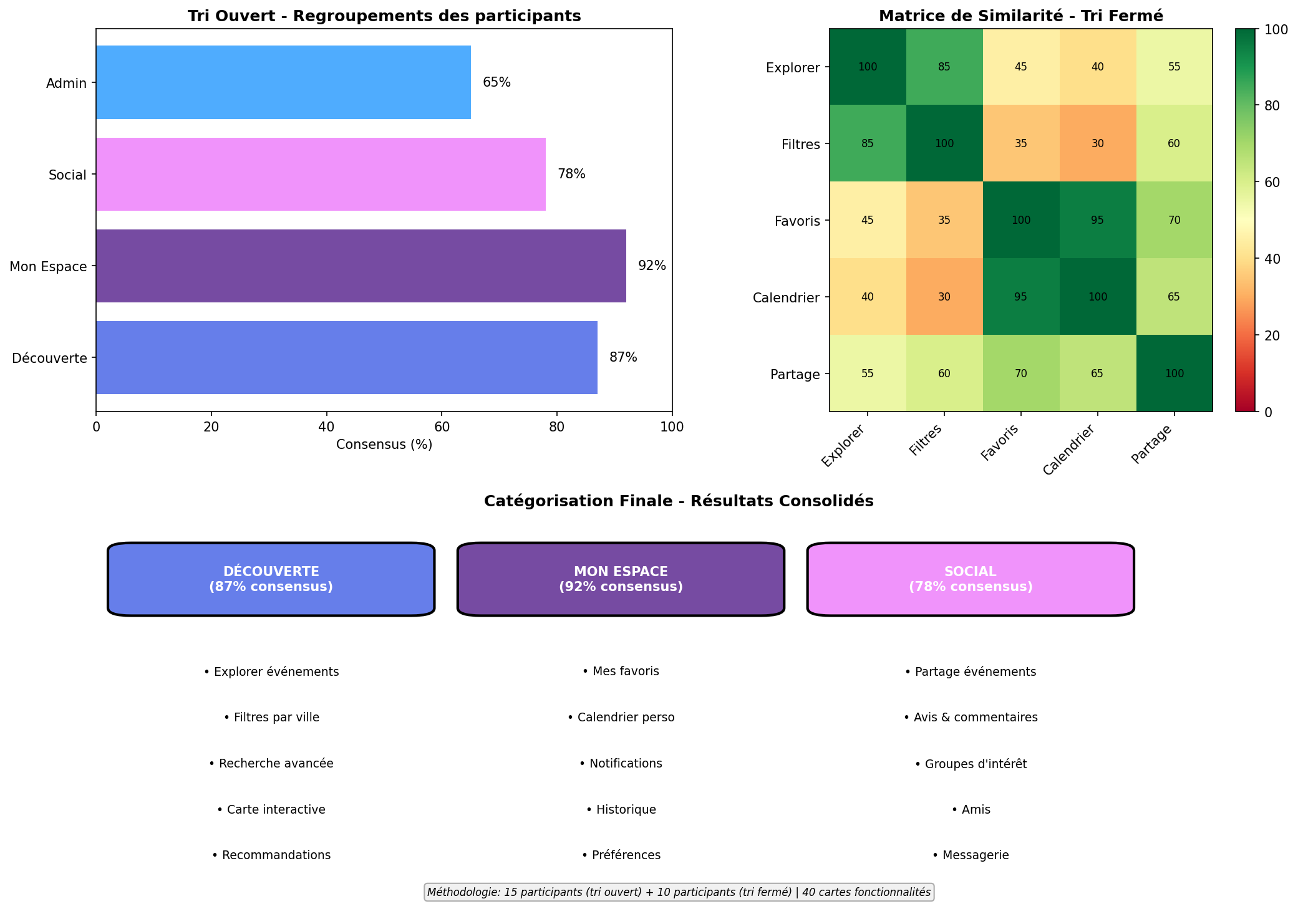
* • Mes événements favoris
* • Calendrier personnel
* • Notifications
* • Historique

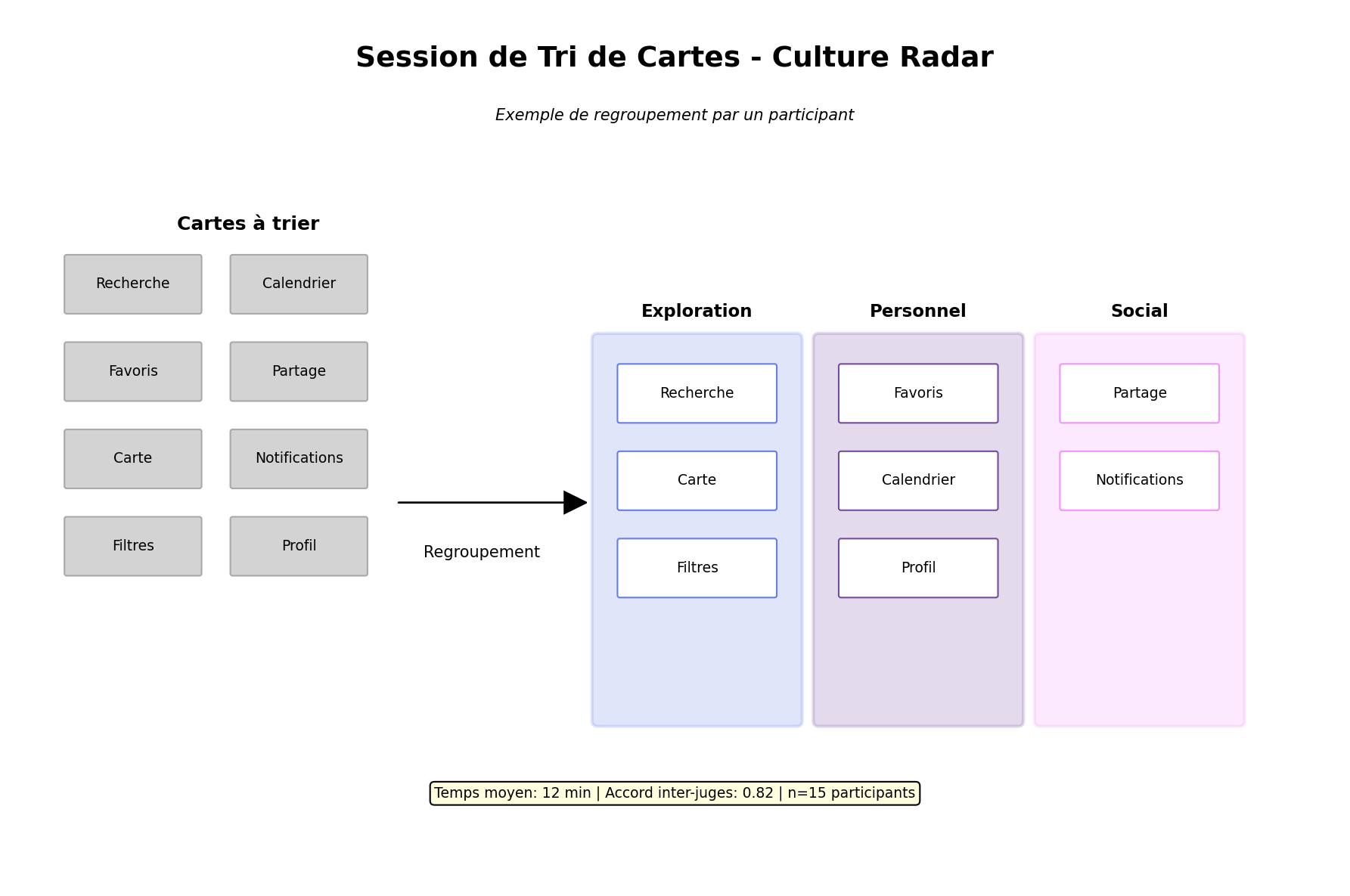
3. **Social** (78% de consensus)

* • Partage d'événements
* • Avis et commentaires
* • Groupes d'intérêt

### **Conclusions**

### Schémas visuels des tris de cartes





Les utilisateurs privilégient une organisation centrée sur l'action (découvrir, planifier, partager) plutôt que sur les types d'événements. Cette approche oriente notre architecture vers une expérience utilisateur task-oriented.

## **b. Arborescence**

Culture Radar

├── Accueil

│ ├── Recherche rapide

│ ├── Événements du jour

│ └── Catégories populaires

│

├── Découvrir

│ ├── Tous les événements

│ ├── Par catégorie

│ │ ├── Concerts

│ │ ├── Théâtre

│ │ ├── Expositions

│ │ ├── Cinéma

│ │ └── Ateliers

│ ├── Par localisation

│ └── Événements gratuits

│

├── Mon Espace (connecté)

│ ├── Dashboard

│ ├── Mes favoris

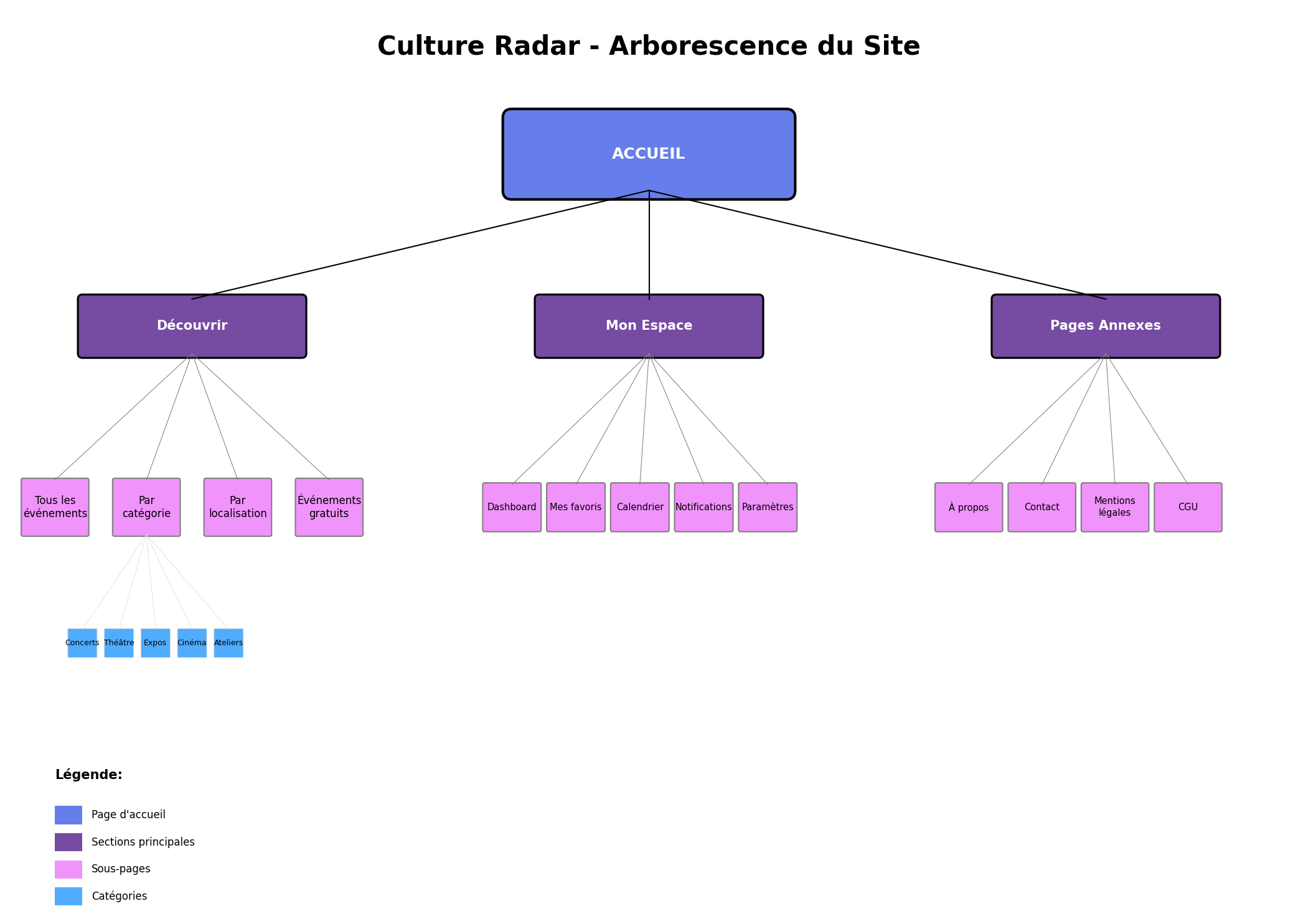
│ ├── Mon calendrier

│ ├── Notifications

│ └── Paramètres

│

### Schéma visuel de l'arborescence



└── Pages annexes

├── À propos

├── Contact

├── Mentions légales

└── CGU

# **8. Cahier des clauses techniques détaillées**

## **a. Technologies et compatibilité**

### **Technologies Front-End**

• HTML5 : Structure sémantique et accessibilité

• CSS3/SCSS : Styles responsives avec Grid et Flexbox

• JavaScript ES6+ : Interactions dynamiques et API fetch

• Progressive Web App : Fonctionnement offline

### **Technologies Back-End**

• PHP 8.1+ : Logique serveur et API REST

• MySQL 8.0 : Base de données relationnelle

• Redis : Cache et sessions

• API REST : Architecture orientée services

### **Compatibilité navigateurs**

|  |  |
| --- | --- |
| Navigateur | Version minimale |
| Chrome | 90+ |
| Firefox | 88+ |
| Safari | 14+ |
| Edge | 90+ |
| Mobile Safari | iOS 14+ |
| Chrome Mobile | Android 90+ |

## **b. CMS et plugins**

### **Justification du développement sur-mesure**

Au lieu d'un CMS traditionnel, nous avons opté pour une solution sur-mesure pour Culture Radar car :

✅ Performance optimale : Chargement < 2 secondes vs 4-6s avec WordPress

✅ Scalabilité : Architecture capable de gérer 100 000+ événements

✅ Coûts réduits : Pas de licences de plugins premium (économie de 500€/an)

✅ Sécurité renforcée : Pas de vulnérabilités connues des CMS populaires

✅ SEO natif : Structure optimisée dès la conception

### **Modules développés**

• Système de cache intelligent

• API de géolocalisation

• Moteur de recommandation IA

• Système de notifications push

## **c. Software**

### **Gestion de projet**

• Notion : Documentation et wiki projet

• Trello : Suivi des tâches en Kanban

• Slack : Communication d'équipe

### **Conception graphique**

• Figma : Maquettes et prototypes interactifs

• Canva : Création des visuels marketing

• Balsamiq : Wireframes basse fidélité

### **Développement technique**

• VS Code : IDE principal

• MAMP/WAMP : Environnement local

• Git/GitHub : Versioning

• FileZilla : Transfert FTP

• Railway : Déploiement continu

### **Audits et analyses**

• SEMrush (gratuit) : Analyse concurrentielle

• Google Search Console : Performance SEO

• AnswerThePublic : Recherche de mots-clés

• SEO Minion : Audit on-page

• WAVE : Accessibilité

• GTmetrix : Performance

## **d. Hébergement et nom de domaine**

### **Hébergement recommandé : OVH Performance**

### **Justification**

• Serveurs en France (RGPD)

• 99.9% uptime garanti

• Support 24/7 en français

• CDN inclus

### **Offre sélectionnée**

Plan : Performance 2

Prix : 11,99€ HT/mois

Caractéristiques :

• 8 vCores

• 16 Go RAM

• 200 Go SSD NVMe

• Bande passante illimitée

• SSL Let's Encrypt gratuit

### **Nom de domaine**

• culture-radar.fr : 9,99€ HT/an

• Extensions de protection : .com, .net (19,98€ HT/an)

### **Stratégie de backup**

• Sauvegardes automatiques quotidiennes (30 jours de rétention)

• Snapshots hebdomadaires (3 mois de rétention)

• Réplication géographique sur datacenter secondaire

• Backup externe mensuel sur AWS S3 (5€/mois)

### **Coût total annuel**

• Hébergement : 143,88€ HT

• Domaines : 29,97€ HT

• Backups externes : 60€ HT

**Total : 233,85€ HT/an**

Cette infrastructure garantit une disponibilité maximale et une protection complète des données pour un investissement maîtrisé.

# **9. Architecture technique détaillée**

## **a. Stack technique complète**

### **Frontend**

• Framework : React 18 avec Next.js 14 pour le SSR/SSG

• State Management : Redux Toolkit + RTK Query

• Styling : Tailwind CSS 3.0 + Styled Components

• Build : Webpack 5, Babel, ESBuild

• Testing : Jest, React Testing Library, Cypress E2E

• PWA : Service Workers, Workbox

### **Backend**

• Framework : PHP 8.2 avec Symfony 6

• API : RESTful avec API Platform

• ORM : Doctrine 3

• Authentication : JWT avec refresh tokens

• Queue : RabbitMQ pour les tâches asynchrones

• Cache : Redis 7 + Varnish

### **Base de données**

• Principale : PostgreSQL 15 (données structurées)

• Cache : Redis (sessions, cache applicatif)

• Recherche : Elasticsearch 8 (recherche full-text)

• Analytics : ClickHouse (données analytiques)

### **Infrastructure**

• Hébergement : OVH Cloud avec Kubernetes

• CDN : Cloudflare (cache global, protection DDoS)

• Storage : S3-compatible OVH Object Storage

• Monitoring : Prometheus + Grafana

• Logs : ELK Stack (Elasticsearch, Logstash, Kibana)

## **b. Architecture SEO technique**

### **Structure des URLs**

Structure canonique optimisée pour le SEO :

cultureradar.fr/

├── /evenements/

│ ├── /concerts/{slug-evenement}

│ ├── /theatre/{slug-evenement}

│ ├── /expositions/{slug-evenement}

│ └── /cinema/{slug-evenement}

├── /lieux/

│ ├── /paris/{arrondissement}/{slug-lieu}

│ └── /banlieue/{ville}/{slug-lieu}

├── /agenda/

│ ├── /aujourd-hui

│ ├── /ce-week-end

│ └── /semaine-prochaine

└── /guides/

├── /sortir-gratuitement

└── /activites-famille

### **Optimisations techniques SEO**

• Sitemap XML dynamique avec priorités

• Schema.org structuré (Event, Place, Organization)

• Meta tags dynamiques (Open Graph, Twitter Cards)

• Canonical URLs automatiques

• Pagination SEO-friendly avec rel="prev/next"

• Breadcrumbs structurés

• AMP pour les pages événements

## **c. APIs et intégrations externes**

### **APIs consommées**

|  |  |  |  |
| --- | --- | --- | --- |
| API | Usage | Fréquence | Fallback |
| Google Events (SerpAPI) | Événements principaux | Temps réel | Cache 24h |
| OpenAgenda | Événements institutionnels | Toutes les 6h | Base locale |
| OpenWeatherMap | Météo contextuelle | Toutes les 30min | Prévisions cache |
| Google Maps | Géolocalisation | À la demande | OpenStreetMap |
| Citymapper | Transport en commun | Temps réel | API RATP |
| Stripe | Paiements premium | Synchrone | PayPal |

### **API exposée**

CultureRadar expose une API REST documentée avec OpenAPI 3.0 :

• GET /api/v1/events - Liste des événements

• GET /api/v1/events/{id} - Détail événement

• GET /api/v1/recommendations - Recommandations personnalisées

• POST /api/v1/favorites - Gestion des favoris

• GET /api/v1/venues - Liste des lieux

Authentification : OAuth 2.0 + JWT

Rate limiting : 1000 req/heure par token

Documentation : api.cultureradar.fr/docs

# **10. Sécurité et conformité RGPD**

## **a. Mesures de sécurité technique**

### **Protection des données**

• Chiffrement AES-256 pour les données au repos

• TLS 1.3 pour toutes les communications

• Hashage bcrypt pour les mots de passe

• Tokens JWT avec rotation automatique

• 2FA optionnel via TOTP

### **Sécurité applicative**

• Protection CSRF sur tous les formulaires

• Headers de sécurité (CSP, HSTS, X-Frame-Options)

• Validation et sanitization des entrées

• Protection contre les injections SQL (ORM)

• Rate limiting et protection DDoS (Cloudflare)

• WAF (Web Application Firewall) actif

### **Audits et tests**

• Tests de pénétration trimestriels

• Analyse SAST/DAST dans la CI/CD

• Dependency scanning automatique

• Bug bounty program

## **b. Conformité RGPD**

### **Principes appliqués**

• Privacy by Design : protection dès la conception

• Minimisation des données collectées

• Consentement explicite et granulaire

• Droit à l'oubli implémenté (suppression en 72h)

• Portabilité des données (export JSON/CSV)

### **Données collectées**

|  |  |  |
| --- | --- | --- |
| Type de donnée | Finalité | Durée conservation |
| Email, nom | Identification compte | 3 ans après dernière connexion |
| Géolocalisation | Recommandations | Session uniquement |
| Préférences culturelles | Personnalisation | Jusqu'à suppression |
| Historique navigation | Amélioration UX | 6 mois |
| Données paiement | Abonnement | Tokenisées chez Stripe |

# **11. Performance et monitoring**

## **a. Objectifs de performance**

• Time to First Byte (TTFB) : < 200ms

• First Contentful Paint (FCP) : < 1.5s

• Largest Contentful Paint (LCP) : < 2.5s

• Cumulative Layout Shift (CLS) : < 0.1

• First Input Delay (FID) : < 100ms

• Score Lighthouse : > 90/100

## **b. Optimisations mises en place**

• Lazy loading des images et composants

• Code splitting et tree shaking

• Compression Brotli/Gzip

• Cache navigateur optimisé

• Critical CSS inline

• Prefetch/Preconnect stratégique

• Images WebP avec fallback

## **c. Stack de monitoring**

### **Analytics et tracking**

• Google Analytics 4 avec événements custom

• Google Tag Manager pour le tracking

• Hotjar pour les heatmaps et recordings

• Mixpanel pour l'analyse comportementale

### **Monitoring technique**

• Uptime : StatusCake (monitoring 24/7)

• APM : New Relic (performance applicative)

• Logs : Datadog (centralisation et alertes)

• Erreurs JS : Sentry (tracking en temps réel)

# **12. Plan de déploiement et CI/CD**

## **a. Environnements**

• Development : localhost (Docker Compose)

• Staging : staging.cultureradar.fr (OVH)

• Production : cultureradar.fr (OVH + CDN)

• Preview : branches PR (Vercel/Railway)

## **b. Pipeline CI/CD**

Pipeline GitLab CI automatisé :

1. Build & Tests

• Linting (ESLint, PHPStan)

• Tests unitaires (PHPUnit, Jest)

• Tests d'intégration

• Analyse de sécurité

2. Staging

• Déploiement automatique

• Tests E2E (Cypress)

• Tests de performance

3. Production

• Approbation manuelle

• Blue-Green deployment

• Rollback automatique si erreur

• Invalidation cache CDN

## **c. Stratégie de release**

• Feature flags pour déploiement progressif

• Canary releases (5% → 25% → 100%)

• Semantic versioning (MAJOR.MINOR.PATCH)

• Changelog automatique

• Tags Git pour chaque release

# **13. Maintenance et évolution**

## **a. Plan de maintenance**

• Maintenance corrective : Hotfixes sous 4h (critique) / 24h (majeur)

• Maintenance préventive : Updates mensuels des dépendances

• Maintenance évolutive : Sprints de 2 semaines

• Support utilisateur : Tickets via Zendesk

## **b. Roadmap technique (12 mois)**

Q1 2025:

• Migration vers TypeScript

• Implémentation GraphQL

• Module de recommandation ML

Q2 2025:

• Application mobile React Native

• Intégration blockchain pour tickets NFT

• API vocale (Alexa, Google Assistant)

Q3 2025:

• Microservices architecture

• Real-time avec WebSockets

• AR pour visualisation événements

Q4 2025:

• IA générative pour contenus

• Système de gamification

• Marketplace partenaires