Student name: MBEMDIA SALIOU Mohamed Mahzou Tariq

Matricule : : ICTU20251574

Course code : 6114

Teacher's name(Dr Fosso)

Project : Design, engineering, and securing of a full software and cloud infrastructure for a nationwide smart tourism platform (NaTourCam).

Project presentation

In the era of digital transformation, Cameroon, often described as “Africa in miniature,” faces the challenge of modernizing its tourism sector. The NaTourCam project was born out of a desire to create an integrated national platform capable of digitizing, promoting, and securing the country's exceptional cultural and natural heritage.

Unlike traditional information systems, NaTourCam is designed as a complex socio-technical system. It must serve a massive user base of 7 million people while respecting cultural diversity represented by more than 250 national languages. The infrastructure is based on strategic geographic distribution across the country's 10 regions, requiring high-precision software engineering to guarantee performance and data sovereignty.

Part 1 : DDD and Agile Architecture

1. Bounded Contexts

The decomposition of NaTourCam into bounded contexts follows Domain-Driven Design principles, ensuring that each domain encapsulates a coherent business responsibility, aligns technical models with cultural realities, and enables independent evolution of services.

Context	Liability	Key entities
Cultural Heritage	● Management of cultural content (stories, rituals, languages, objects) ● Approval by cultural	● CulturalArtifact ● Language ● Translation ● CulturalAuthority

	authorities	
Tourism Sites Context	• Management of tourist sites (parks, museums, chiefdoms) • Geographic and logistical information	• TourismSite • SiteCategory • OpeningSchedule • Region
User Profiles Context	• User management (tourists, partners, administrators)Préférences linguistiques et culturelles	• User • Role • Preference • AccessibilitySettings
Ticketing & Payments Context	• Ticket sales • Integration of mobile payments (Mobile Money)	• Ticket • PaymentTransaction • PaymentProvider • Invoice
Recommendation Engine Context	• Personalized suggestions • User behavior analysis	• RecommendationModel • UserBehavior • ScoringRule
Geo-Services Context	• Geolocation • Itineraries • Security alerts	• GeoLocation • Itinerary • SafetyAlert
Partner Integration Context	• APIs for hotels, agencies, and transport companies • Integration contract management	• Partner • PartnerAPIKey • Contract
Analytics & Governance Context	• Dashboards • Statistics for the Department of	• KPI • Report • DataAggregation

	Tourism	

2. Ubiquitous Language

The Ubiquitous Language establishes a shared vocabulary between developers, cultural experts, tourism operators, and government stakeholders, reducing ambiguity and ensuring semantic consistency across the system.

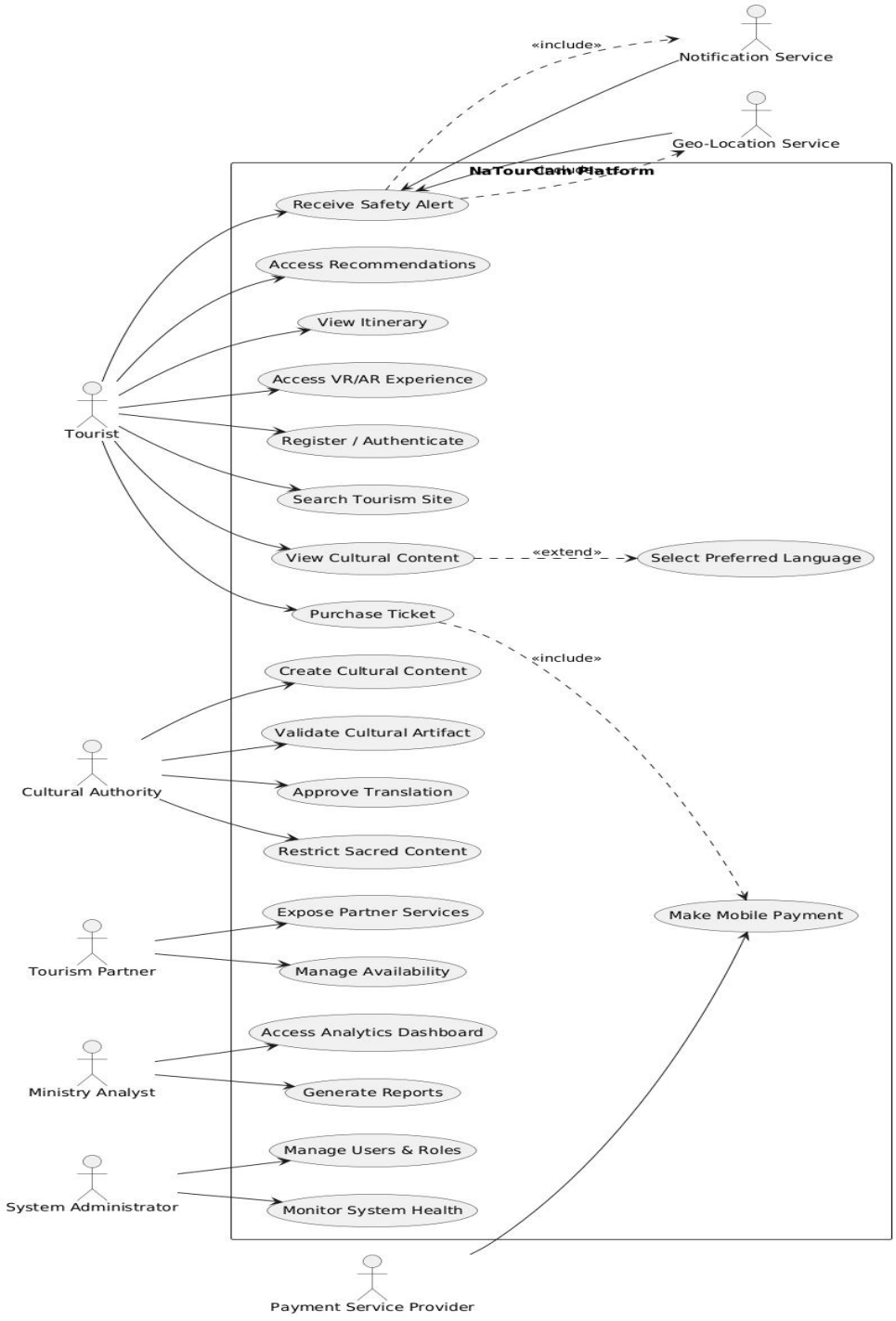
Contexts	Terms	Job Definition
Cultural Heritage	Cultural Artifact	Validated cultural element representing local identity
	Cultural Authority	Institution or chiefdom authorizing dissemination
	Translation	Official language version of content
	Sacred Content	Restricted access content
Ticketing & Payments Context	Ticket	Digital right of access to a website
	Payment Transaction	Traceable financial transaction
	Mobile Payment	Payment via Mobile Money
	Refund	refund check
Geo-Services	Itinerary	Recommended geolocated route
	Safety Alert	Real-time risk notification
	Geo-Fence	Virtual zone triggering an alert

3. Choix de la Méthodologie Agile : SCRUM

The Scrum framework is selected as the primary Agile methodology for the NaTourCam project due to its adaptability, strong stakeholder engagement mechanisms, and suitability for large-scale, distributed systems.

- Scalability : Scrum allows the use of the “Scrum of Scrums” framework to coordinate teams distributed between Yaoundé, Buea, Douala, and Maroua..
- Uncertainty :Short cycles (Sprints) allow for rapid testing of user acceptance of VR/AR services.
- Stakeholder Diversity : Sprint Reviews provide a structured framework for incorporating feedback from cultural guardians, government, and tourism agencies.
- Incremental Delivery : Allows you to launch basic services (ticketing) first before adding complex modules such as the recommendation engine.

4. Major Interactions



5. Contribution of Agility to Collaboration

The implementation of Scrum for NaTourCam is not only about coding faster, but also about building bridges between very different worlds (technology, tradition,

administration). According to Sommerville, the success of a complex system depends on the involvement of stakeholders throughout the process.

a. With Linguists and Cultural Guardians

- **Sprints Reviews & Démos :** At the end of each sprint (2 to 4 weeks), we present the translation prototypes or 3D models to the heritage guardians. This allows them to immediately validate whether the representation of a sacred artifact or local language (e.g., Ewondo, Duala, Fufulde) is accurate and respectful.
- **Product Backlog Refinement :** Linguists help refine user stories to ensure that àcultural nuances are incorporated from the design phase onwards, thereby avoiding costly errors at the end of the project.

B. With Local Tour Operators

- **Incremental Delivery :** Instead of waiting two years, tourism agencies receive functional modules quickly (e.g., the booking module). This allows them to adjust their commercial offerings based on the features actually available on the platform.
- **Feedback Loops :** Local guides test the beta versions in the field (in national parks) and report any connectivity or usability issues directly to the development team.

C. With Government Agencies (Regulation)

- **Transparency and Visibility :**The Product Owner (often a representative of the Ministry) has a clear view of progress via the Information Radiator (Kanban/Scrum Board). This facilitates accountability and alignment with national digitization policies.
- **Adaptability to legislative changes:** If the government changes a law on data

protection or tourist taxes, agility allows this change to be incorporated into the next Sprint without disrupting the entire project.

D. Among Developers (Technique)

- **Daily Scrums :** Technical teams distributed across 10 regions synchronize daily. This reduces silos and ensures that developers in Maroua work to the same standards as those in Douala.

In a project like NaTourCam, a rigid approach (Waterfall) would be a failure because cultural needs are too nuanced to be set out in a 500-page document at the start of the project. Agility allows for co-construction of the system: the technology adapts to Cameroonian culture, not the other way around.

Partie 2 : SOFTWARE ARCHITECTURE and MICROSERVICES

A. Microservices Architecture

1. Decomposition into microservices

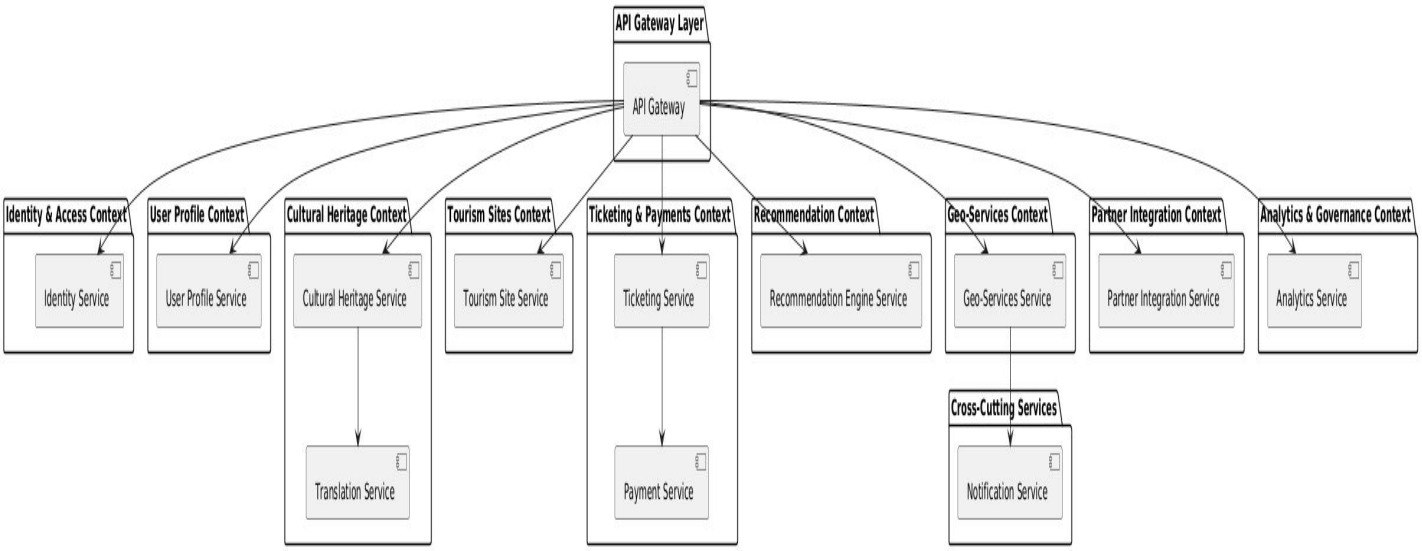


Image 1 : decomposition into microservices

2. Component diagram

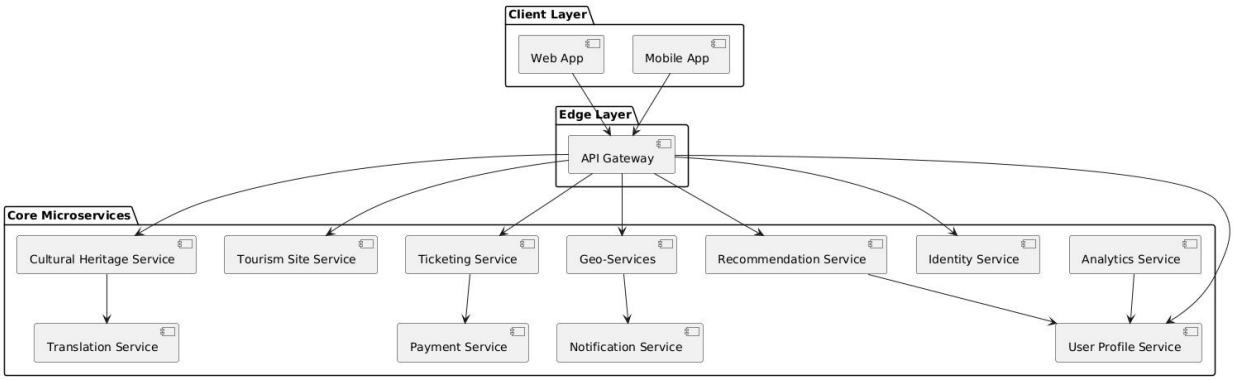


Image 2 : component diagram

This component diagram illustrates a cloud-native microservices architecture where each service encapsulates a specific business capability and communicates through well-defined APIs via an API Gateway.

3. Support for multilingualism and cultural diversity

Architectural approach

- Strict separation:
  - **Cultural Heritage Service ->** source content
  - **Translation Service** -> language versions
- Each piece of content is associated with:
  - a language
  - a validation status
  - a cultural authority
- Automatic fallback :
  - Local language → French/English

Key advantage :

- No data duplication
- Compliance with cultural norms

4. Role of the API Gateway and Service Mesh

API Gateway:

- Centralized authentication
- Traffic limitationAgrégation de réponses
- Protection against DDoS attacks

Service Mesh :

- Internal TLS encryption
- Circuit breaking
- Observability (metrics, traces)
- Detailed management of security policies

B. Data Management and APIs

5. Distributed data architecture (SQL + NoSQL)

Fundamental principle: Polyglot Persistence — each microservice chooses the database that best suits its needs.

Proposed architecture

Micorservice	Type de DB	Justification
User, Ticketing, Payment	PostgreSQL (SQL)	ACID Transactions
Cultural Content	MongoDB (NoSQL)	Semi-structured data
Geo-Services	PostGis	Spatial queries
Recommendation Engine	NoSQL / Vector DB	Quick scoring
Analytics	Data Warehouse	Massive aggregation
Cache	Redis	Performance

6. OpenAPI — APIs critiques

a. Tourism Site Search API

```
GET /api/v1/sites/search
parameters:
  - name: region
    in: query
    schema:
      type: string
  - name: language
    in: query
    schema:
      type: string
responses:
  '200':
    description: List of tourism sites
```

b. Cultural Content API

```
GET /api/v1/cultural-content/{siteId}
parameters:
  - name: language
    in: query
    schema:
      type: string
responses:
  '200':
    description: Cultural content in selected language
```

C. Performance and Scalability

7. Scalability and performance

a) Horizontal vs Vertical Scaling

- Horizontal : adding pods
- Vertical : increase in resources (limited)

b) Caching Strategies

- Redis : sessions, profils
- CDN : images, videos, AR/VR contents
- HTTP Cache Headers

C) Management of 500,000 simultaneous visitors

- Auto-scaling Kubernetes
- Load Balancers multi-régions
- CDN mondial
- Architecture event-driven
- Graceful degradation

The system prioritizes availability and user experience through elastic scaling and intelligent caching.

PART III : CI/CD, DEVOPS and SOFTWARE QUALITY

1. Design of the complete CI/CD pipeline

The CI/CD pipeline ensures continuous validation of code quality, security, and deployability, enabling frequent and reliable releases.

a) Linting

- ESLint / SonarQube
- Early error detection

b) Automated tests

- Unit tests
- Integration tests

c) Security scanning

- Scan dependencies
- Scan Docker images

d) Container build

- Build Docker images
- Semantic versioning

e) Deployment to kubernetes

- Automated deployment via Helm
- Environments: dev -> staging -> production

2. Example of a CI/CD pipeline (GitHub Actions)

```
name: NaTourCam CI/CD Pipeline
```

```
on:
  push:
    branches: [ "main", "develop" ]

jobs:
  build-test-deploy:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v4

    - name: Set up Node.js
      uses: actions/setup-node@v4
      with:
        node-version: '18'

    - name: Install dependencies
      run: npm install

    - name: Run linting
      run: npm run lint

    - name: Run unit tests
      run: npm test

    - name: Security scan
      uses: aquasecurity/trivy-action@master
      with:
```

```
    image-ref: myservice:latest


- name: Build Docker image
  run: docker build -t natourcam-service .


- name: Push to registry
  run: docker push natourcam-service


- name: Deploy to Kubernetes
  run: helm upgrade --install natourcam ./helm
```

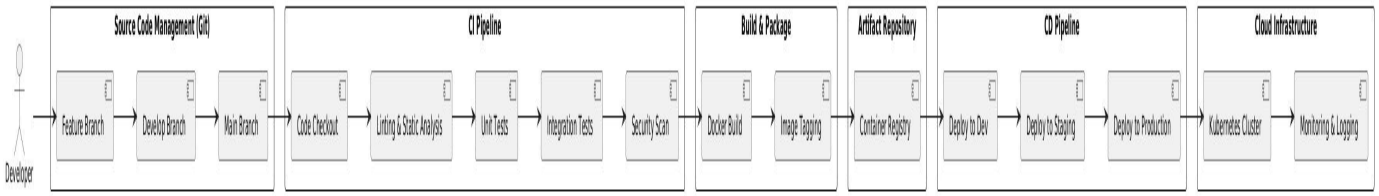


Image 3 : CI/CD Pipeline UML Diagram

3. Testing strategy

Model-Based Testing ensures consistency between system models and implementation, reducing regression risks in complex distributed systems.

a. Units Tests

- Target isolated services
- Executed on every commit

b. Integration tests

- Verify communication between microservices
- Use ephemeral containers

c. Load tests

- Tools: JMeter / k6
- Simulation of 500,000 users
- Response time measurement

d. Tests UI/UX

- Cypress / Playwright
- Tests multi-langues

e. Model-Based Testing (MBT)

- Generation of scenarios based on UML models
- Validation of critical business flows (ticketing, payment)

4. Software quality KPIs

KPI	Description	Target
Availability	Time of availability	>= 99.9%
MTTR	Average repair time	< 30 min
Test Coverage	Test coverage	>= 80%
Response Time	API response time	< 300 ms
Deployment Frequency	Deployments	Daily
Change Failure Rate	Post-deployment failures	<5%

5. Contribution of DevOps for distributed teams

Key benefits:

- **Automation -> fewer human errors**

- **Shared visibility -> same metrics for all regions**
- **Rapid feedback -> early detection of problems**
- **Standardization -> same pipelines everywhere**

PART IV : CLOUD ARCHITECTURE, CONTAINERS and ORCHESTRATION

Cloud and Virtualization

1. Select the appropriate cloud model for NaTourCam: Hybrid Cloud

2. Justification for the choice::

a) Cost:

- The public cloud absorbs seasonal peaks (holidays, festivals)
- The private cloud hosts sensitive data
- CapEx/OpEx optimization

B) Performance:

- Data close to regional users
- Reduced latency through edge computing

C) Sovereignty :

- Sensitive cultural data remains hosted in Cameroon
- Sensitive cultural data remains hosted in Cameroon

D) Cultural Data Protection:

- Logical separation of sacred data
- Control by cultural authorities

3. Multi-region deployment architecture

The multi-region deployment architecture enhances resilience, fault tolerance, and regional autonomy.

Conceptual view :

- 1 National Private Cloud (Yaoundé – core services)

- 9 Satellite regions (edge nodes)
- Controlled data replication
- Geographic load balancing

Containerization & Kubernetes

4. Dockerfile — Ticketing Service

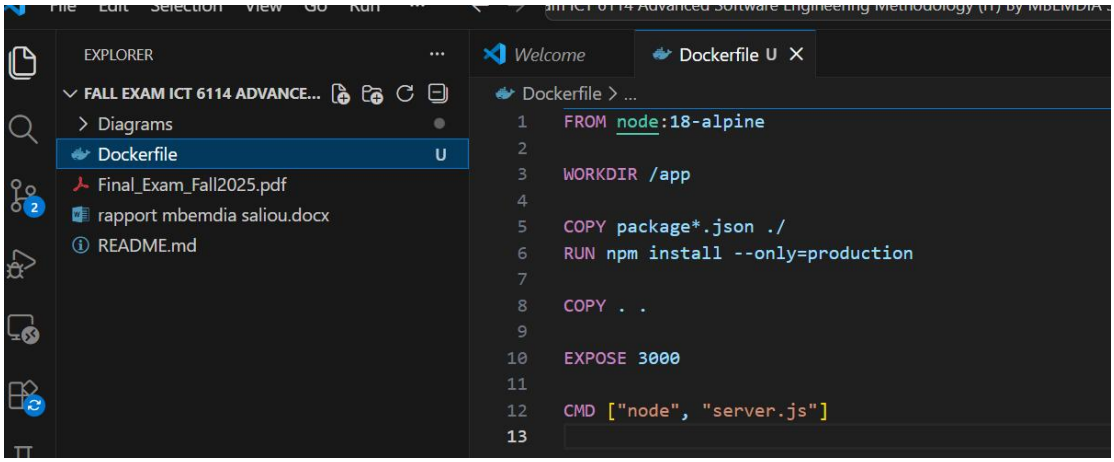


Image 4: dockerfile image

5. Kubernetes Deployment & Service

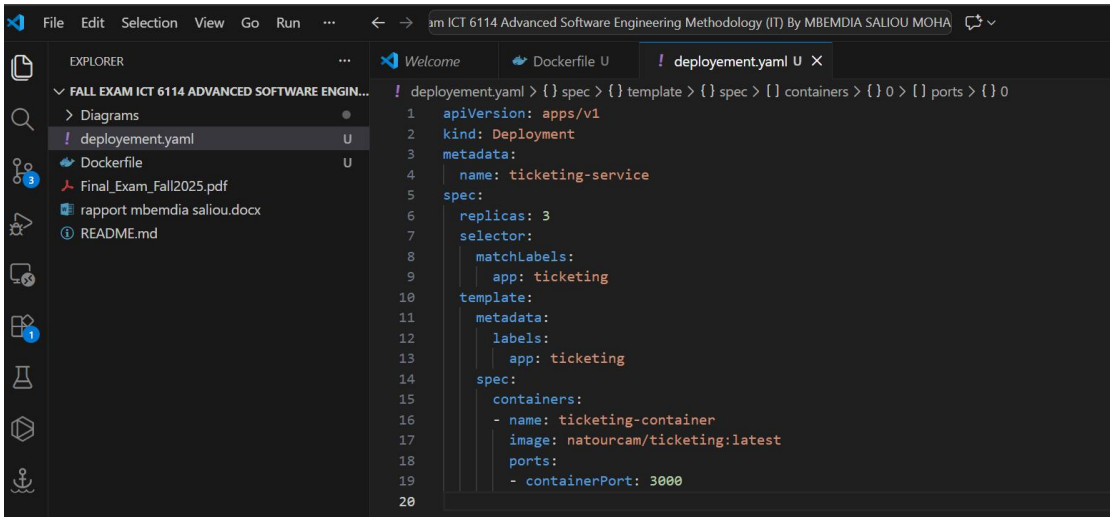


Image 5 : deployement.yml file

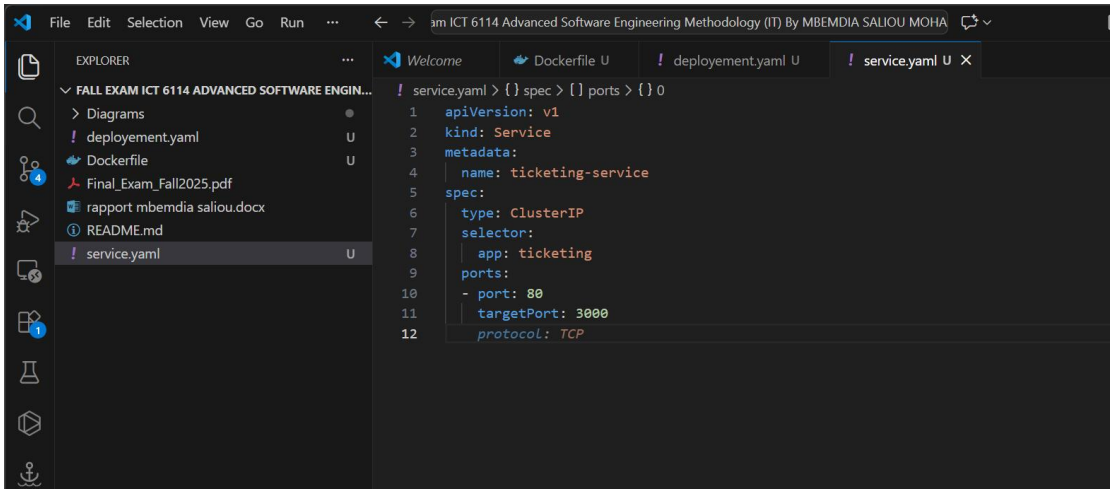


Image 6: service.yaml file

6. Kubernetes concepts explained

a. Auto-scaling

Auto-scaling refers to the system's ability to automatically adapt its resources based on actual load.

In the context of NaTourCam, this means that when the number of visitors increases suddenly (for example, during a cultural festival or holiday period), the platform automatically creates new service instances in order to maintain an acceptable response time.

b. Rolling Updates

Rolling updates are a strategy for gradually updating services without interrupting the system.

Rather than shutting down the entire platform to deploy a new version, Kubernetes gradually replaces old instances with new ones, while keeping the service accessible to users.

In a critical system such as NaTourCam, this approach is essential because it allows :

- introduce new features,,
- fix security vulnerabilities,
- or improve performance,

c. Self-Healing

Self-healing refers to the system's ability to automatically repair itself in the event of a failure.

If a system component fails (for example, a container that stops responding), Kubernetes detects this anomaly and automatically restarts a new instance of the affected service.

This feature is particularly important for NaTourCam, as it ensures high availability even in a distributed environment spanning multiple regions.

Thanks to self-healing, users continue to access tourism services without noticing the underlying technical incidents.

Together, auto-scaling, rolling updates, and self-healing enable NaTourCam to deliver a resilient, scalable, and continuously available platform, capable of supporting national-scale tourism services.

7. Impact of latency on the recommendation engine

Latency is the response time between a user action and the system's reaction.

In the context of NaTourCam, high latency can have a direct and negative impact on the relevance and effectiveness of the tourist recommendation engine.

When recommendations (sites to visit, itineraries, cultural activities) are generated with a significant delay, they may become obsolete or ill-suited to the tourist's actual situation. For example, a visitor on the move may receive a suggestion too late to be useful, reducing the perceived value of the service.

To limit this impact, NaTourCam adopts several strategies:

- Regional deployment of services to bring recommendation calculations closer to users,
- Use of cache mechanisms to quickly provide pre-calculated recommendations,
- use of asynchronous processing, allowing suggestions to be updated without blocking the user interface.

By controlling latency, NaTourCam's recommendation engine remains responsive,

relevant, and contextual, contributing to a better tourist experience and effectively promoting Cameroon's heritage.

PART V : SECURITY, ETHICS and DATA GOVERNANCE

1. Zero Trust security model adapted to NaTourCam

The Zero Trust model is based on a fundamental principle : “Never trust, always verify.”.

In NaTourCam, no user, service, or component is considered trustworthy by default, even if it is located within the network.

In practical terms, this means that :

- Each request is authenticated
- Each access is explicitly authorized
- Each action is logged

This model is particularly well suited to NaTourCam, as the platform :

- is distributed across multiple regions,
- exposes APIs to external partners,
- and handles sensitive cultural data.

2. Safety mechanisms

a. Authentication (OIDC / OAuth2)

NaTourCam uses :

- **OAuth2 for access delegation**,
- **OpenID Connect (OIDC) for user authentication**.

Each user receives a secure token after authentication, which is then verified by the API Gateway before any interaction with the microservices.

b. Access Control (RBAC / ABAC)

Two complementary models are used :

- RBAC (Role-Based Access Control)

Rights are assigned according to role (tourist, cultural authority, administrator, partner).

- ABAC (Attribute-Based Access Control)

Access decisions also take into account: the region, the type of content (sacred or public), and the context of use

This combination is essential for managing complex cultural rules.

c. Data Encryption

- **Encryption in transit** : HTTPS / TLS
- **Encryption at rest** : bases de données chiffrées
- **Secure key management** : Key Management Service (KMS)

This means that even in the event of unauthorized access to the infrastructure, the data remains unusable.

3. Ethical risks associated with the digitization of cultural heritage

The digitization of Cameroon's heritage raises several major ethical risks. :

A. Cultural appropriation

Distribution of content without the consent of local communities

B. Excessive distortion or simplification

Loss of the real cultural meaning of artifacts

C. Unfair commercial exploitation

Unregulated monetization of sacred content

D. Monitoring and tracking of tourists

Invasion of privacy through geolocation

4. mitigation strategies

a. Data Misuse

- Strong authentication
- Continuous monitoring
- Network segmentation

b. Misrepresentation of Cultural Artifacts

- Mandatory approval by cultural authorities

- Possibility of content removal

c. Unauthorized Access

- Strong authentication
- Continuous monitoring
- Network segmentation

d. Geo-location Tracking Risks

- Data anonymization
- Can be disabled by the user
- Time limit on storage

5. Security incident response plan

In the event of a leak or compromise of cultural data :

- ✓ Step 1 — Detection: Automatic alerts, Log analysis
- ✓ Step 2 — Containment: Isolation of compromised services, Revocation of access
- ✓ Step 3 — Eradication: Fixing the vulnerability, updating systems
- ✓ Step 4 — Recovery: Secure restoration, Data integrity verification
- ✓ Step 5 — Communication: Notification of relevant authorities, Controlled transparency

By integrating security, ethics, and governance into its architecture, NaTourCam ensures that technological innovation respects cultural identity, user privacy, and national sovereignty.