

PROJET STORM – MT4

Système de Messagerie Temps Réel Distribué

Objectifs cibles

- 100 000 connexions simultanées
 - 500 000 messages / seconde
 - Budget maximal : 700 € (AWS ou équivalent)
-

Concept pédagogique

Les étudiants conçoivent et développent le **backend d'une application de messagerie temps réel** (type Slack / Discord).

Pas de cours magistraux :

- Apprentissage par la pratique
 - Feedback continu
 - Accompagnement par un **professionnel en rôle de Tech Lead**
-

Organisation des notes (EU)

- 4 matières dans l'UE
 - 3 notes de contrôle continu (QCM rapides – contrôle des connaissances)
 - 1 note majeure basée sur le projet STORM
 - Le barème détaillé est inclus dans le document
-

Stack technique (suggestions – non imposée)

Langage (au choix de l'équipe)

- Go
- Rust
- Node.js
- Java
- Python

- C#

Orchestration / Déploiement

- Kubernetes (EKS / GKE)
- Docker Swarm
- Serverless (AWS Lambda)
- Machines virtuelles classiques

Message Broker

- Kafka
- RabbitMQ
- Redis Streams
- NATS
- Amazon SQS / SNS
- Pulsar

Base de données

- PostgreSQL
- MongoDB
- DynamoDB
- ScyllaDB
- Cassandra
- CockroachDB

Cache

- Redis
- Memcached
- KeyDB
- Hazelcast

Temps réel

- WebSocket natif
- Socket.io
- Server-Sent Events (SSE)
- gRPC streaming
- Pusher / Ably

Authentification

- AWS Cognito
- Auth0
- Keycloak
- Firebase Auth
- JWT custom

Observabilité

- Prometheus / Grafana
- Datadog
- AWS CloudWatch
- ELK Stack
- Jaeger / Zipkin

Infrastructure & CI/CD

- Terraform
- Pulumi
- CloudFormation
- GitHub Actions
- GitLab CI
- Jenkins

Phases du projet

| Phase | Contenu | Matière |
|-------------------|--|-----------------------|
| Architecture | Design système, IaC, CI/CD, contrat API | Ingénierie logicielle |
| Développement | Microservices, tests (>80% coverage), scans sécurité | Qualité & Sécurité |
| Performance | Tests de charge, profiling, optimisations | Performance |
| Chaos Engineering | Pannes simulées (services tués, latence, spikes) | Tests & Sécurité |
| STORM DAY | Lancement viral simulé (100K users + incidents) | Toutes |
| Analyse finale | Post-mortems, rapport technique, soutenance | Revues de code |

STORM DAY

Journée clé du projet :

- Simulation d'un **lancement viral**
 - Montée rapide à **100 000 utilisateurs simultanés**
 - Déclenchement volontaire d'**incidents techniques**
 - Évaluation sur la **résilience, la communication et la réaction de l'équipe**
-

Rôle du professionnel (Tech Lead)

L'intervenant agit comme un **Tech Lead senior**, et non comme un développeur à la place des étudiants :

- Stand-ups réguliers (suivi & déblocage)
 - Sessions de pair-programming à la demande
 - Code reviews et feedback architecture
 - Game Master lors du Storm Day (incidents)
 - Accompagnement à la montée en compétences
-

Évaluation du projet

| Critère | Poids | Matière |
|---|-------|--------------------|
| Qualité de l'architecture et du code | 35 % | Qualité logicielle |
| Tests & sécurité (couverture et pertinence) | 20 % | Tests & sécurité |
| Résultats du Storm Day (SLO atteints) | 25 % | Performance |
| Documentation & post-mortems | 10 % | Revues |
| Collaboration & communication | 10 % | Transverse |