

Auteurs : BINAN Yohan – BERNARD Cédric – SIBOUT Matthis

## CONTEXTE ET OBJECTIF

SPEEDERNET (50 p à Lyon) est spécialisée dans le **Digital Learning par pédagogie immersive**. Elle édite à la fois des contenus pédagogiques et leurs outils SPHERE de conception Web en réalité virtuelle ou vidéo 360.

Pour homogénéiser le développement en cours de nombreux services SPHERE, cette 3<sup>ème</sup> collaboration avec l'Esisar répond à 2 objectifs distincts :

- gérer les licences utilisateurs des différents modules SPHERE (actuels ou futurs)
- proposer une solution de partage des contenus immersifs associés.



## MÉTHODES ET DÉVELOPPEMENTS

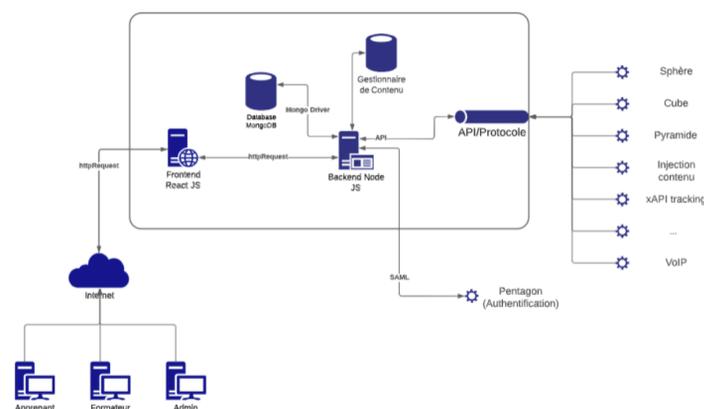
Le projet est divisé en cinq grandes phases, autour d'une réalisation en mode agile SCRUM :

- **Analyse et cadrage** itératif de la problématique face aux nombreux modules, cas d'usage, rôles et droits associés définis par le CdC fonctionnel initial de Speedernet
- Conception d'une **architecture fonctionnelle** puis **technique** compatible avec la plateforme historique SPEEDERNET et ses évolutions cloud, autour d'une authentification SSO (Single Sign On) SAML
- Développement d'une **interface Web** permettant aux administrateurs d'interagir avec le gestionnaire de droit et de contenus et de tester itérativement les modules.
- **Développement d'une API** appelée par les modules SPEEDERNET pour connaître le rôle, les droits et les accès de chaque utilisateur d'une flotte client.
- **Test** avec les modules historiques de SPHERE, ainsi que rédaction de la **documentation technique**.

Challenges principaux : gestion de la base de données NoSQL utilisée (MongoDB) et communication SAML avec le serveur d'authentification de SPEEDERNET.

### Outils :

- Gestion de projet : Gitea (code), et Jira (SCRUM)
- Interfaces web : Bootstrap, React
- Serveur : Nodejs, Typescript
- Base de Données : MongoDB

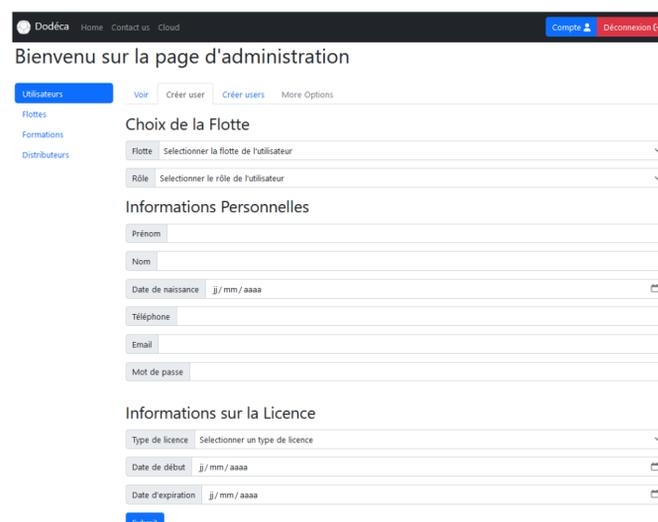


## RÉSULTATS ET CONCLUSION

Aujourd'hui, chaque catégorie d'utilisateur peut définir depuis une plate-forme homogène et centralisée les droits associés aux différents rôles, du super-Administrateur Speedernet à l'apprenant final d'une flotte client.

Chacun des modules SPHERE peut requérir par l'API les données dont il a besoin pour autoriser ses fonctionnalités et l'accès aux contenus associés.

L'architecture et son API sont conçues pour permettre aux nouveaux modules d'accéder à la plate-forme en gérant les futures fonctionnalités qui leurs sont spécifiques (Multi-utilisateurs, injection de contenus, mesure des apprentissages ...):



**MOTS-CLÉS** : TypeScript, MongoDB (NoSQL) , SCRUM, SAML, React, Nginx