

Auteurs : Sofian AZHARI – Yann GRANDGIRARD – Julien HAGNIER

CONTEXTE ET OBJECTIF

Speedernet est une entreprise spécialisée dans le **Digital Learning** basée à Lyon. Depuis 2017, elle développe Sphere, un outil-auteur de création de contenu de formation mono-utilisateur **immersif** et **scénarisable**.

Avec la crise sanitaire et la généralisation du **télétravail**, l'entreprise a amorcé le développement d'un mode **multi-utilisateur** à distance. La voix sur IP ne faisant pas partie de leurs domaines d'expertise, l'objectif du projet est de concevoir et réaliser une **brique logicielle de VoIP** pour Sphere.



MÉTHODES ET DÉVELOPPEMENTS

Le projet est divisé en quatre grandes phases :

- **Montée en compétence** et **veille technologique** : l'équipe-projet a pu acquérir des **connaissances** de base sur la VoIP et isoler plusieurs **Media Servers** open source au-dessus desquels construire la brique ;
- **Benchmarking** : une étude de **performance** a été menée afin de choisir le meilleur Media Server ;
- **Développement, tests de fonctionnement** et **documentation** ;
- **Tests de performances** : l'équipe a mesuré les **performances** finales de la brique logicielle, et son impact sur le **réseau** dans plusieurs configurations représentatives des **cas d'usages**.

L'équipe a adopté la méthodologie **SCRUM**, bien connue de Speedernet. Les **incréments** logiciels ont été livrés directement sur un **dépôt Bitbucket** accessible à l'entreprise.



RÉSULTATS ET CONCLUSION

La brique logicielle est intégrée à Sphere et permet des conversations audio jusqu'à **20 utilisateurs** en simultané et de les répartir dans différentes « **Rooms** ». Il est également possible de communiquer à l'écrit grâce à un **tchat textuel**.

Elle fonctionne sur tous les **navigateurs majeurs** récents. Les débits et latences mis en œuvre sont suffisamment faibles pour permettre la communication sur des **réseaux mobiles**.



Exemple de contenu créé avec Sphere

MOTS-CLÉS : Node.js, WebSocket, VoIP, WebRTC, Media Server, SCRUM, Intégration continue