

CONTEXTE ET OBJECTIF



RSACOSMOS

RSA Cosmos, fabricant de solutions optiques, numériques, full dômes 3D, hybrides et portables, a déjà équipé plus de 60 planétariums dans le monde.

Actuellement les planétariums fournis par RSA Cosmos sont **autonomes et indépendants**, ils ne peuvent être pilotés de l'extérieur, toute séance nécessite la présence d'un conférencier dans le planétarium qui coordonne la séance via leur logiciel de simulation astronomique 3D temps réel.

Le projet de **DomeCasting** sera présenté à l'**IPS 2018** (International Planetarium Society) et ajouté à la solution de leurs clients actuels. Il doit répondre aux besoins suivants :

- Un planétarium « maître » doit être capable de contrôler un ou plusieurs planétariums « esclaves ».
- Le contenu d'une conférence doit être retransmis depuis le maître vers les esclaves en direct.
- Les participants doivent être en capacité de communiquer via un système de messagerie instantanée.



MÉTHODES ET DÉVELOPPEMENTS

Rédaction du cahier des charges
(1 mois)

Veille technologique - Rédaction des spécifications - Autoformation
(2 mois)

Développement - Intégration au logiciel existant - Tests
(2 mois et demi)

Recette - Finalisation et documentation
(2 semaines)

Technologies imposées :

- API : PHP – Laravel
- GUI : Qt – QML
- Client et serveur : C++



Technologies choisies :

- Visioconférence : GStreamer, libVLC
- Chat : Boost - Asio

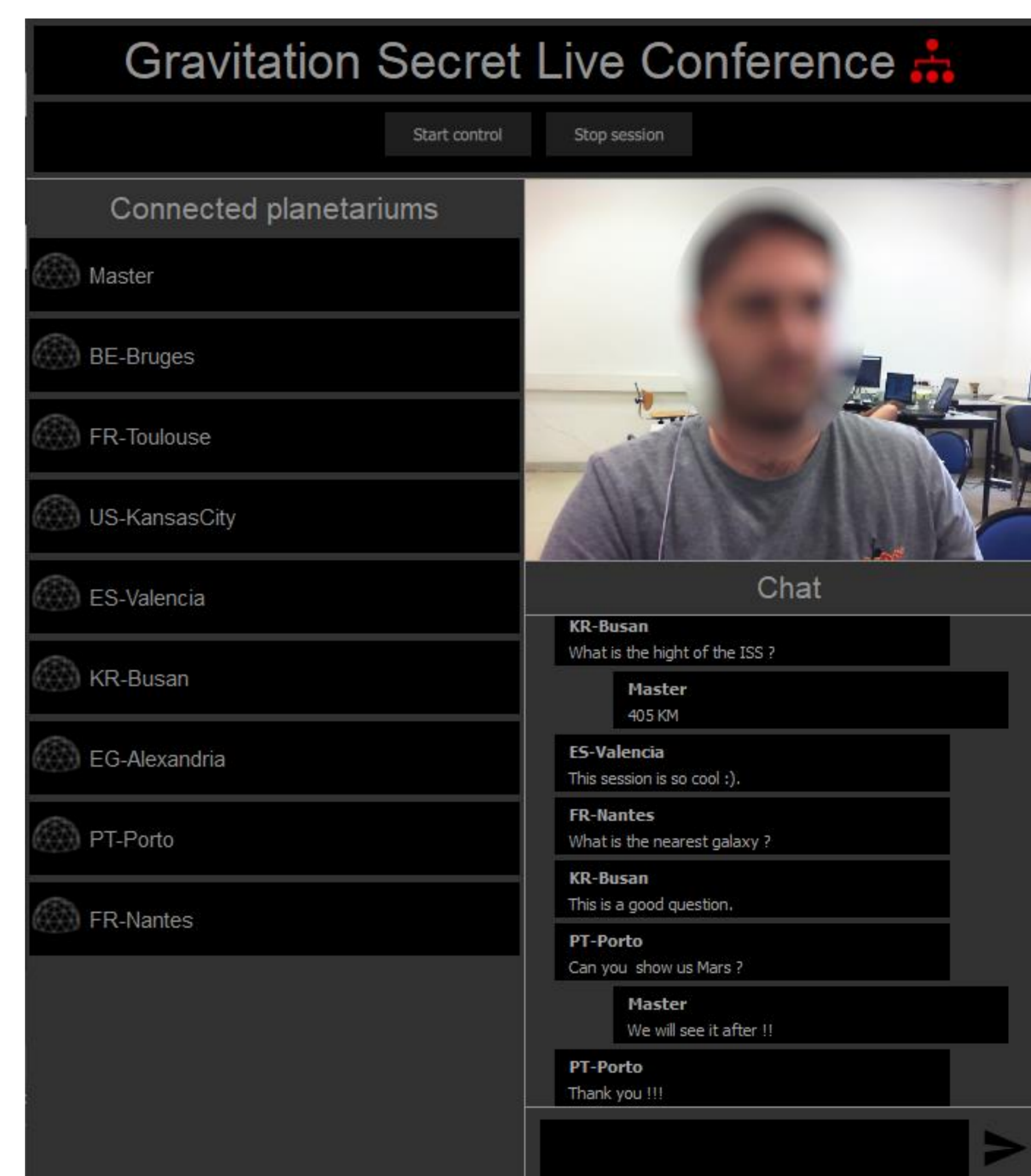


RÉSULTATS ET CONCLUSION

La fonctionnalité de DomeCasting possède :

- Une partie **visioconférence** et **messagerie instantanée**
- Une solution de **contrôle à distance d'un planétarium**
- Une **API** gérant les appels à une **base de données** contenant les informations des sessions de DomeCasting
- Une **interface graphique** intégrée au logiciel existant

Cette solution intégrée au logiciel existant pourra être complétée par l'ajout futur de nouvelles fonctionnalités par l'entreprise.



MOTS-CLÉS : C++, PHP, Laravel, Qt-QML, GStreamer, libVLC, Boost, réseaux