



Utilisation de Jovo dans un projet

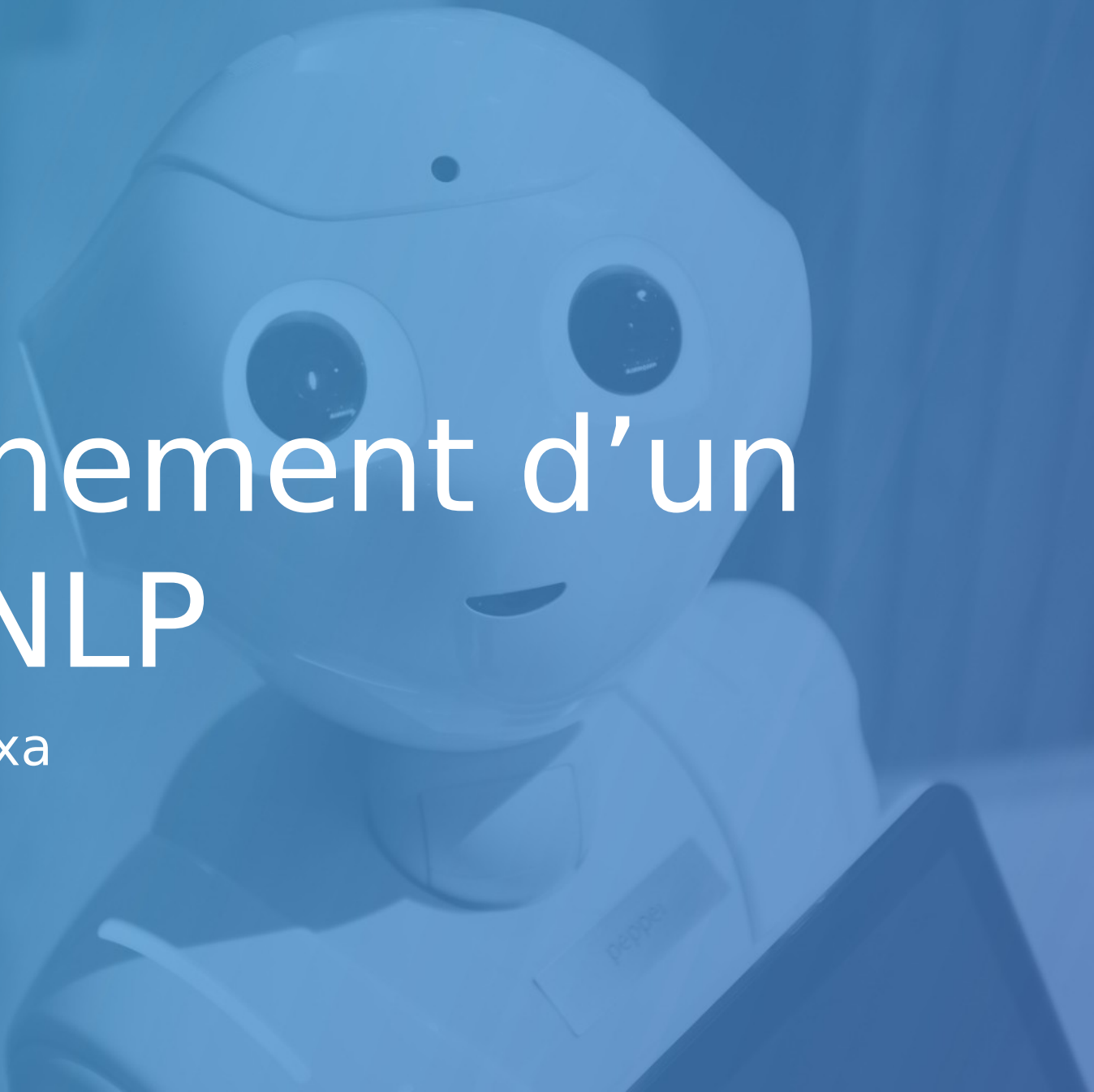
Basé sur le projet « chatbot Rennes Métropole »

Plan de la présentation

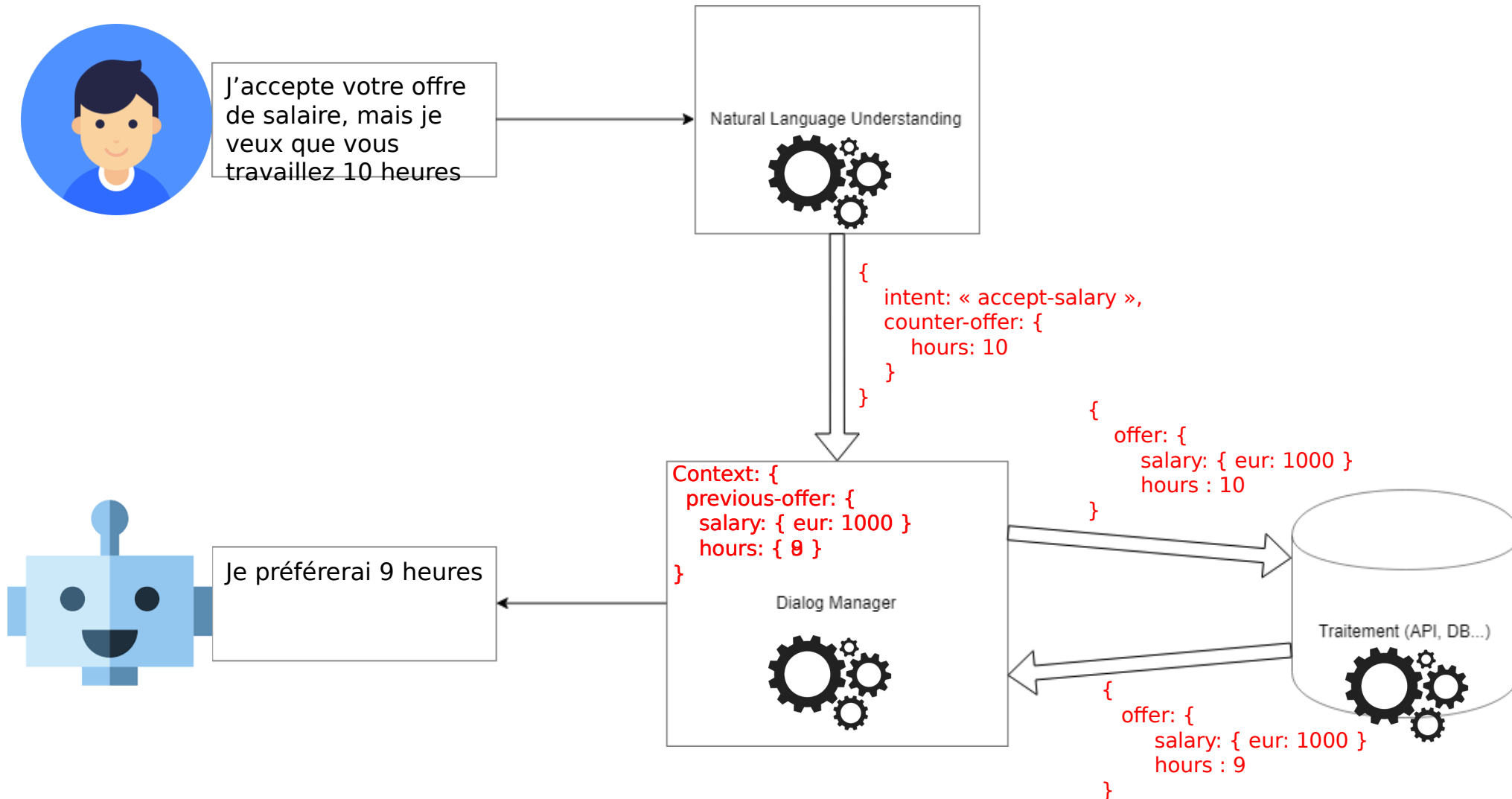
- Fonctionnement d'un chatbot NLP
- Les plateformes supportées
- Logique de la conversation sous Jovo
- Déploiement avec Jovo
- Le futur de Jovo

Fonctionnement d'un chatbot NLP

Google Assistant / Alexa



Fonctionnement général d'un chatbot NLP

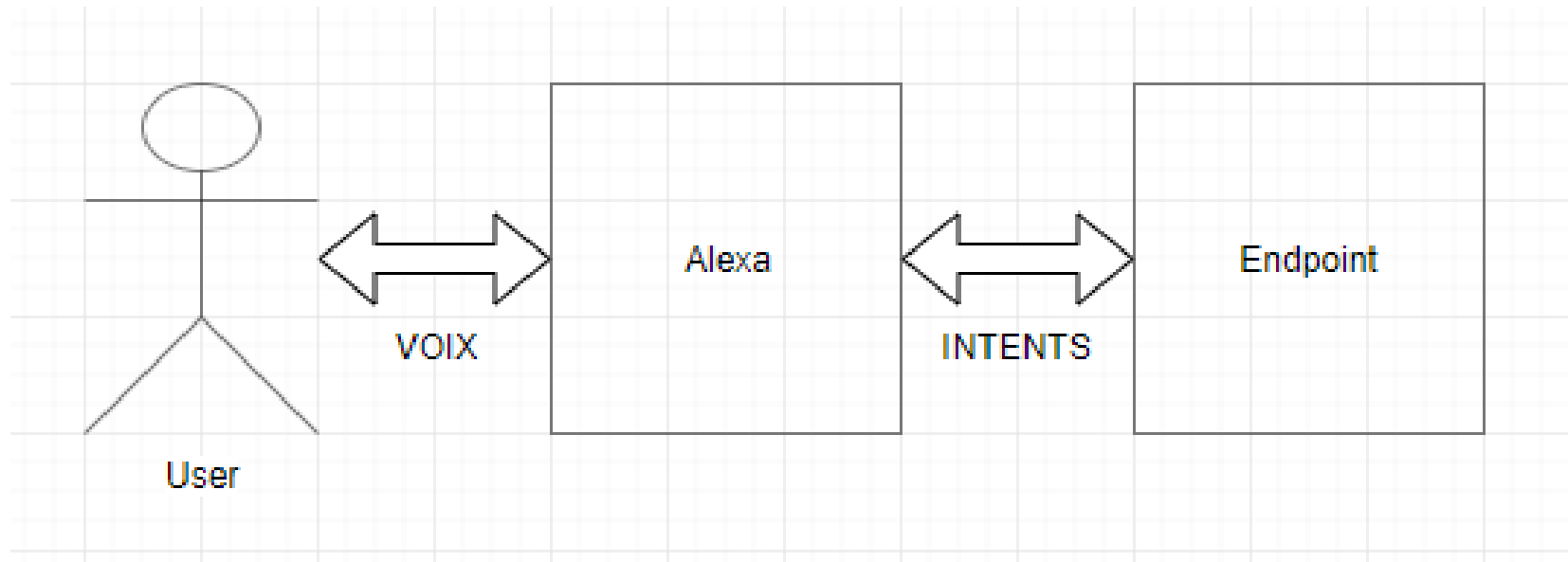




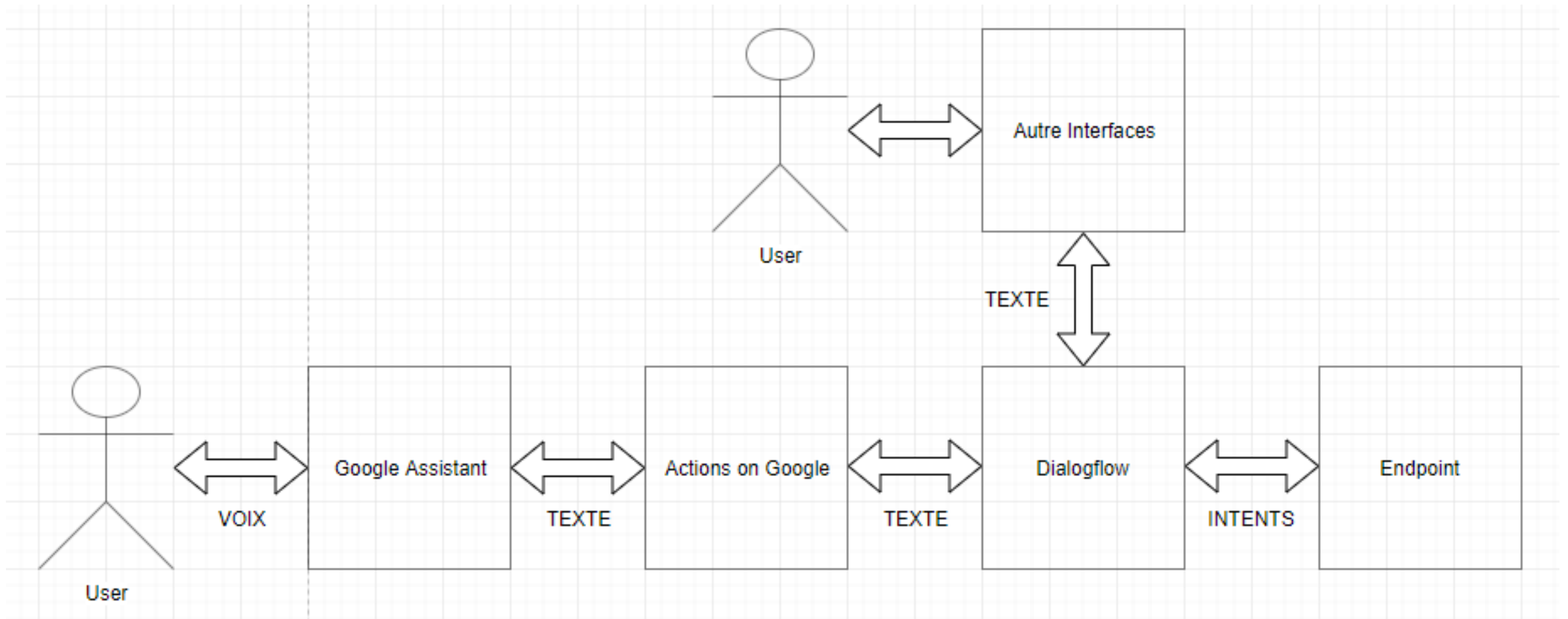
Les plateformes

Google Assistant / Alexa

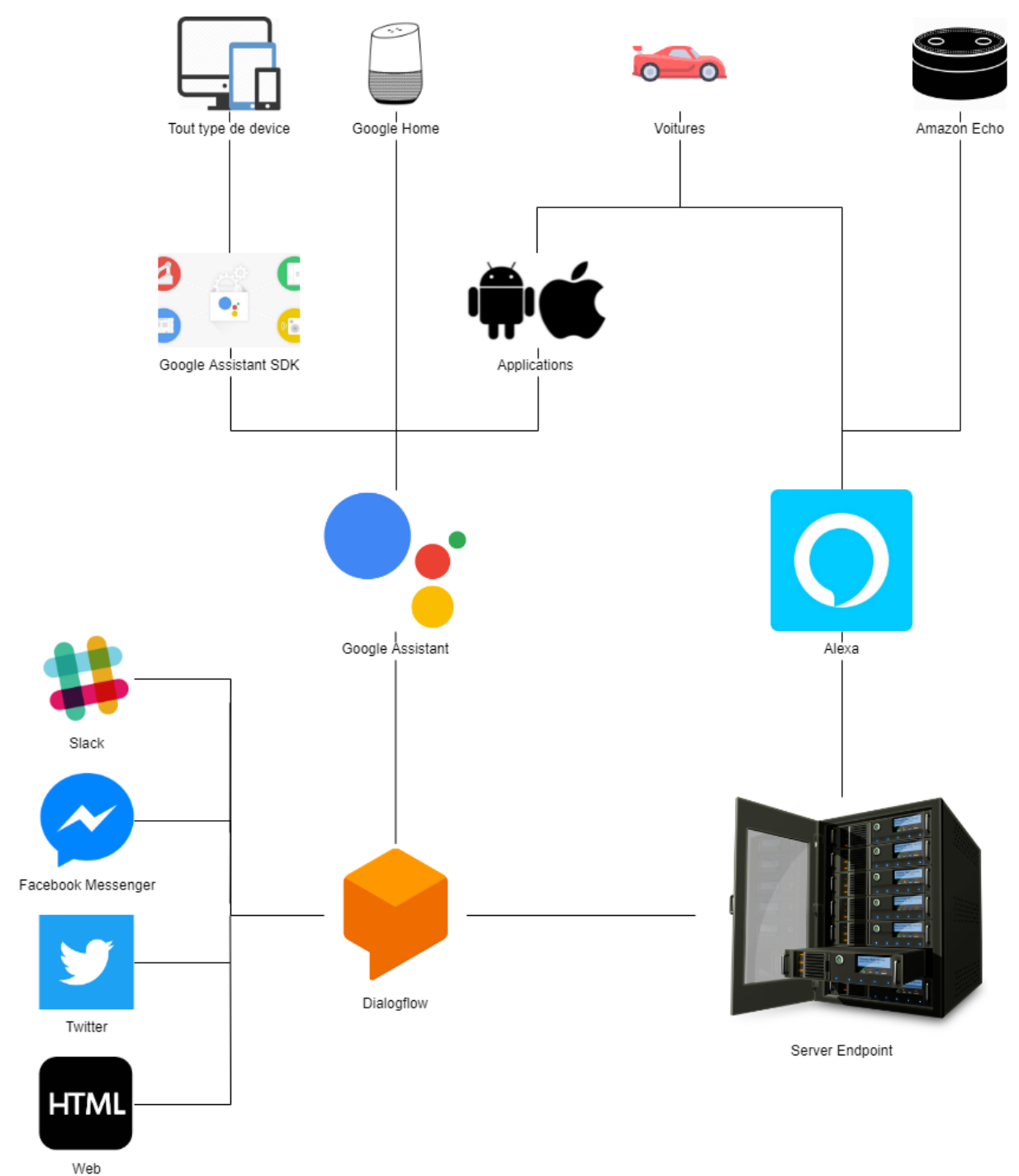
Avec Alexa



Avec Google assistant + Dialogflow



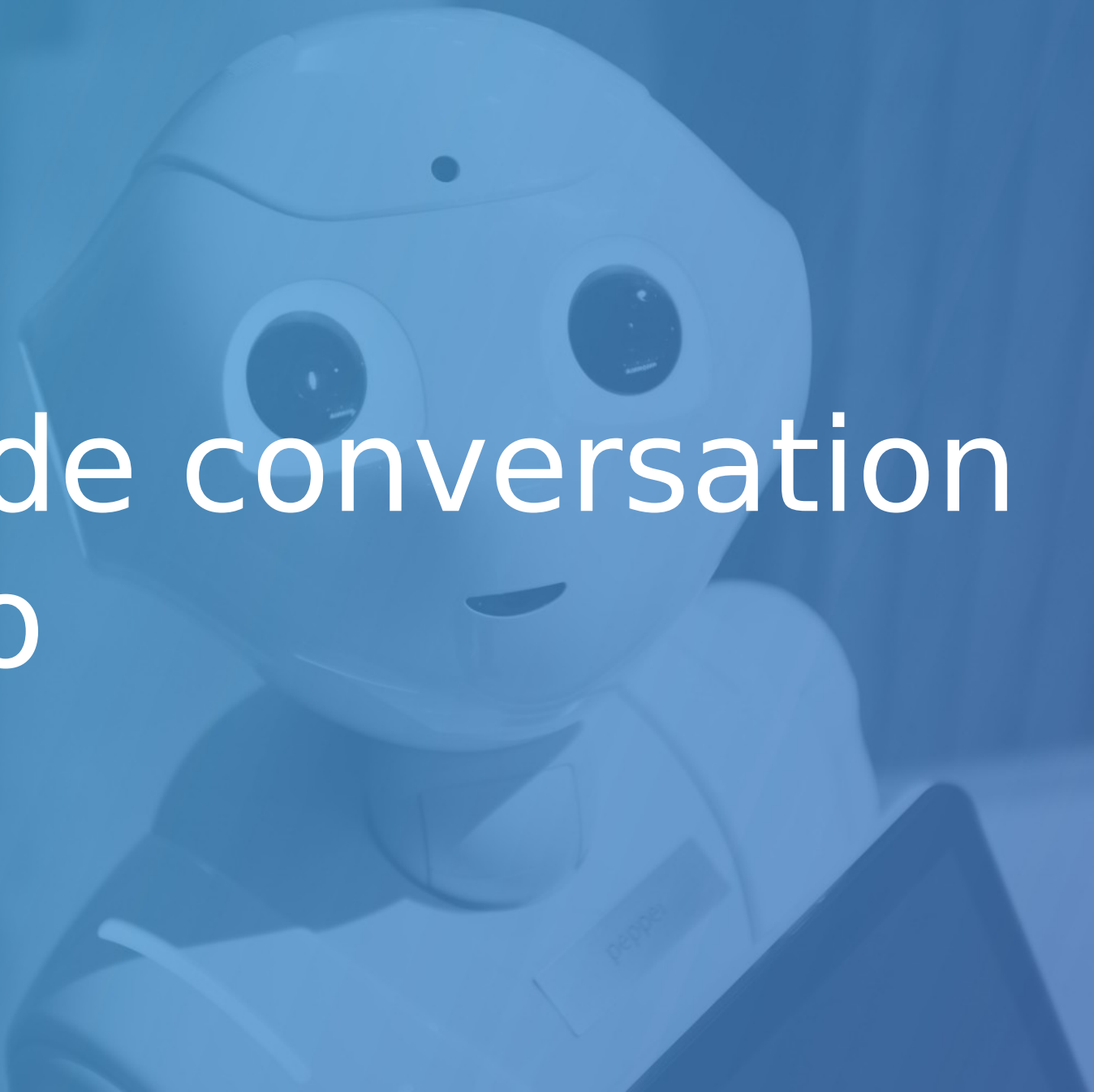
Notre objectif : une seule application, Plusieurs plateformes



Jovo supporte actuellement :

- Google :
 - Dialogflow (NLU)
 - Google Assistant (assistant)
- Amazon :
 - Alexa Skill Kit (NLU + assistant)

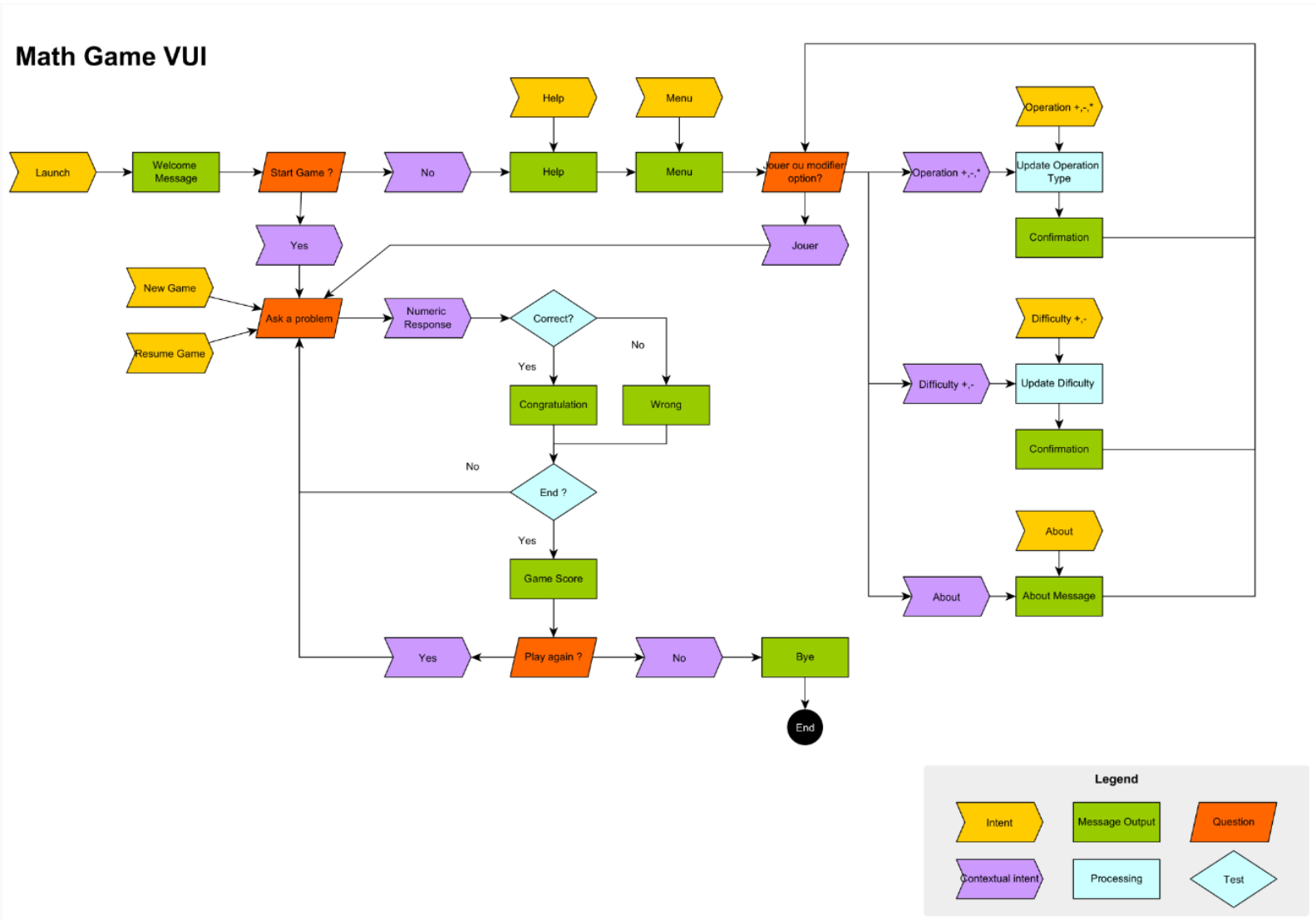
Logique de conversation avec Jovo



Les requêtes reçues

- Format différent pour google / amazon
- Jovo fait la conversion automatiquement
- Informations communes :
 - Intents
 - Entities (slots)
 - Contexte
 - Session ID
 - User ID
 - ...

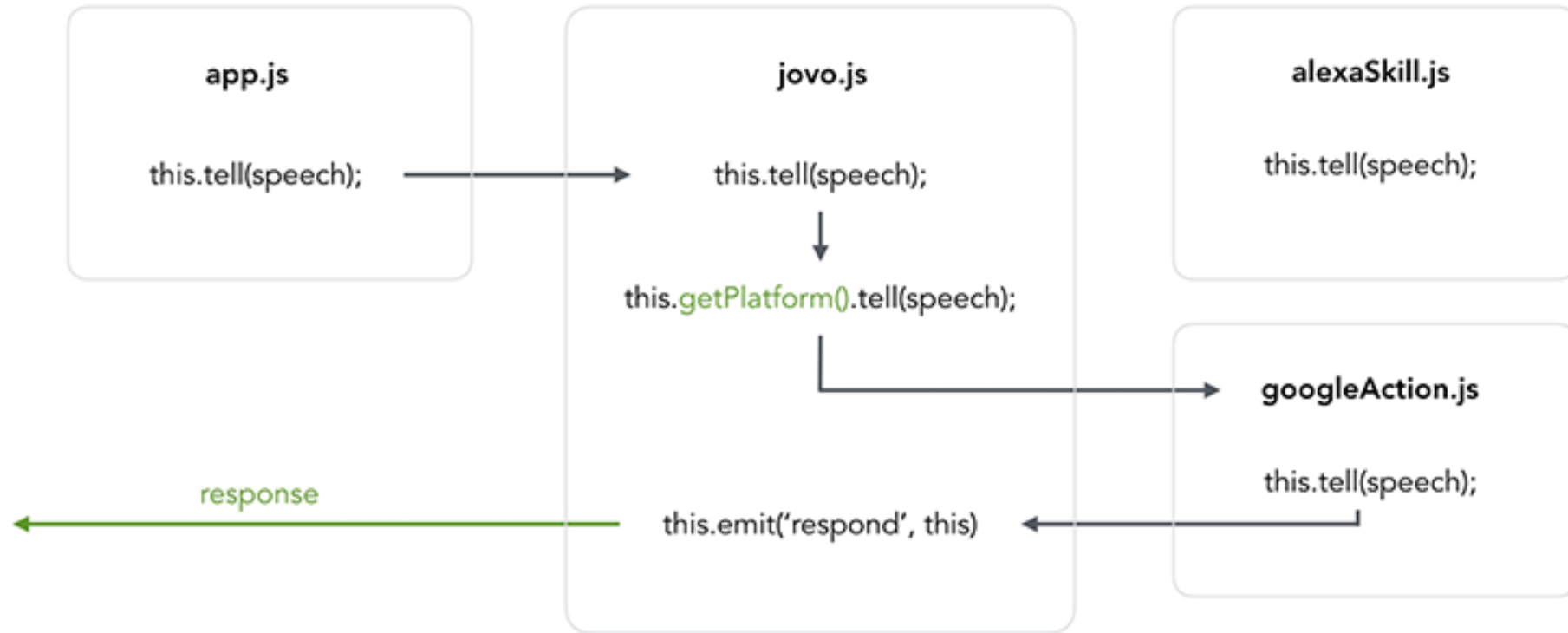
Modélisation de la conversation



Le routing

- On reçoit une requête pour l'intent « getMeteo » une variable (entité) de type « ville », valeur « Rennes »
- Jovo route la requête vers une fonction « handler ».
- Il est possible de :
 - faire de la redirection
 - créer des « events listener » (par exemple NEW_SESSION)
 - définir des états dans la conversation
 - router vers une intent par défaut
 - ...

Réponses



Testing

- L'un des points difficiles dans les projet de chatbot.
- Jovo unit-testing pour tester le lien handler -> réponse.
 - Permet la consistance des réponses dans le projet
- Pas encore installé, mais il existe jovo model-testing pour le lien phrase -> handler
 - Permettrai de garantir la performance du NLU



déployer avec Jovo

Etapes du Déploiement

Build

- Les formats d'entraînement d'Alexa et de Google différents
- Jovo utilise un format tiers (jovo-model) transformé dans le format respectif des deux plateformes

Deploy

- Permet d'entraîner Dialogflow & Alexa en une commande
- Gère les étapes de développement

Le futur de Jovo



REX auprès des développeurs de JOVO

Jovo développe eux même des chatbots, et ils ont des idées similaires

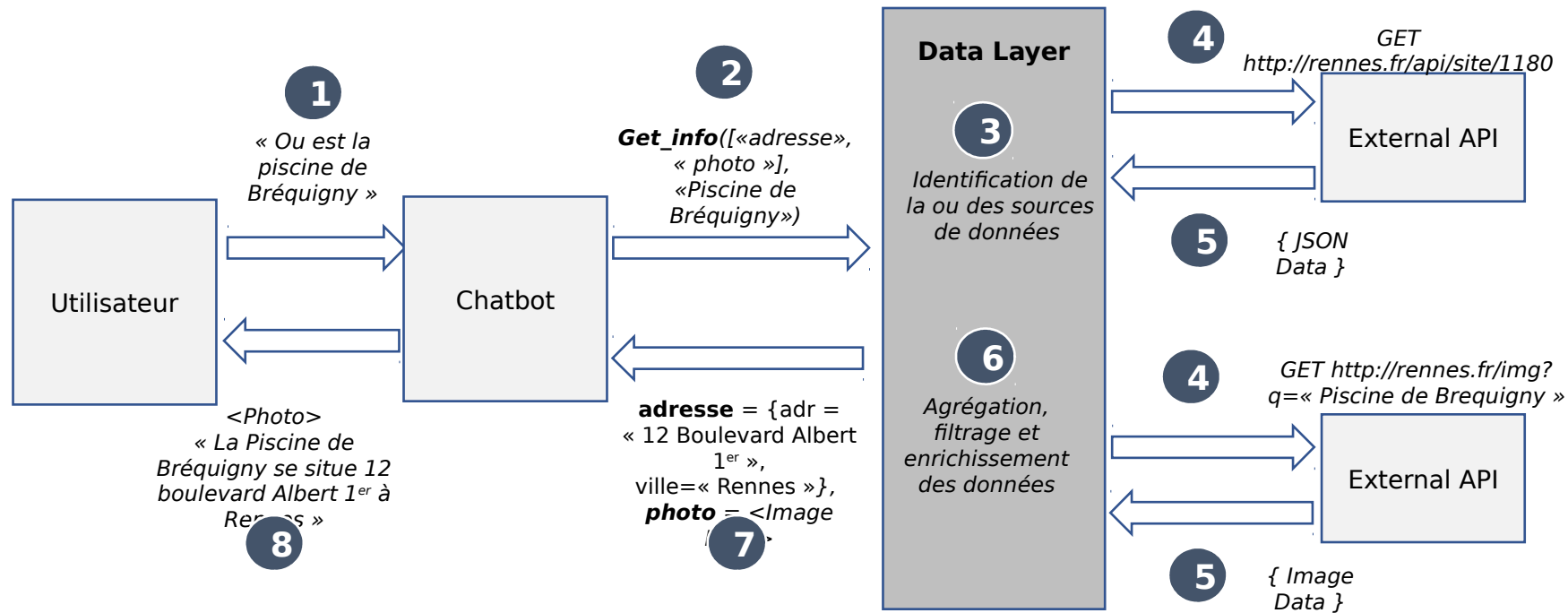
- Approche modulaire de la conversation
 - Pouvoir séparer la conversation en modules qu'on peut partager entre projet
- Approche modulaire du déploiement
 - Permettre de créer des plugins pour contrôler profondément le déploiement
 - En particulier, export Docker

Ils veulent intégrer Cortana (et des NLU open source)

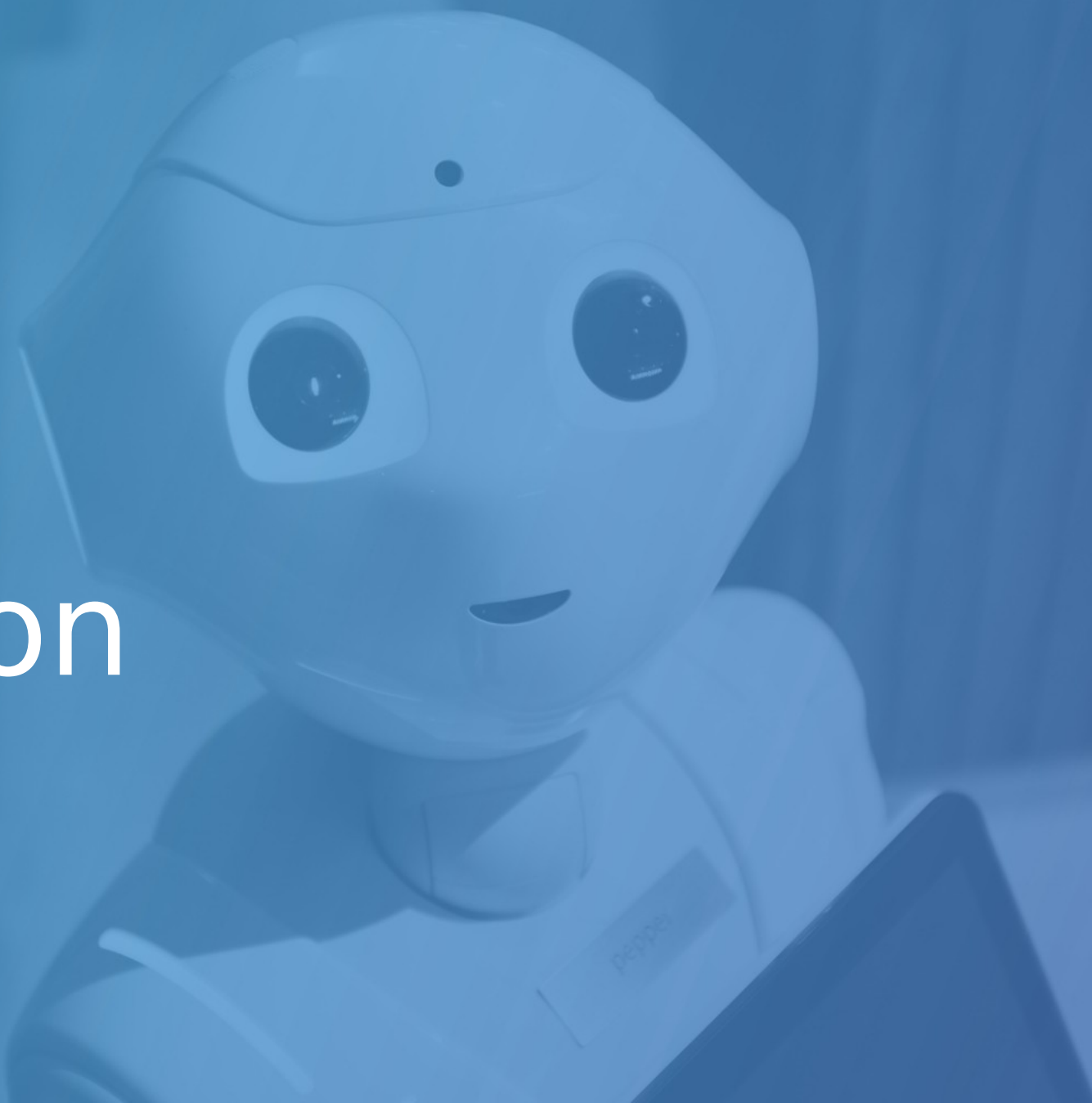


Data-connector

Data-Connector



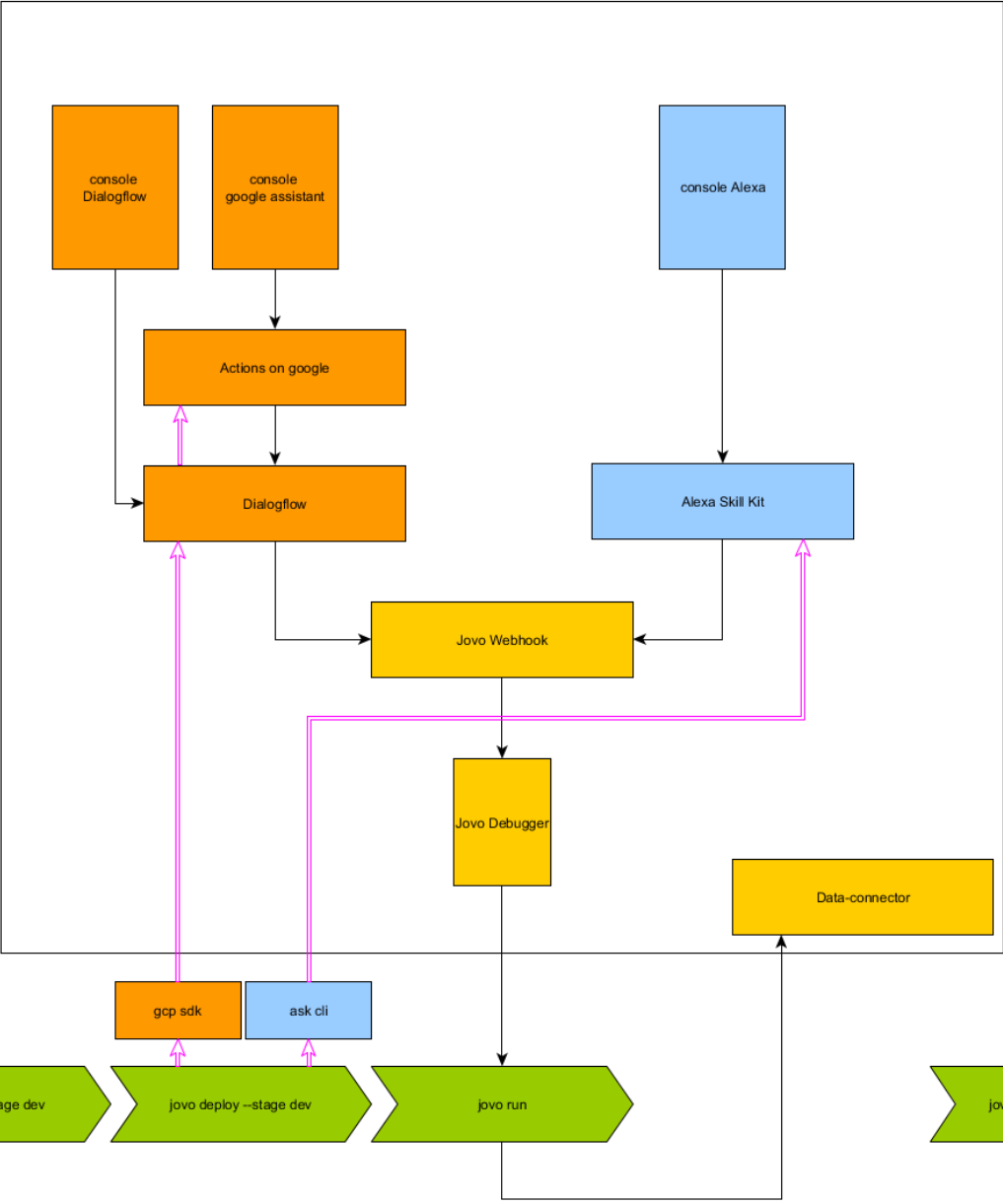
Conclusion



Conclusion :

- Jovo permet de supporter beaucoup plus de plateformes à partir d'un seul projet
 - est plus facile à intégrer aux outils type 'git'
 - permet d'appliquer l'approche devops
 - Mais il n'est pas encore complet, mais se développe rapidement
-
- La partie base de données / API peu toujours poser des problèmes, en particulier dans le cas où les données ne sont pas vraiment « propres »

stage "dev"



exemple stage "prod"

