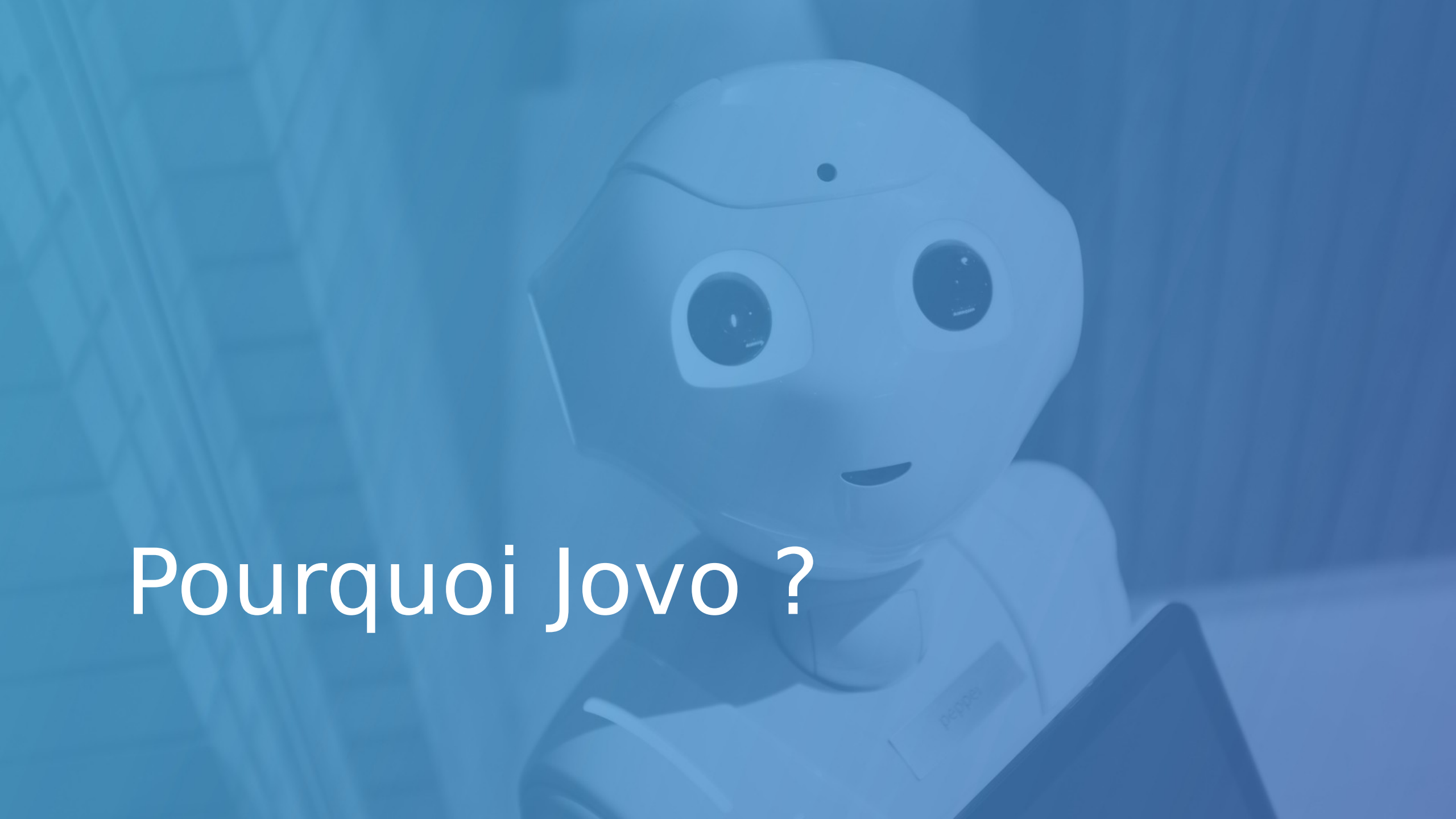
Jovo

# Plan

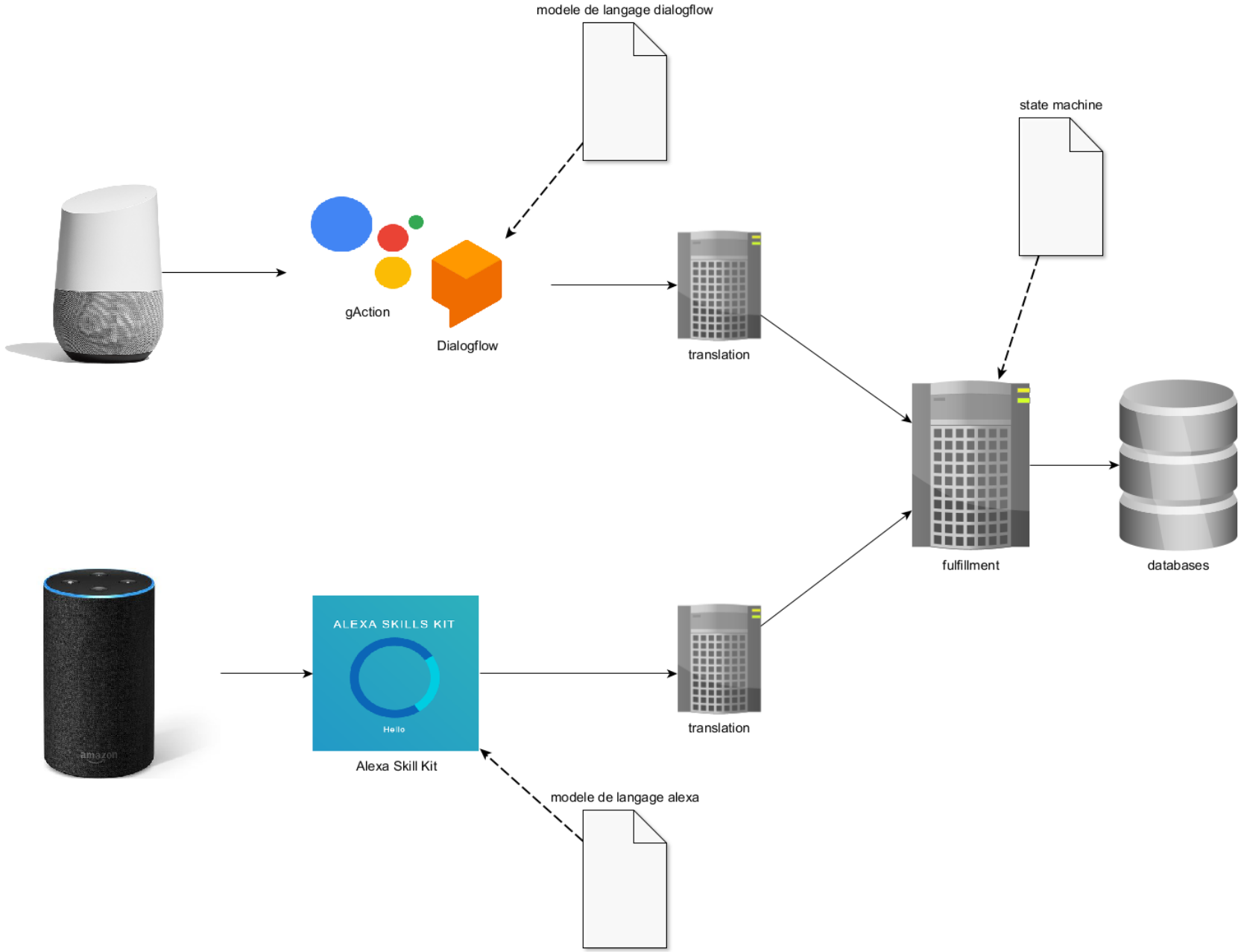
- Introduction
- Pourquoi Jovo ?
- Fichiers d’un projet Jovo
- Les étapes du déploiement
- Autres fonctionnalités

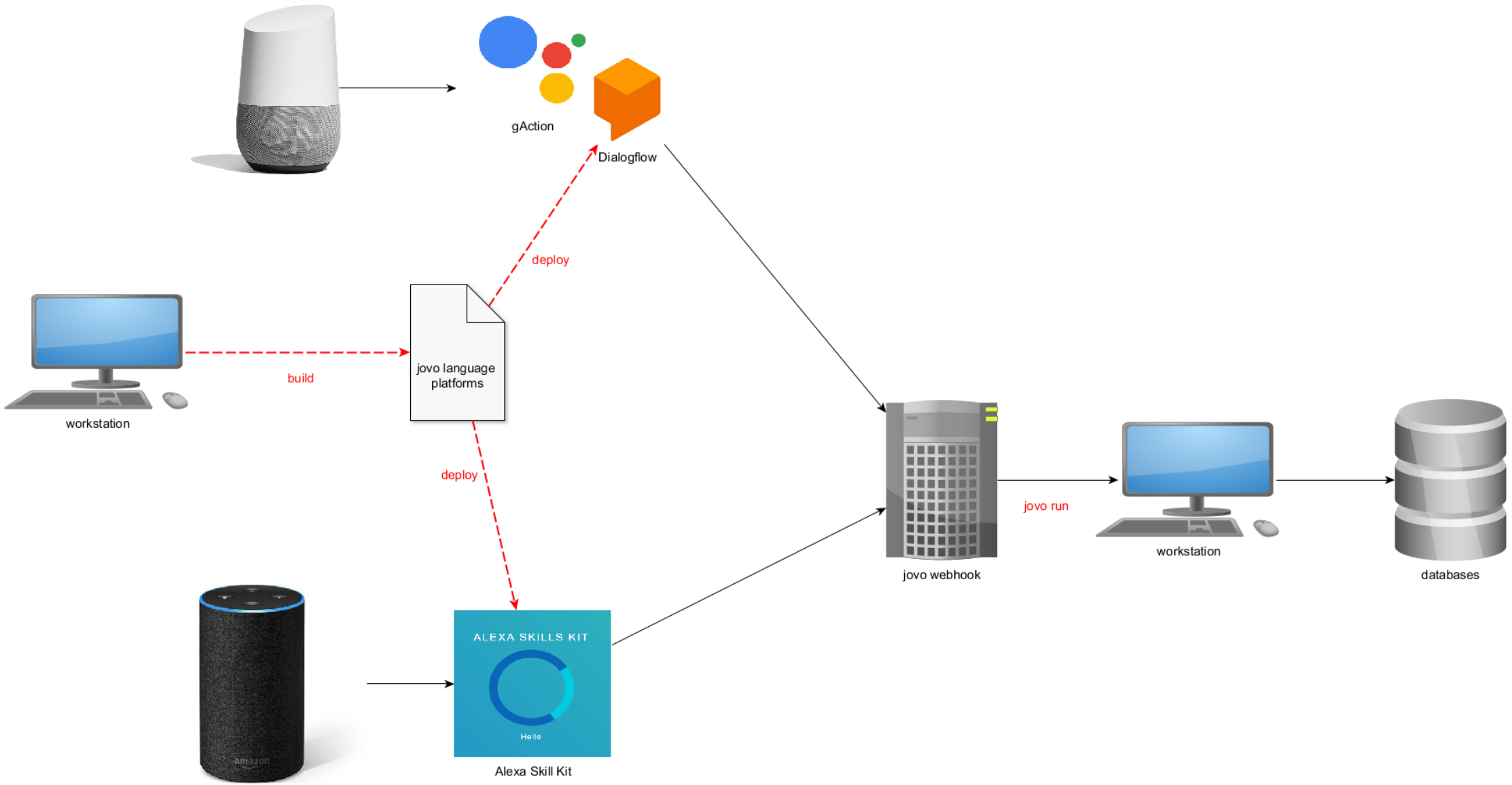
# Introduction

- Présentation Jovo-CLI v1.2, Jovo-Framework v1.3.4
- Jovo est le premier framework qui permet de créer des applications à la fois pour Amazon Alexa et Google Assistant.
- Open source. La prochaine version de Jovo sera modulaire pour permettre la création de plus de plugins et d’intégrer de nouveaux NLU.
- Jovo permet :
  - Intégration
  - Staging du déploiement
  - Test
  - Support i18n
  - …



# Pourquoi Jovo ?

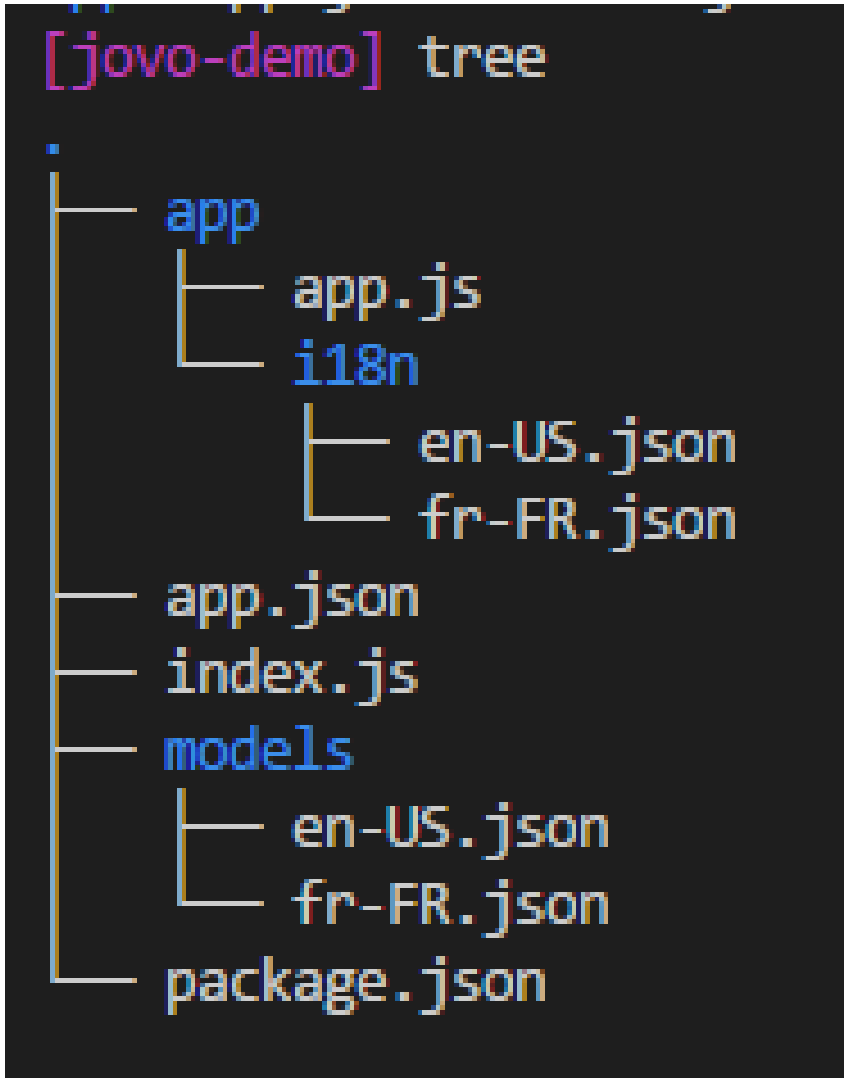

modele de langage dialogflow
state machine
gAction
Dialogflow
translation
fulfillment
databases
ALEXA SKILLS KIT
Hello
Alexa Skill Kit
translation
modele de langage alexa


gAction
Dialogflow
deploy
workstation
build
jovo language platforms
deploy
jovo webhook
jovo run
workstation
databases
ALEXA SKILLS KIT
Hello
Alexa Skill Kit

# Fichiers d’un projet Jovo

# Fichiers

| | |
|---|---|
| **app/** | handlers |
| **app/i18n/** | Templates de réponse |
| **app.json** | Fichier de configuration du déploiement |
| **index.js** | Fichier de lancement |
| **models/** | Modèles de langage jovo |



# Fichiers de modèles

- Les fichiers sont dans le dossier « models/ »
- 1 fichier par variante de langage (fr-FR, fr-CA …)
- 4 parties dans chaque fichier :
  - Intents communes
  - Entities communes
  - Alexa uniquement
  - Dialogflow uniquement

# Fichiers du server « fulfillment »

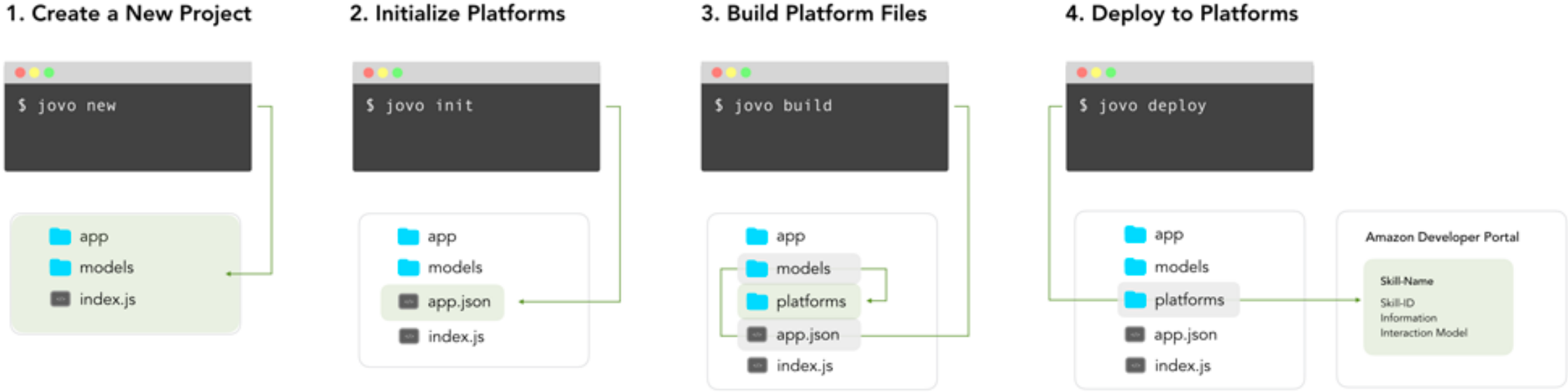
- Fichier app.js dans app/
- Gestion de la conversation par machine à état
- Speechbuilder() permettant de créer des réponses audio (pauses, etc…)
- Possibilité de faire du routage entre les intents
- Fonctions pour renvoyer des éléments différents selon les plateformes (listes, carrousel, …)
- …

# Fichiers de templates

- Les fichiers sont dans le dossier app/i18n/
- 1 fichier par variante de langue (fr-FR.json, fr-CA.json, ...)
- Passage automatique de l’un a l’autre selon la langue de l’utilisateur

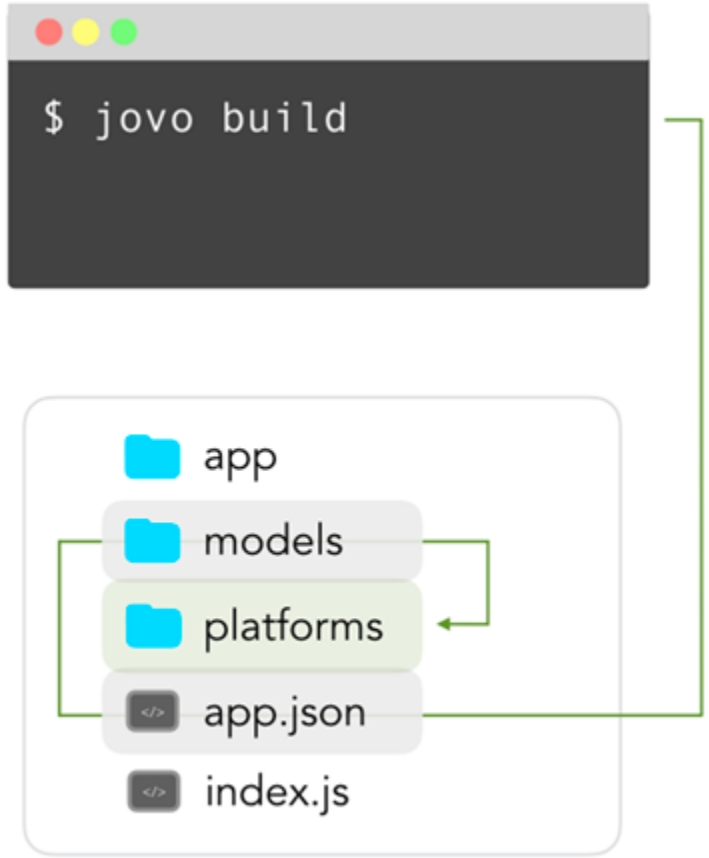
# Utiliser Jovo-cli

# Workflow


1. Create a New Project
$ jovo new
app
models
index.js
2. Initialize Platforms
$ jovo init
app
models
app.json
index.js
3. Build Platform Files
$ jovo build
app
models
platforms
app.json
index.js
4. Deploy to Platforms
$ jovo deploy
app
models
platforms
app.json
index.js
Amazon Developer Portal
Skill-Name
Skill-ID
Information
Interaction Model

# Build

- Permet de créer les fichiers de plateformes à partir des fichiers de modèles (suivant la configuration dans app.json)



# En savoir plus sur jovo build

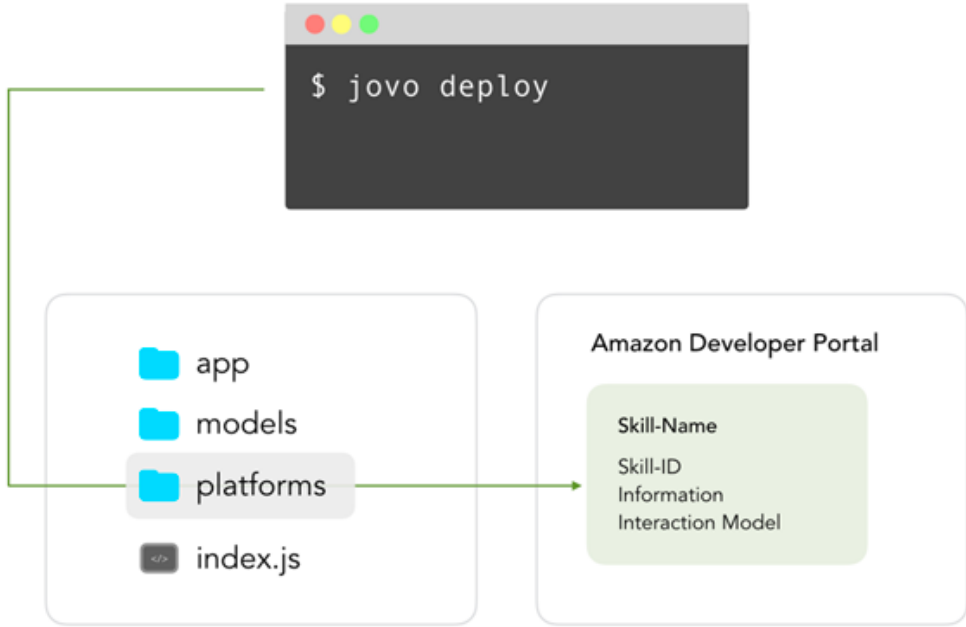
- Il est possible de build les fichiers de models à partir des fichiers de platforms : `jovo build –-reverse`

- Il est possible d’importer les fichiers de platformes à partir de Alexa Skill kit (dialogflow work in progress) : `jovo get`

- Pour plus d’informations : https://www.jovo.tech/docs/cli#jovo-build

# Deploy

- Jovo deploy permet d’upload les platformes vers les consoles de développement respectives



# Deploy (Alexa)

- `Jovo deploy -p alexaSkill`

- Pour déployer sur Alexa en ligne de commande, il faut installer et configurer « ASK CLI », l'outil d'Amazon pour la configuration en ligne de commande d'Alexa Skill Kit : https://developer.amazon.com/fr/docs/smapi/quick-start-alexa-skills-kit-command-line-interface.html

- Console Alexa : https://developer.amazon.com/alexa/console/ask

# Deploy (googleAction)

- `Jovo deploy -p googleAction`

- Pour déployer sur googleAction, il faut créer un service account sur Google Cloud Platform, et installer / configurer le Cloud SDK, l’interface en ligne de commande de google : https://www.jovo.tech/blog/deploy-dialogflow-agent-jovo-cli/

- Console Dialogflow : https://console.dialogflow.com

# Pour en savoir plus sur jovo deploy

- https://www.jovo.tech/docs/cli#jovo-deploy

- Il est possible de déployer l'application fulfillment directement sur AWS Lambda https://www.jovo.tech/guides/deploy-lambda-cli

- Il est possible de définir plusieurs environnements (staging) de déploiement, pour en savoir plus : https://www.jovo.tech/guides/staging-examples

# Jovo run

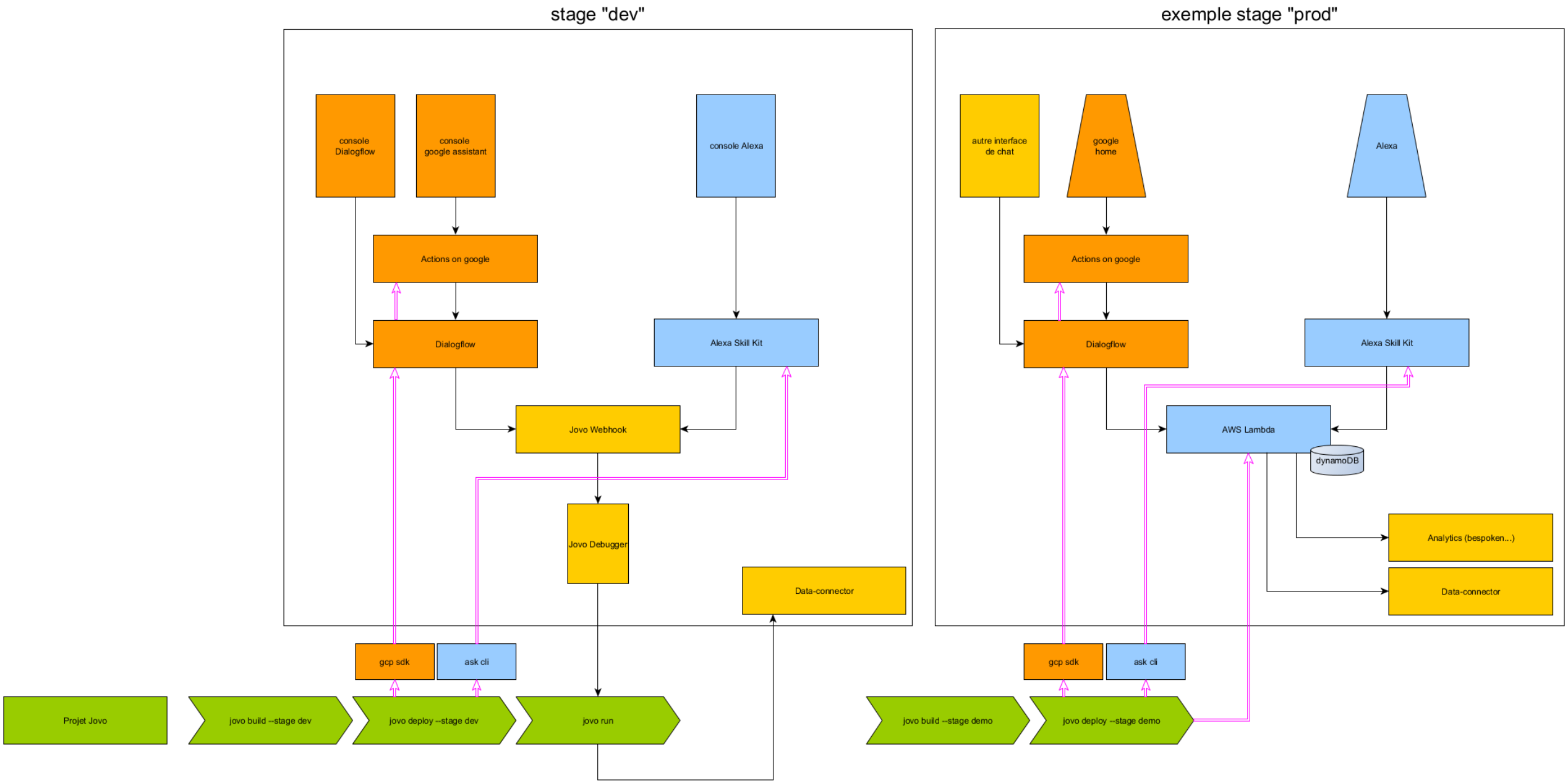
- Démarre le server de développement dans le fichier index.js et active le webhook

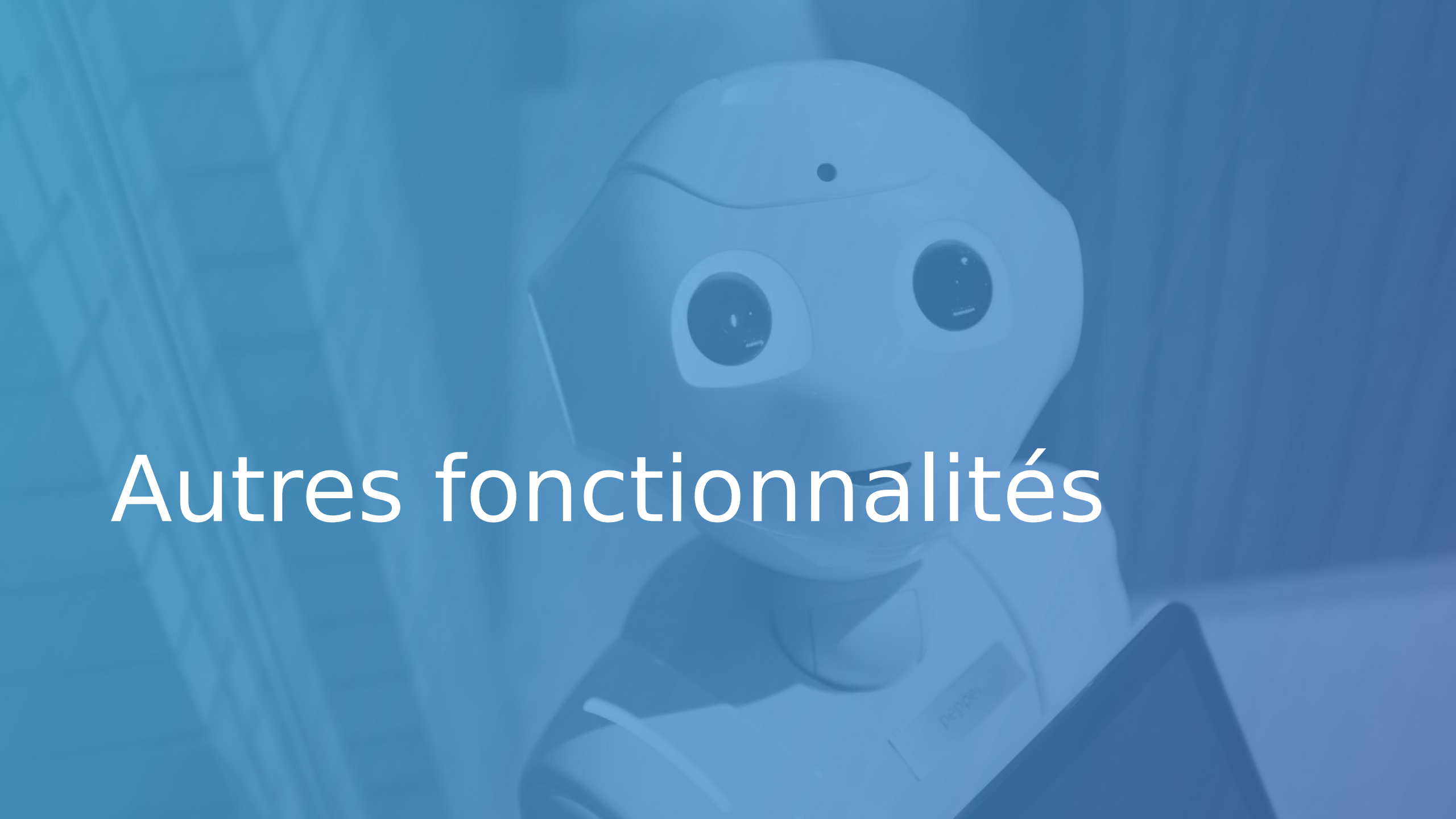
# Pour en savoir plus sur jovo run

- https://www.jovo.tech/docs/cli#jovo-run
- Il est possible d’utiliser Jovo avec nodemon : `jovo run –watch`
- Il est possible de passer par un autre webhook que celui de jovo

- Il est possible d’accéder à un débuggeur https://www.jovo.tech/debugger

- Il existe aussi une interface de test https://www.jovo.tech/blog/debug-alexa-audioplayer-skills-jovo-debugger/


stage "dev"
exemple stage "prod"
console Dialogflow
console google assistant
console Alexa
Actions on google
Dialogflow
Alexa Skill Kit
Jovo Webhook
Jovo Debugger
Data-connector
gcp sdk
ask cli
Projet Jovo
jovo build --stage dev
jovo deploy --stage dev
jovo run
autre interface de chat
google home
Alexa
Actions on google
Dialogflow
Alexa Skill Kit
AWS Lambda
dynamoDB
Analytics (bespoken...)
Data-connector
gcp sdk
ask cli
jovo build --stage demo
jovo deploy --stage demo



# Autres fonctionnalités

- Jovo permet de définir des tests unitaires https://www.jovo.tech/templates/unit-testing
- Jovo persistence layer pour intégrer les bases de données https://www.jovo.tech/docs/databases
- Jovo analytics layer pour les outils d’analyse https://www.jovo.tech/docs/analytics
- Jovo plugins (peu de plugins pour l’instant) https://github.com/jovotech/jovo-plugins

# Ressources

- Website https://www.jovo.tech/
- Documentation https://www.jovo.tech/docs/
- Guides https://www.jovo.tech/guides
- Github https://github.com/jovotech/jovo-framework-nodejs
- Community (slack, ...) https://www.jovo.tech/community