

# Architecture logicielle réutilisable pour la manipulation d'objets déformables



Victor Giraud

Institut Pascal – Sigma Clermont

# Motivation



- Projet Softmanbot : Consortium de 11 partenaires, Industriel et Académique
- But : automatiser la manipulation d'objets déformables
- Problématique : Créer un système qui permet
  - De résoudre la tâche industrielle de Michelin
  - Tester les stratégies de recherche
  - Généralisable aux autres membres du consortium



# Contexte : projet Softmanbot



# Contexte : Objet déformable



- Nombre infini de degré de liberté
- Champ émergent de la robotique et du contrôle
- Applications : Industries (textile, plastique), robotisation de la cuisine.



# Besoins



- Application avec une formulation skilled-based
- Permettre de tester les pistes de recherche des collègues
- Intégrable aux machines industrielles pour tester
- Ré-utilisabilité des composants

# État de l'art



- SkiRos : Skill Based. Supervision possible par non-experts.  
Application typique : tâches de kiting
- RobMoSys : Model Based. Ecosystème de modules interchangeables pour construire un robot industriel spécifique à coûts réduits
- Beaucoup d'applications construisent un logiciel jetable

# Présentation de l'architecture

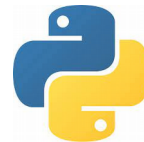


- Architecture basée sur ROS. Tourne sous Unix.

- Développée en C++



- Interfaçable avec Python et Shell



- Formée de 4 nœuds. Un nœud comporte une partie générique et une partie spécifique.
- Séparations des responsabilités entre les nœuds.



# Elaboration des choix



- ROS /Unix : Standard de facto dans la recherche

- C++ : Language qui offre le plus de liberté



- Python : utilisé par des collègues



- 4 nœuds :
  - 1 nœud pour le très haut niveau : machine d'état, GUI.
  - 1 nœud pour la planification
  - Besoin d'un nœud pour l'abstraction du Hardware
  - Séparé en 2 : GPU nécessaire pour la vision peut être sur une autre machine

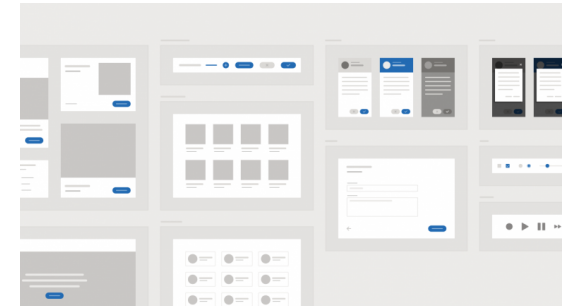




# Présentation de l'architecture : Division des responsabilités



- Supervisor : Planification, IHM, gestion d'erreur



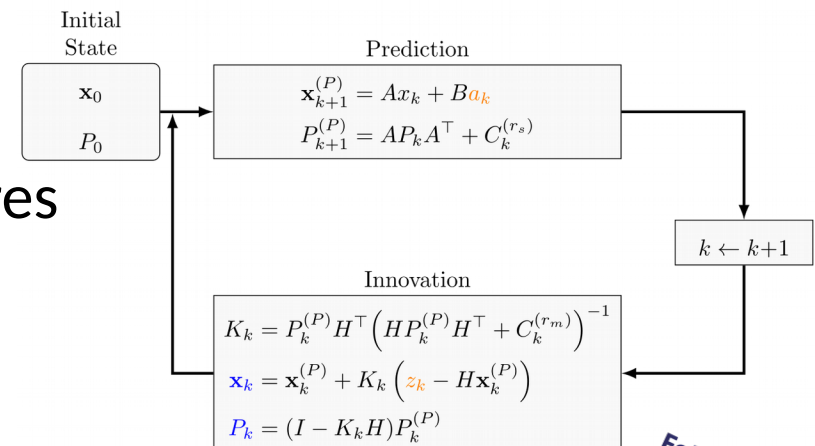
- Skilled Workcell : tous les drivers de toutes les machines utilisées



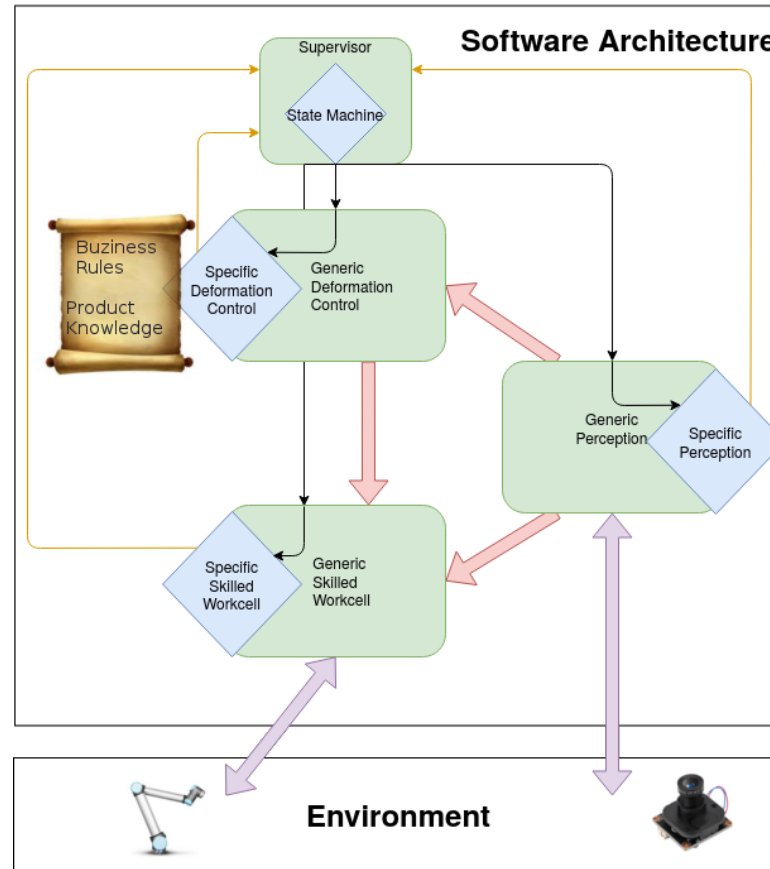
- Perception : drivers des capteurs, algorithmes de fusion de données



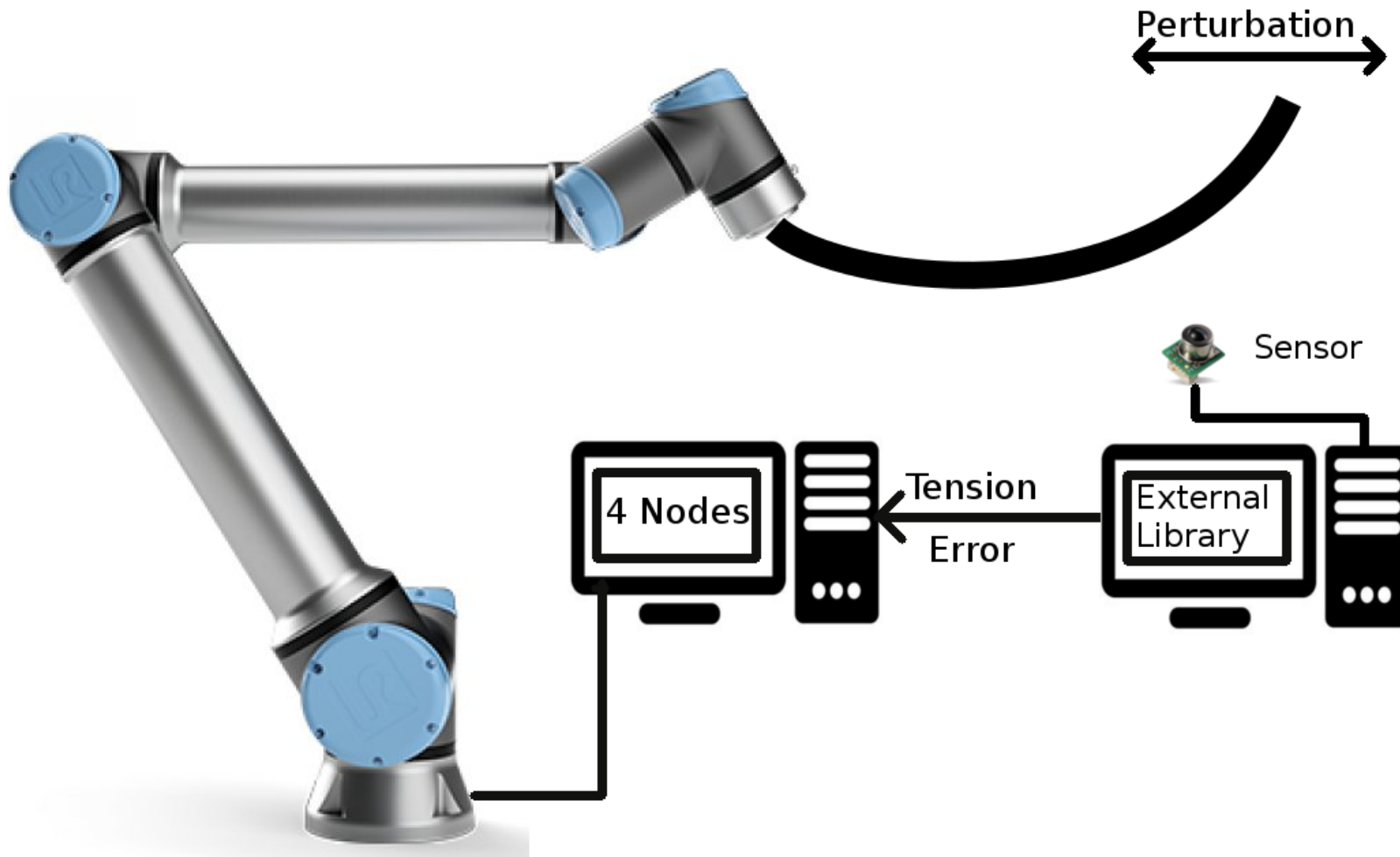
- Shape Control : Générations des trajectoires des robots pour contrôler la forme de l'objet



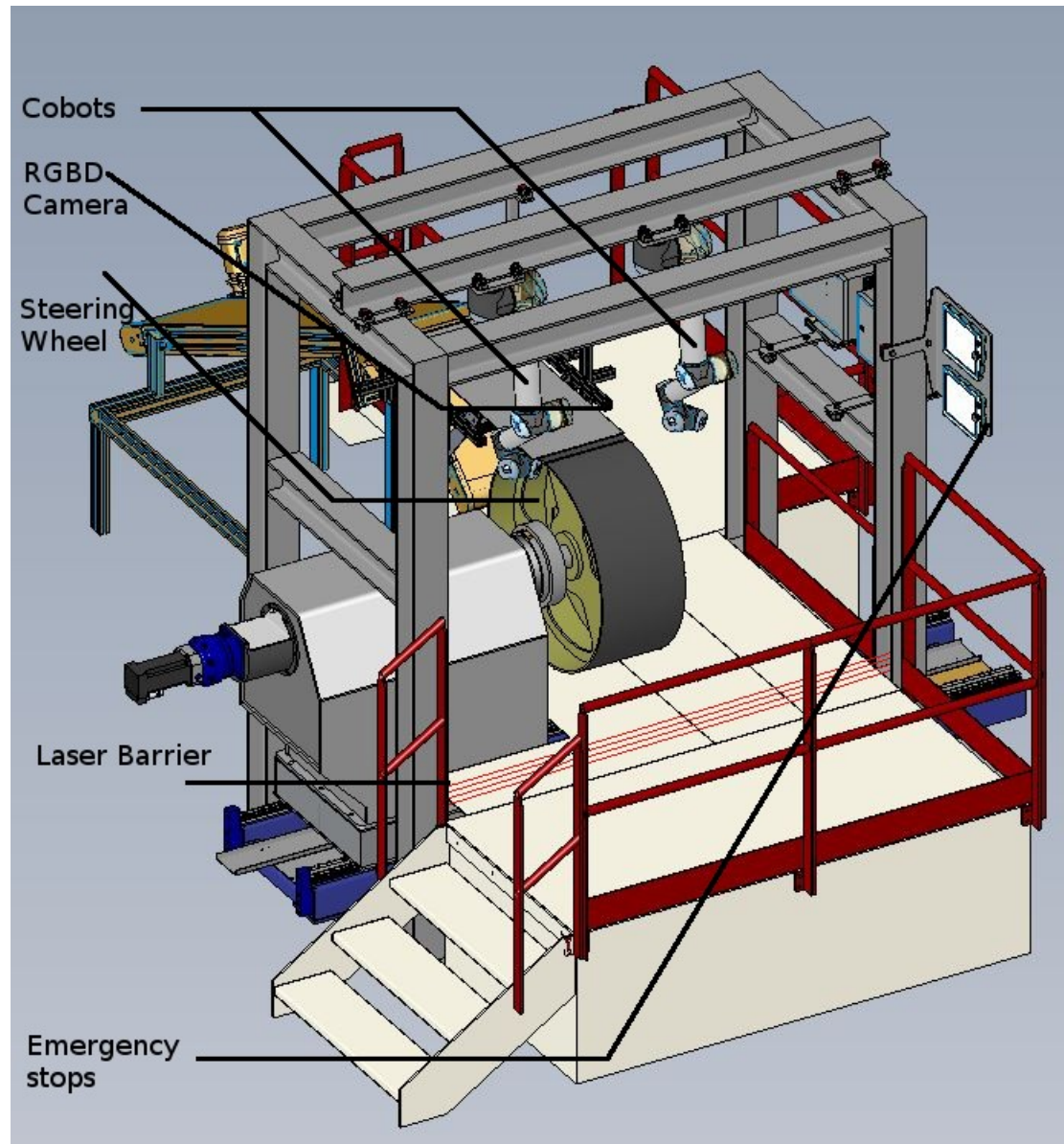
# Présentation de l'architecture



# Validation de l'architecture



# Banc de test



# Contributions



- Architecture Open source pour la manipulation d'objets déformables
- Beaucoup de flexibilité dans l'implémentation pour ne pas restreindre les axes de recherche
- Modularisation du hardware pour la ré-utilisabilité
- Le dépôt Git sera mis à jour régulièrement avec les différents modules

# Conclusion



- Création d'une architecture logicielle pour les besoins du projet.
- Efforts mis sur la généricité et la ré-utilisabilité
- Tests concluants.

# Questions ?

