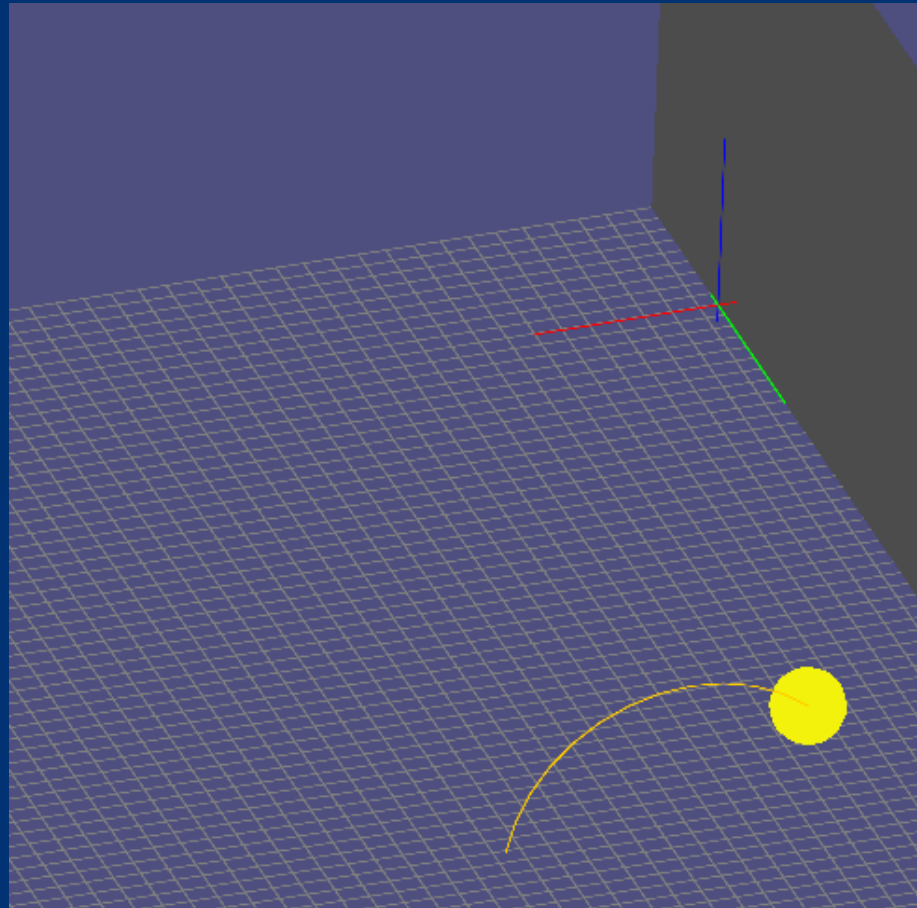


Ball catcher

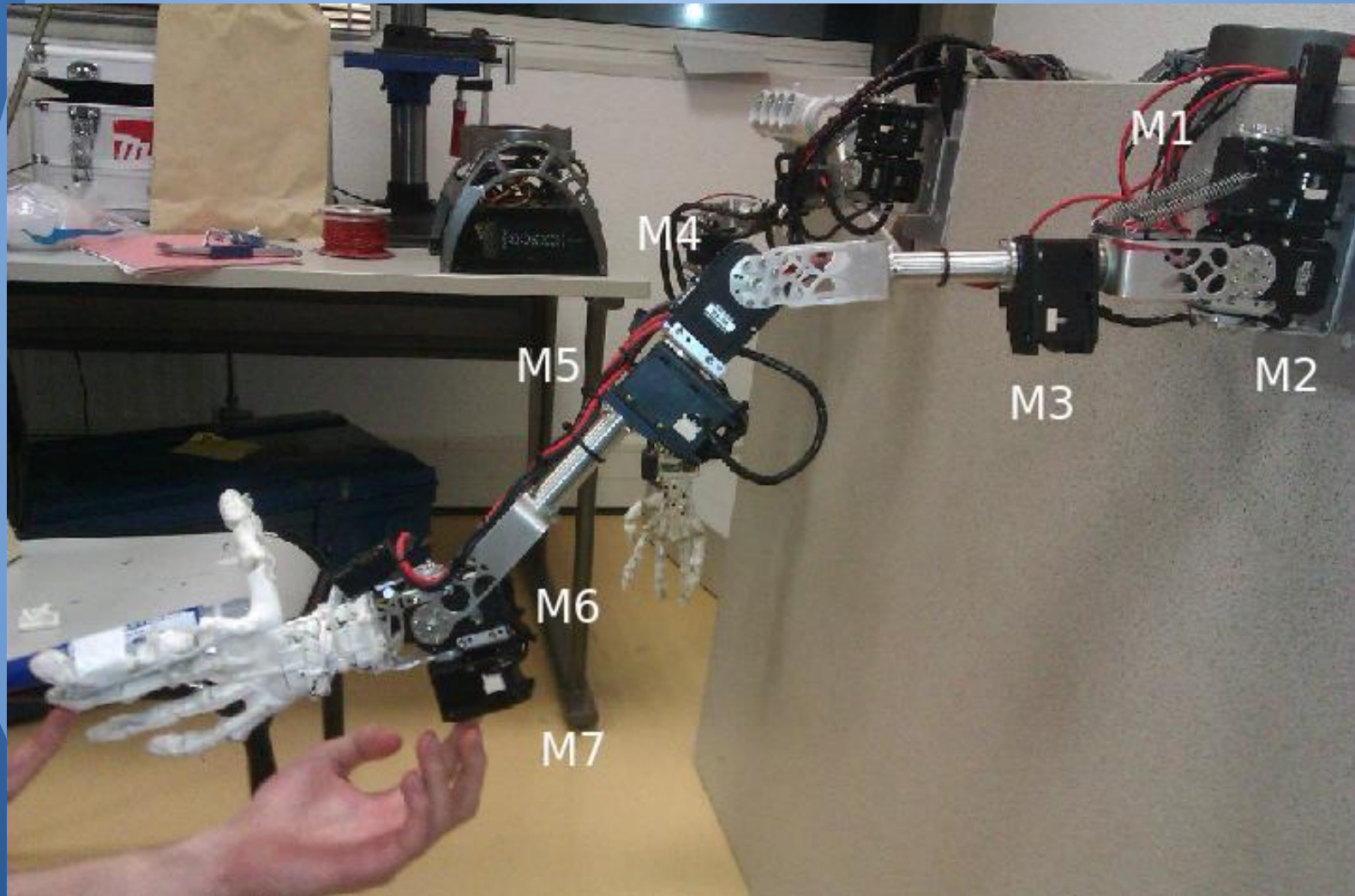
Thibaud Cheippe, Victor Cheimanoff, Yoan Mollard



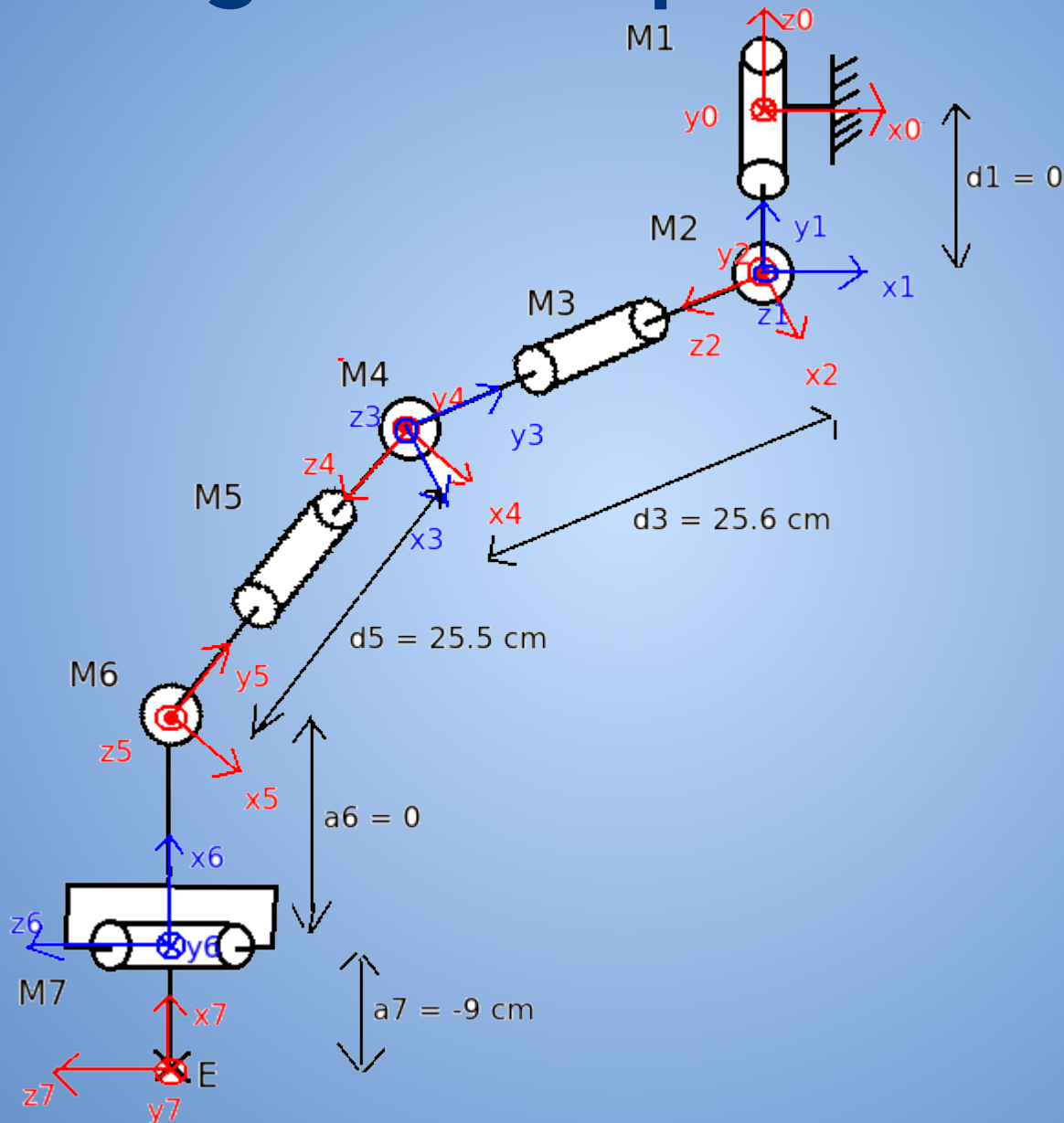
MLP Ball catcher

- Modèle géométrique
- Calcul de la trajectoire
- Le système Optitrack
- Le visualisateur

Modèle géométrique



Modèle géométrique



Modèle géométrique

- M4 est déterminé dans le triangle M2 M4 M6
- M3 est imposé pour figer le système
- M1 et M2 sont déterminés par deux équations de Paul de type 2
- M5, M6 et M7 sont déterminés par des équations trigonométriques

-> Deux solutions pour chaque moteur

Prédiction de trajectoire

$$x = A_x t + B_x$$

$$y = A_y t + B_y$$

$$z = -\frac{1}{2}gt^2 + A_z t + B_z$$

Prédiction de trajectoire

$$A_x = \frac{x_j - x_i}{t_j - t_i}$$

$$A_y = \frac{y_j - y_i}{t_j - t_i}$$

$$A_z = \frac{(z_j + \frac{1}{2}gt_j^2) - (z_i + \frac{1}{2}gt_i^2)}{t_j - t_i}$$

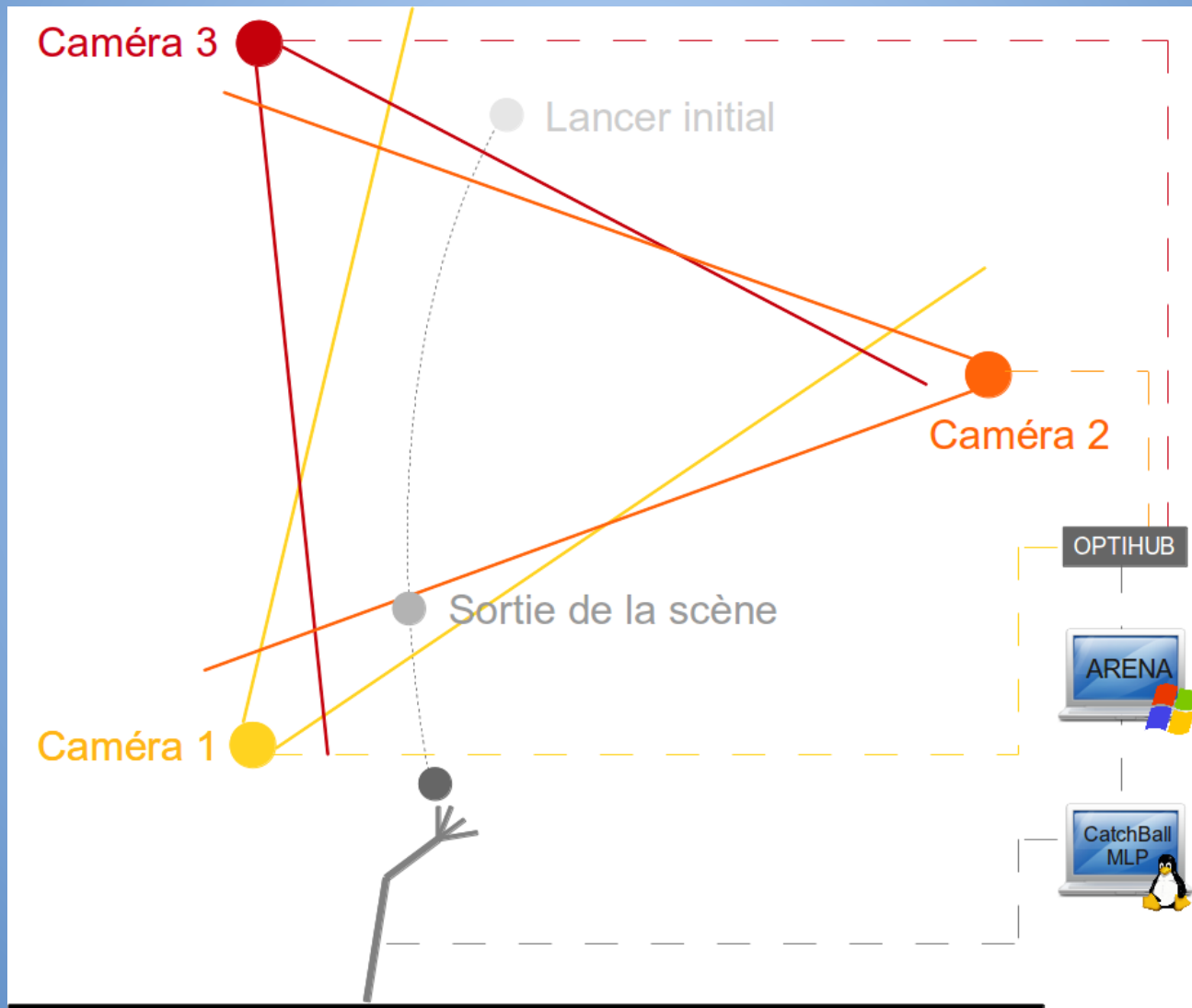
Prédiction de trajectoire

$$B_x = x_i - A_x t_i$$

$$B_y = y_i - A_y t - i$$

$$B_z = z_i + \frac{1}{2} g t_i^2 - A_z t_i$$

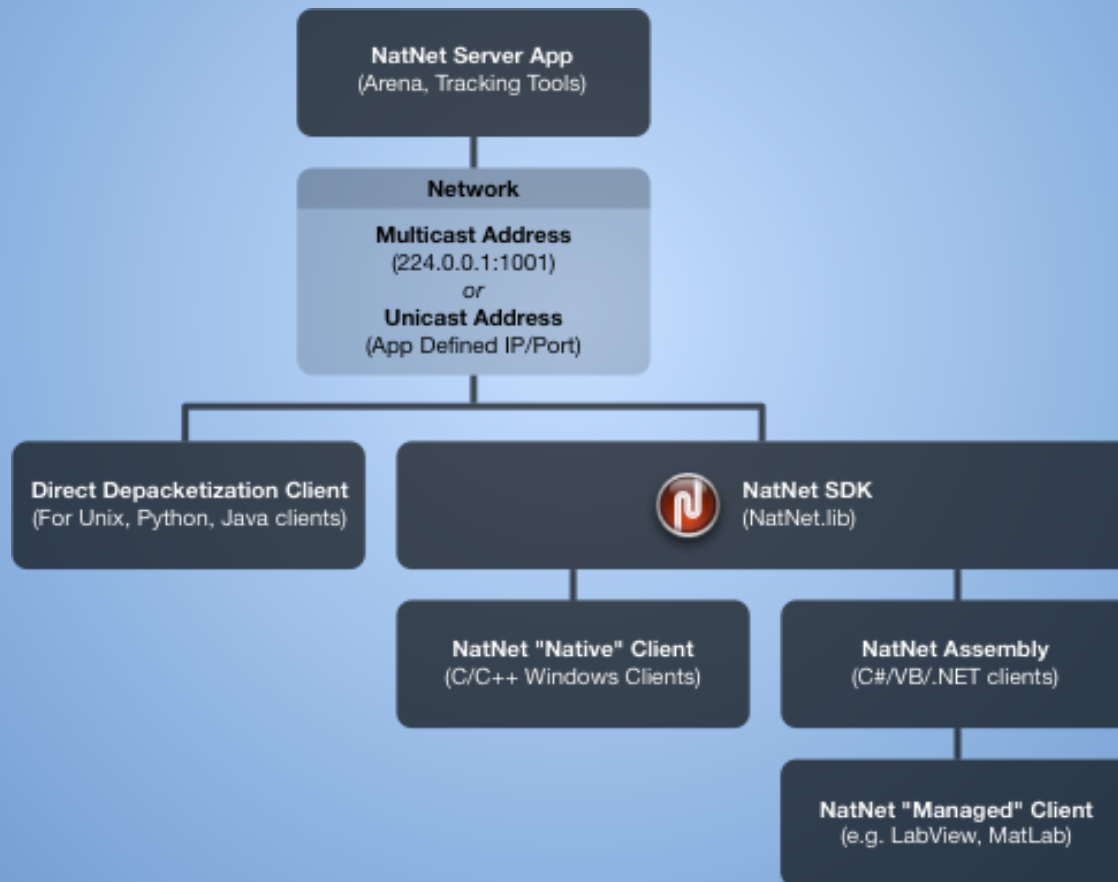
Le système Optitrack



Le système Optitrack

- Calibration des Caméras
- Création d'un Corps Rigide
- Exportation des données
- Calcul de l'erreur

Le SDK NatNet

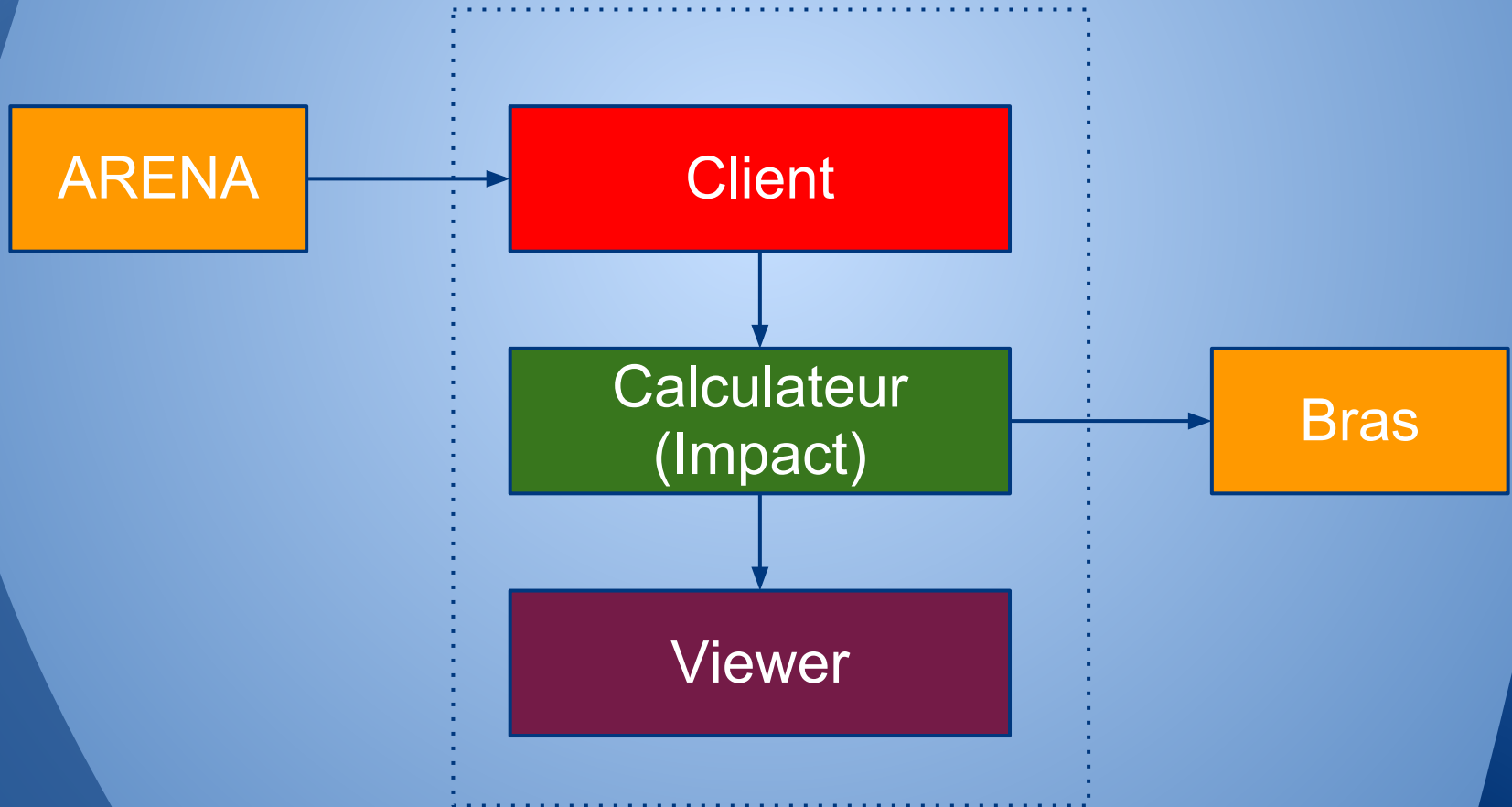


Le visualisateur 3D

- Idée = visualiser la trajectoire perçue et le point d'impact prédit
- Simuler voire se substituer au bras matériel
- Basé sur OpenGL et Glut pour QT

Architecture globale

- ~\$./Client | ./Impact | ./Viewer



To be done...

- Client : Déboguage + Astuce pour utiliser la balle sans marqueur
- Calculatoire : Calcul trajectoire ok. Finir implémentation C++ MGI
- Visualisateur : Reste à implémenter le bras et le lier à la partie calculatoire (protocole)