

gz-unitree: Reinforcement learning en robotique avec validation par moteurs de physique multiples pour le robot *H1v2* d'Unitree

Gwenn Le Bihan <gwenn.lebihan@etu.inp-n7.fr>

6 Novembre 2025

Reinforcement Learning

Et son application à la robotique

Bases du RL

Agent

Environnement

Score

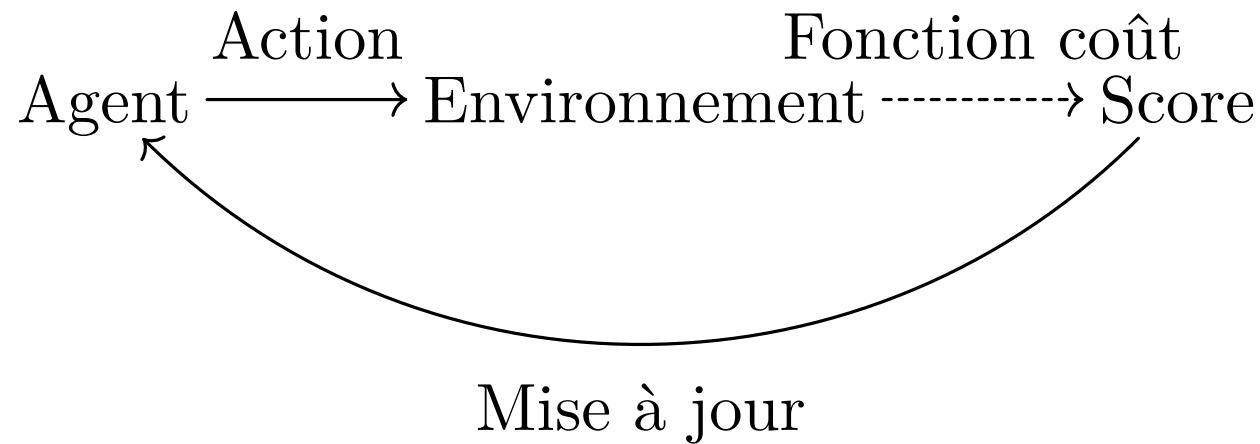
Bases du RL



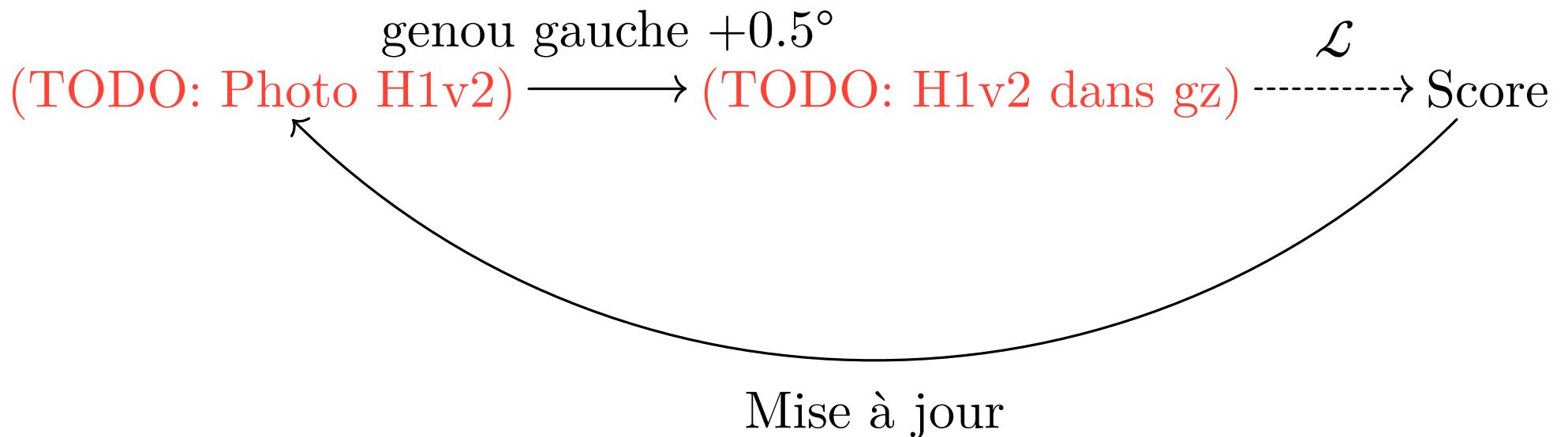
Bases du RL



Bases du RL



RL en robotique



C'est quoi \mathcal{L} ?

C'est très simple:

$$\mathcal{L}_r(\pi', \pi) := \mathbb{E}_{(s_t, a_t)_{t \in \mathbb{N}} \in \mathcal{C}} \sum_{t=0}^{\infty} \frac{Q_{\pi}(s_t, a_t)}{Q_{\pi'}(s_t, a_t)} A_{\pi, r}(s_t, a_t)$$

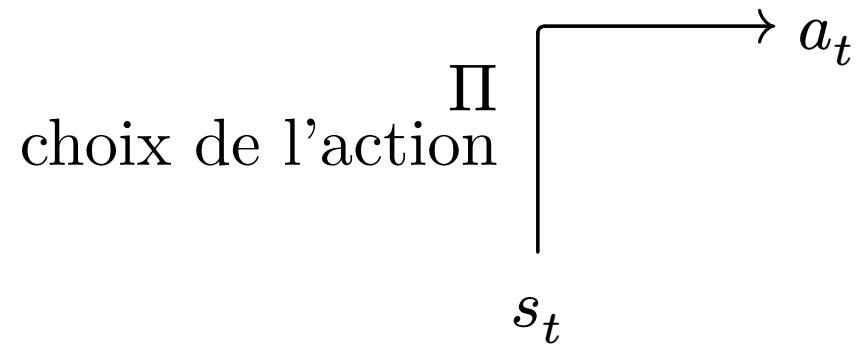
Comparaison des politiques

En Reinforcement Learning

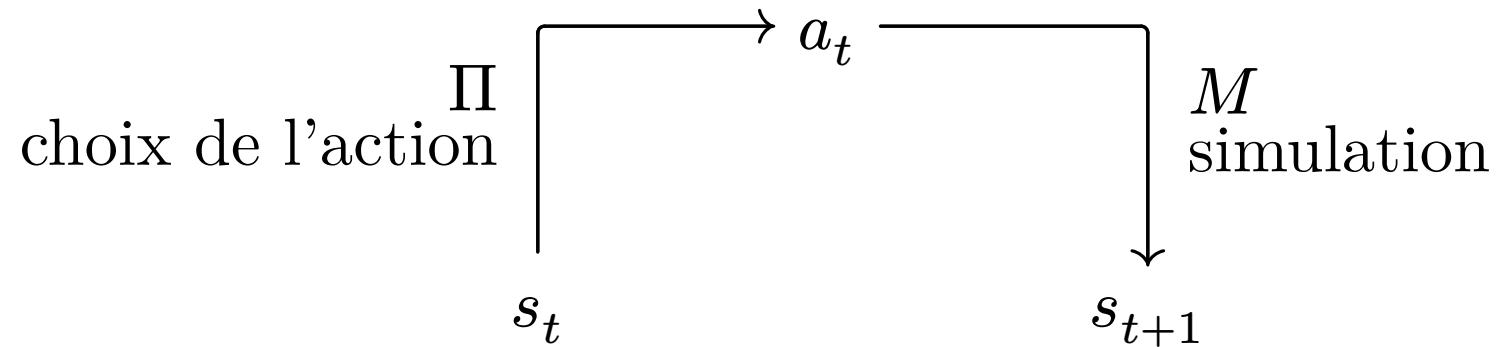
Comparaison des politiques

s_t

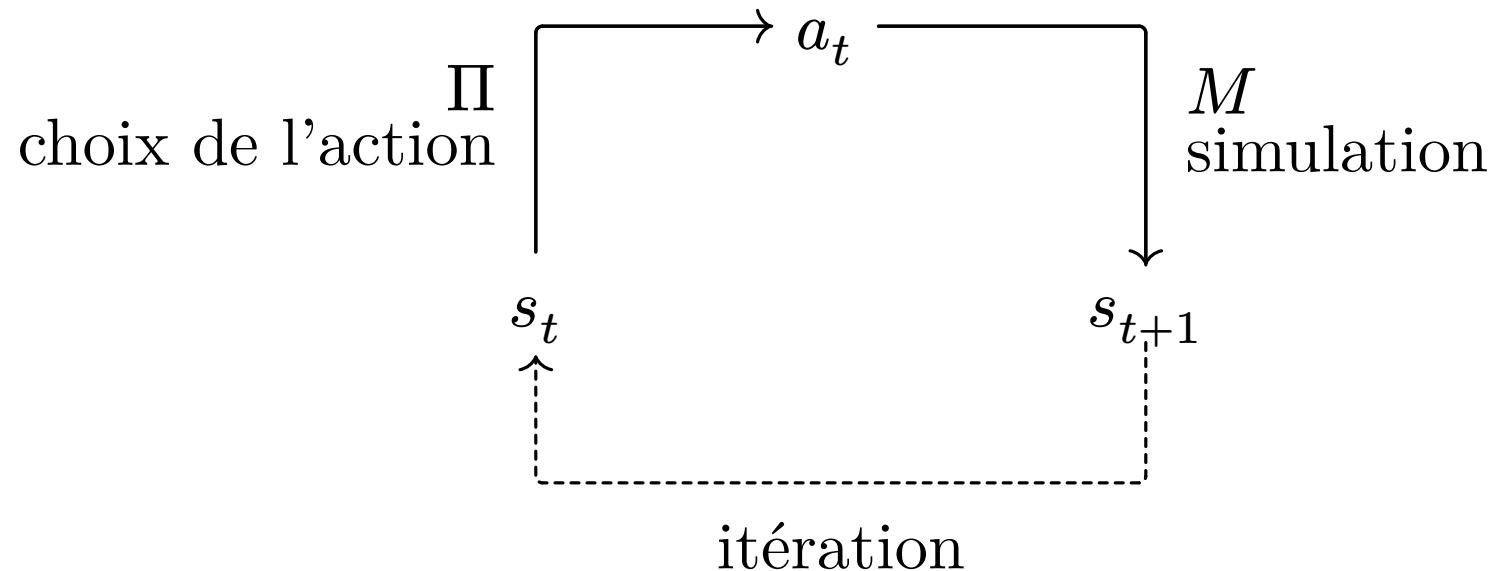
Comparaison des politiques



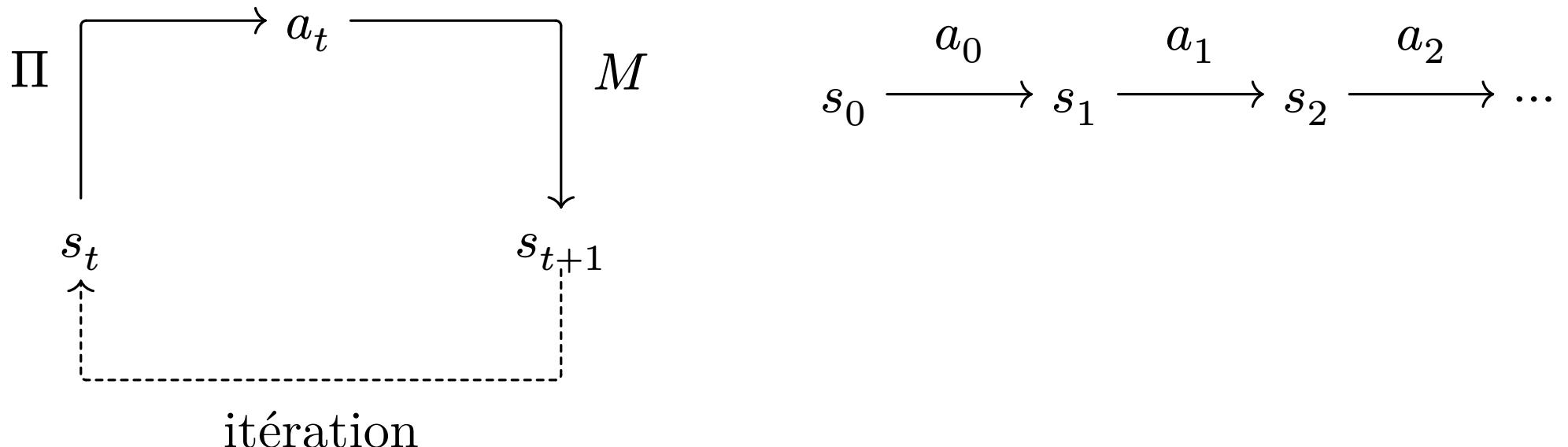
Comparaison des politiques



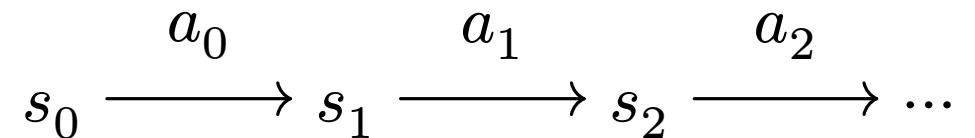
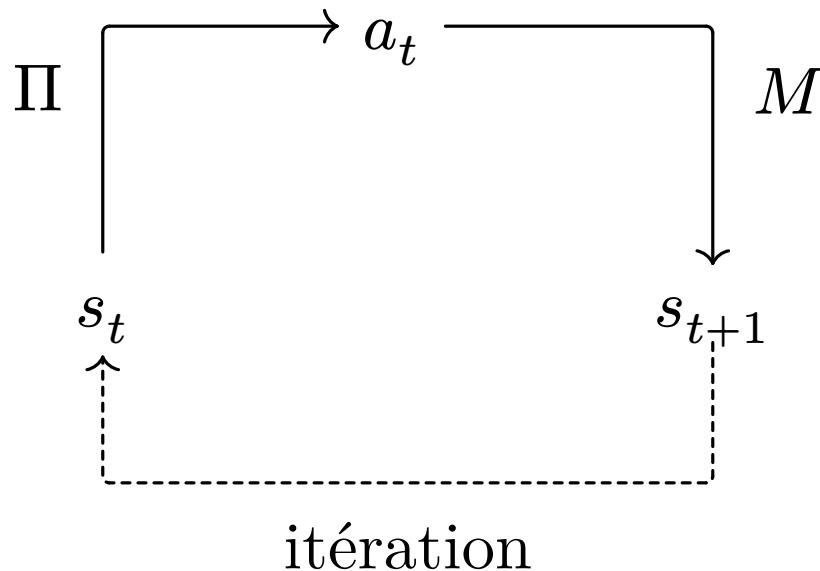
Comparaison des politiques



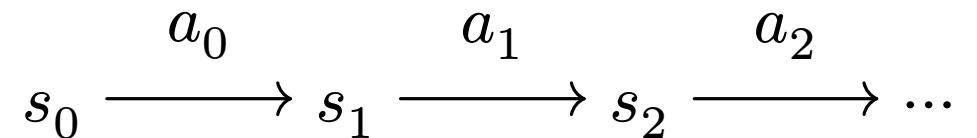
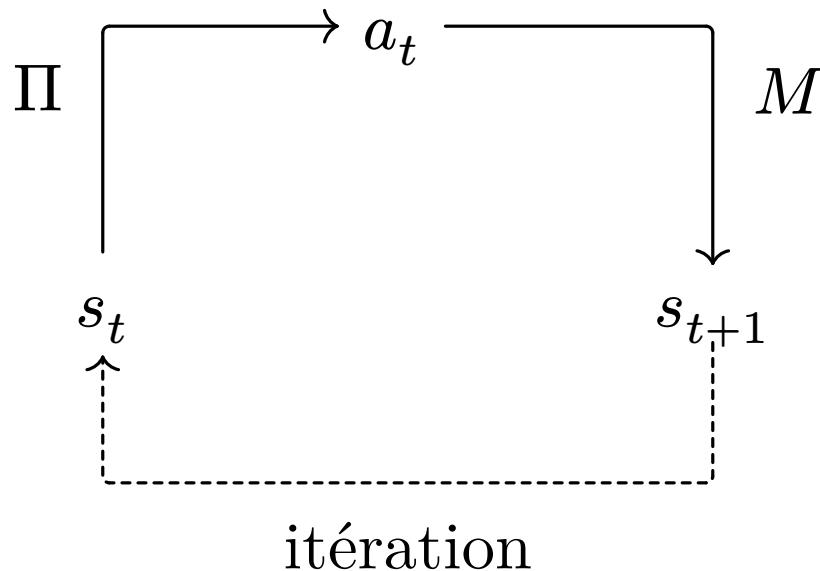
Comparaison des politiques



Comparaison des politiques



Comparaison des politiques



$$((s_0, a_0), (s_1, a_1), (s_2, a_2), \dots) \in \mathcal{C}$$

Comparaison des politiques

A := actions possibles

S := états possibles

$$\mathcal{C} := \left\{ \left\{ \forall t \in \mathbb{N} \quad \begin{array}{l} c_0 = (s_0, a_0) \\ c_{t+1} = (M(c_t), a_t) \end{array} \right| (s_0, a) \in S \times A^{\mathbb{N}} \right\}$$

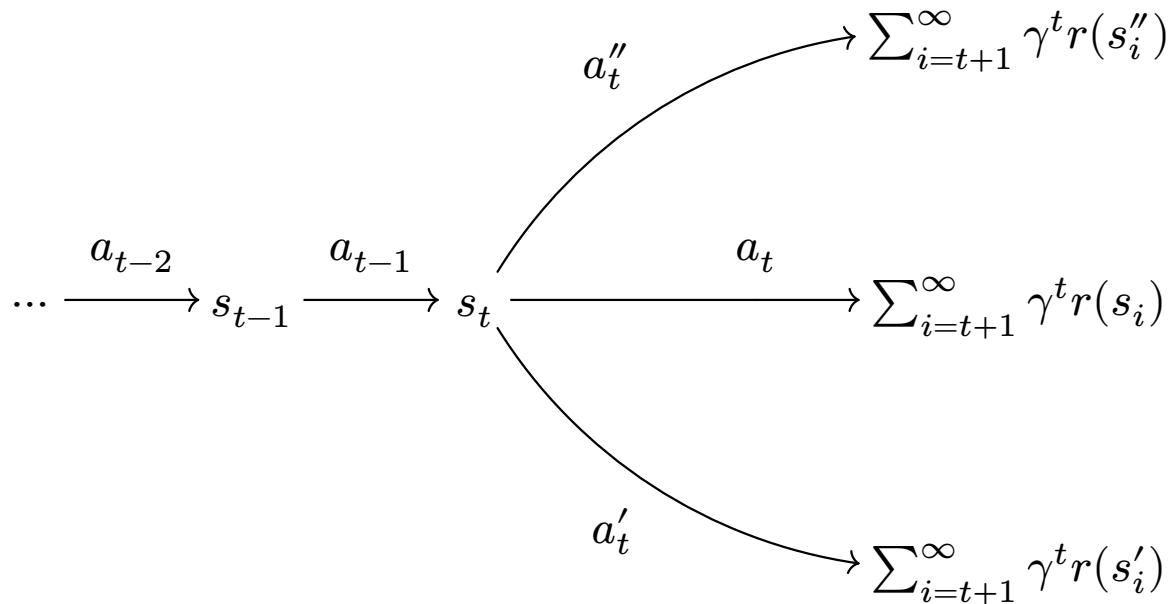
Comparaison des politiques: Avantage A

À quel point est-il mieux de choisir a_t plutôt qu'une autre action?

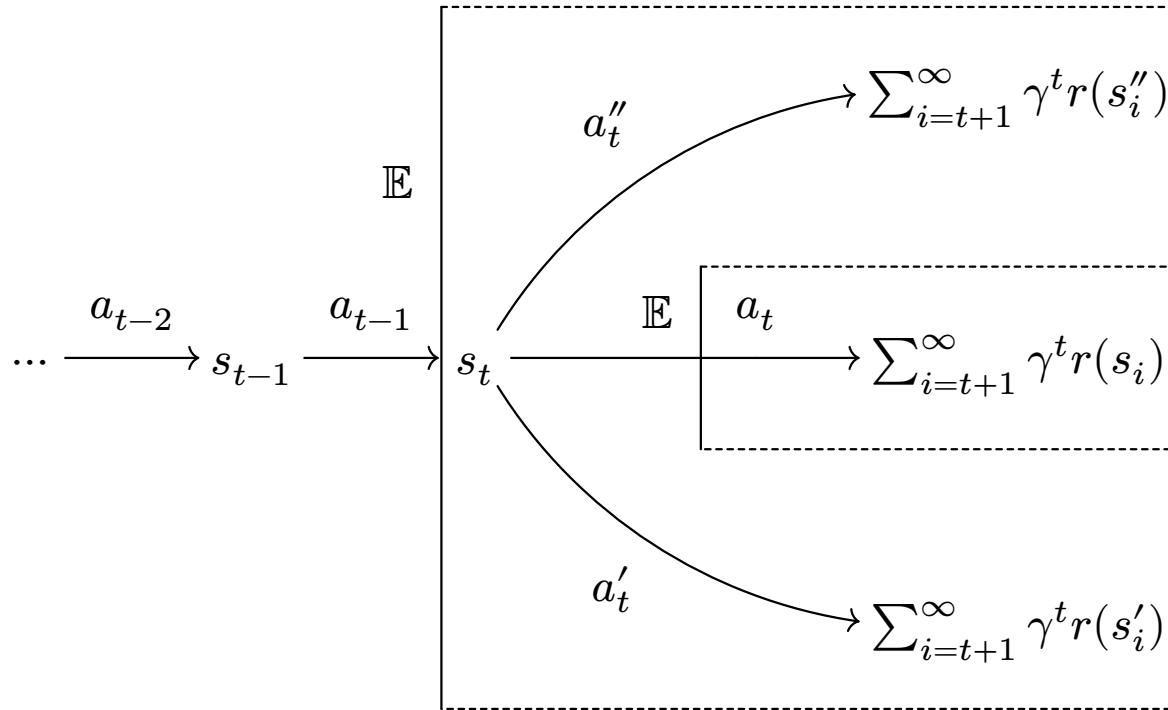
Comparaison des politiques: Avantage *A*

$$\dots \xrightarrow{a_{t-2}} s_{t-1} \xrightarrow{a_{t-1}}$$

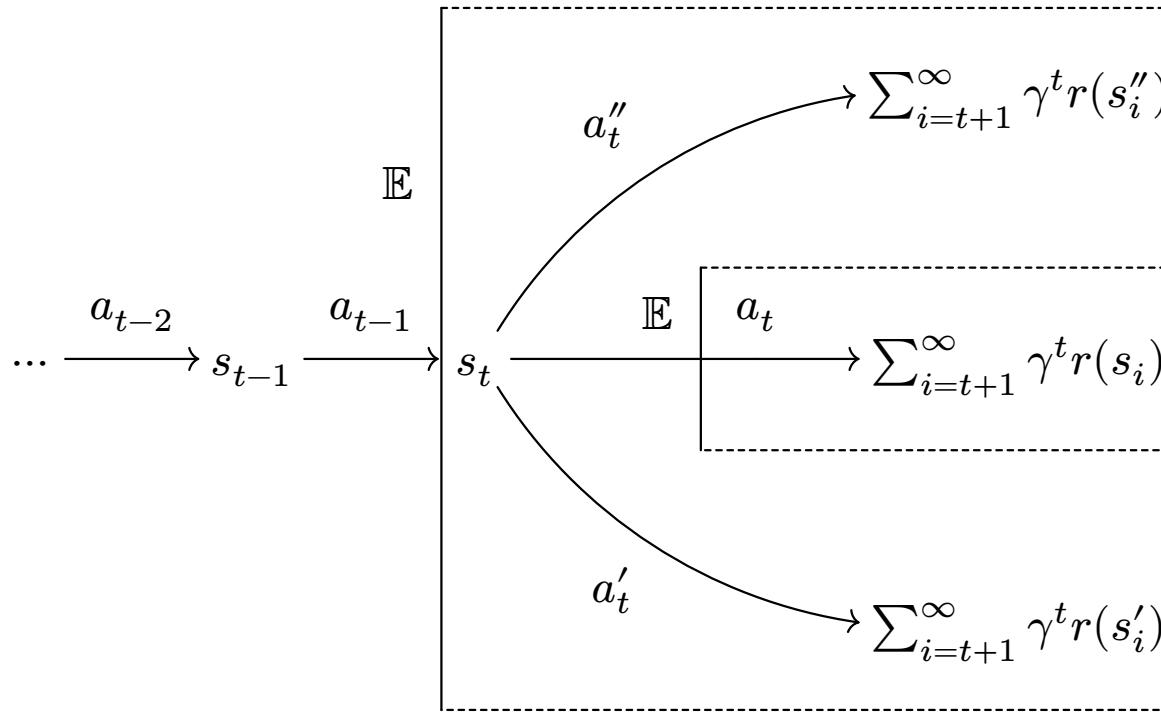
Comparaison des politiques: Avantage A



Comparaison des politiques: Avantage A



Comparaison des politiques: Avantage A



$$A_{\pi,r}(s, a) := \mathbb{E}(\text{avec } a_t) - \mathbb{E}(\text{à } t-1)$$

C'est quoi \mathcal{L} ?

$$\mathcal{L}_r(\pi', \pi) :=$$

C'est quoi \mathcal{L} ?

$$\mathcal{L}_r(\pi', \pi) := \mathbb{E}_{(s_t, a_t)_{t \in \mathbb{N}} \in \mathcal{C}}$$

C'est quoi \mathcal{L} ?

$$\mathcal{L}_r(\pi', \pi) := \mathbb{E}_{(s_t, a_t)_{t \in \mathbb{N}} \in \mathcal{C}} \sum_{t=0}^{\infty}$$

C'est quoi \mathcal{L} ?

$$\mathcal{L}_r(\pi', \pi) := \mathbb{E}_{(s_t, a_t)_{t \in \mathbb{N}} \in \mathcal{C}} \sum_{t=0}^{\infty} \frac{Q_{\pi}(s_t, a_t)}{Q_{\pi'}(s_t, a_t)} A_{\pi, r}(s_t, a_t)$$

Mise à jour de Π

$$\Pi' = \begin{cases} \operatorname{argmax}_{\pi} \mathcal{L}_r(\pi, \Pi) \\ \text{s.c. } \operatorname{distance}(\Pi', \Pi) < \delta \end{cases}$$

Mise à jour de Π : distance entre politiques

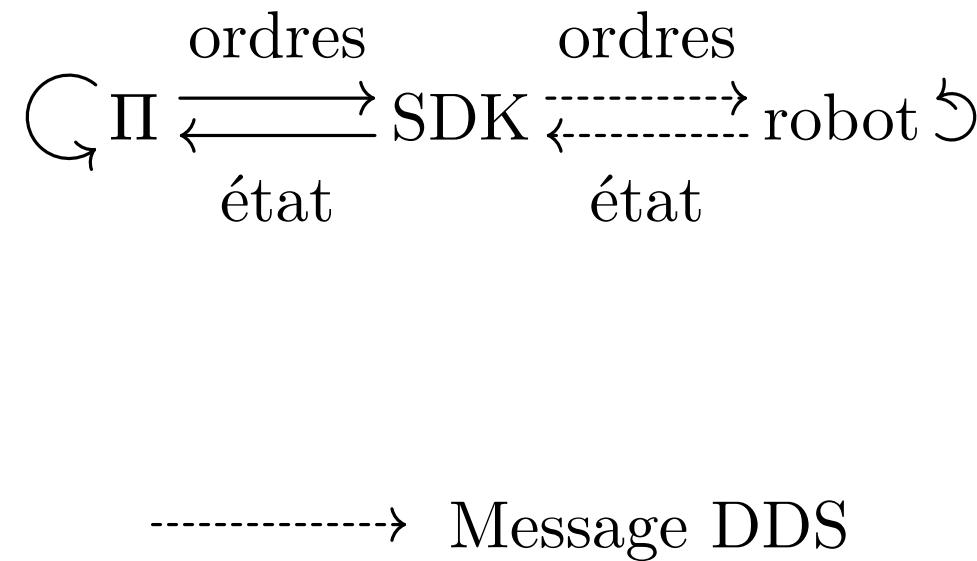
$$\text{distance}(\Pi', \Pi) := \max_{s \in S} D_{\text{KL}}(Q_{\Pi'}(s, \cdot) \parallel Q_{\Pi}(s, \cdot))$$

$$D_{\text{KL}}(P \parallel P') := \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{P'(x)}$$

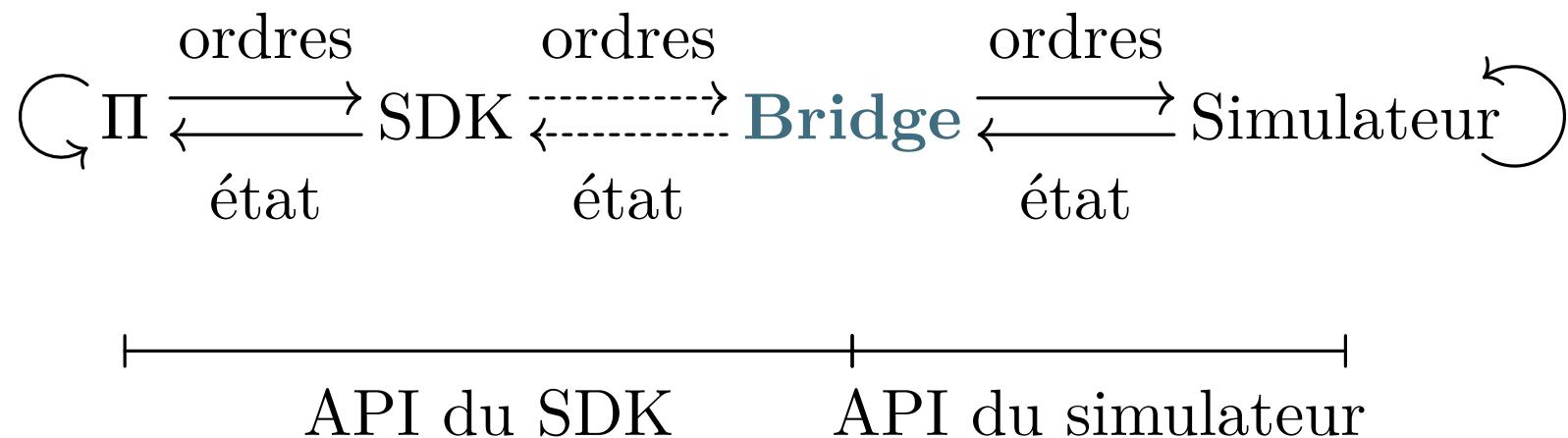
Le *SDK*¹ d'Unitree

¹Software Development Kit

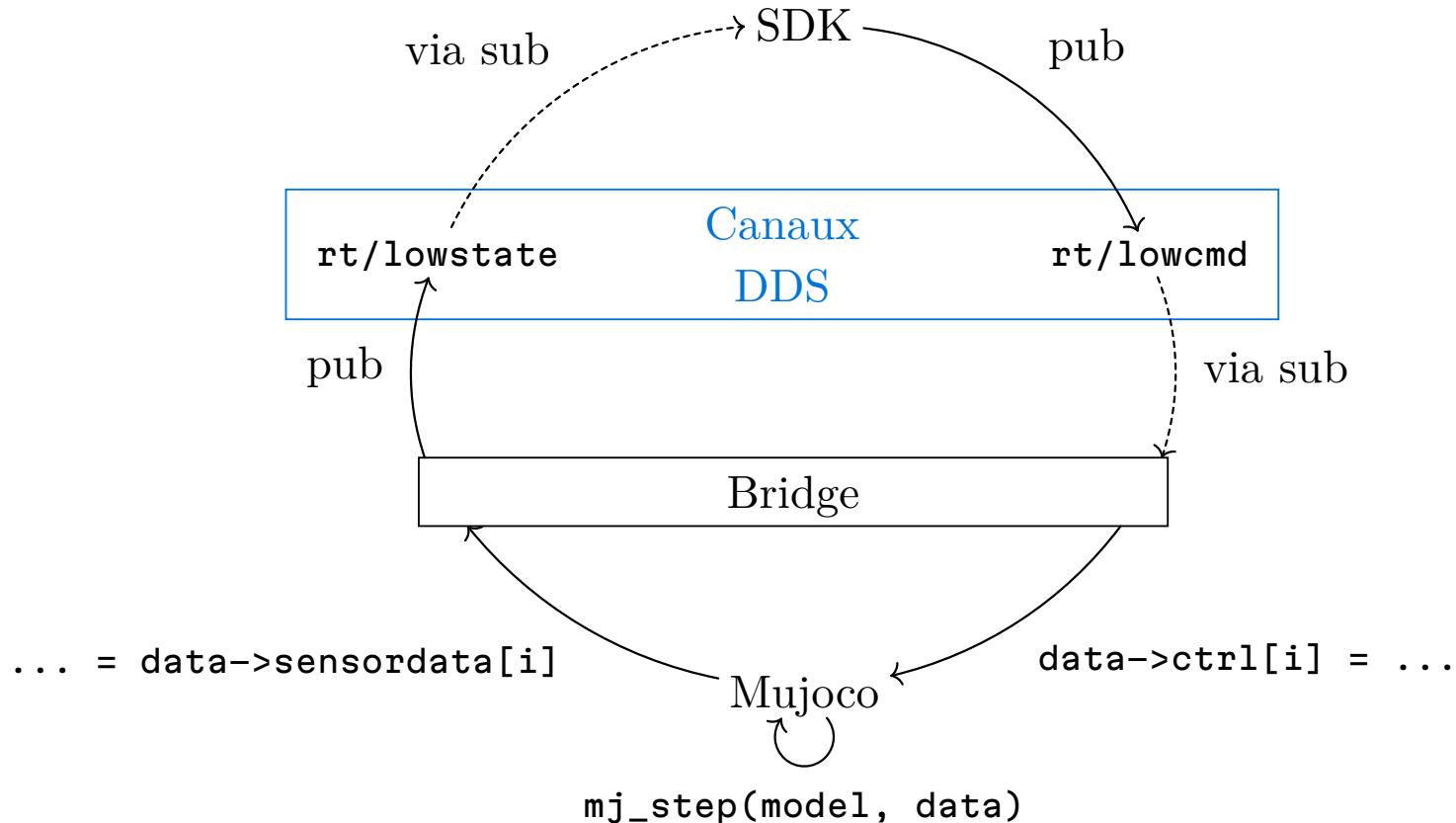
Le SDK d'Unitree



Le SDK d'Unitree



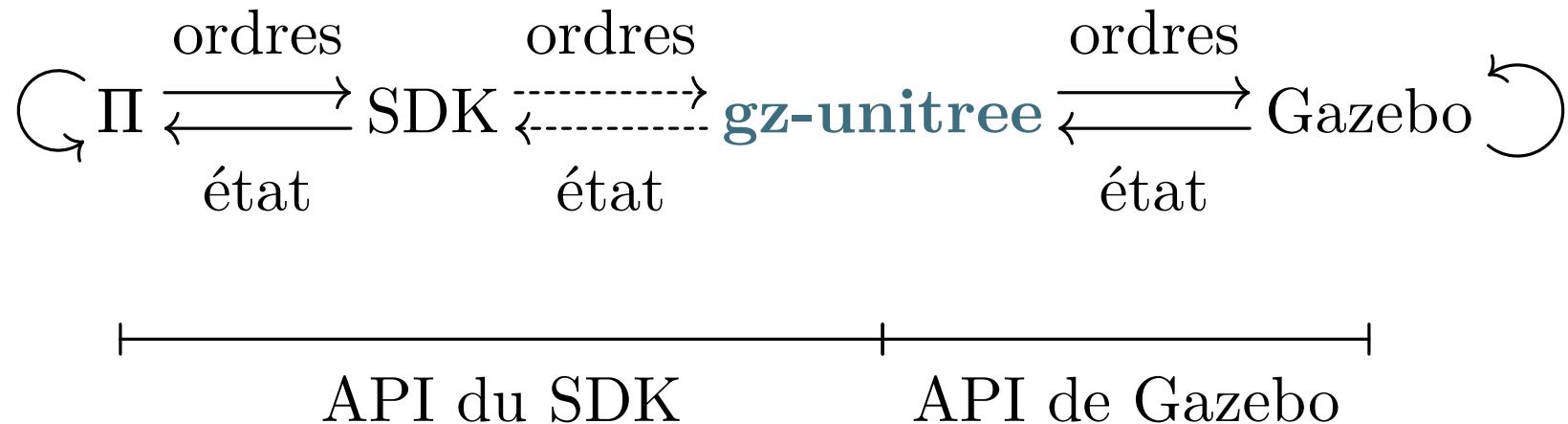
unitree_mujoco



Développement de *gz-unitree*

Un bridge pour Gazebo

Développement de *gz-unitree*



Développement de *gz-unitree*

```
#include <gz/sim/System.hh>
namespace gz_unitree
{
    class UnitreePlugin :
        public gz::sim::System,
        public gz::sim::ISystemPreUpdate
    {
public:
    UnitreePlugin();
public:
    ~UnitreePlugin() override;
public:
    void PreUpdate(const gz::sim::UpdateInfo &_info,
                  gz::sim::EntityComponentManager &ecm) override;
    };
}
```

Développement de *gz-unitree*

```
#include <gz/plugin/Register.hh>
...
... // class implementation
GZ_ADD_PLUGIN(
    UnitreePlugin,
    gz::sim::System,
    UnitreePlugin::ISystemPreUpdate)
```

```
<sdf version='1.11'>
<world name="default">
    <plugin filename="gz-unitree"
name="gz_unitree::UnitreePlugin">
        </plugin>
    </world>
    <model name='h1_description'>
        <link name='pelvis'>
            <inertial>
                ...
            </inertial>
        </link>
    </model>
</sdf>
```

(Π)

(Π)

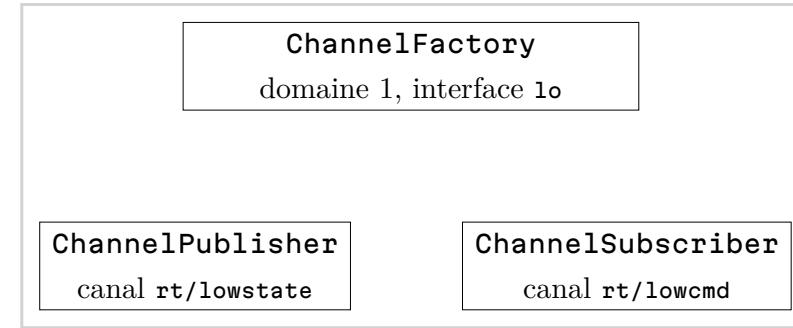
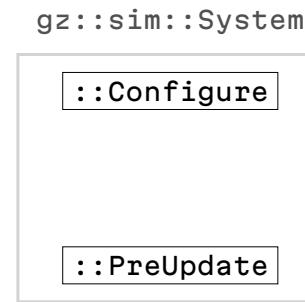
`gz::sim::System`

`::Configure`

`::PreUpdate`

Π

Unitree SDK



Π

Unitree SDK

`gz::sim::System`

`::Configure`

`::PreUpdate`

`ChannelFactory`

domaine 1, interface 1o

`ChannelPublisher`

canal `rt/lowstate`

`ChannelSubscriber`

canal `rt/lowcmd`

`::LowStateWriter`

`::CmdHandler`

State buffer

`statebuf`

Commands buffer

`cmdbuf`

`::TickHandler`

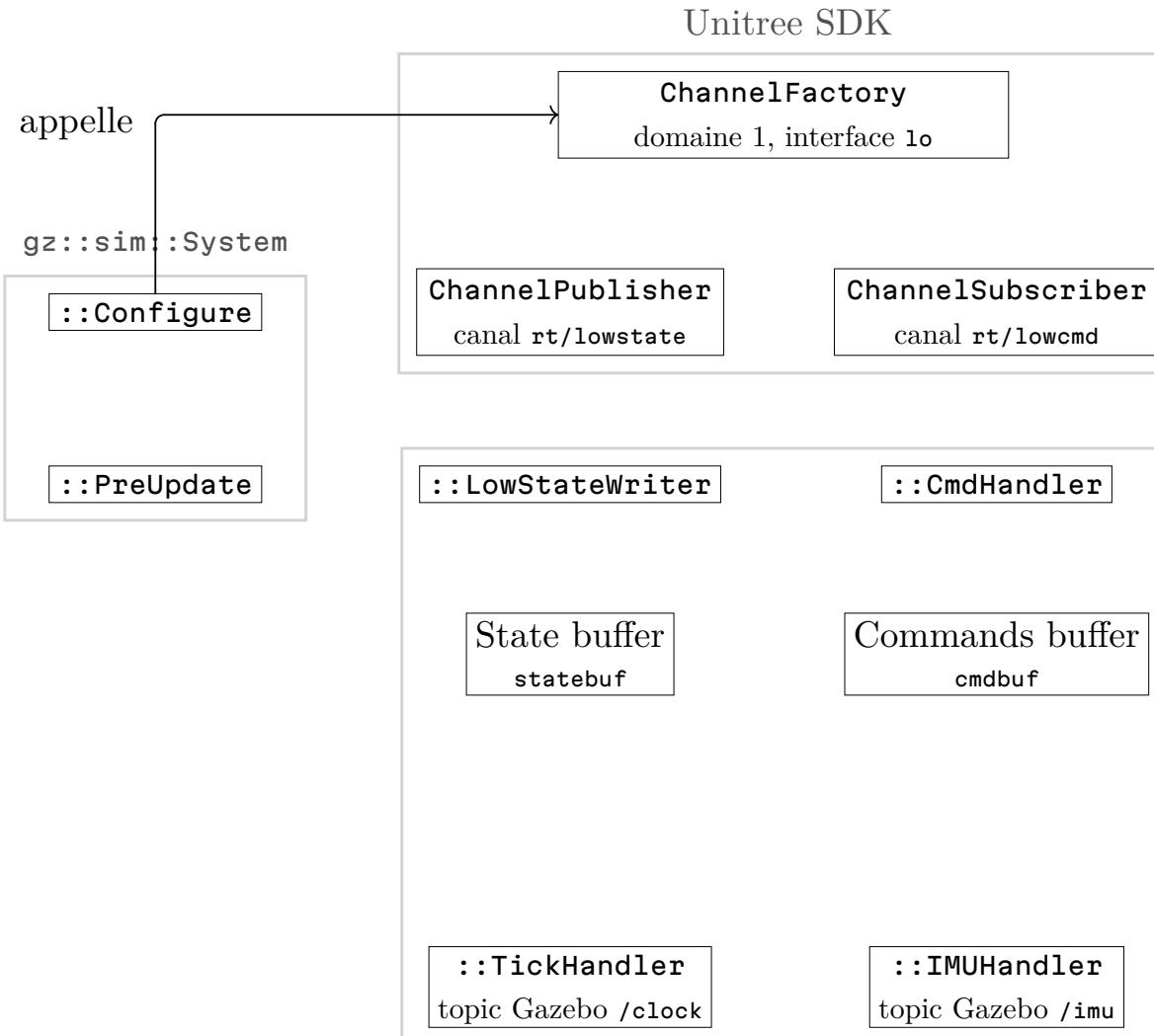
topic Gazebo `/clock`

`::IMUHandler`

topic Gazebo `/imu`

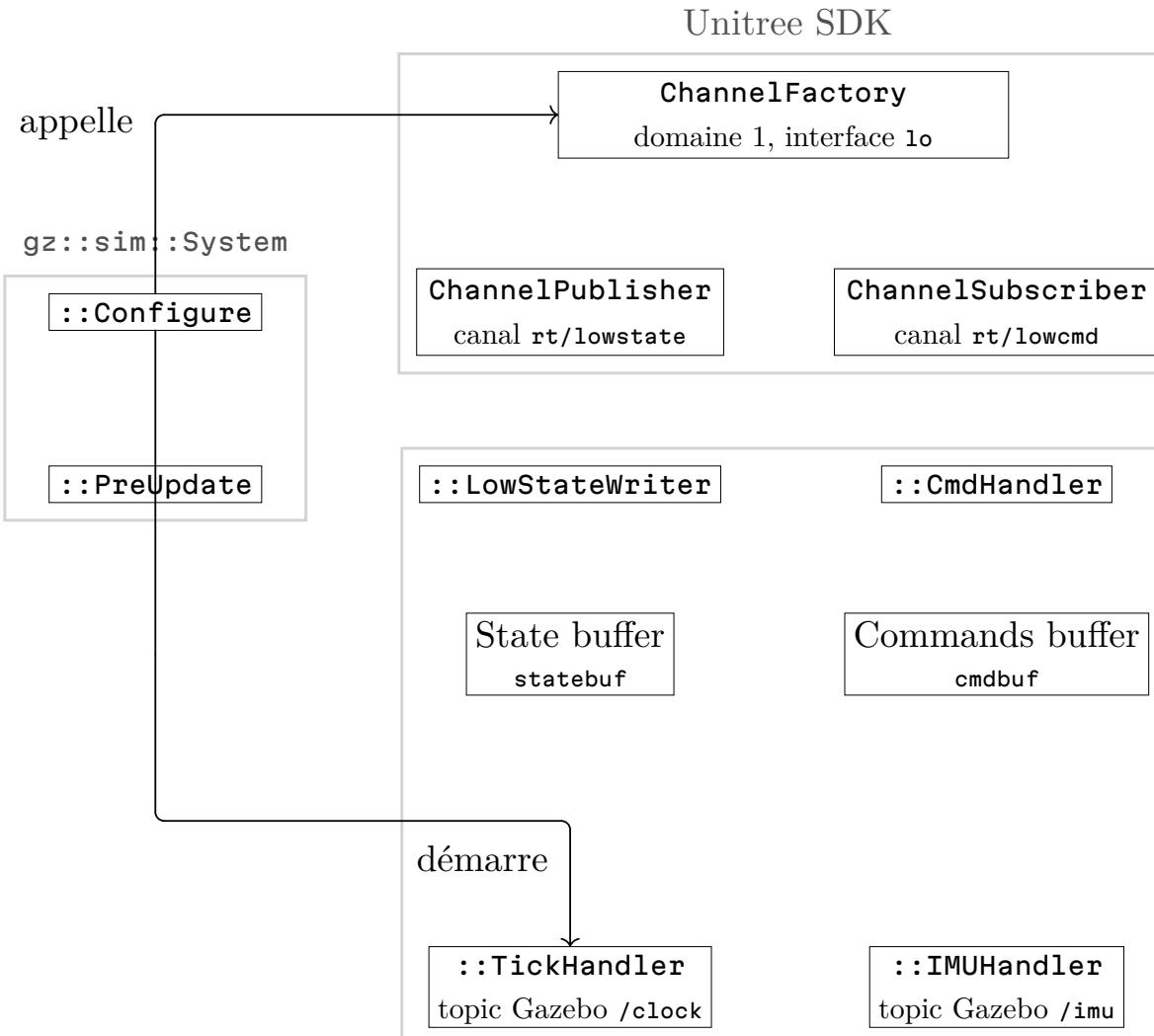
Plugin internals

Π

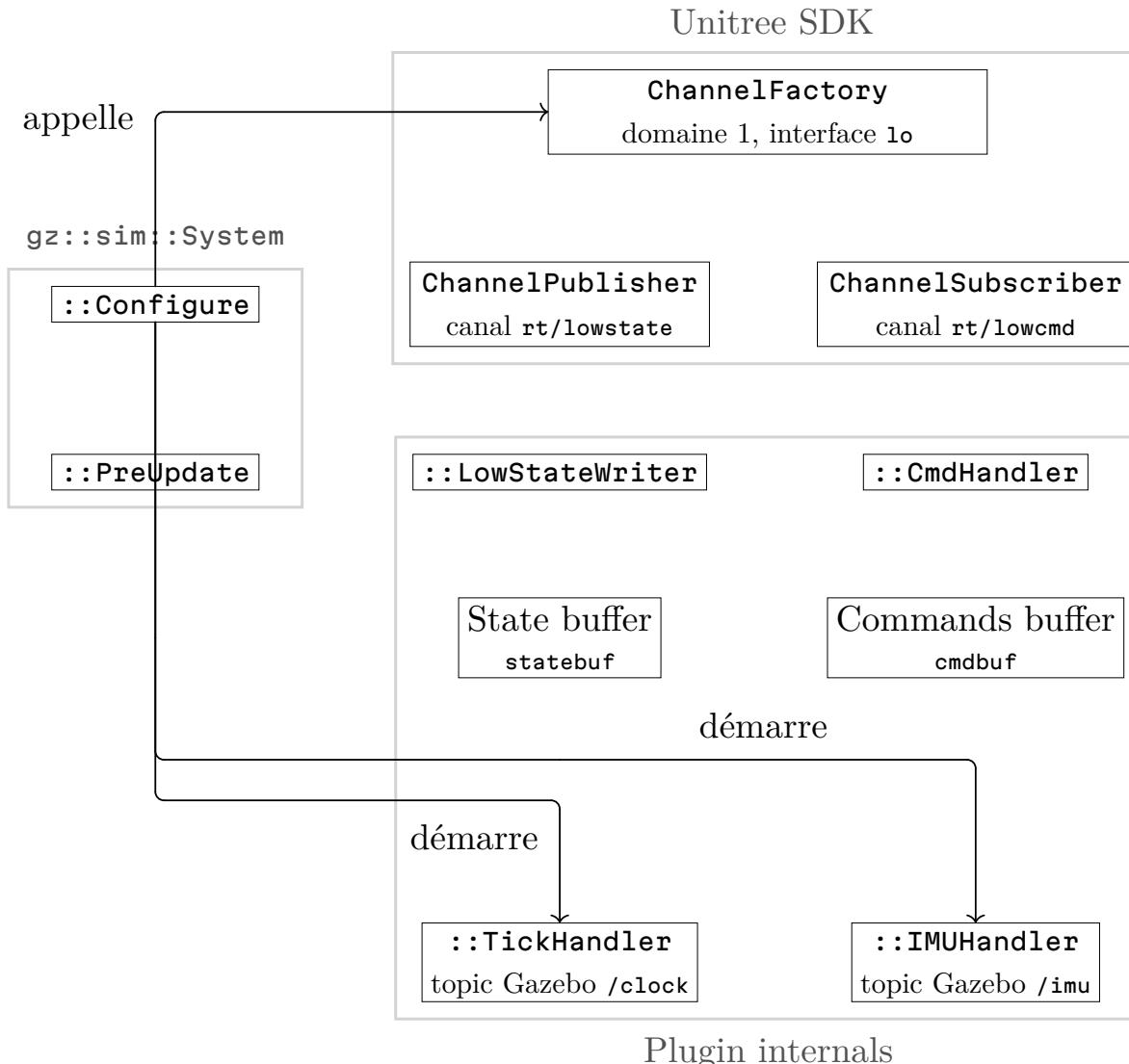


Plugin internals

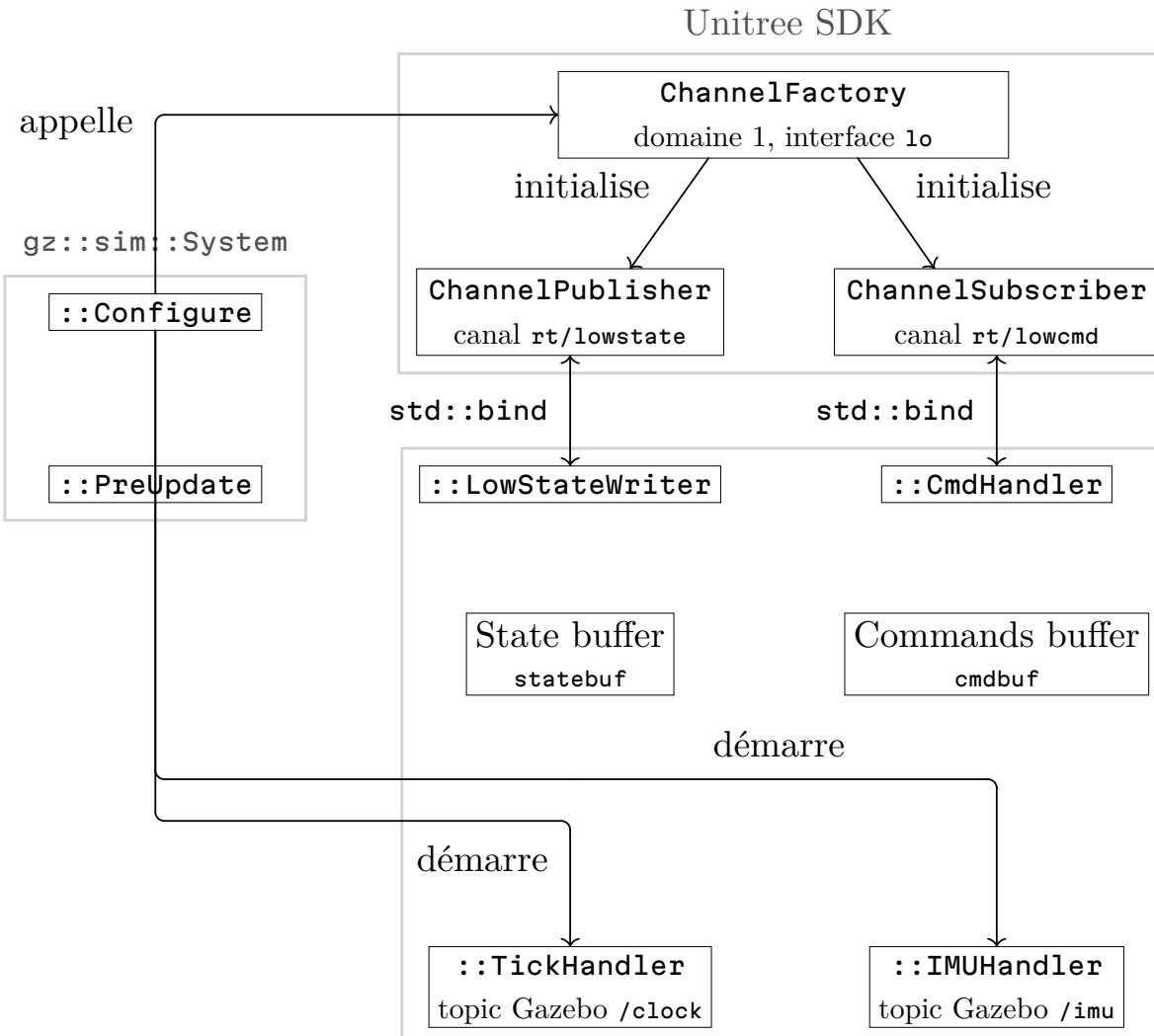
Π



II



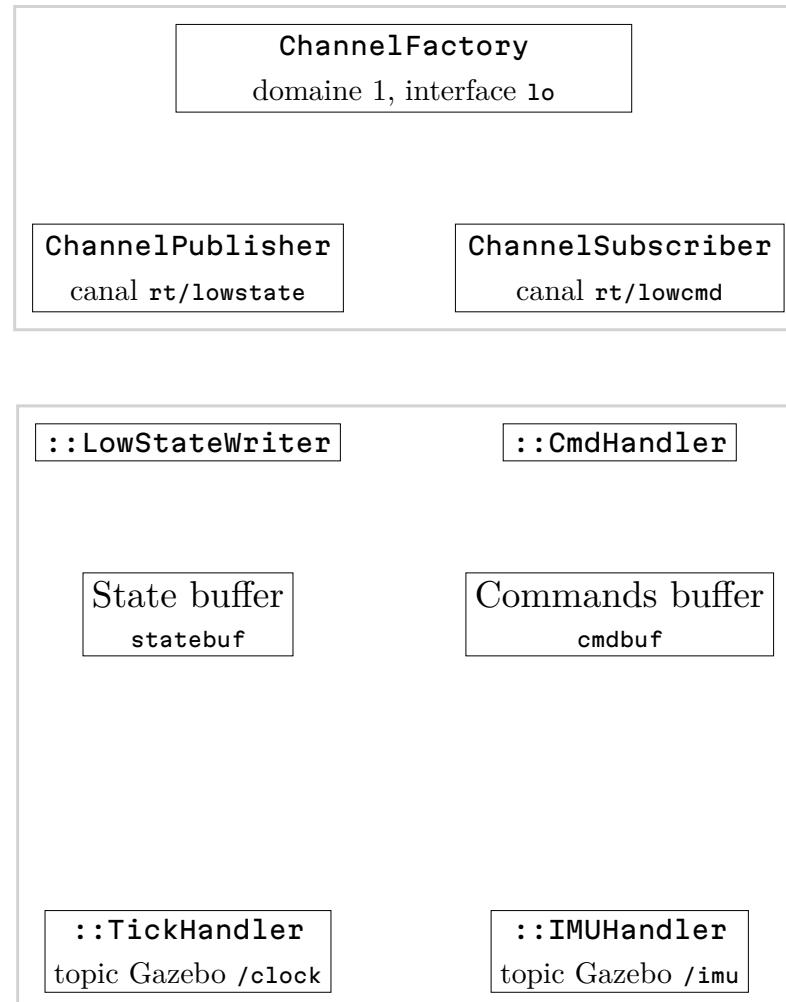
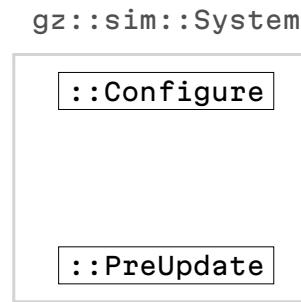
II



Plugin internals

II

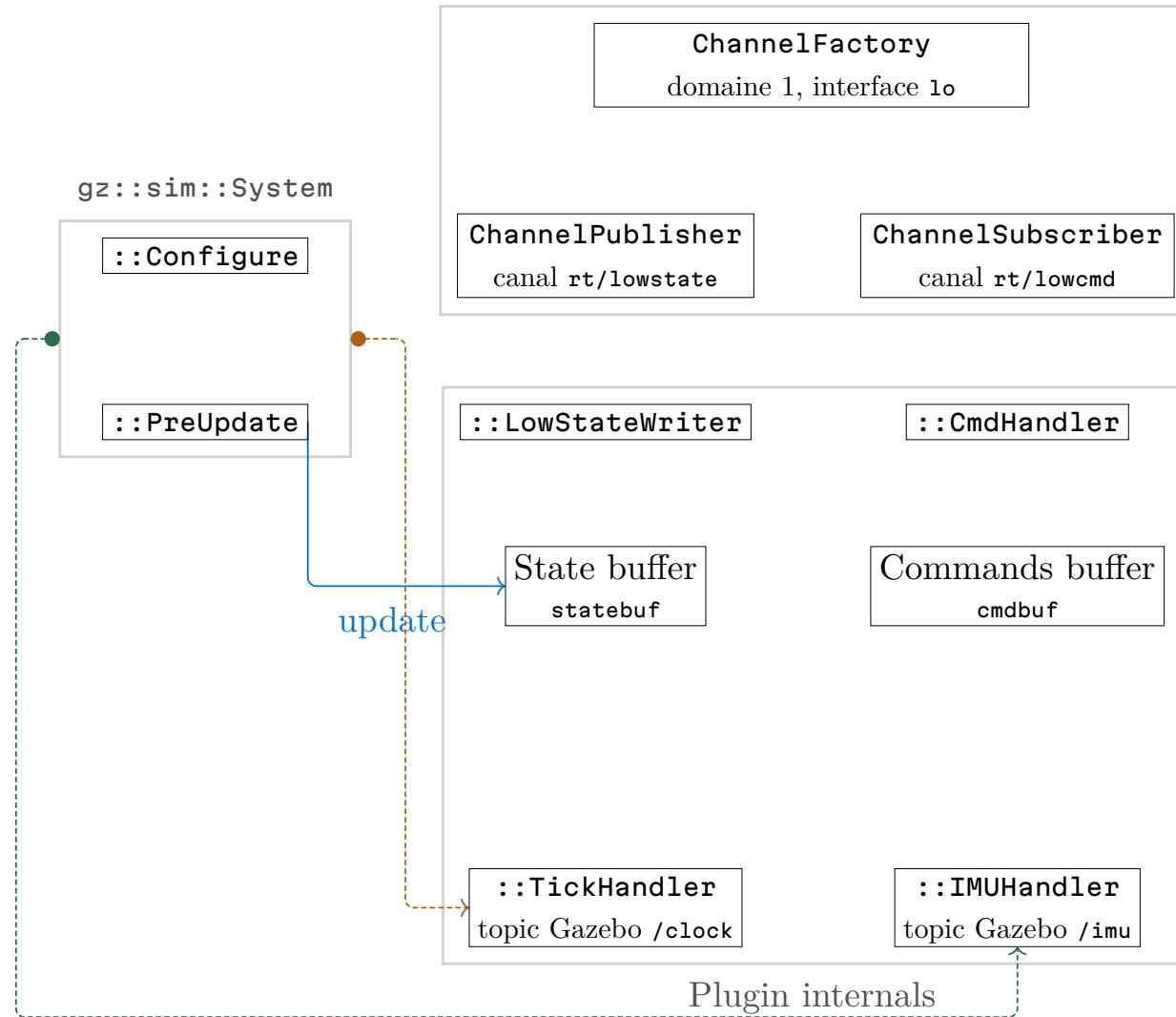
Unitree SDK



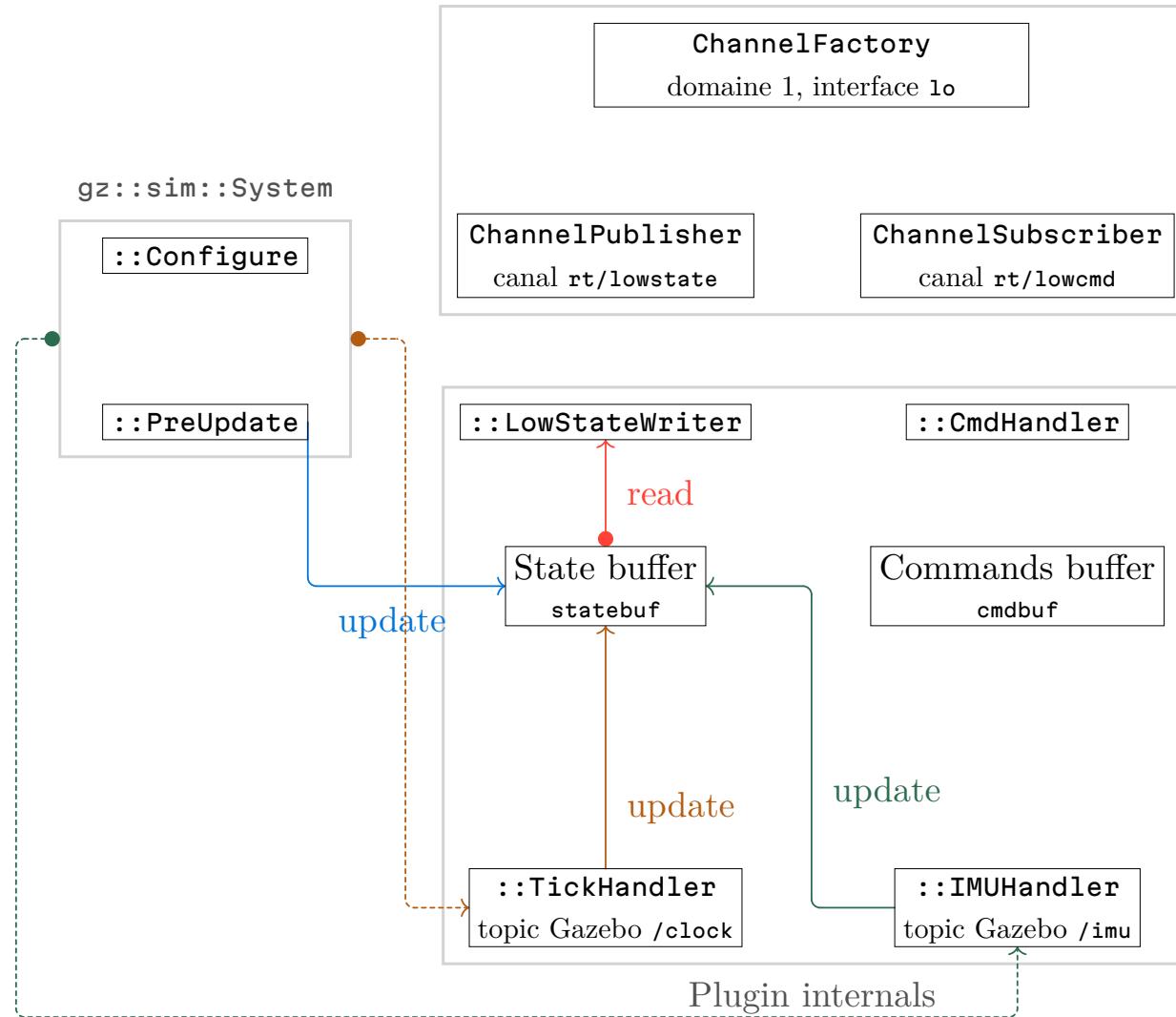
Plugin internals

II

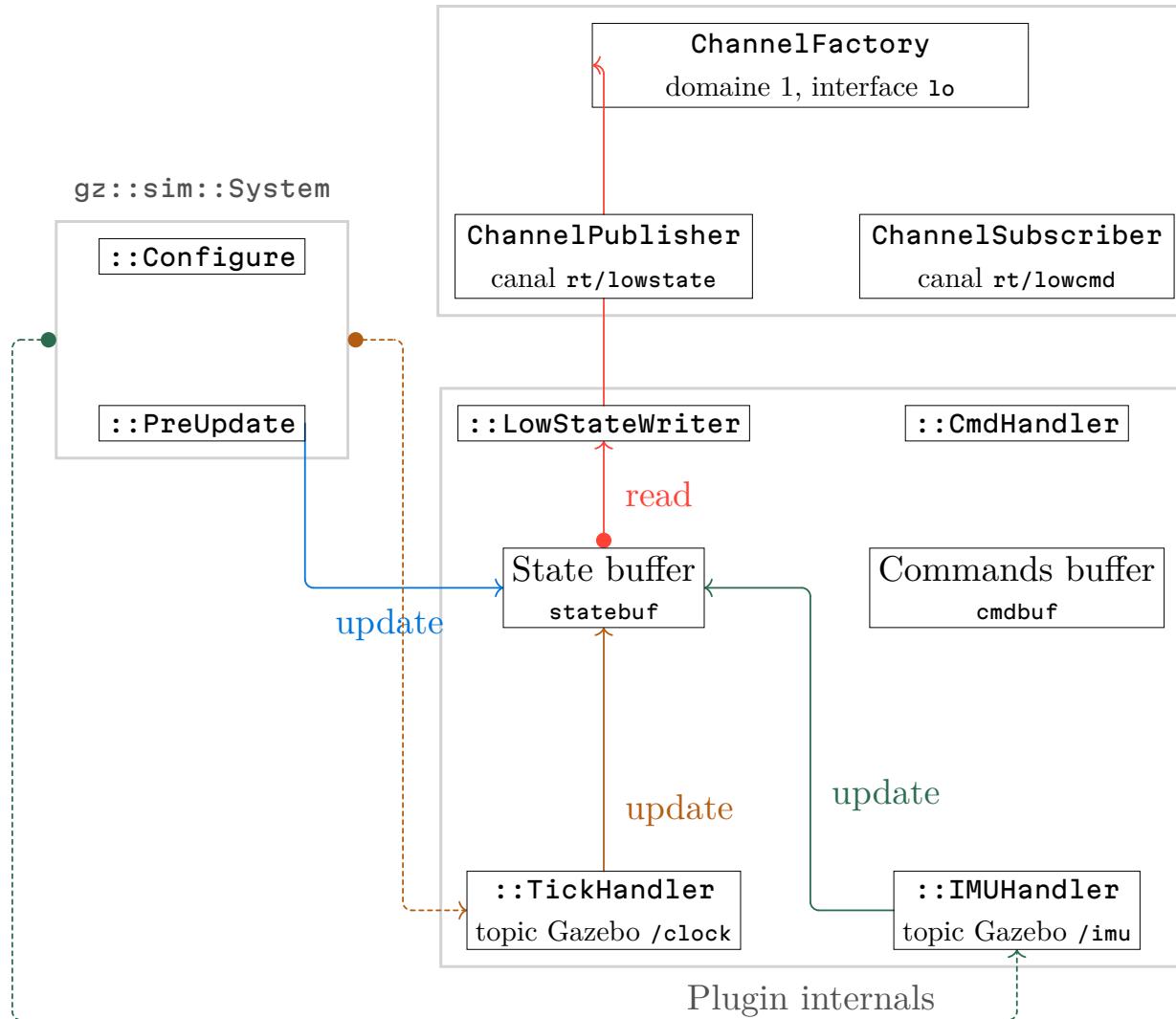
Unitree SDK

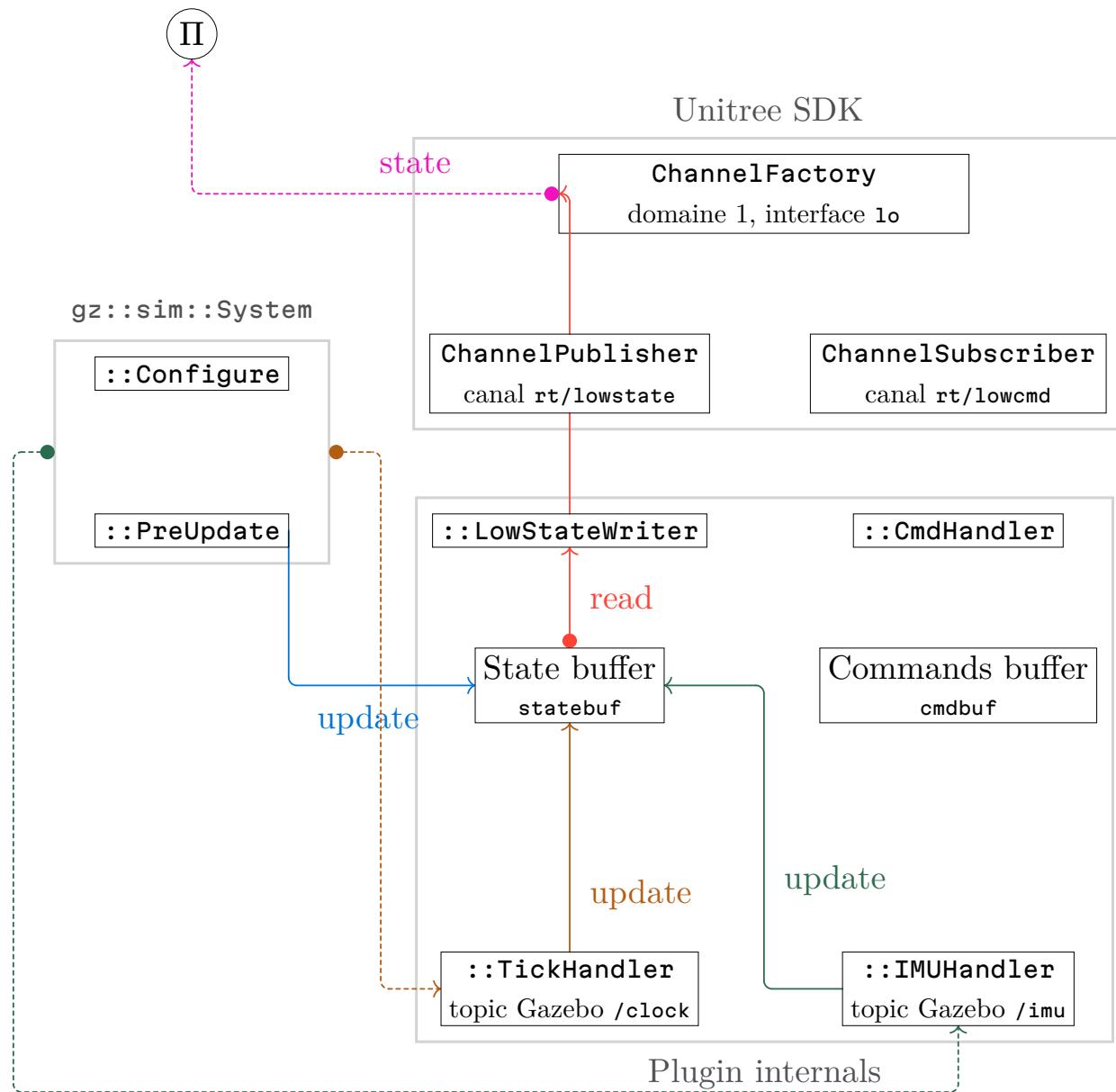


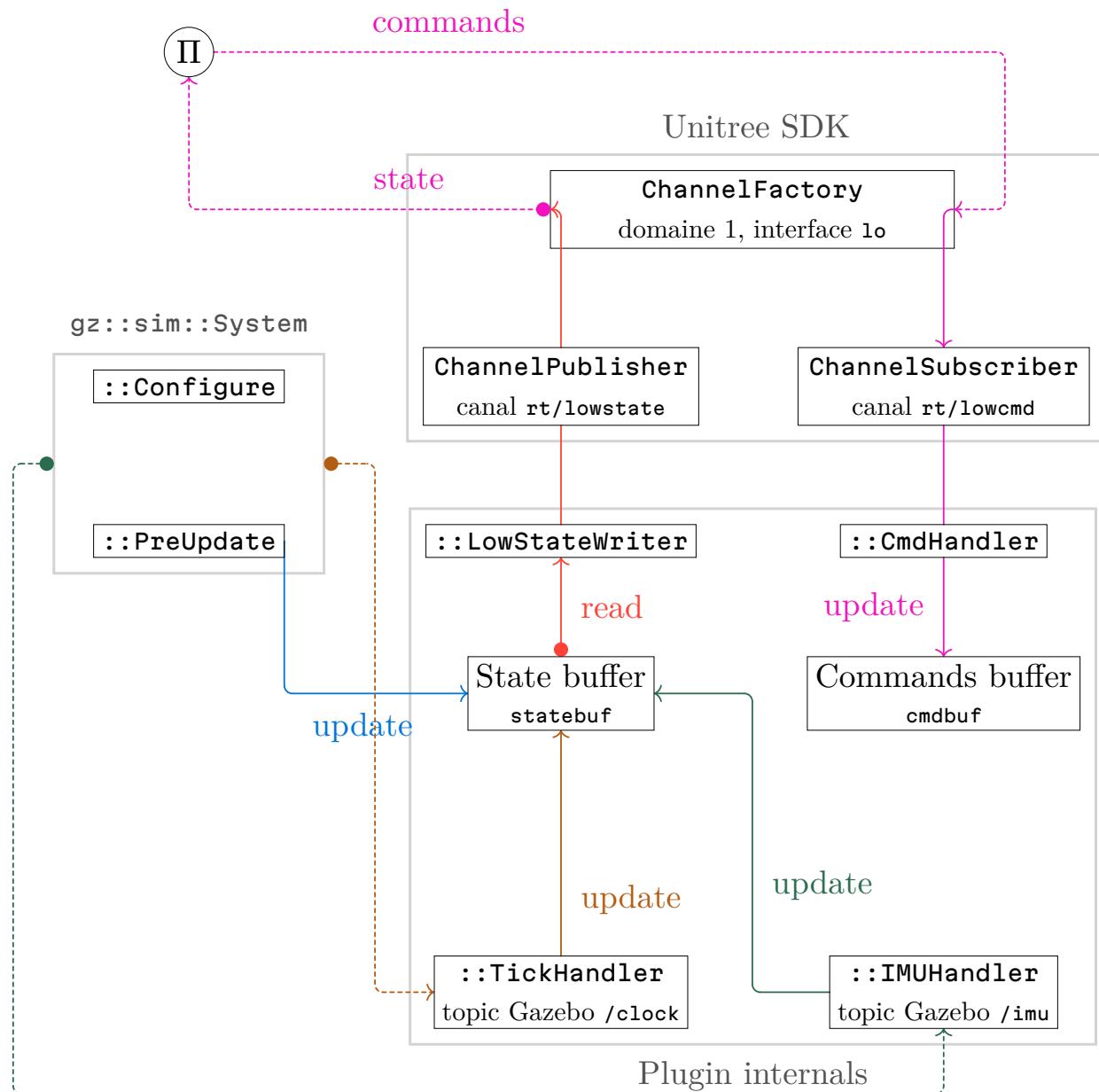
Unitree SDK

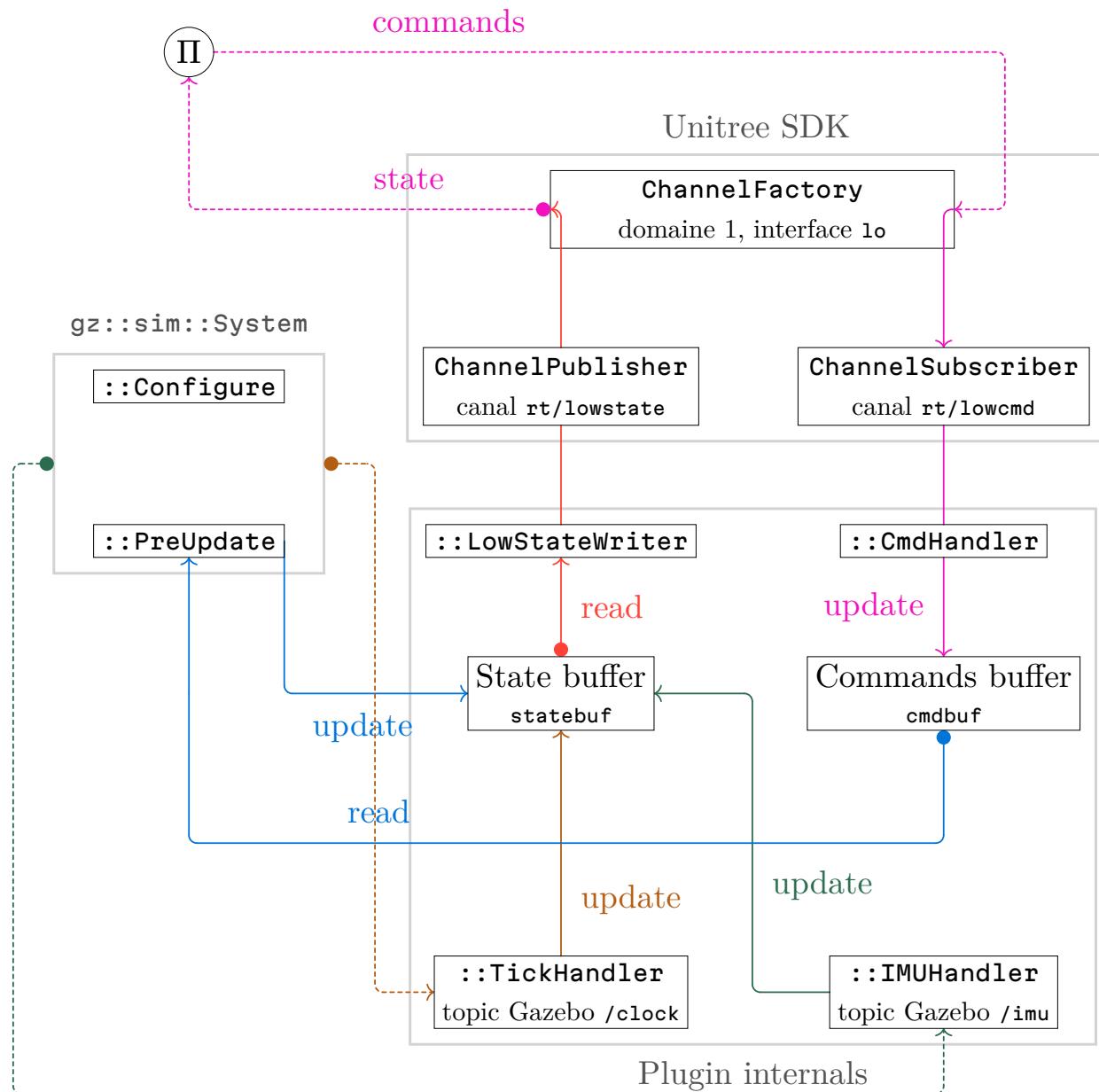


Unitree SDK









Reproductibilité

Avec Nix