

authors :

- S. LAVAZAIS
- J. SPICHT
- A. QUERE

sources:

- <https://excalidraw.com>
- <https://nx.dev>

# PRÉSENTATION NX

Par



le plan de la présentation

1. Qu'est-ce qu'un mono-repository ?
2. Cas appliqué : un projet micro service
  - présentation du cas
  - le cycle de dev
    - en multi-repo
    - en mono-repo
  - l'organisation des releases
    - en multi-repo
    - en mono-repo
3. Et NX dans tout ça ?
  - décrire le principe du fonctionnement de NX
  - parler de ce qui a été réalisé sur le projet DavLab
  - Demo !
4. Questions
5. Remerciements

# SOMMAIRE

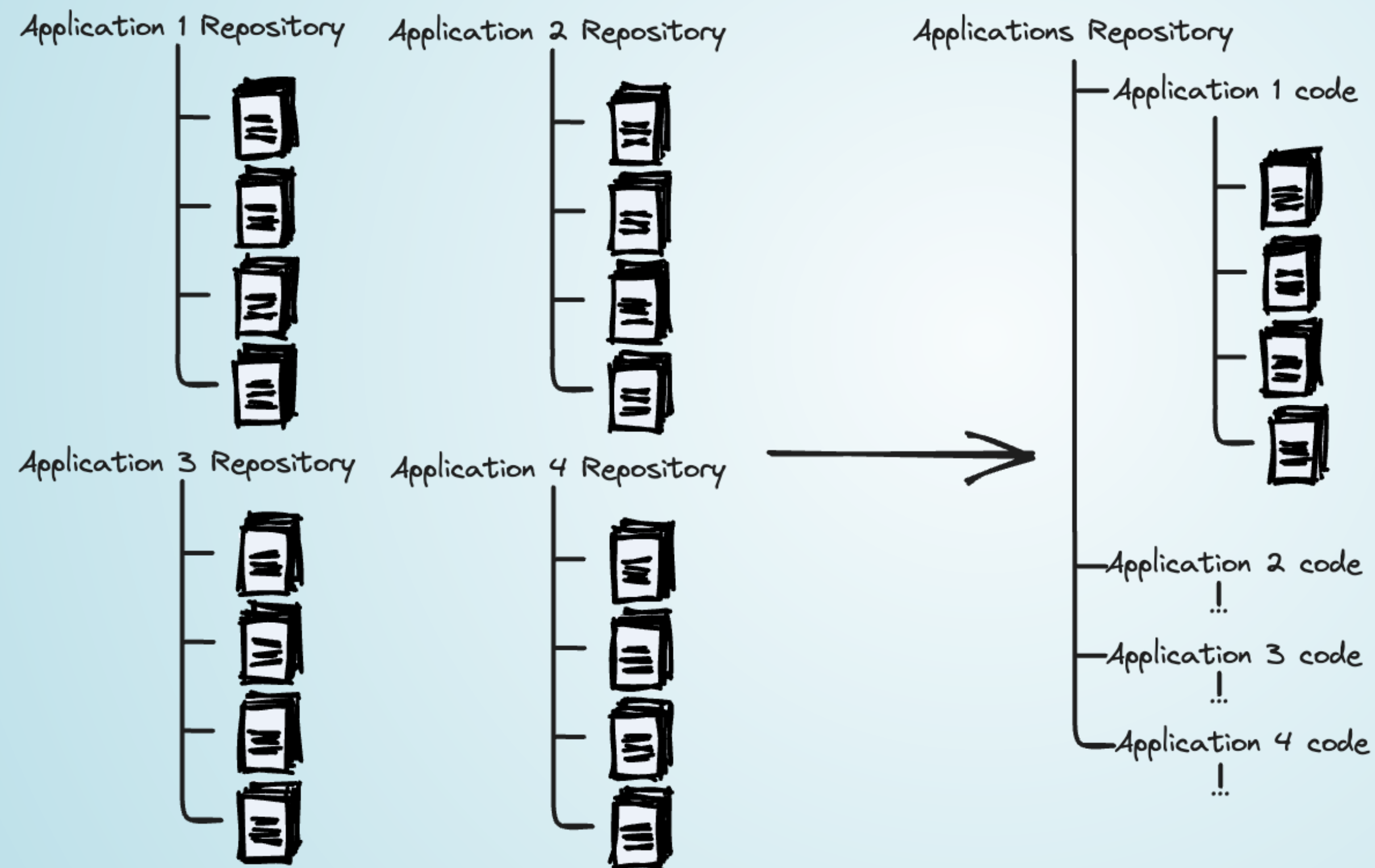
1. Qu'est-ce qu'un mono-repository ?
2. Cas appliqué : un projet micro service
3. Et NX dans tout ça ?

Lorsque l'on gère le code d'un projet type "micro-service", on range souvent le code dans plusieurs dépôts de code. (figure de gauche)

Le principe de base d'un mono-repository est de mettre le contenu de tous ces dépôts dans un seul dépôt de code afin de garder une seule et même arborescence. (figure de droite)

# QU'EST-CE QU'UN "MONO-REPOSITORY" ?

Le principe de base

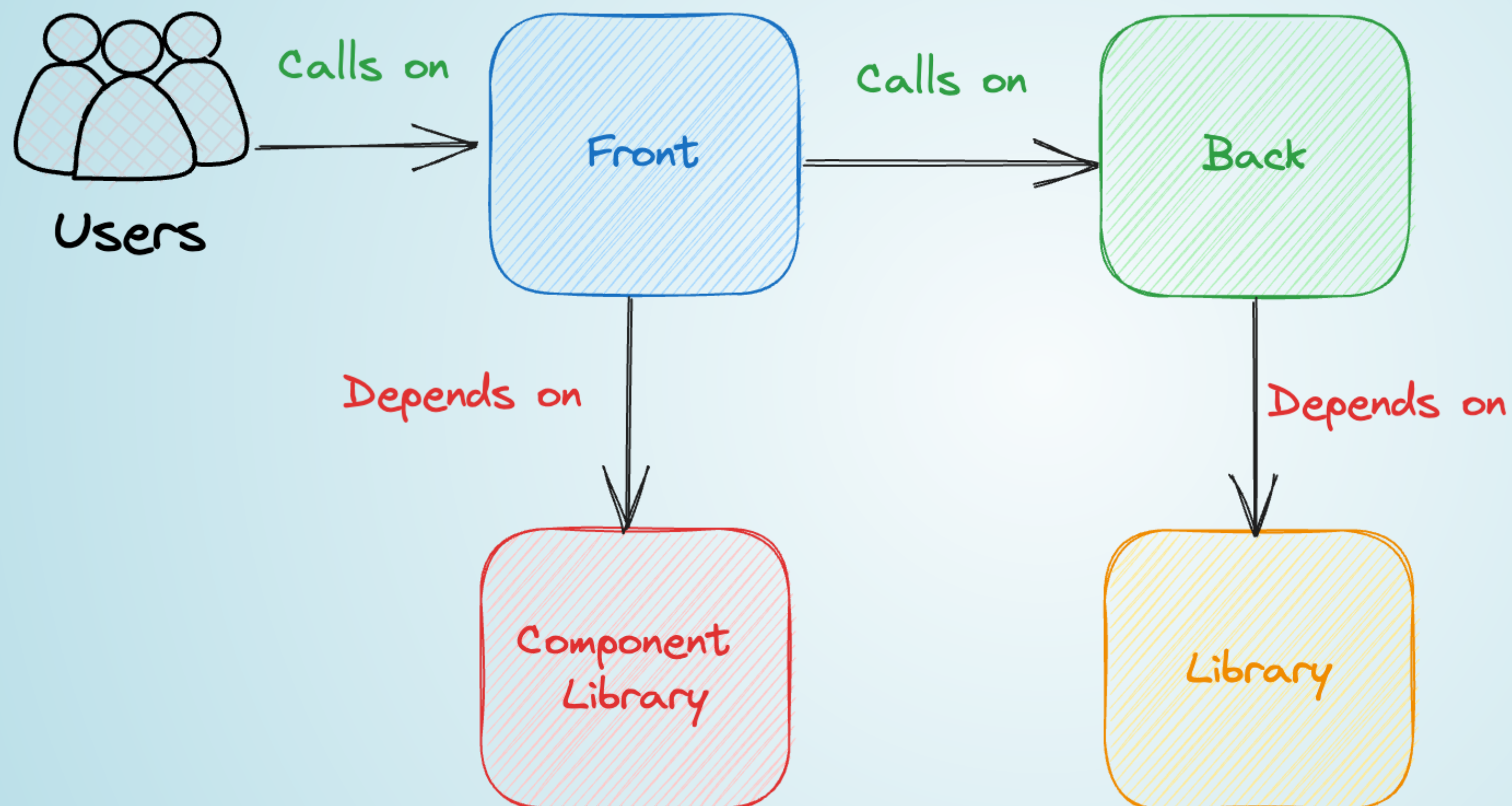




Exemple en tête, un projet de micro-service typique, avec un back, un front, une librairie pour le back et une librairie de composant pour le front.

Maintenant, on va voir la différence concrète entre une organisation de code "classique" et une organisation en mode "mono-repository"

## CAS APPLIQUÉ : UN PROJET MICRO SERVICE

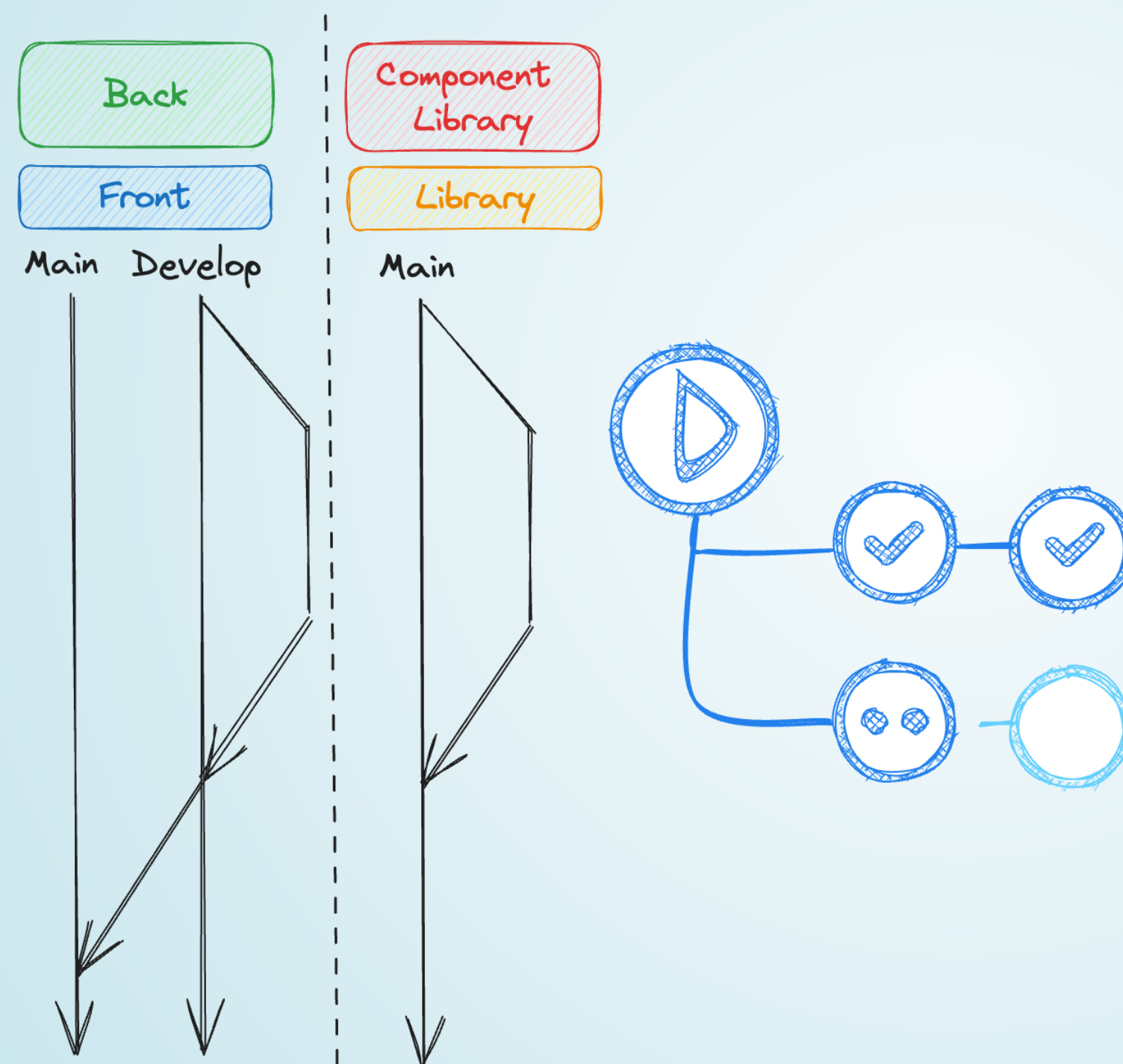


Comme on a pu le voir précédemment, le code est donc découpé en quatre dépôts (un pour chaque partie de ce micro-service).

Et Chacun de ces composants va avoir son propre cycle de développement

# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

## Le cycle de développement - multi





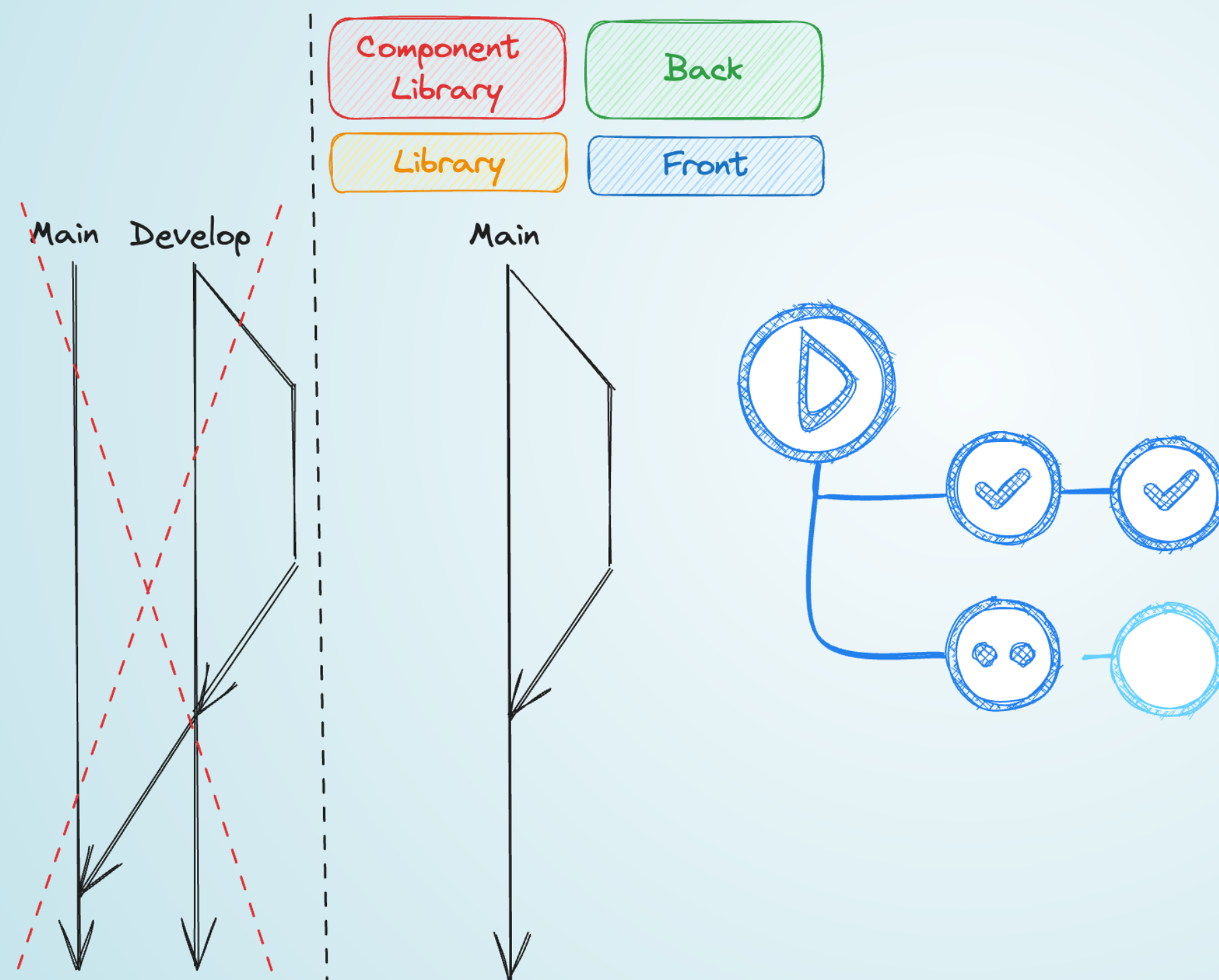
Dans le cas d'un mono repository, on ne peut pas avoir deux cycles de développements différent.

Il faut donc en choisir un et faire en sorte que chaque composant s'inscrive dans ce nouveau cycle de développement.

Par soucis de simplicité, on peut choisir le trunk base, mais sur le papier, le git flow fonctionne également.

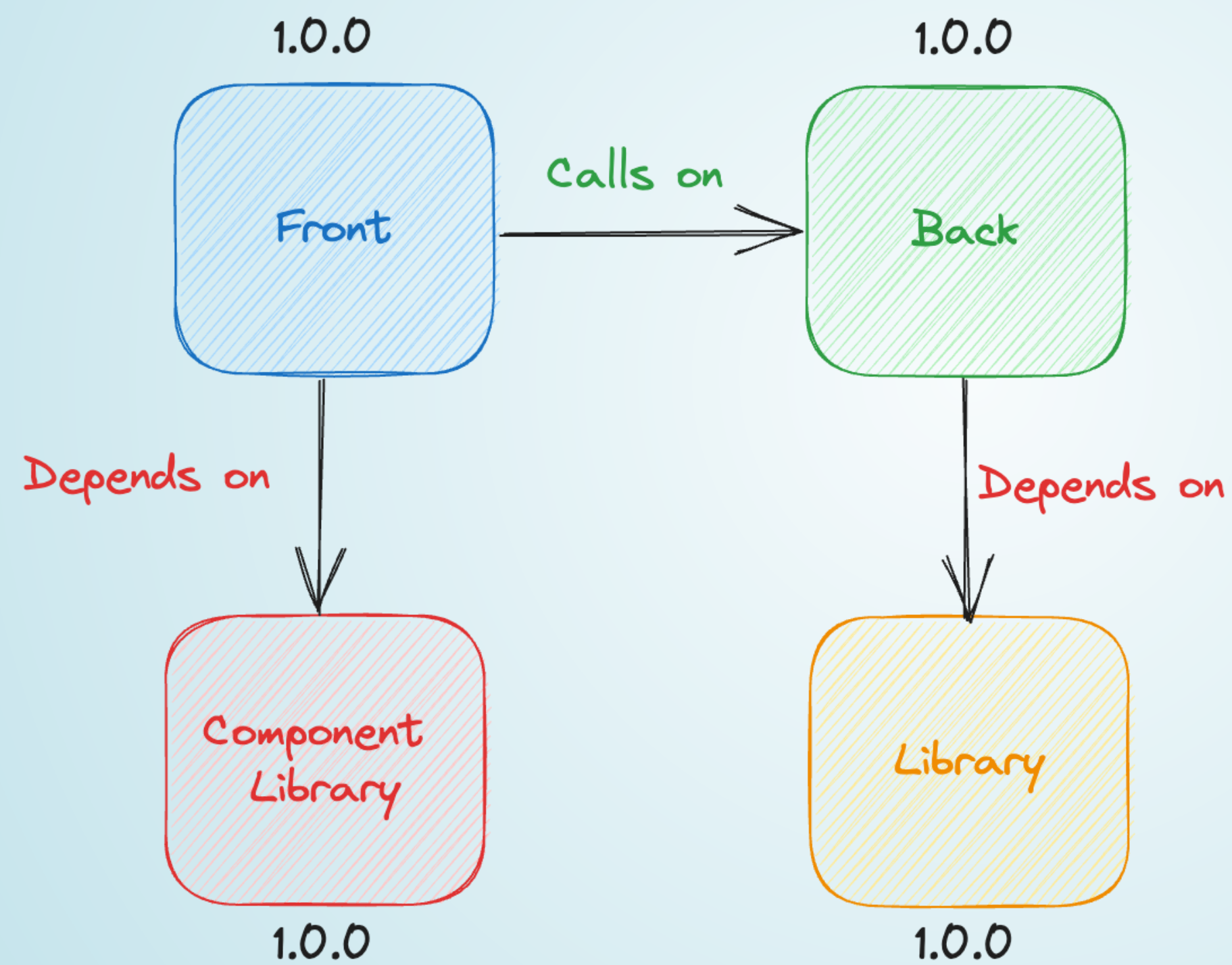
# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

## Le cycle de développement - mono



# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

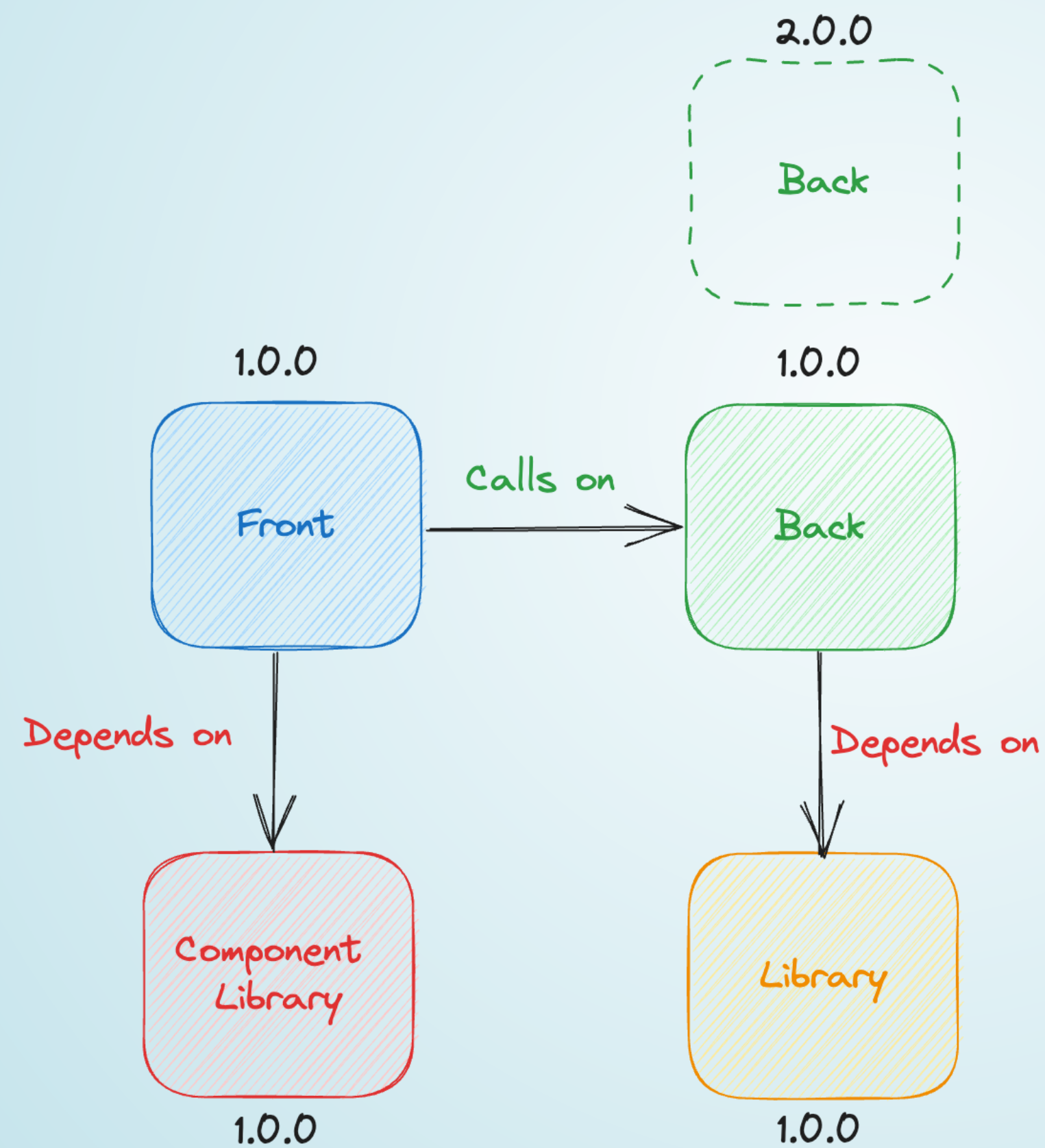
L'organisation des releases - multi





# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

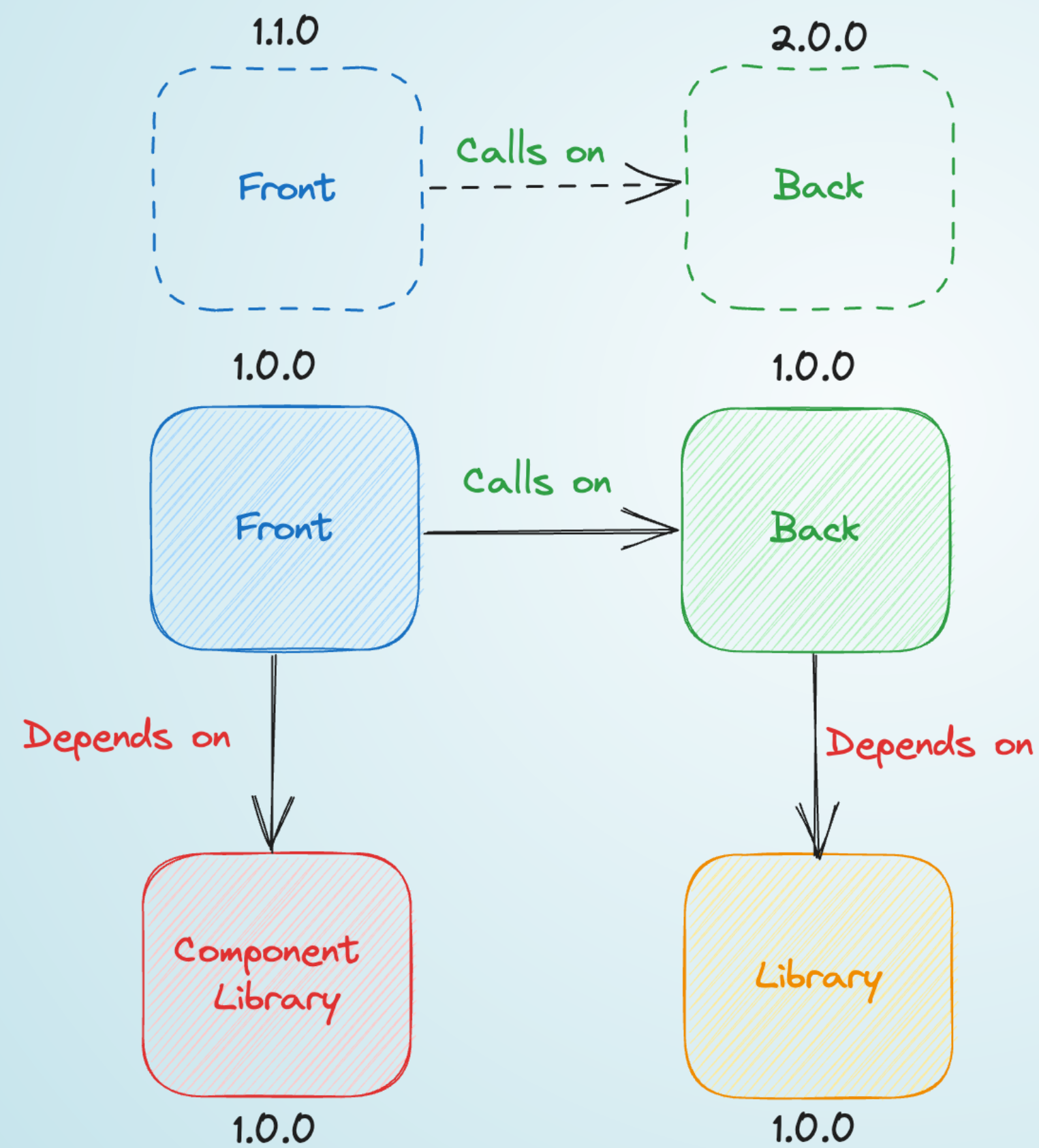
## L'organisation des releases - multi





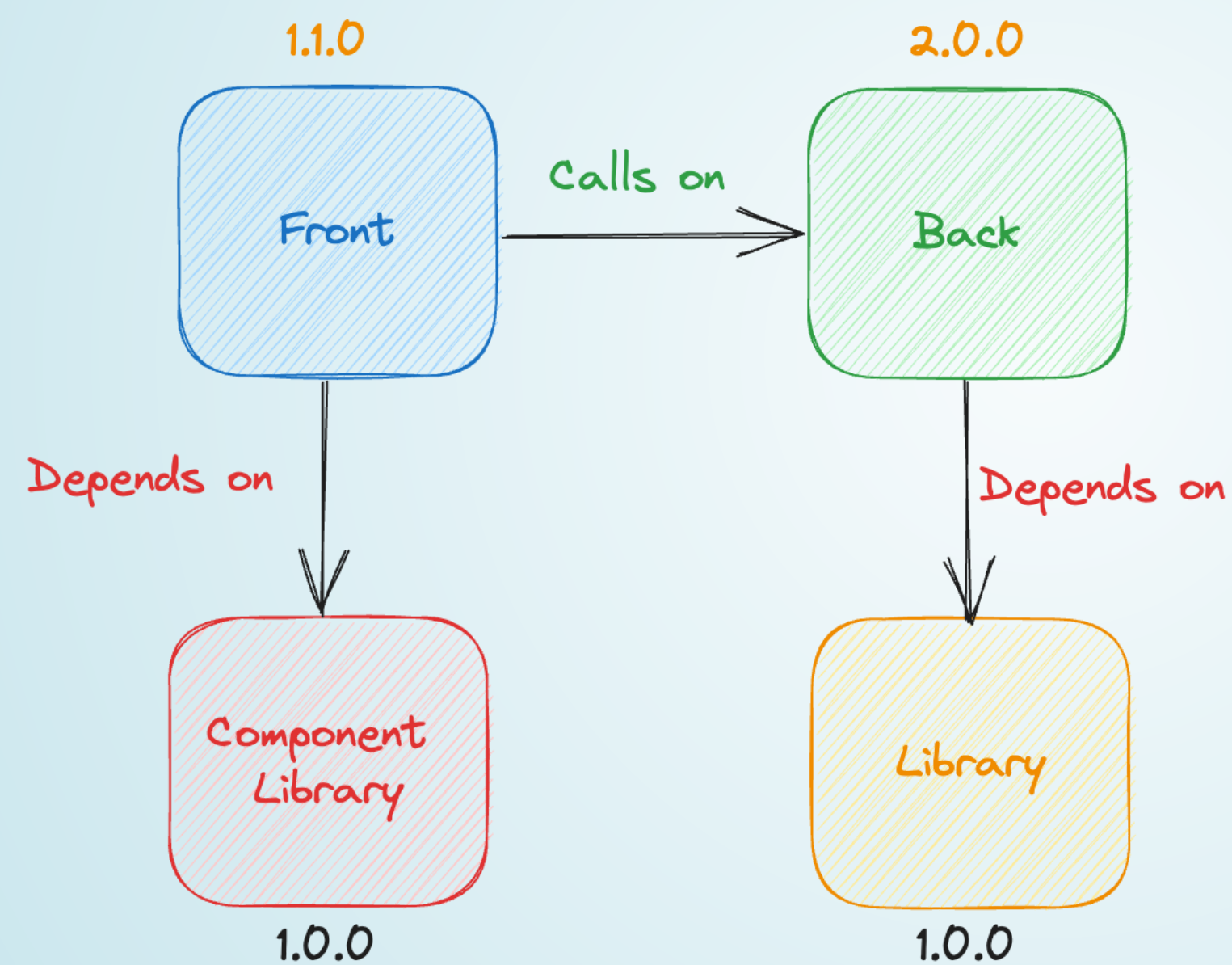
# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

## L'organisation des releases - multi



# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

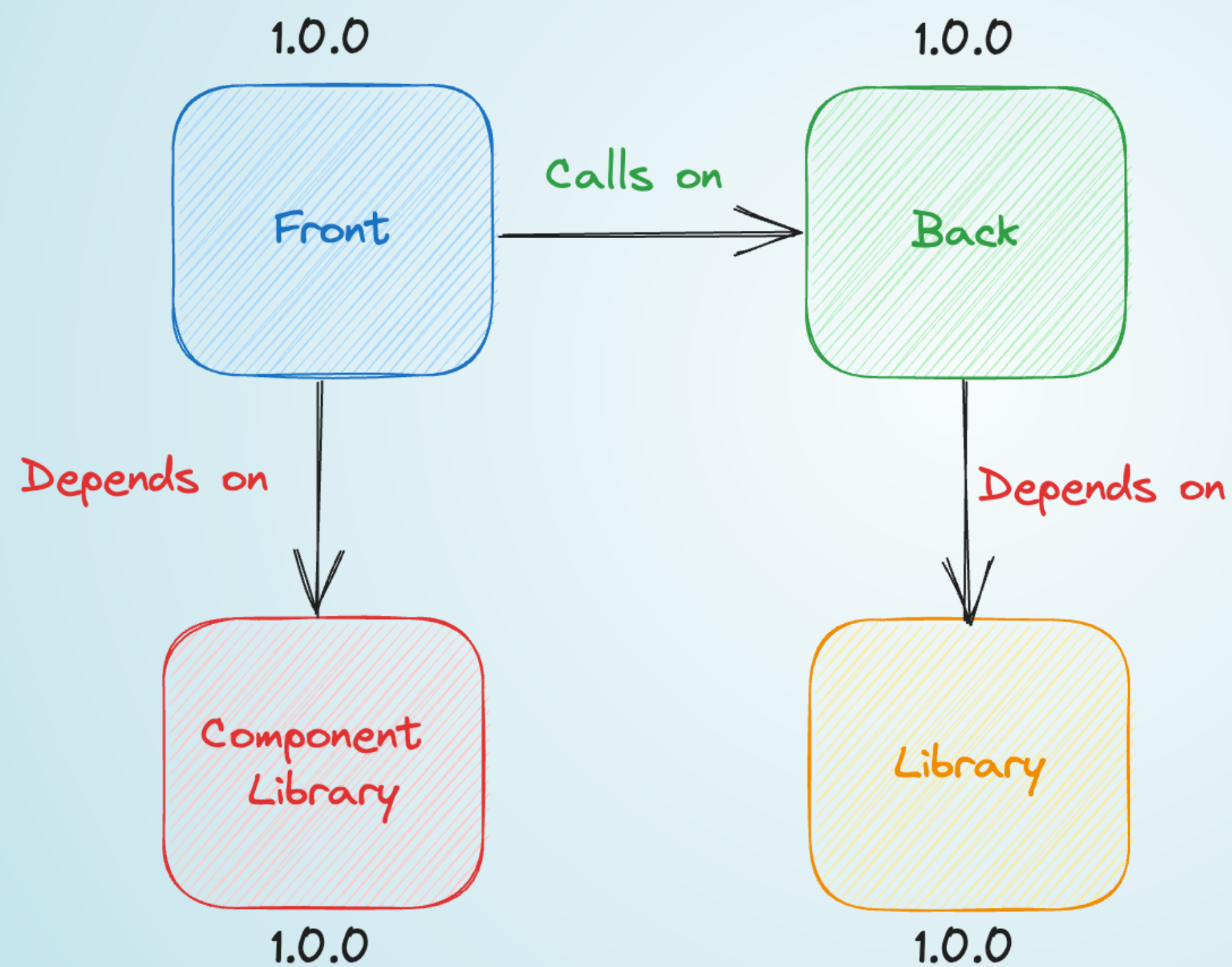
L'organisation des releases - multi





# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

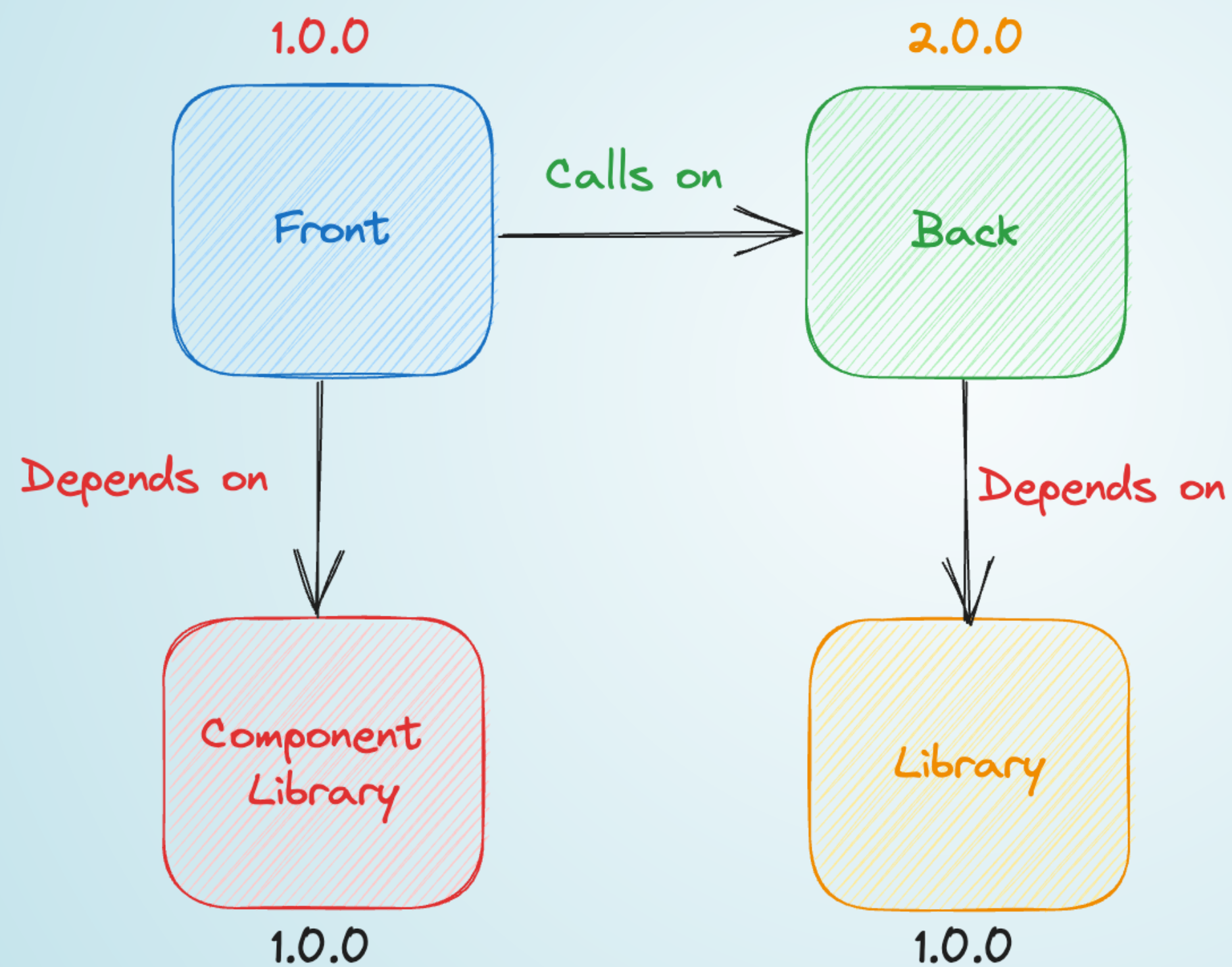
L'organisation des releases - mono





# CAS APPLIQUÉ : UN PROJET MICRO SERVICE

L'organisation des releases - mono



NX est un framework conçu par "NRWL" (narval) comportant des outils et des techniques simples permettant d'accélérer grandement la productivité et la rendre plus simple du point de vue de l'expérience développeur.

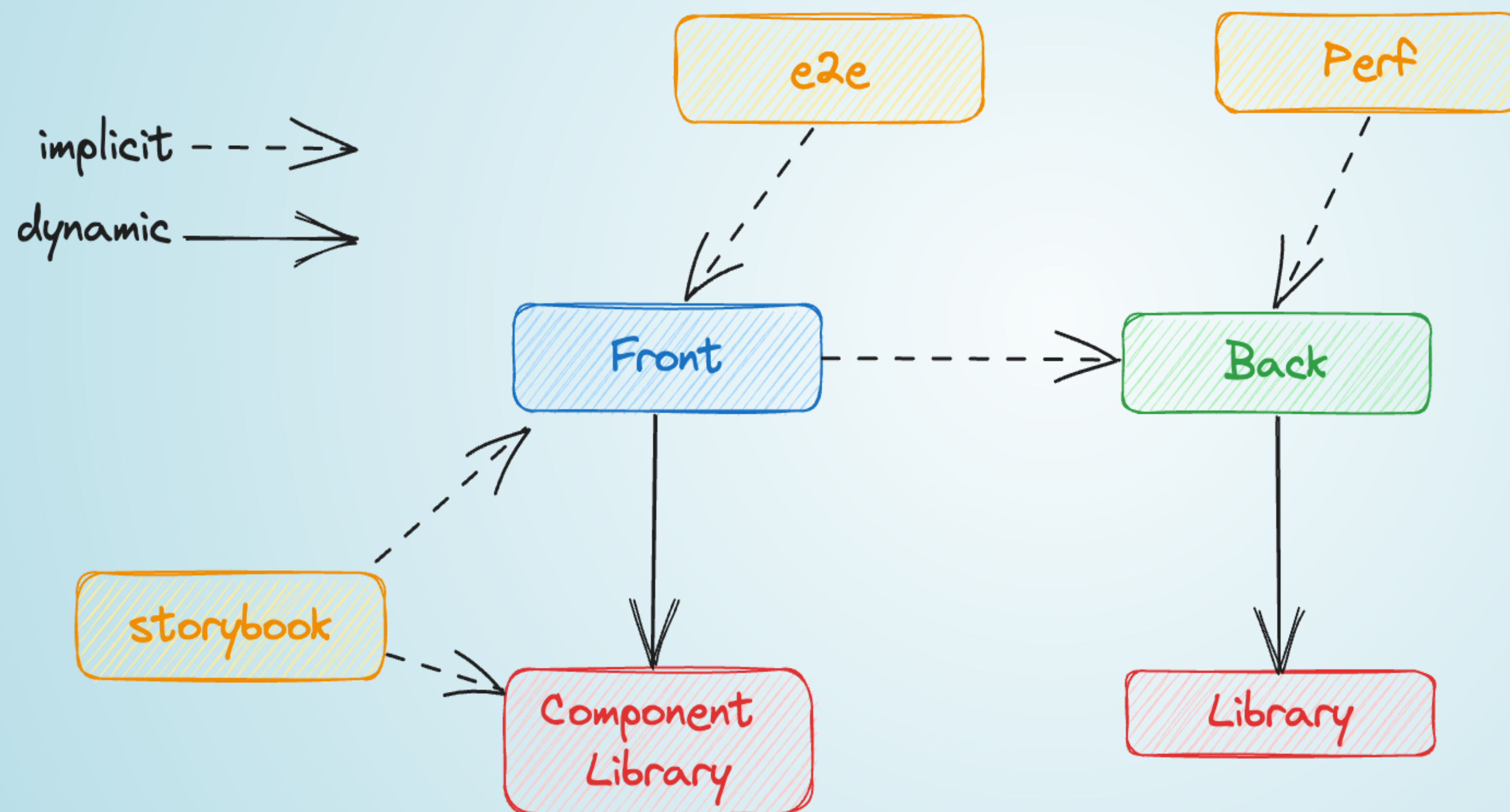
Dans notre cas de projet micro-service NX est l'outil idéal pour mettre en pratique un mono-repository presque parfait

ET NX DANS TOUT ÇA ?



# ET NX DANS TOUT ÇA ?

Le graph de dépendance



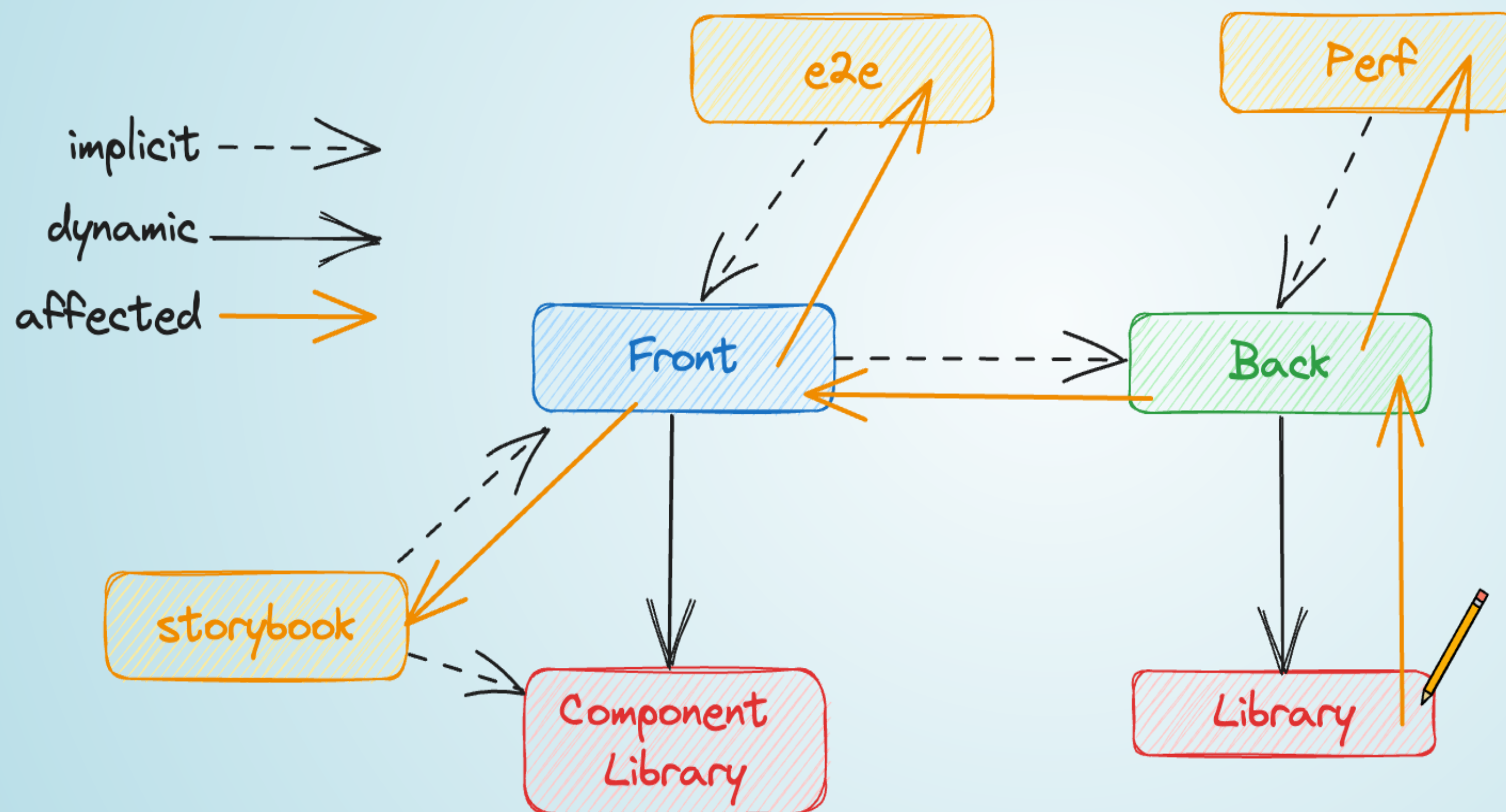


NX est capable de déduire les projets impactés par des modifications. À partir des dépendances entre projets, il est capable de reconstituer toute la chaîne de projets impactés.

Grâce à ce mécanisme, NX est capable de garantir la cohérence de l'ensemble du mono-repo

## ET NX DANS TOUT ÇA ?

Le graph de dépendance - les "affected"



# ET NX DANS TOUT ÇA ?

Les langages



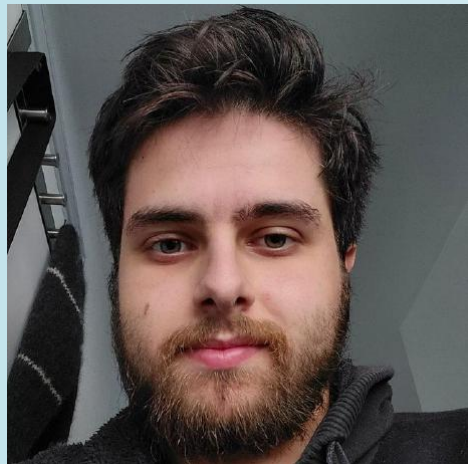
ET NX DANS TOUT ÇA ?

La Démo



# MERCI POUR VOTRE TEMPS!

Présenter par



Anthony QUÉRÉ   Jules SPICHT   Sylvain LAVAZAIS

QRCode

Sources:

- <https://excalidraw.com>
- <https://nx.dev>