Compartimentation dynamique imbriquée pour objets contraints

Nicolas Dejon, *Orange Labs, Université de Lille*
Chrystel Gaber, *Orange Labs*
Gilles Grimaud, *Université de Lille*

Scénario cible

La Memory Protection Unit (MPU)

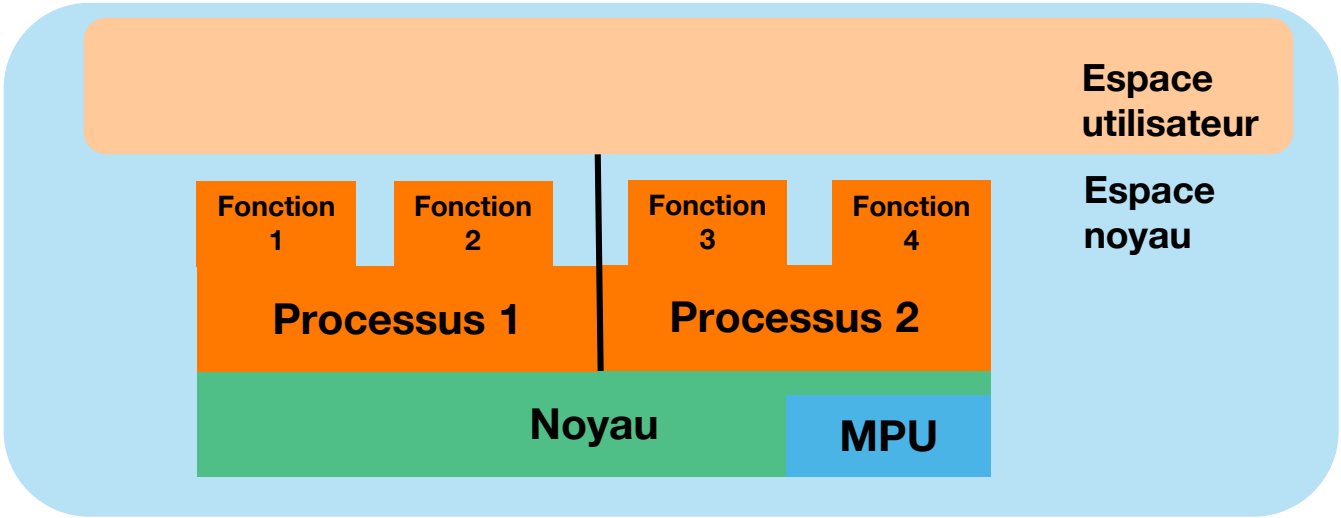
Etat de l'art des systèmes basés sur MPU

Framework proposé et API

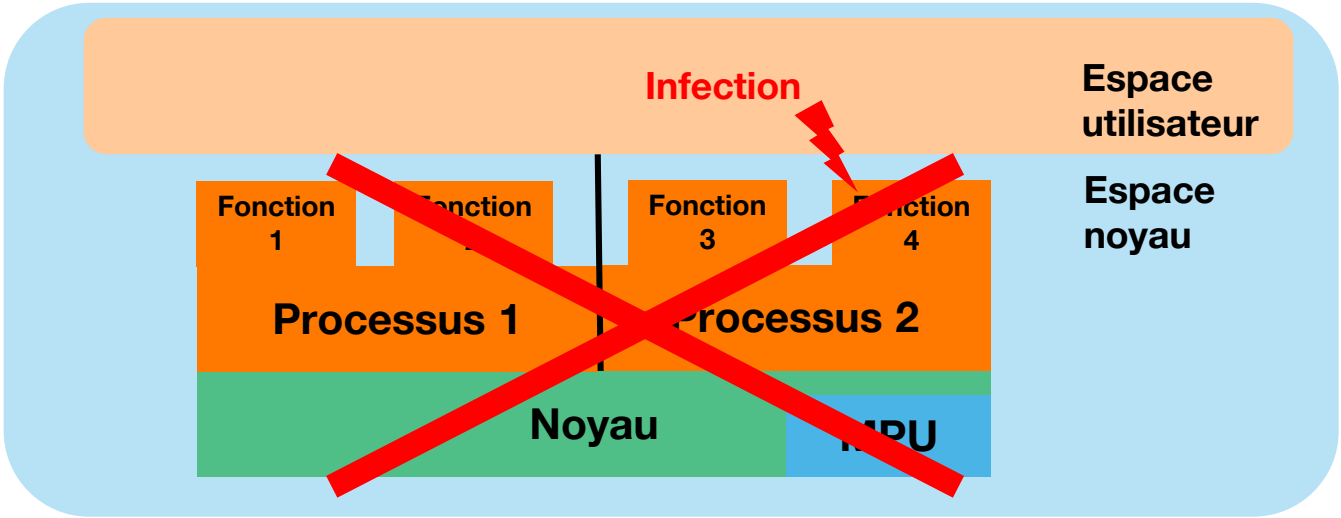
Application à 3 systèmes d'exploitation

08/07/21
COMPAS (Conférence francophone d'informatique en Parallélisme, Architecture et Système) 2021

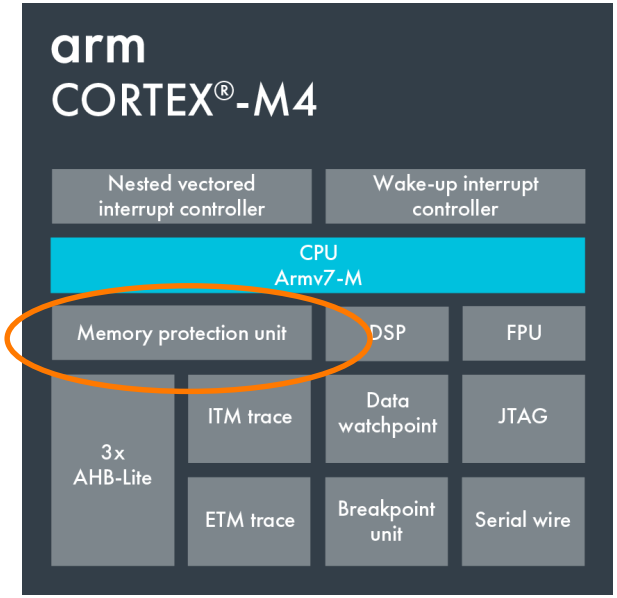
Problématique : isolation mémoire dans les objets contraints



Problématique : isolation mémoire dans les objets contraints



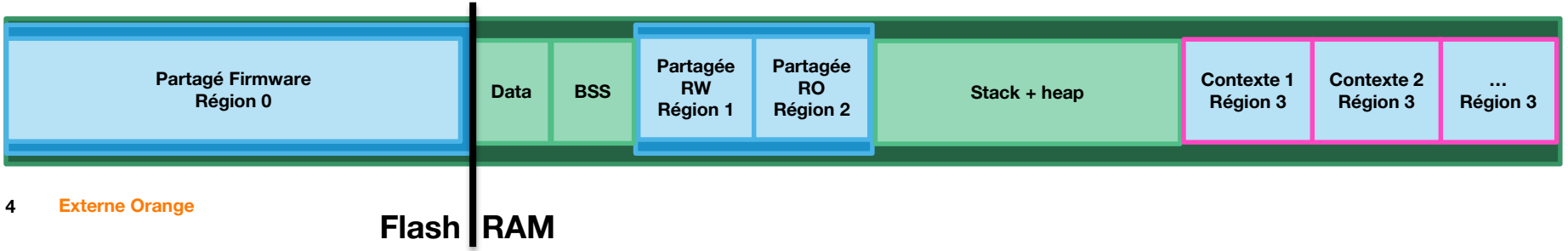
La *Memory Protection Unit* ou MPU



Unité facultative dans les ARM Cortex-M

- Protection matérielle (permissions RWX)
- 8/16 régions mémoire protégées simultanément
- 1 région MPU = (@début, @fin, RWX)
- Peut être reconfigurée

https://developer.arm.com/ip-products/processors/cortex-m/cortex-m4



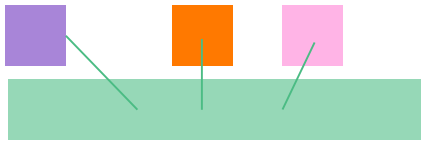
Etat de l'art et objectif

TABLE 1 – Comparaison de la compartimentation dans des systèmes basés MPU.

SE/noyau/ hyperviseur/outil	*Dynamisme*			*Flexibilité*	
	Reconfiguration MPU	Modèle de permission	Mémoire extensible	Nature de la compartimentation	Compartimentation imbriquée
TrustLite	✗	immuable	✗	processus	✗
Choupi-OS	✓	immuable	✗	processus	✗
EwoK	✓	immuable	✗	processus	✗
µVisor	✓	immuable	✗	*thread*	✗
ACES	✓	immuable	✗	générique	✗
TockOS	✓	immuable	(✓)	processus	✗
Zephyr (MPU)	✓	(variable)	(✓)	*thread*	✗
FreeRTOS-MPU	✓	(variable)	(✓)	tâche	✗

+ de fléxibilité

Permissions immuables
+ compartimentation générique
+ extension mémoire limitée



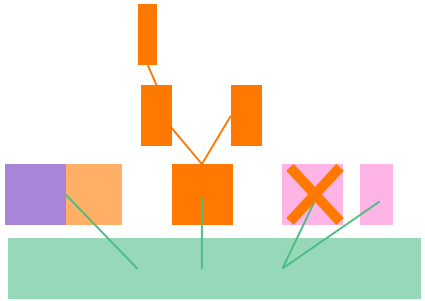
Etat de l'art et objectif

TABLE 1 – Comparaison de la compartimentation dans des systèmes basés MPU.

SE/noyau/ hyperviseur/outil	*Dynamisme*			*Flexibilité*	
	Reconfiguration MPU	Modèle de permission	Mémoire extensible	Nature de la compartimentation	Compartimentation imbriquée
TrustLite	✗	immuable	✗	processus	✗
Choupi-OS	✓	immuable	✗	processus	✗
EwoK	✓	immuable	✗	processus	✗
µVisor	✓	immuable	✗	*thread*	✗
ACES	✓	immuable	✗	générique	✗
TockOS	✓	immuable	(✓)	processus	✗
Zephyr (MPU)	✓	(variable)	(✓)	*thread*	✗
FreeRTOS-MPU	✓	(variable)	(✓)	tâche	✗
framework proposé	✓	variable	✓	générique	✓

+ de fléxibilité

Permissions variables
+ compartimentation générique
+ extension mémoire ~~limitée~~
+ imbrication de sous-domaines protégés



Compartimentation imbriquée avec la MPU

Contrôle généralisé

- **Entité dédiée pour la gestion de la MPU**
- **Accessible par chaque abstraction (noyau, processus, *thread*…)**

Configuration centralisée de la MPU par appel système

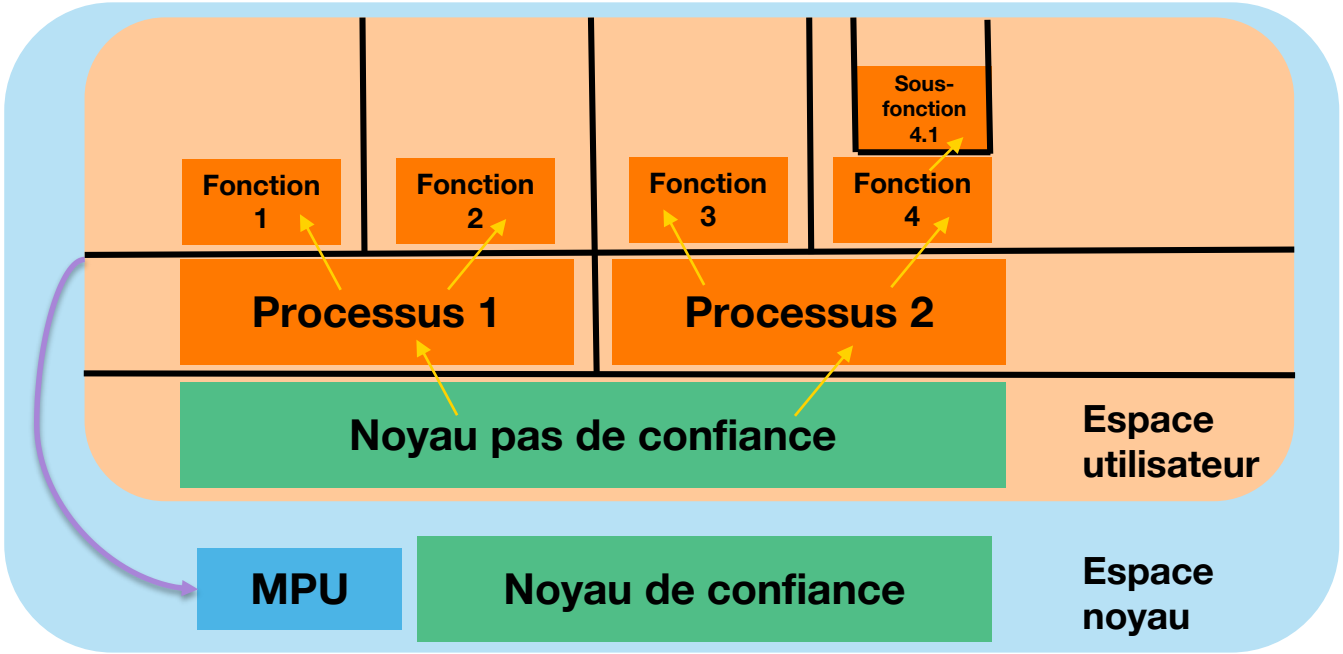
- **Entité dédiée *privilégiée***
- **Toutes les abstractions doivent être dans l'espace utilisateur**

Politique de sécurité personnalisée

- **par l'implémentation**
- **Exemples :**
  - on ne donne à un sous-domaine que de la mémoire qu'on détient
  - application du principe W^X
  - 1 seul bloc partagé entre plusieurs sous-domaines

Application du *framework* au scénario cible

Appels système pour modifier la configuration de la MPU

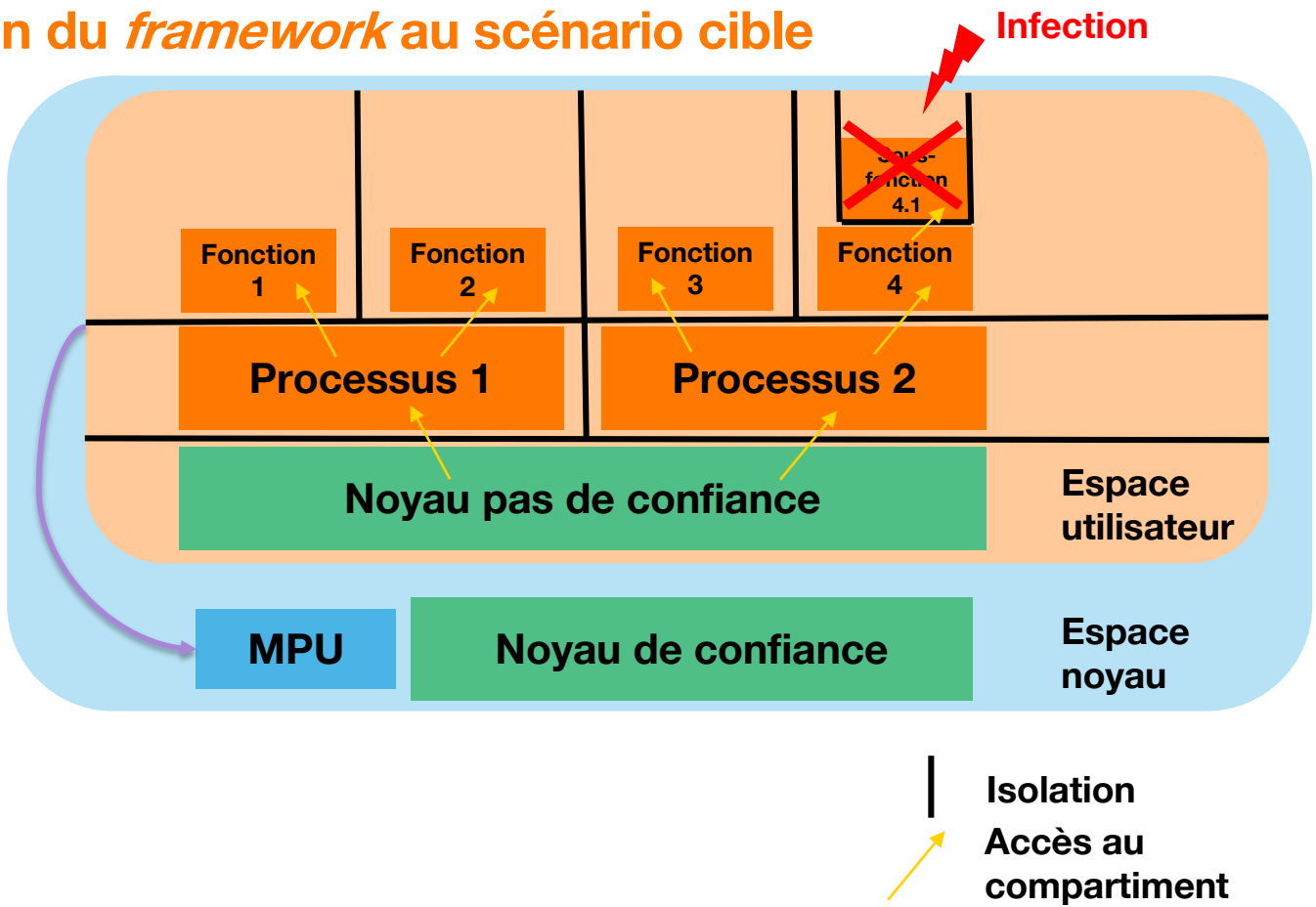


Isolation

Accès au compartiment

Application du *framework* au scénario cible

Appels système pour modifier la configuration de la MPU



IPA/API de gestion dynamique et flexible de compartiments

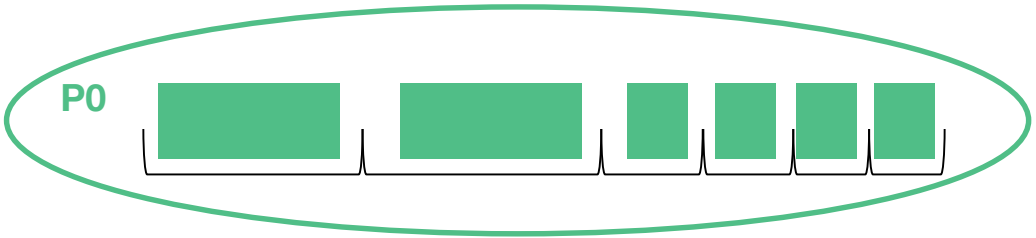
Appels système dynamique (gestion d'un sous-domaine, extension mémoire, modification des permissions d'accès) :

- **add**
- **remove**
- **prepare**
- **collect**
- **cut**
- **merge**

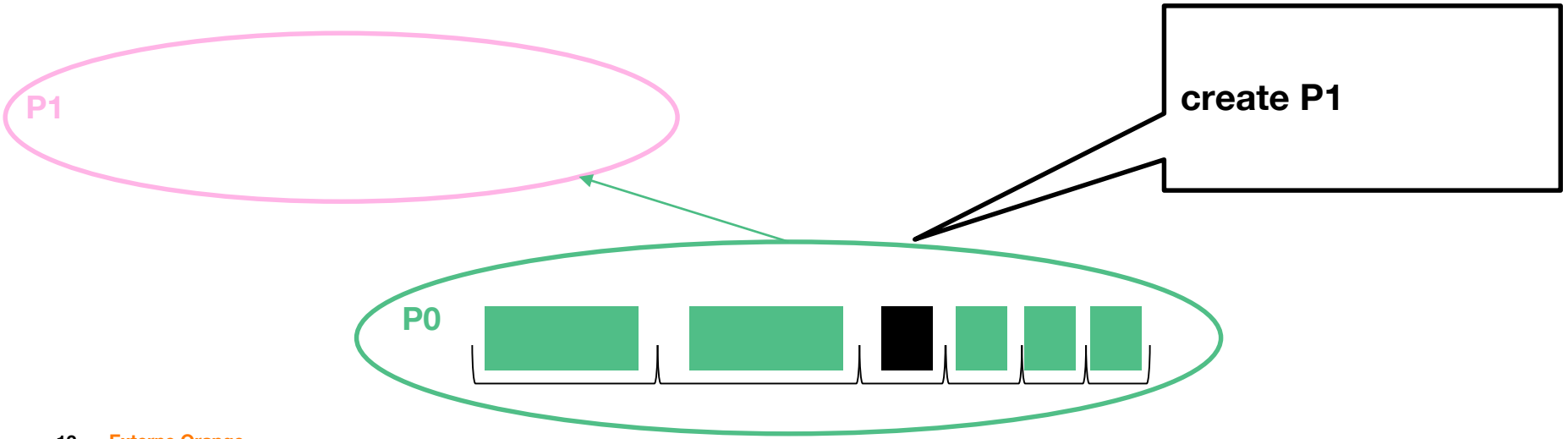
Appels système flexible (création/destruction de sous-domaines protégés) :

- **create**
- **delete**

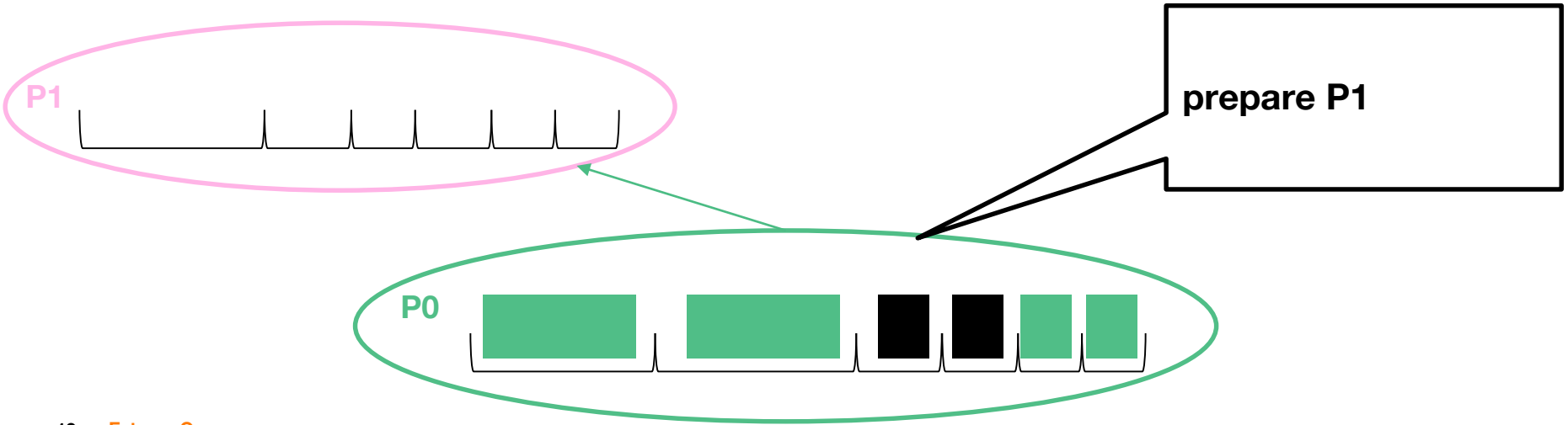
Exemple d’utilisation de l’API



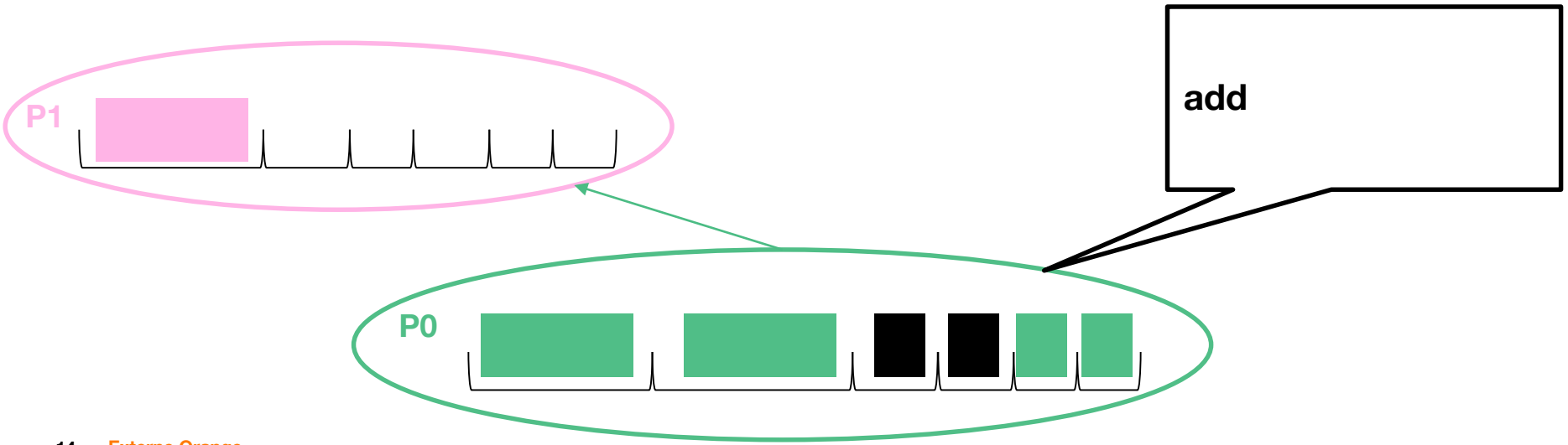
Exemple d’utilisation de l’API



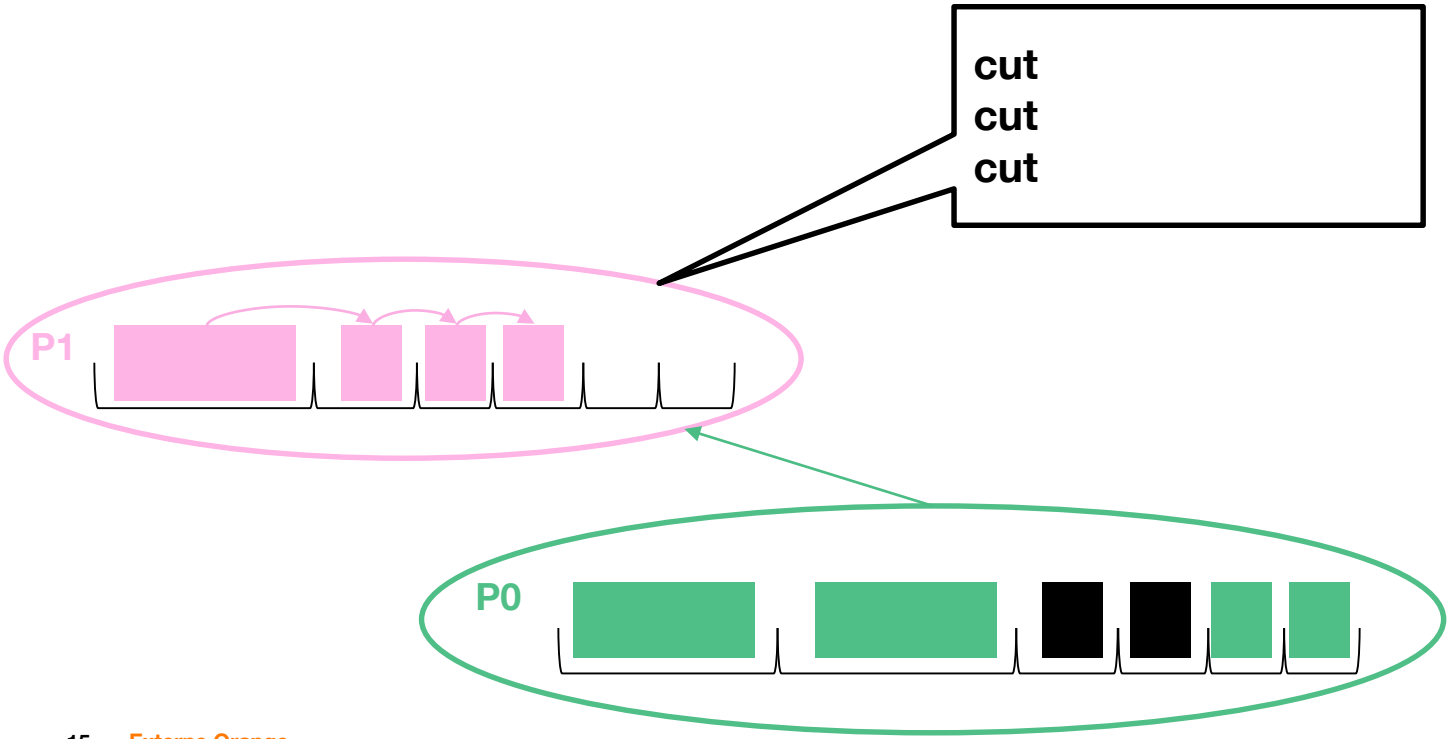
Exemple d’utilisation de l’API



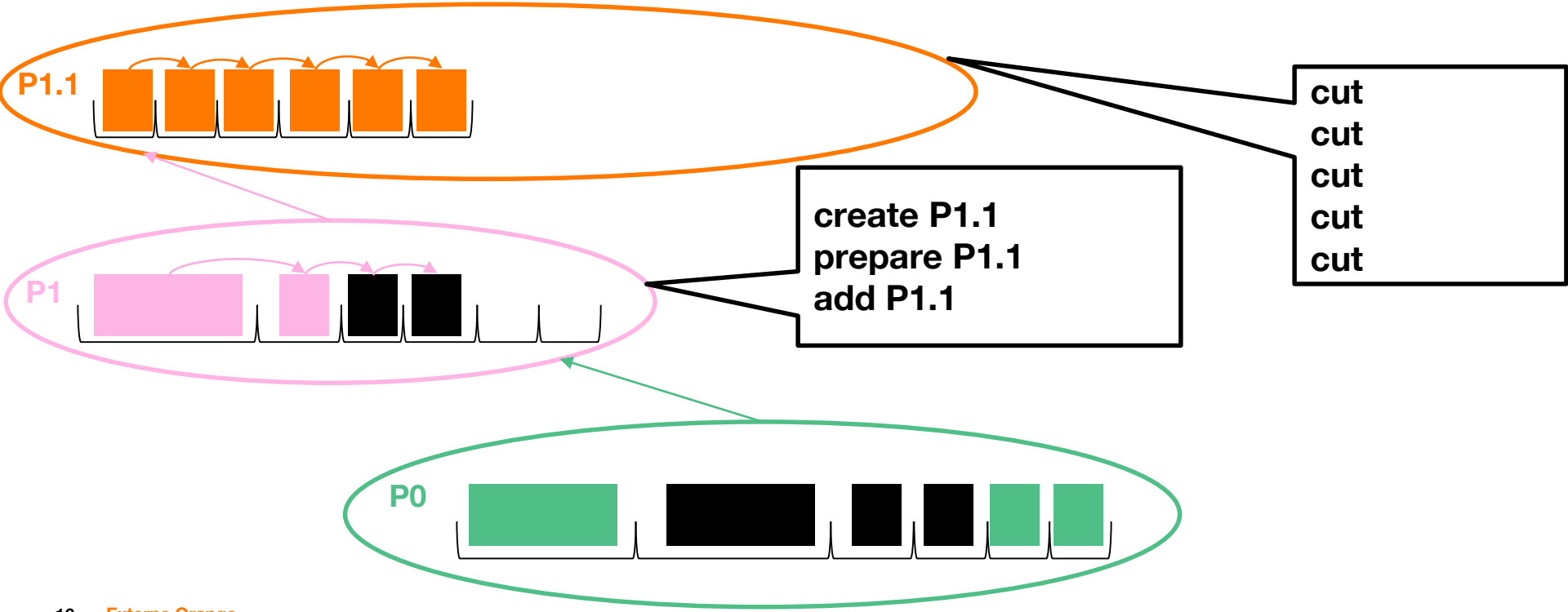
Exemple d’utilisation de l’API



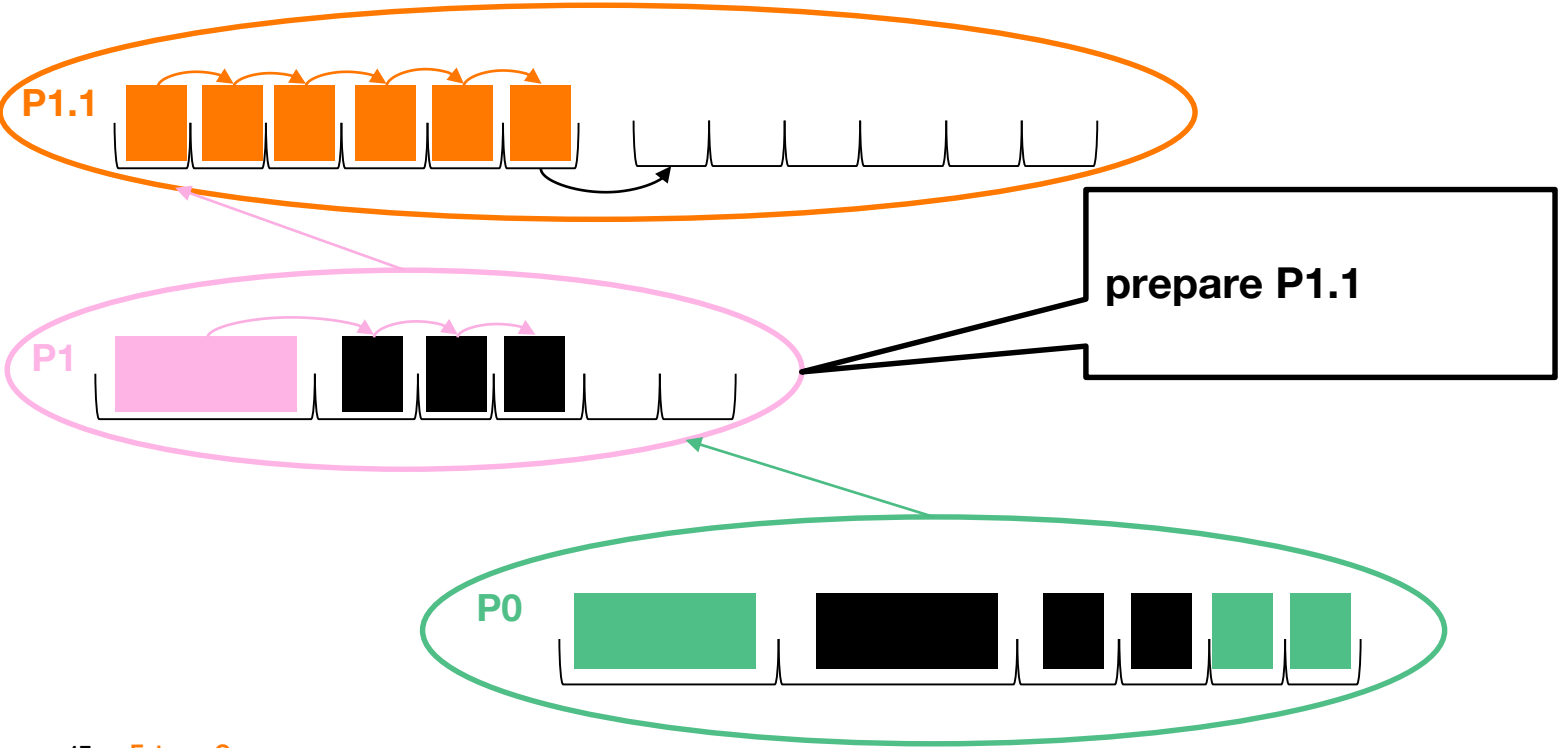
Exemple d’utilisation de l’API



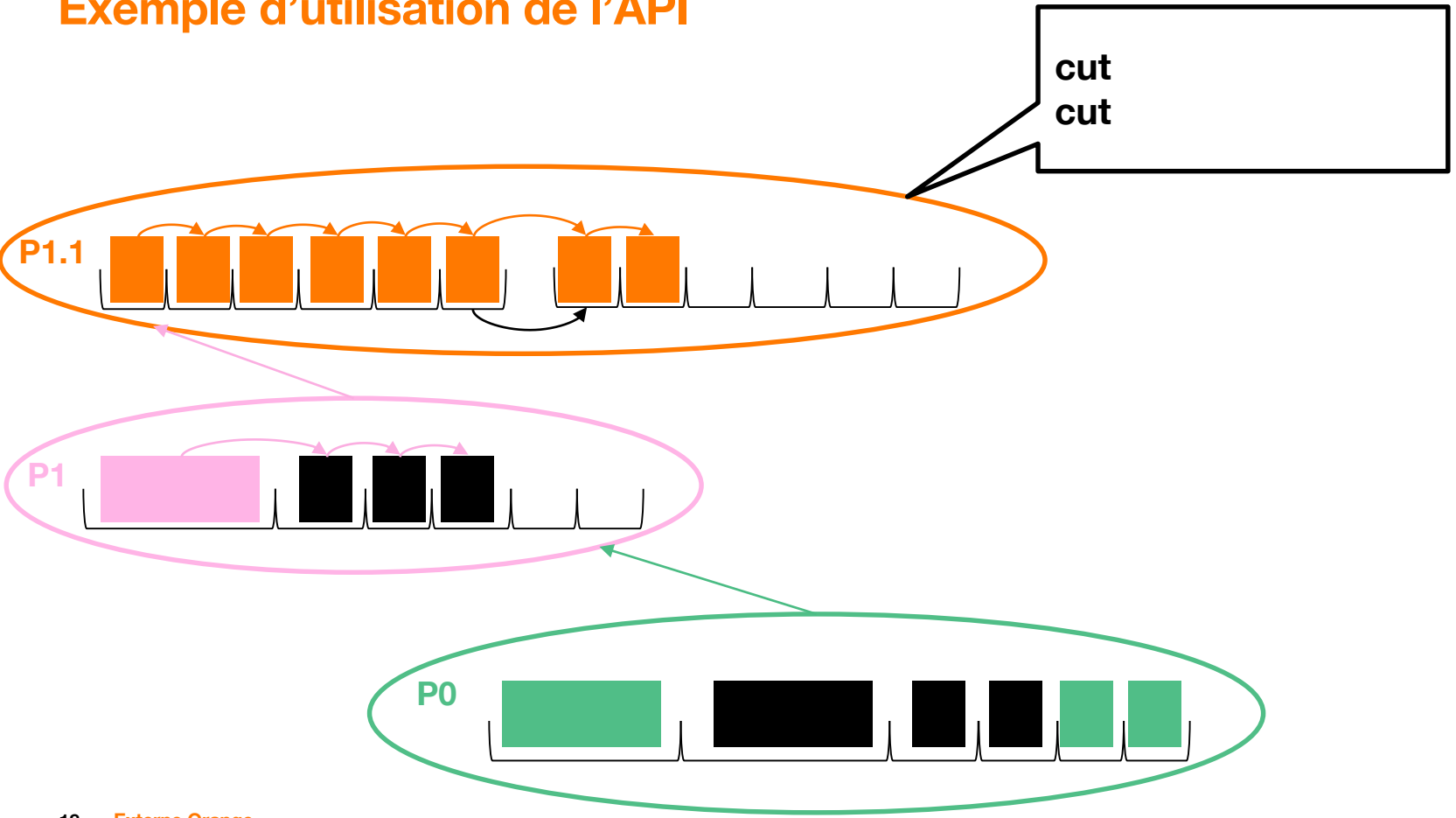
Exemple d’utilisation de l’API



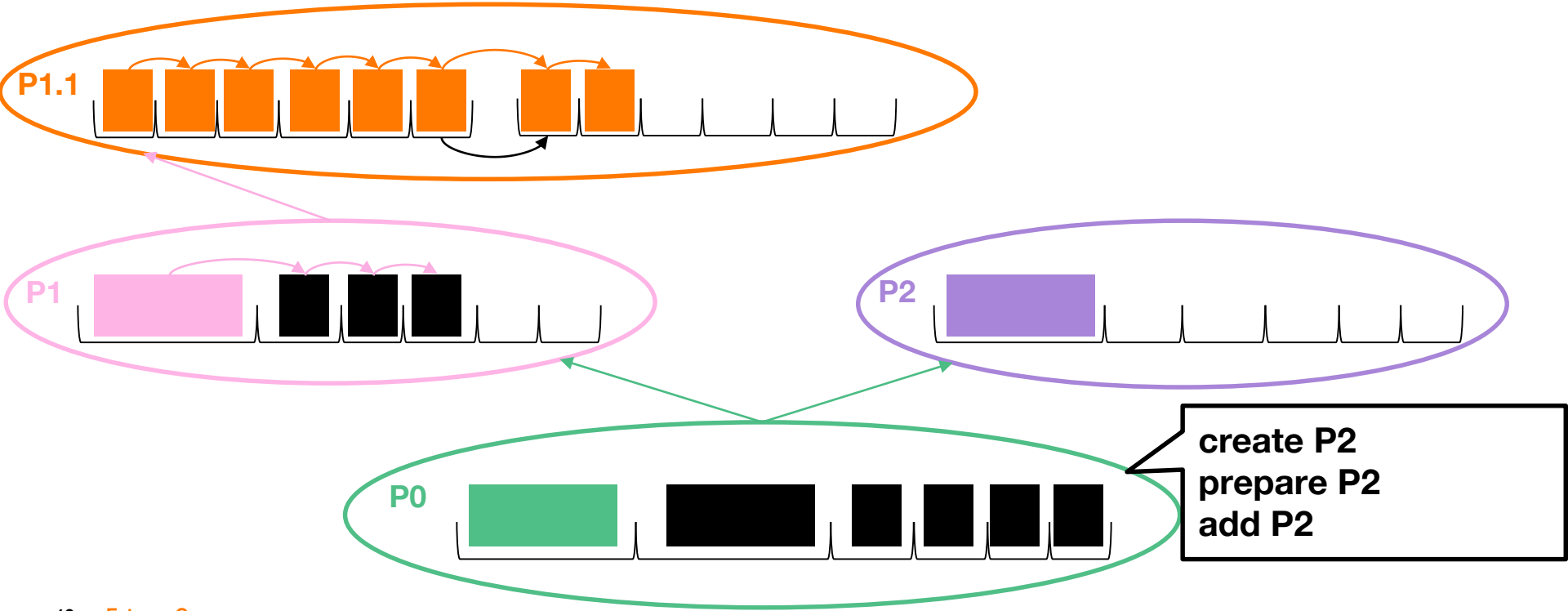
Exemple d’utilisation de l’API



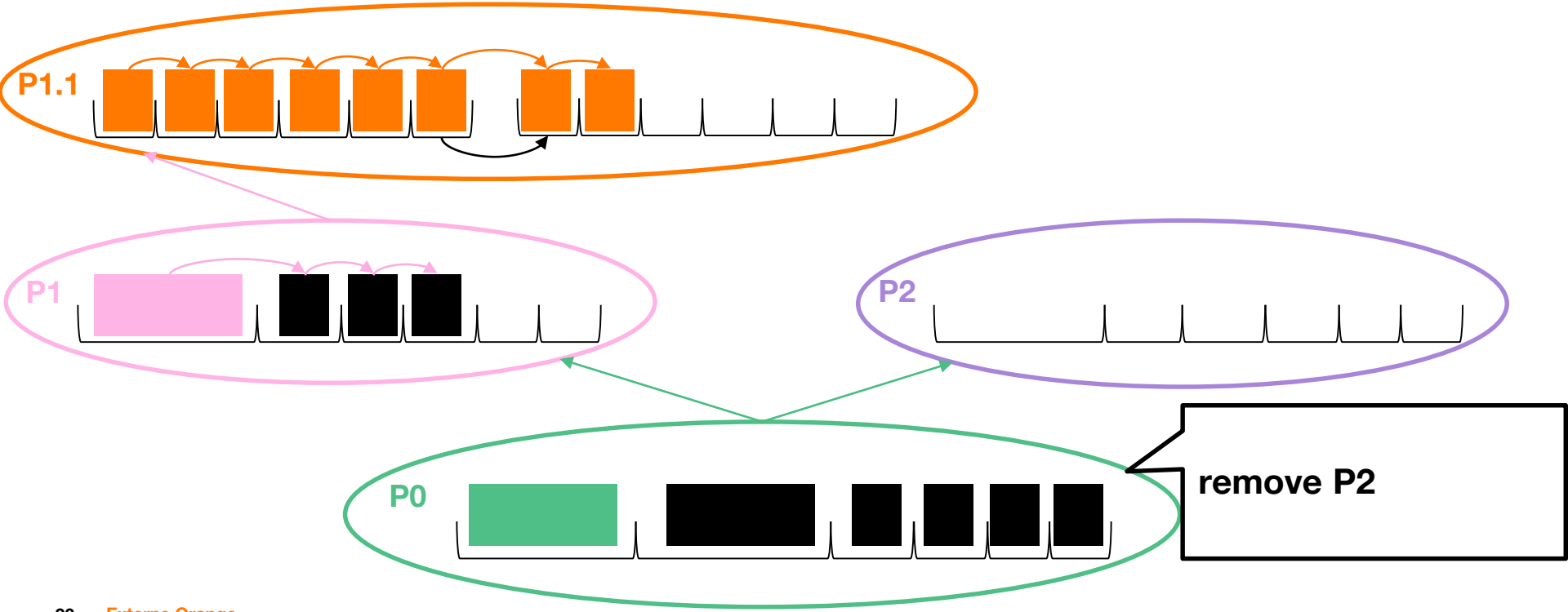
Exemple d’utilisation de l’API



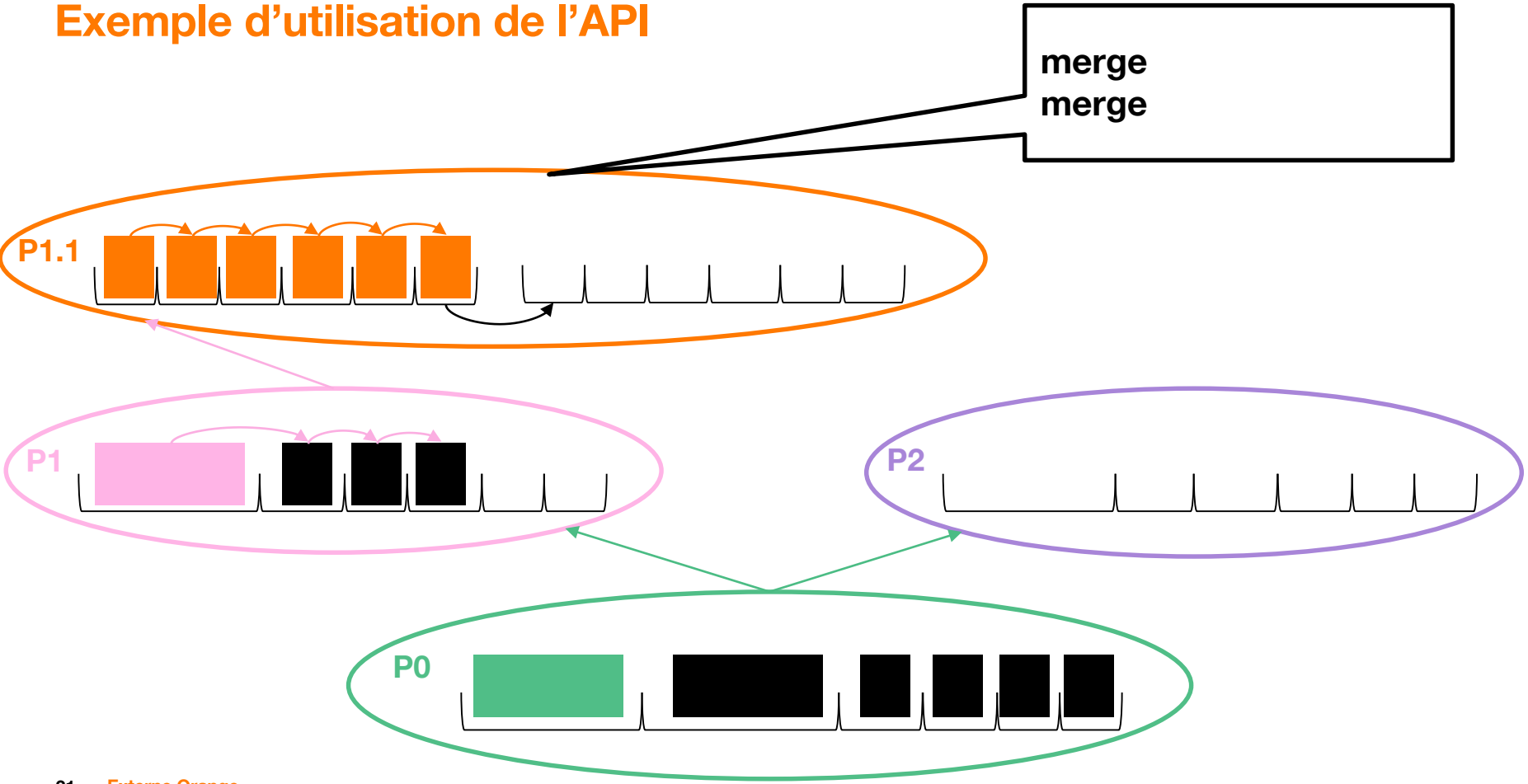
Exemple d’utilisation de l’API



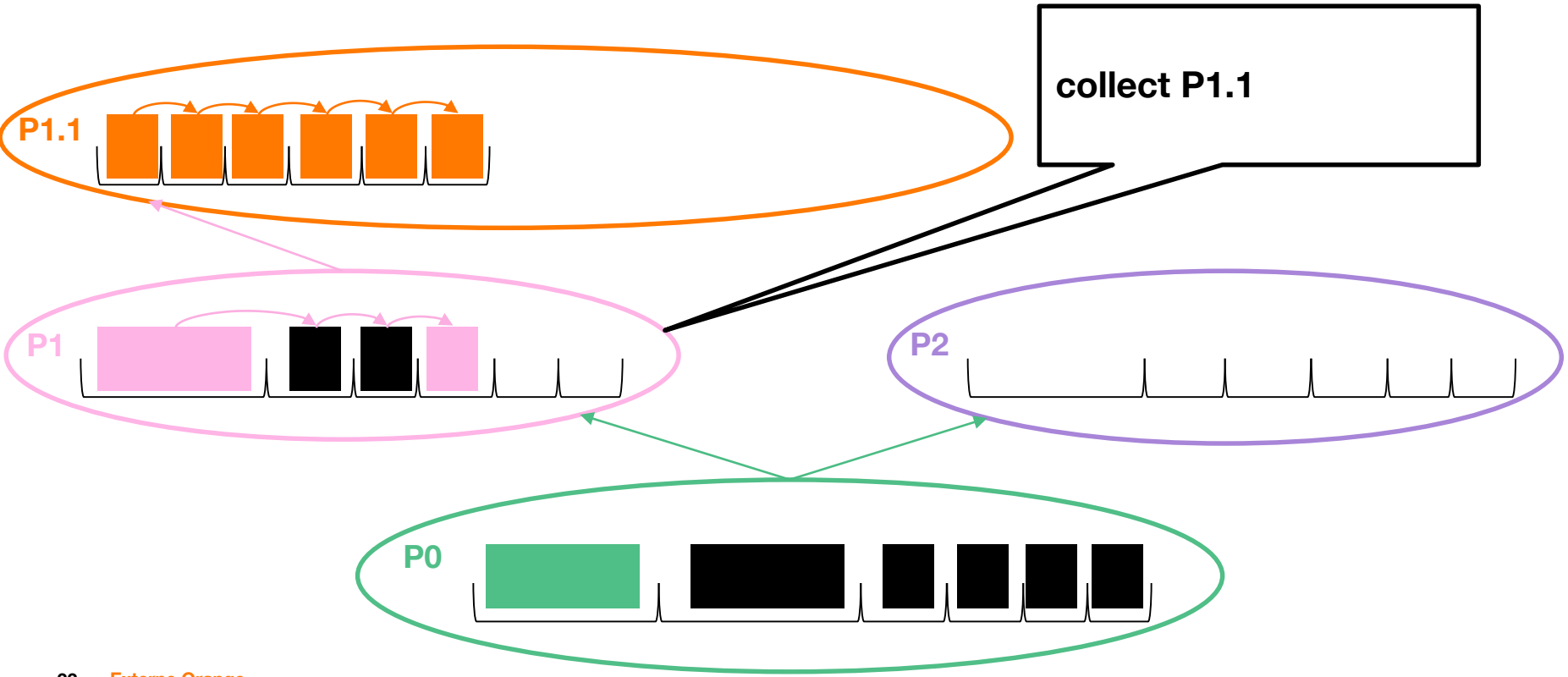
Exemple d’utilisation de l’API



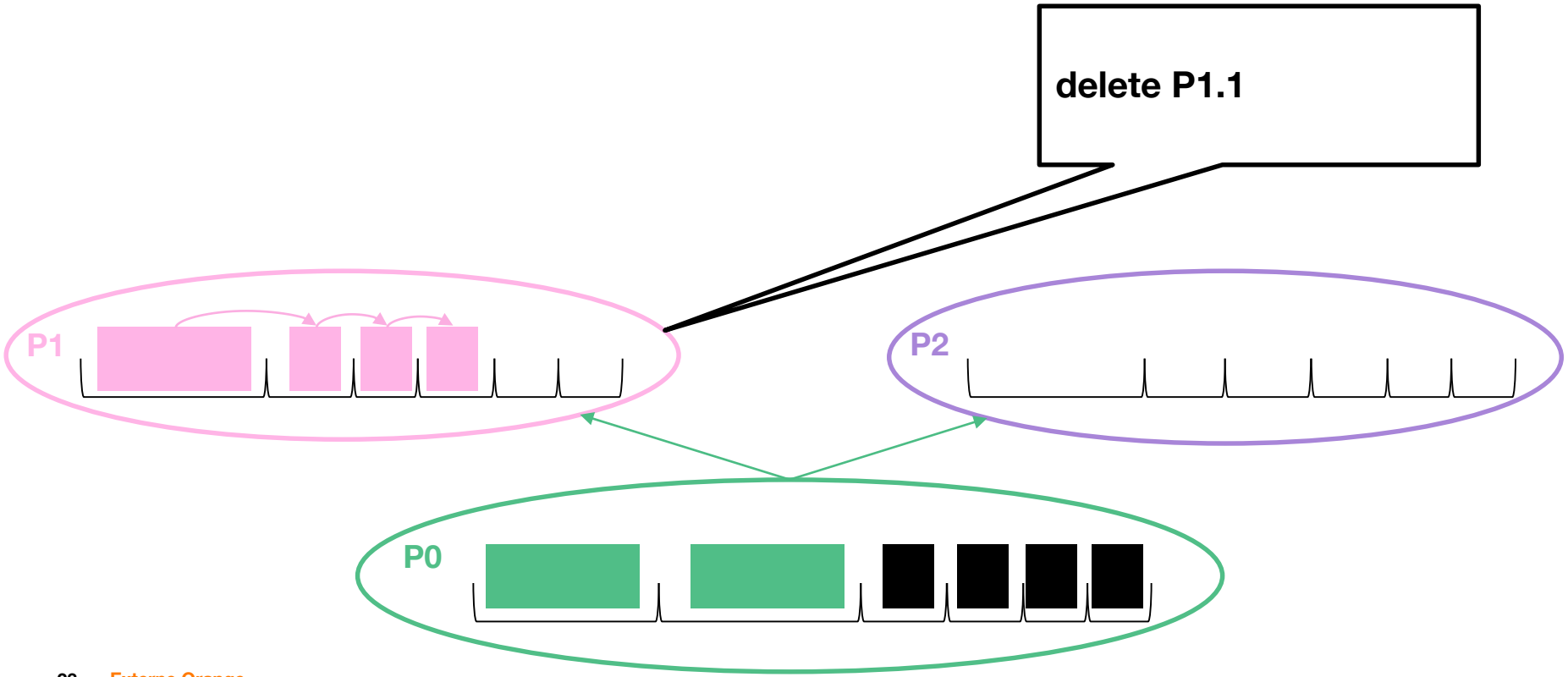
Exemple d’utilisation de l’API



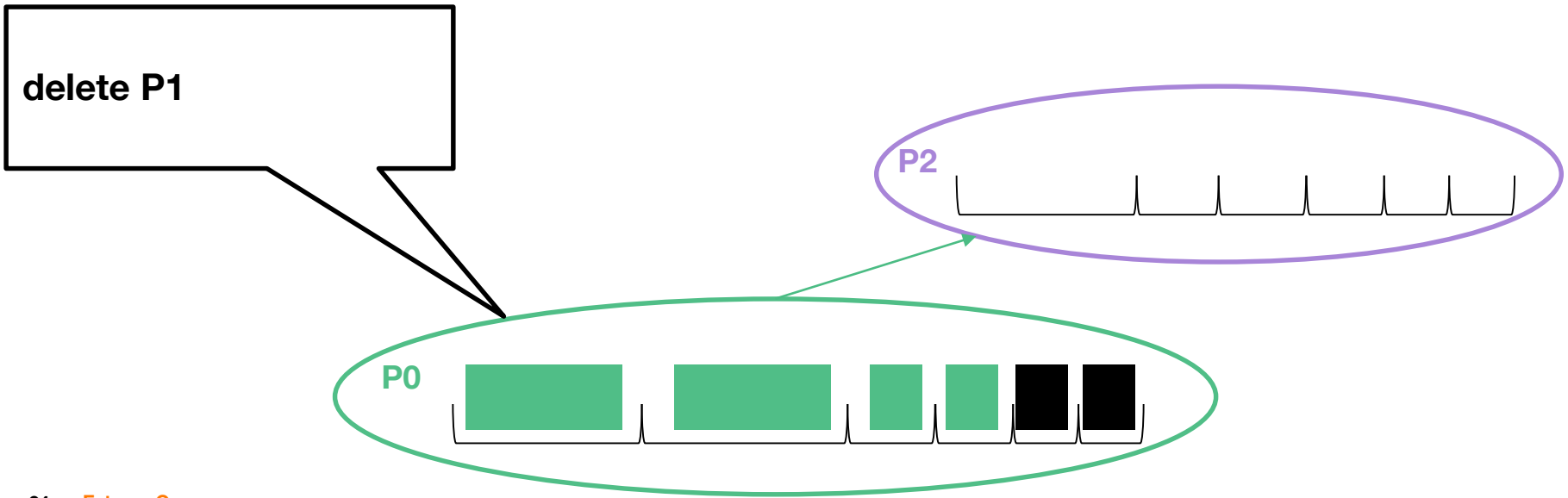
Exemple d’utilisation de l’API



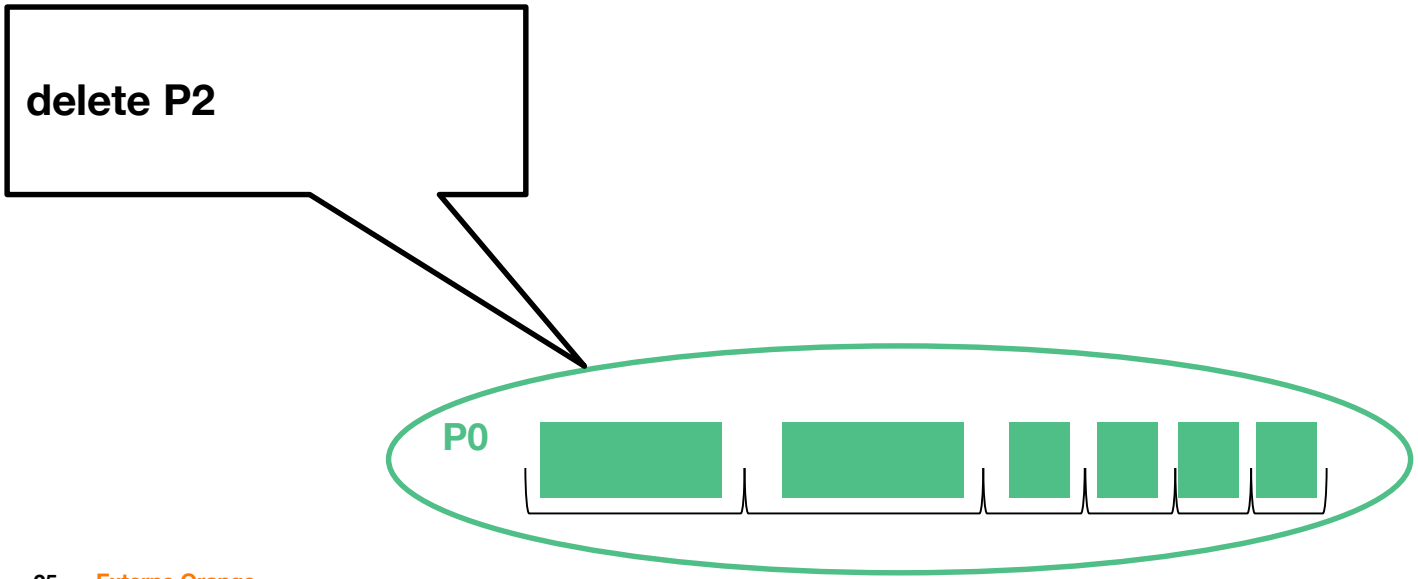
Exemple d’utilisation de l’API



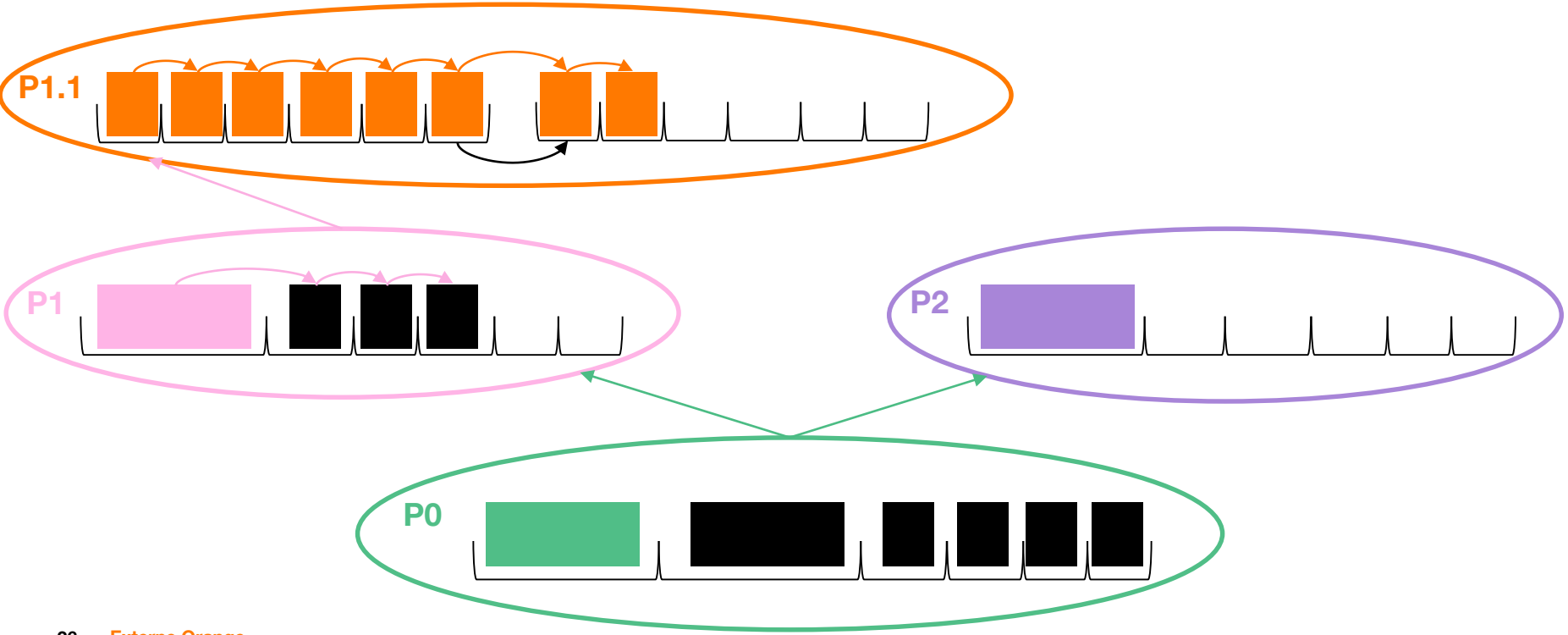
Exemple d’utilisation de l’API



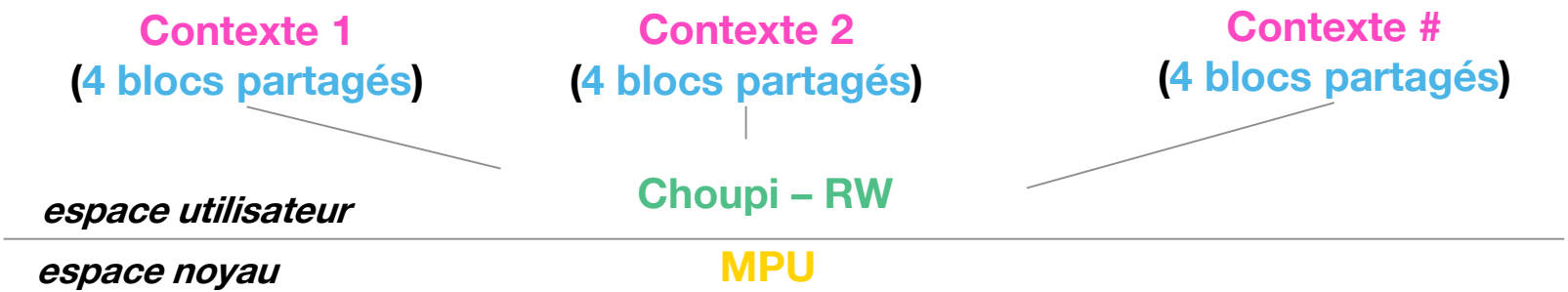
Exemple d’utilisation de l’API



Exemple d’utilisation de l’API - Récapitulatif

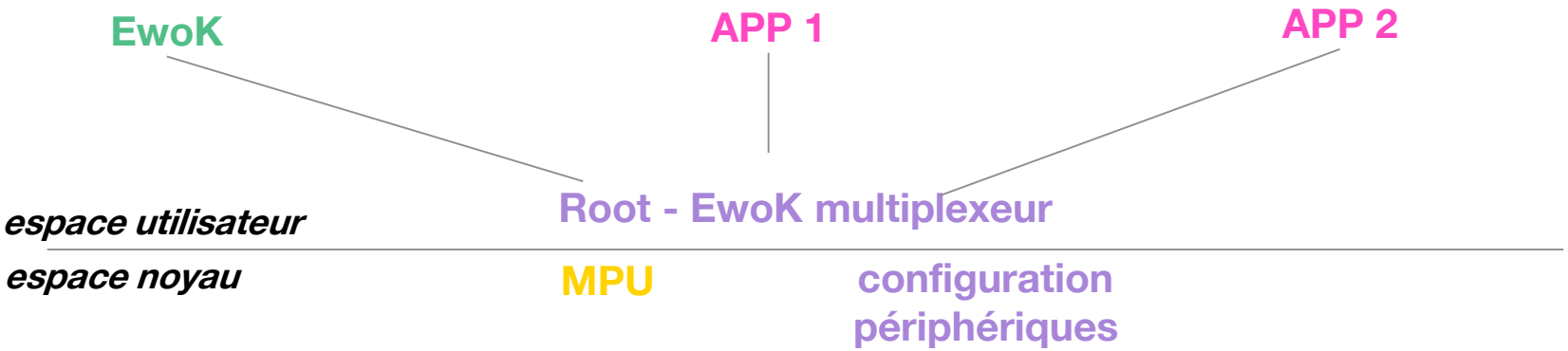


Exemples de mise en place du *framework* MPU avec Choupi-OS



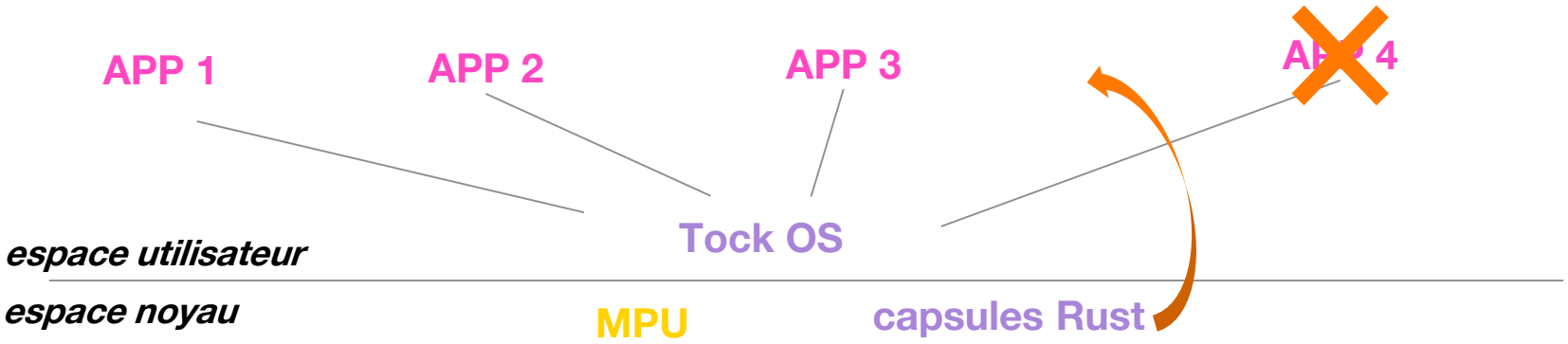
Isolation des contextes, mémoire partagée
+ extension mémoire
+ creation de plus de contextes

Exemples de mise en place du *framework* MPU avec WooKey (EwoK)



isolation des applications,
isolation d'EwoK
+ extension mémoire
+ creation de plus d'applications

Exemples de mise en place du *framework* MPU avec TockOS



isolation des applications, extension mémoire limitée, capsules “de confiance”, chargement/déchargement d’applications
+ création de plus d’applications
+ protection des capsules
+ isolation du noyau

Conlusion

Proposition

- Mise en place d'une compartimentation imbriquée ciblant les objets contraints
- basée sur la *Memory Protection Unit* (MPU) sans modification matérielle
- par l'intermédiaire d'une entité privilégiée dédiée pour la gestion de la MPU

Compatible avec les noyaux existants moyennant modifications

- Plus de flexibilité, plus de granularité de protection, avec les mêmes garanties de sécurité
- Clarification et simplification de la gestion de la MPU

Considérations supplémentaires

- Protection des métadonnées
- Virtualisation des régions MPU : automatique ou pilotée

Efficace contre notre adversaire

- Amélioration de la sécurité possible en suivant la philosophie du proto-noyau Pip

Quelques références

- Narjes Jomaa, David Nowak, and Paolo Torrini. Formal Development of the Pip Protokernel. 2018.
- Ryad Benadjila, Arnauld Michelizza, Mathieu Renard, Philippe Thierry, and Philippe Trebuchet. Wookey: Designing a trusted and efficient USB device. *ACM International Conference Proceeding Series*, pages 673–686, 2019.
- Guillaume Bouffard and Léo Gaspard. Hardening a Java Card Virtual Machine Implementation with the MPU. 2018.
- Tock development team . Website of: Tock. https://www.tockos.org/, 2020. [Online; accessed November 20, 2020].

Merci de votre écoute
Thank you for listening

Contacts

- **Nicolas Dejon**
  - nicolas.dejon@univ-lille.fr
  - nicolas.dejon@orange.com
- **Chrystel Gaber, PhD**
  - chrystel.gaber@orange.com
- **Prof. Gilles Grimaud**
  - gilles.grimaud@univ-lille.fr

orange™

Université de Lille