Le Projet OP-TEE

Implementation libre d’un TEE
Status en 2021

Etienne CARRIERE
Ingénieur logiciel embarqué, STMicroelectronics

Etienne Carrière,
pour les Journées SIF sur les environements d’exécutions sécurisés, 14 & 15 octobre 2021.

Ingénieur logiciel embarqué, STMicroelectronics
e-mail: etienne.carriere@st.com

Actuellement assigné par STMicroelectronics dans l’organisation Linaro
e-mail: etienne.carriere@linaro.org

Avatar gihub.com: etienne-lms

life.augmented

Linaro

Agenda

- Historique, cadre et composants
- Détails d’architecture
- Services disponibles

Cadre et composants

Cadre du projet OP-TEE

https://www.op-tee.org

Implémentation des spécifications GPD TEE (Client & Internal Core APIs) sous license libre

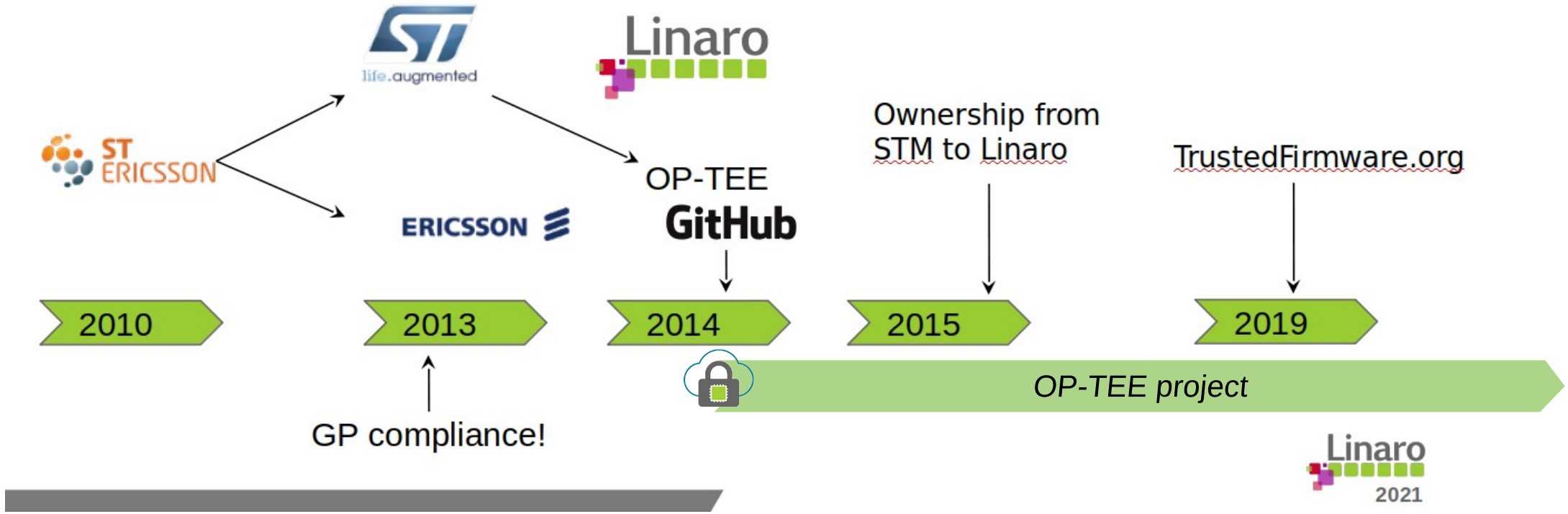
Supporte les architectures Arm/Trustzone (cortex-A, 32/64bit)

https://github.com/OP-TEE/optee_os.git

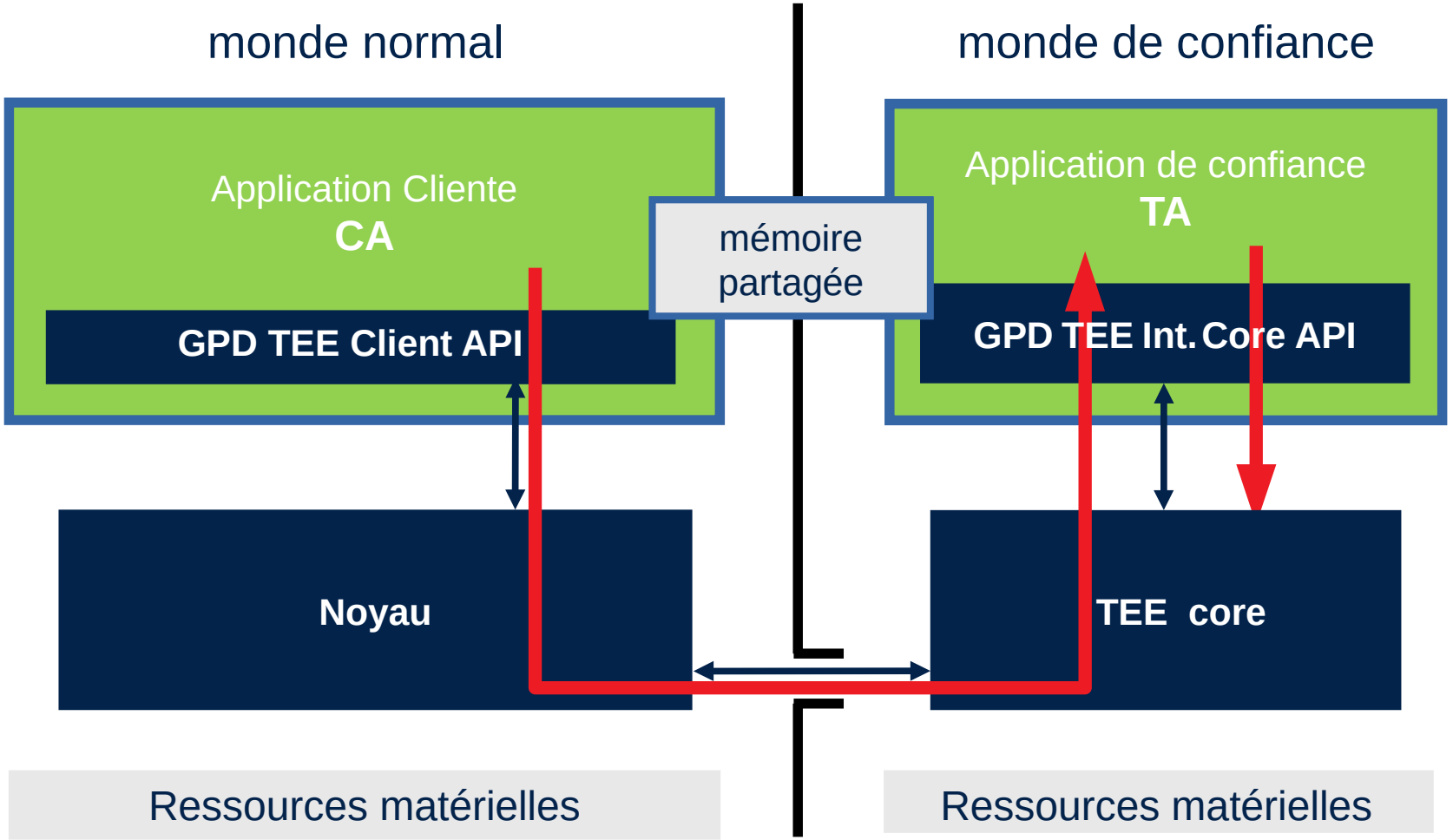
https://optee.readthedocs.io/en/latest/

https://github.com/OP-TEE/optee_os/security/advisories?state=published

Historique

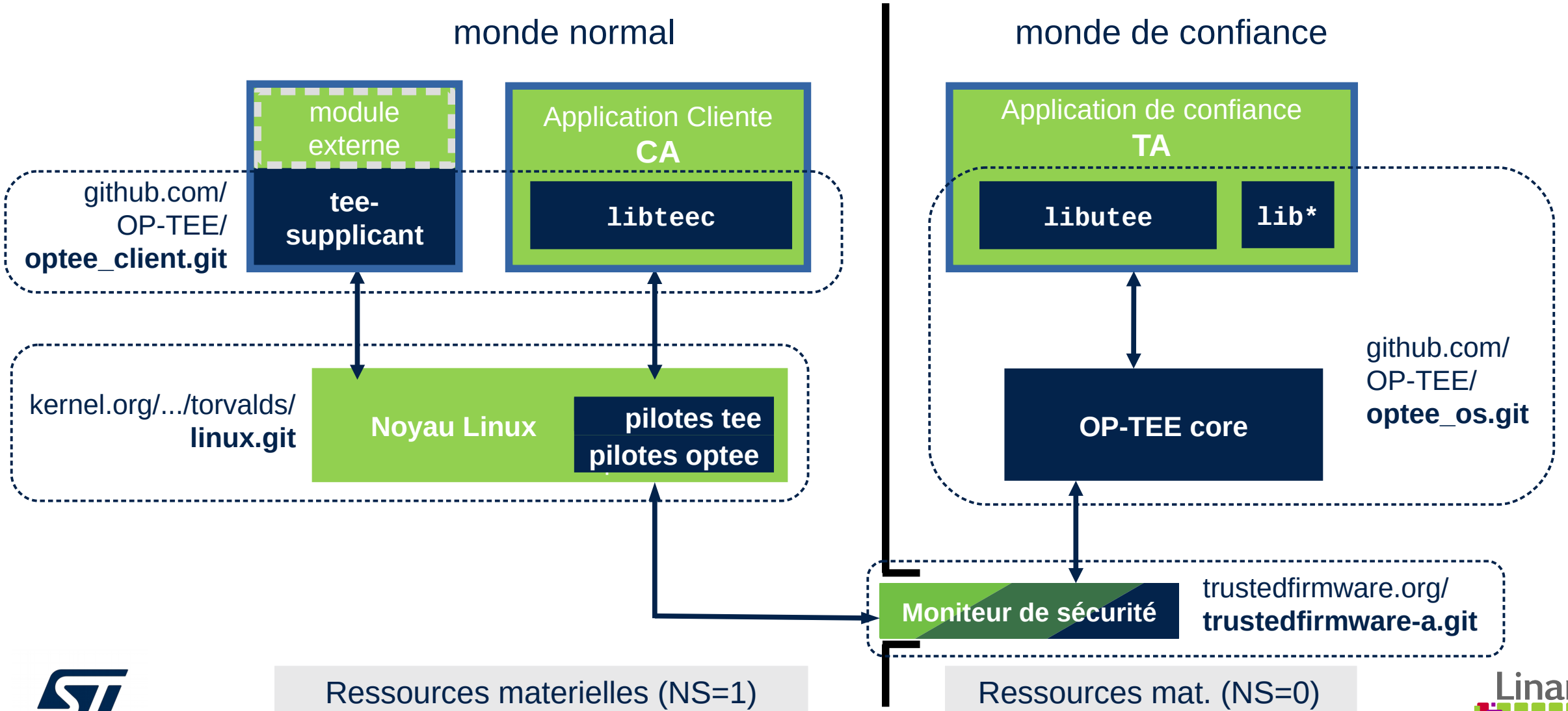

life.augmented
Linaro
ST ERICSSON
ERICSSON
OP-TEE
GitHub
Ownership from STM to Linaro
TrustedFirmware.org
2010
2013
2014
2015
2019
GP compliance!
OP-TEE project
Linaro 2021

L’architecture GPD TEE



https://globalplatform.org/specs-library/

Les composants de OP-TEE / Arm/TZ©



Les composants de OP-TEE

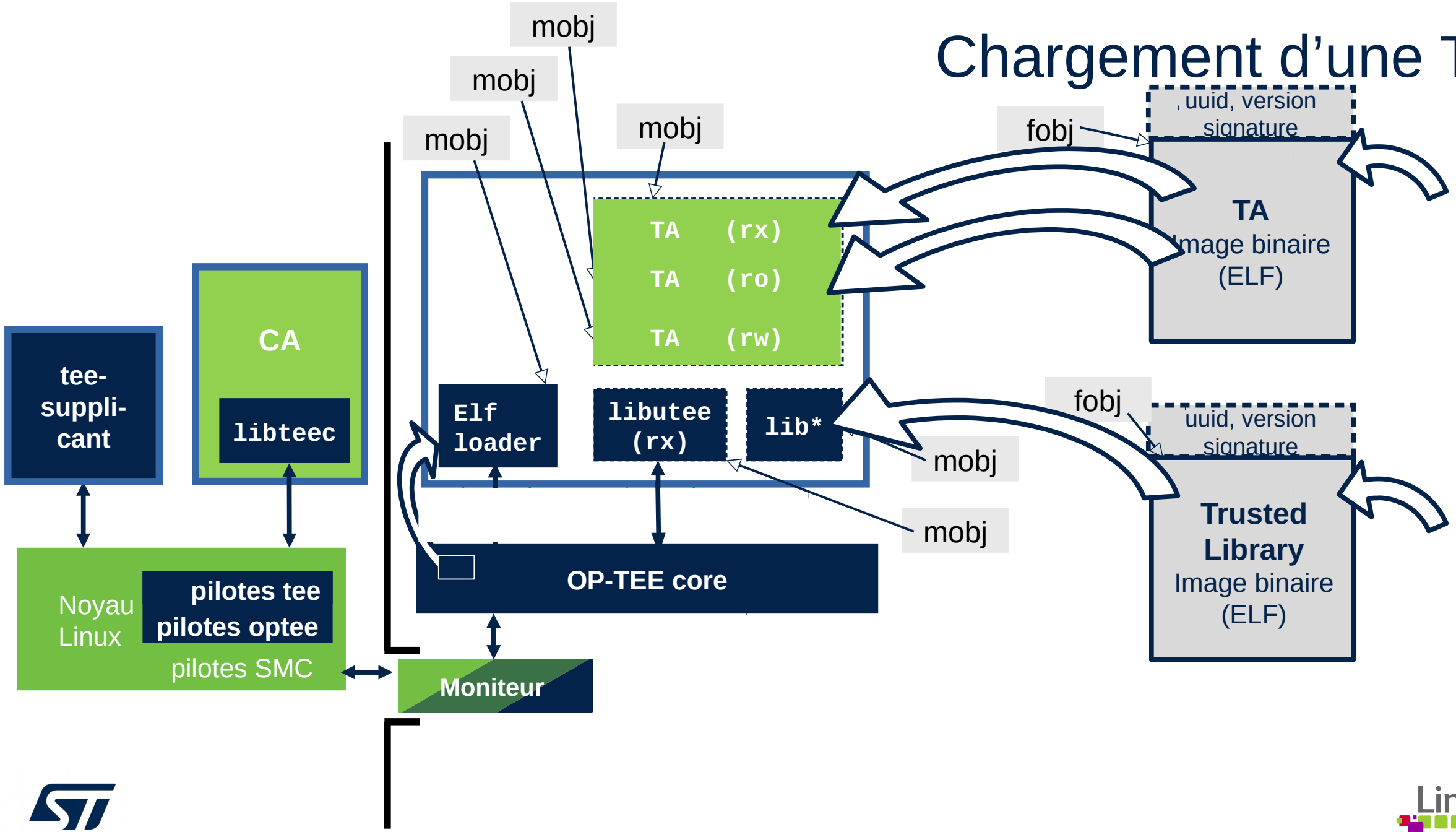
https://github.com/OP-TEE/	optee_os.git	BSD-2-Clause
	optee_client.git	LGPLv2 (librairies), BSD-2-Clause (tee-supplicant)
	optee_test.git	GPLv2 (xtest), BSD-2-Clause (TAs)
	optee_docs.git	BSD-2-Clause
https://github.com/linaro-swg/	linux.git	GPLv2
	optee_examples.git	BSD-2-Clause
	optee_benchmark.git	BSD-2-Clause
https://github.com/OP-TEE/	manifest.git	
	build.git	

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git	GPLv2
https://source.denx.de/u-boot/u-boot.git	GPLv2

...

Détails d’architecture

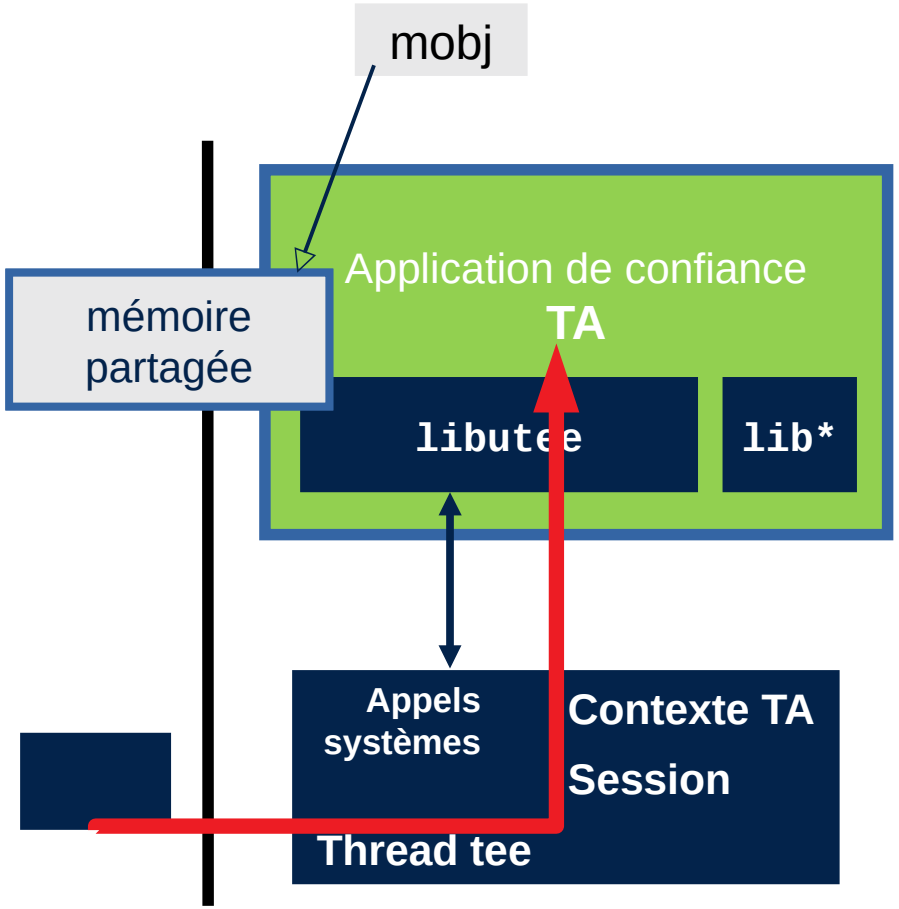
Chargement d'une TA



life.augmented

Linaro

Contexte d’éxecution



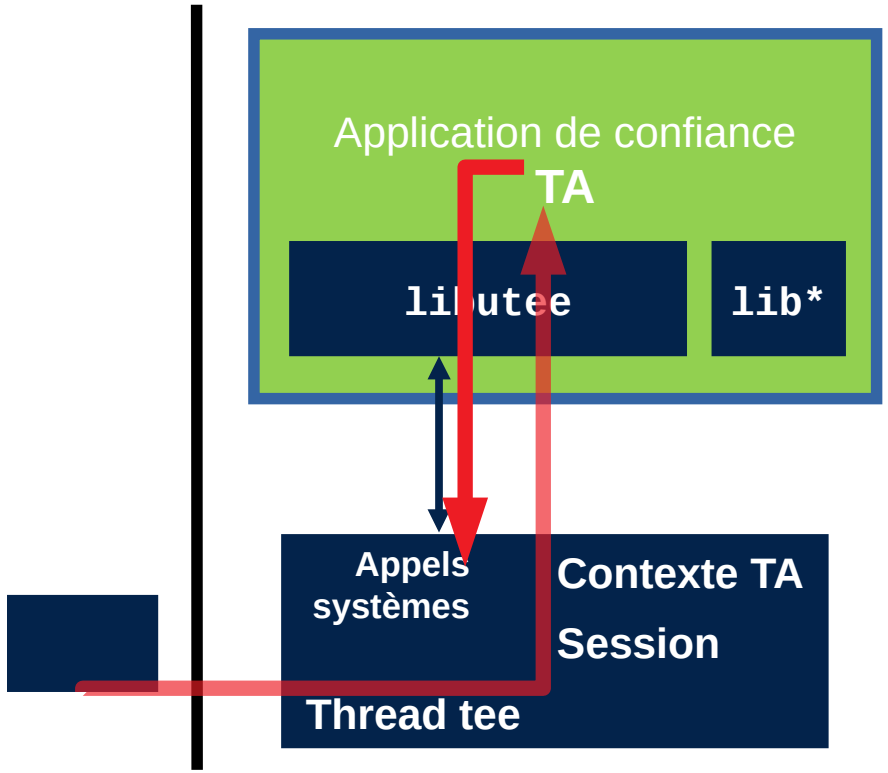
Contexte applicatif
- Interruptible
- Préemptible
- Thread *tee*, lié au thread non-secure d’entrée

OP-TEE core n’a pas de séquenceur de tâche/thread.

Invocation dans le cadre d’une session TEE
- Client identifié

Arguments d’invocation
- 1 Identifiant commande
- 4 paramètres d’invocation:
  valeur ou réference mémoire partagée

Services systèmes



Capture de temps

Stockage sécurisé (coffre-fort)
- eMMC/RPMB
- Fichiers dans Linux
- `struct tee_file_operations` (*a la* POSIX)

Generation de nombre aléatoire

Objects (clés, certificats) et operations cryptographiques
- Import (provision), génération
- Chiffrement, signature, AE, dérivation
- Pilotes et libraries cryptographiques

Invocation d'une autre TA

Extensions OP-TEE:
- Dérivation de la clé unique du produit (HUK)
- Libraries dynamiques/partagées (dlopen, dlsym)
- Invocation de service tee-supplicant
- ...

life.augmented

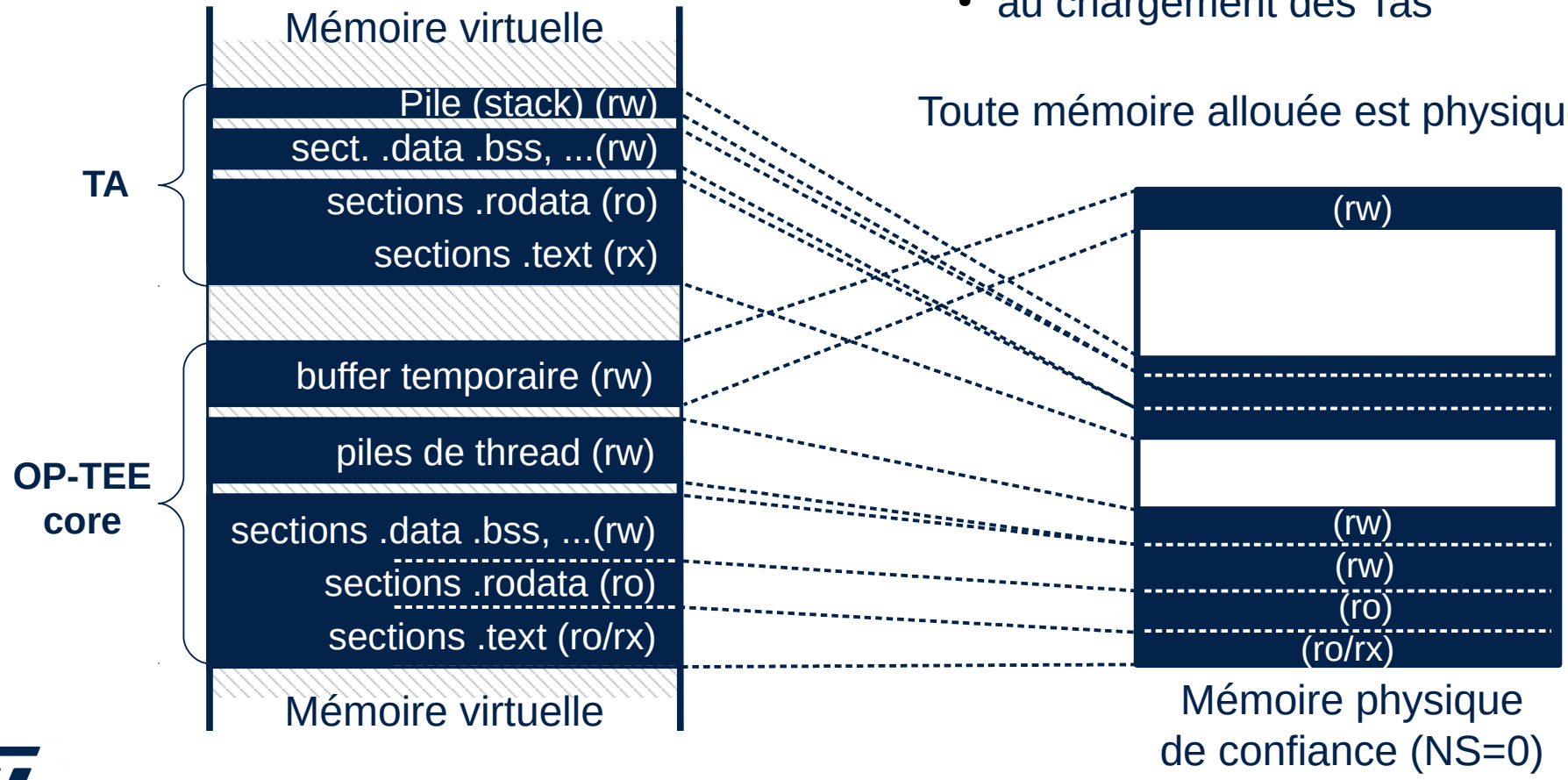
Linaro

Configuration nomale: “pager” désactivé

La mémoire physique de confiance est provisionnée:

- au démarrage de OP-TEE core
- au chargement des Tas

Toute mémoire allouée est physiquement disponible.

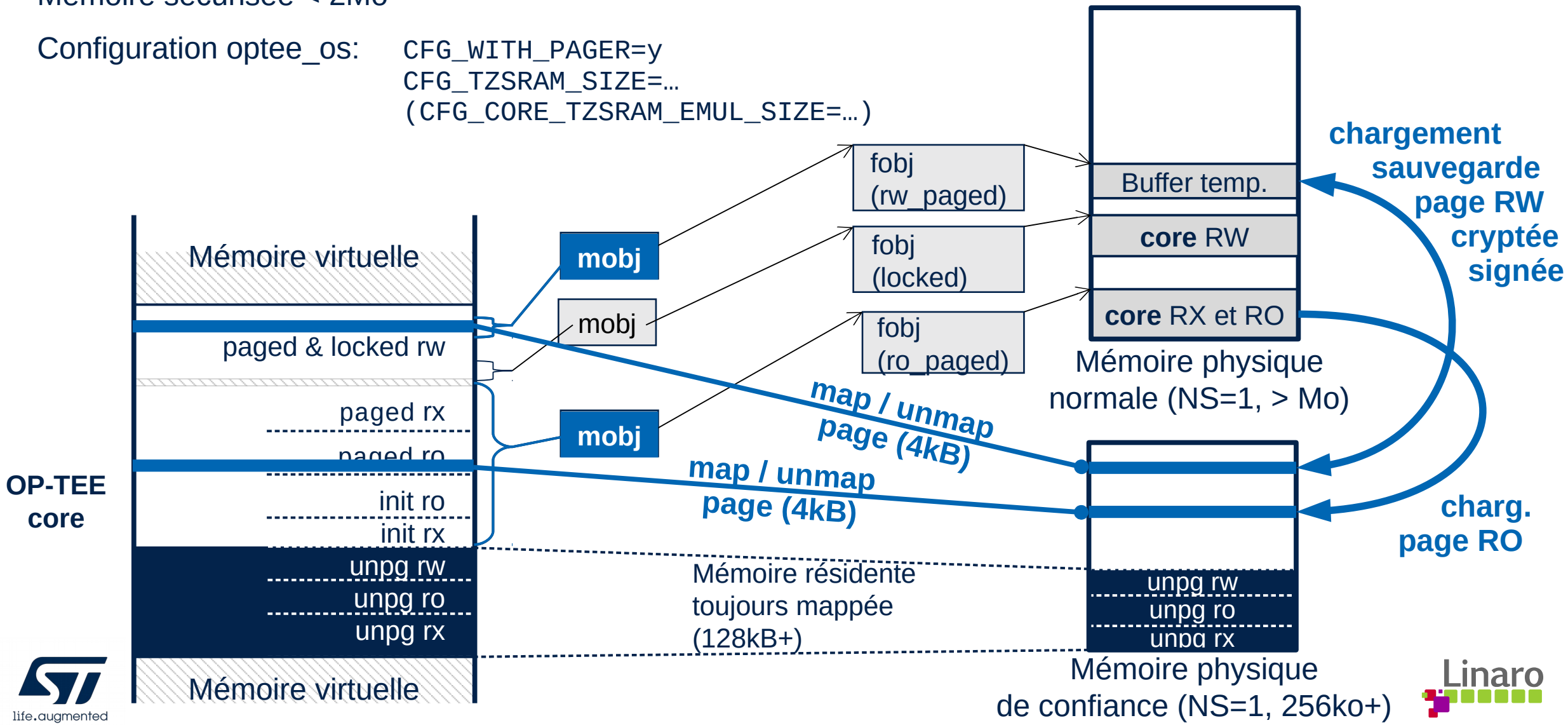


Pager & core

Mémoire sécurisée < 2Mo

Configuration optee_os:

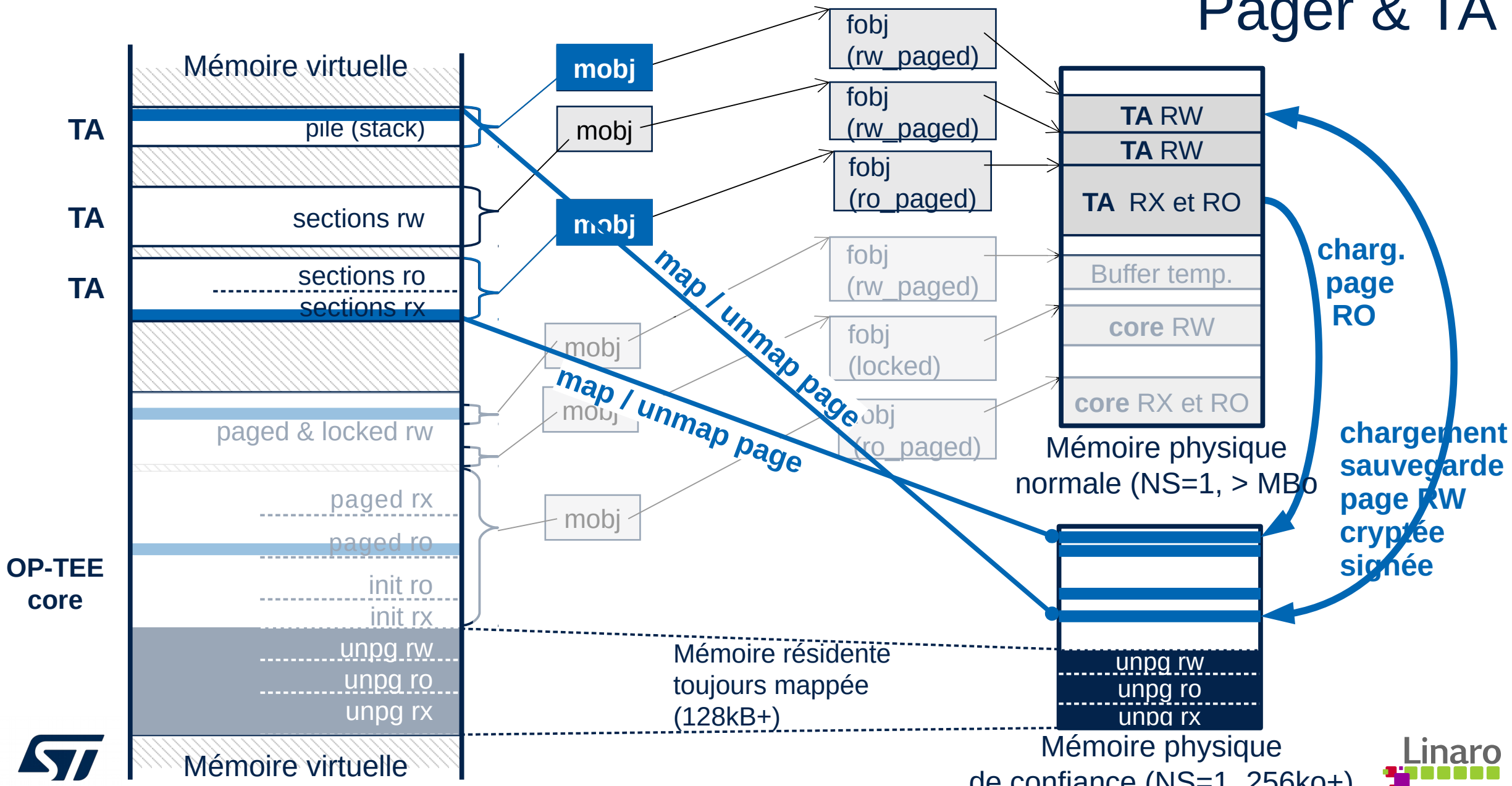
```
CFG_WITH_PAGER=y
CFG_TZSRAM_SIZE=…
(CFG_CORE_TZSRAM_EMUL_SIZE=…)
```



life.augmented

Linaro

Pager & TA

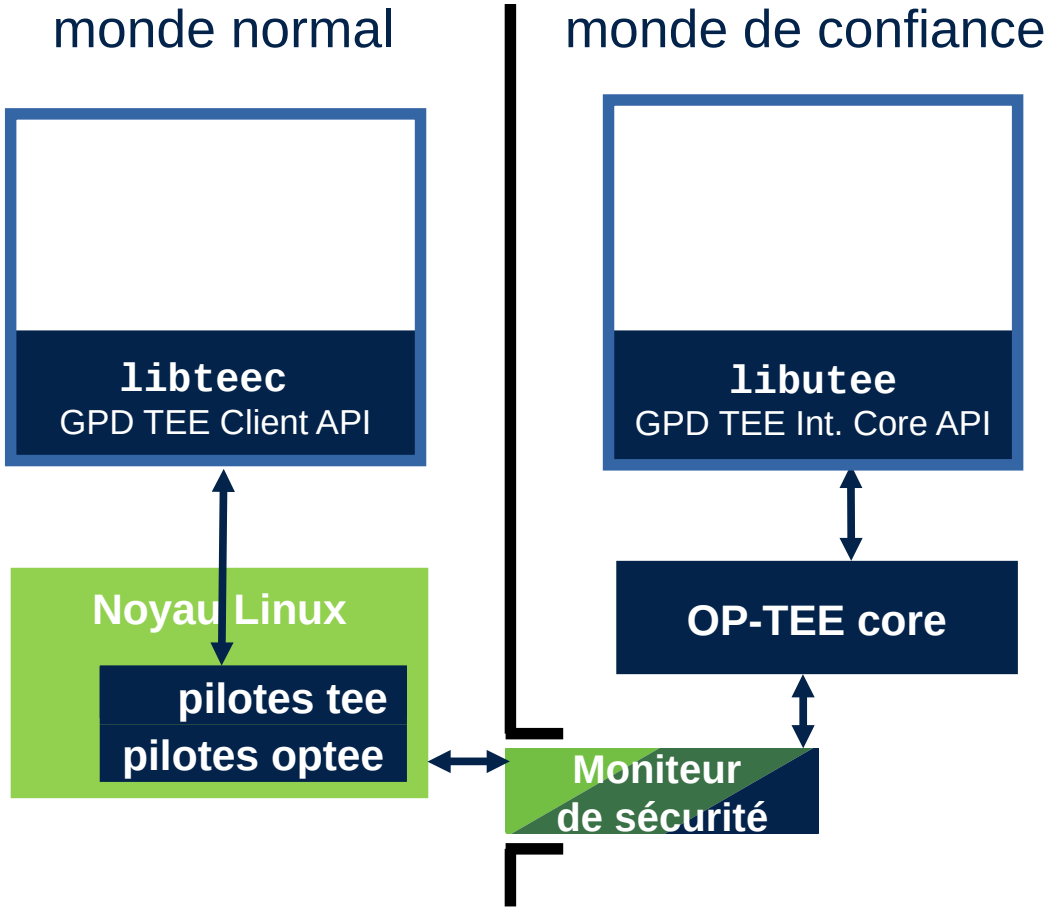


life.augmented

Linaro

Services dans l’écosystème OP-TEE

optee_os.git: dévelopement de TAs / CAs



optee_client.git
- Libraries libteec (GPD TEE Client API)
- Service tee-supplicant

optee_os.git
- OP-TEE core
- **TA devkit (SDK)**, inclus libutee (GPD TEE Internal Core API)

Livraison OP-TEE / qemu:
- Emulation de coeurs Armv7-A (32bit) et Armv8+ (32/64bit) sur PC avec Qemu. La distribution inclue OP-TEE et un OS Linux basé sur Buildroot.
- https://optee.readthedocs.io/en/latest/building/index.html
- Inclus optee_test (xtest) et optee_examples.

Services du projet OP-TEE

- Trusted keys: optee_os.git (3.9.0) `ta/trusted_keys/`
  linux.git (v5.13) `security/keys/trusted-key/trusted-tee.c`

- hwrng: optee_os.git (3.14.0) `core/pta/hwrng.c`
  linux.git (v5.1) `drivers/char/hw_random/optee-rng.c`

- PKCS#11: optee_os.git (3.15.0) `ta/pkcs11/`
  optee_client (3.15.0) `libckteec/`

- Android Verified Boot (AVB)
  optee_os.git (3.3.0) `ta/avb/`
  u-boot (v2018.07) `common/avb_verify/`, `lib/libavb/`

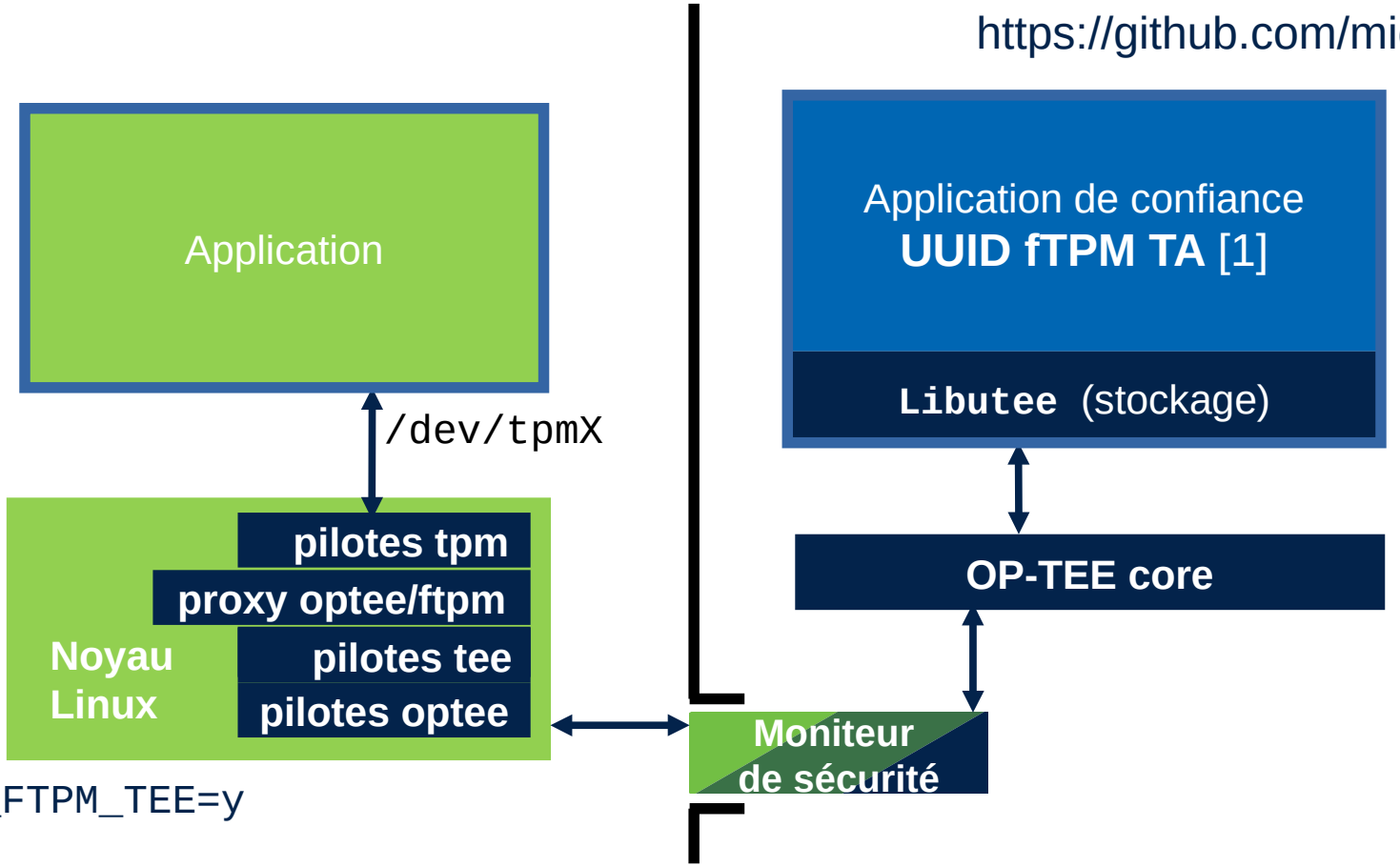
- Android KeyMaster & GateKeeper (magasin de clés d'Android)
  https://github.com/linaro-swg/kmgk.git

life.augmented

Linaro

Firmware TPM (fTPM)

https://github.com/microsoft/ms-tpm-20-ref



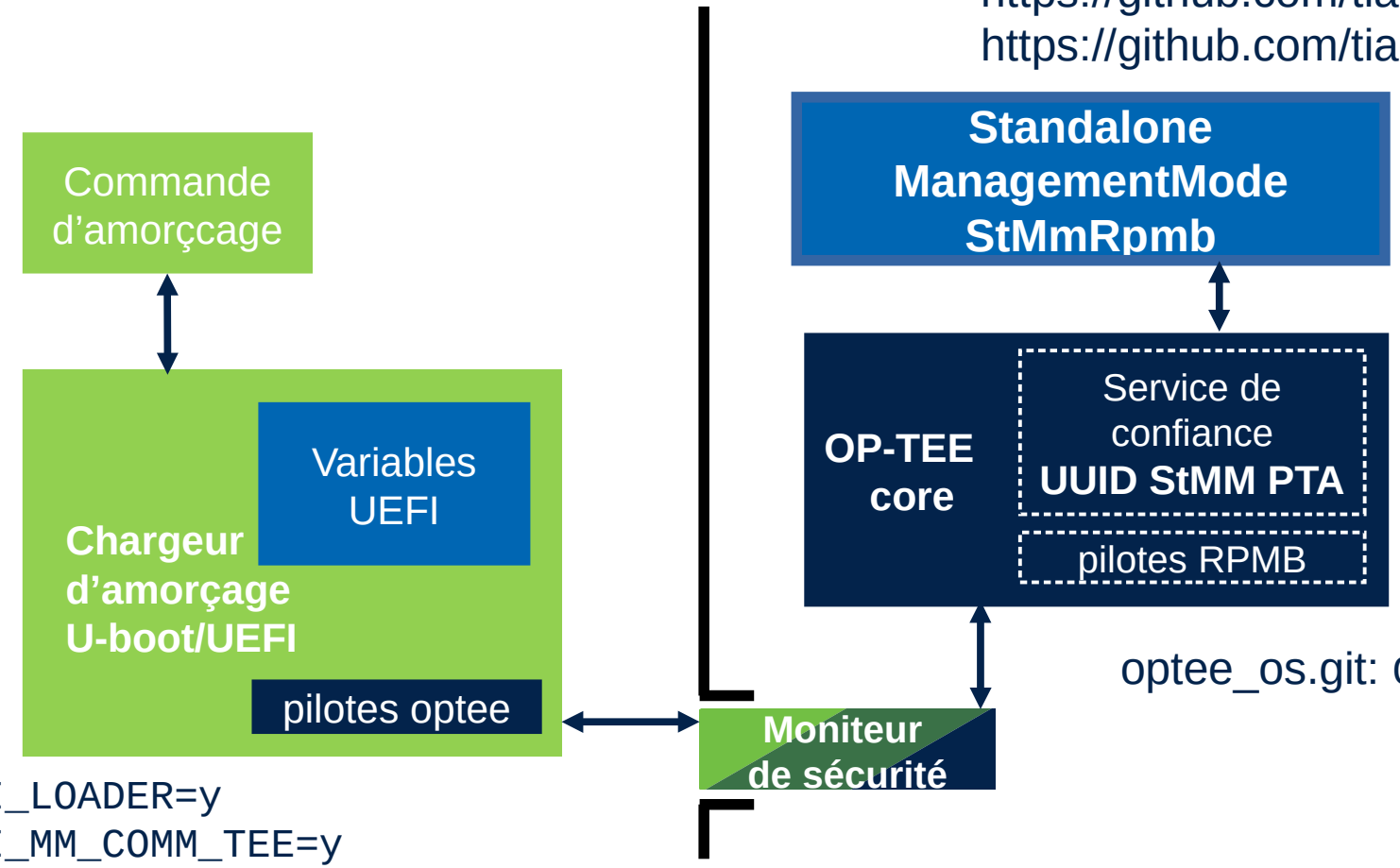
linux.git

`CONFIG_TCG_FTPM_TEE=y`

[1] https://github.com/microsoft/ms-tpm-20-ref/blob/master/Samples/ARM32-FirmwareTPM/README.md

EDK2: Variables UEFI sécurisées

https://github.com/tianocore/edk2
https://github.com/tianocore/edk2-platforms



u-boot.git

```
CONFIG_EFI_LOADER=y
CONFIG_EFI_MM_COMM_TEE=y
```

optee_os.git: `CFG_WITH_STMM_SP=y`

C++, Rust

OP-TEE core, les libraries (TAs et CAs), les exemples et tests (CAs et TAs) sont implémentés en C.

Implémentation de TA en C++

- Supporté depuis OP-TEE 3.10.0
- Restriction sur les libraries: statiques uniquement

Implémentation de CAs et Tas en Rust

- Teaclave incubator : https://github.com/apache/incubator-teaclave-trustzone-sdk
- En cours d'integration dans OP-TEE: tag 3.15.0 pour platforme qemu/armv8-a

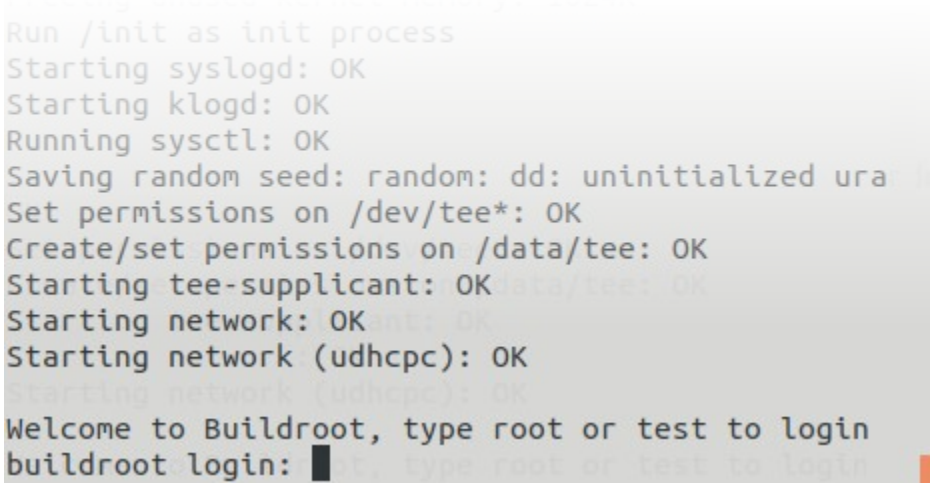
Essayez OP-TEE avec Qemu

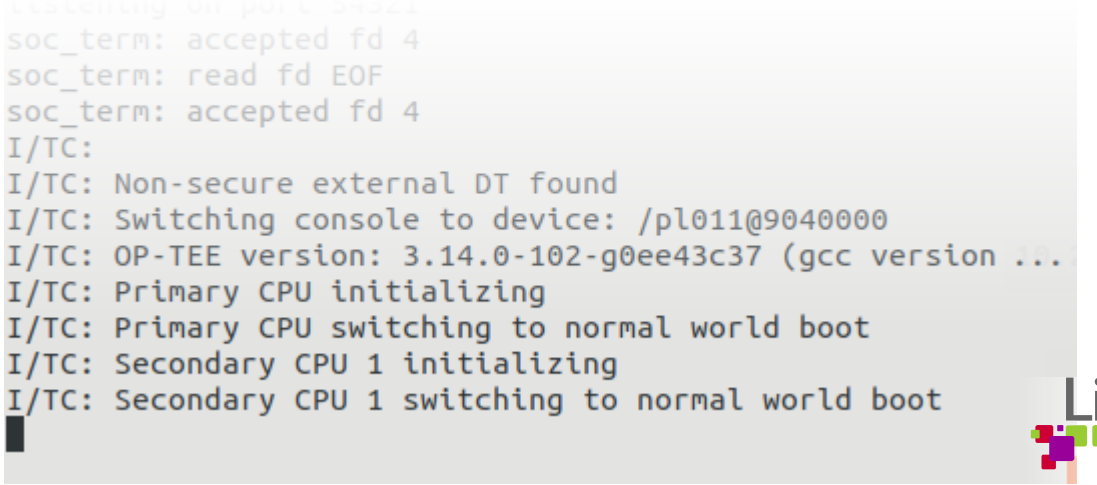
Instructions dans la documentation: https://optee.readthedocs.io/en/latest/building/devices/qemu.html

Sur une machine type Ubuntu ou Fedora:

```
$ repo init -u https://github.com/OP-TEE/manifest.git
$ repo sync
$ cd build
$ make toolchains
$ make run
```

```
QEMU 6.0.0 monitor - type 'help' for more information
(qemu) c
```





life.augmented

Linaro

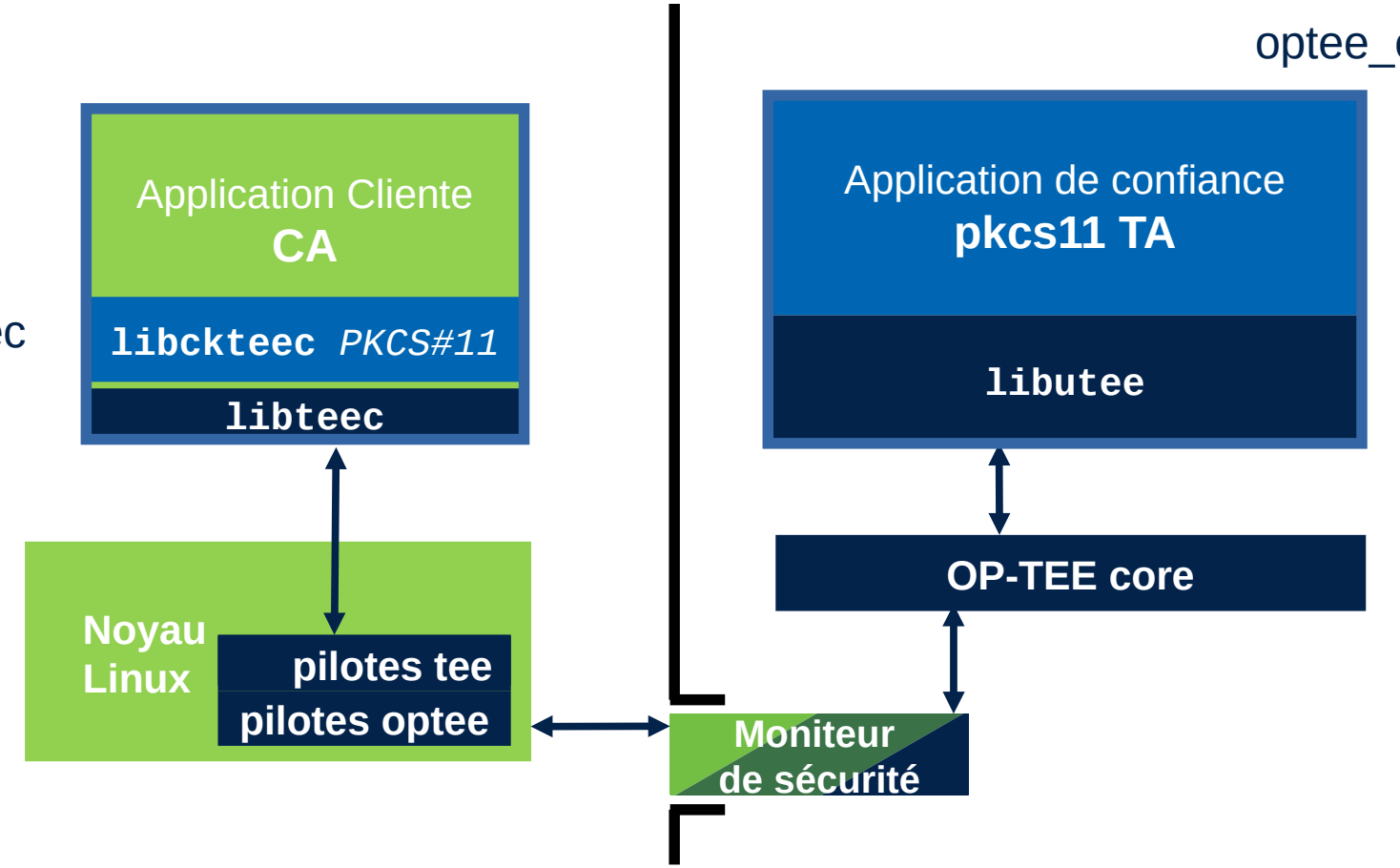
Merci

Annexes

optee_os.git: PKCS#11

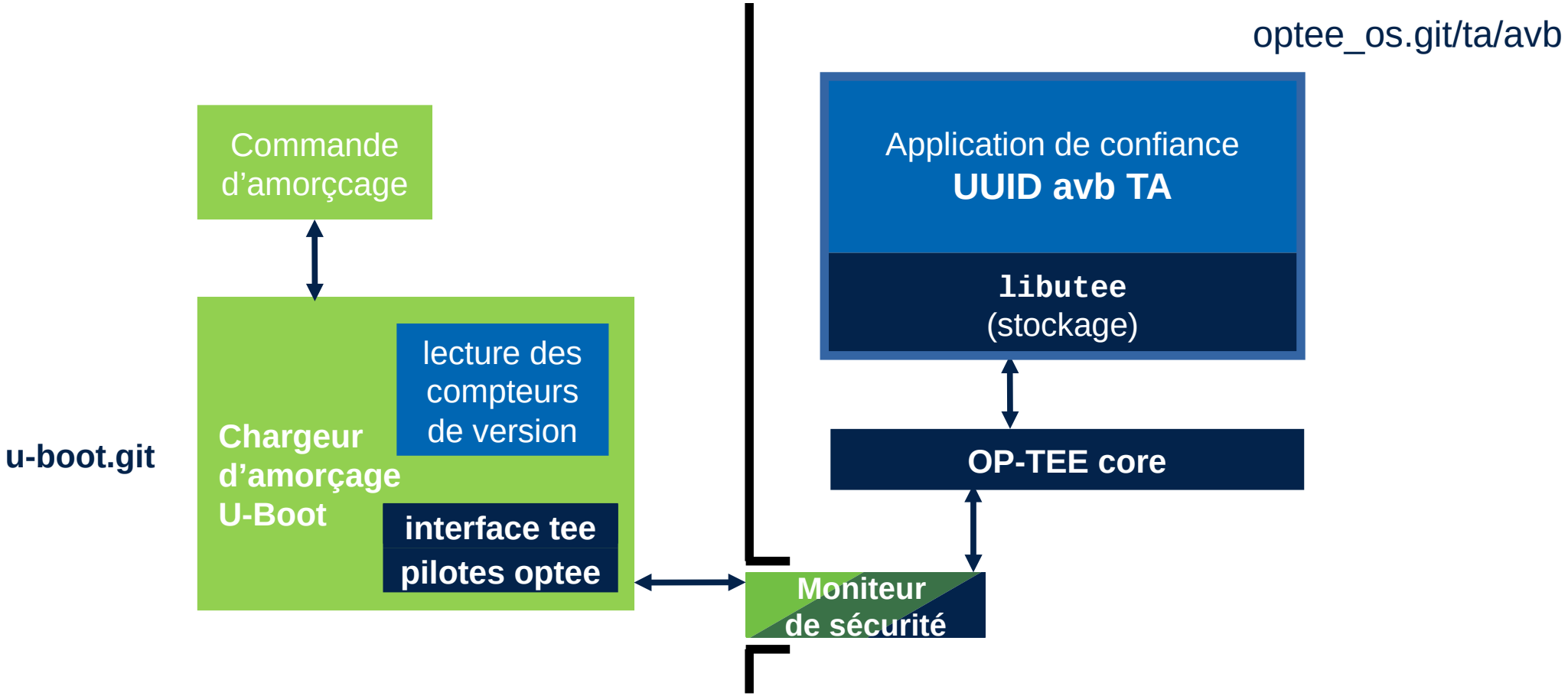
optee_os.git / ta/pkcs11

optee_client.git / libckteec

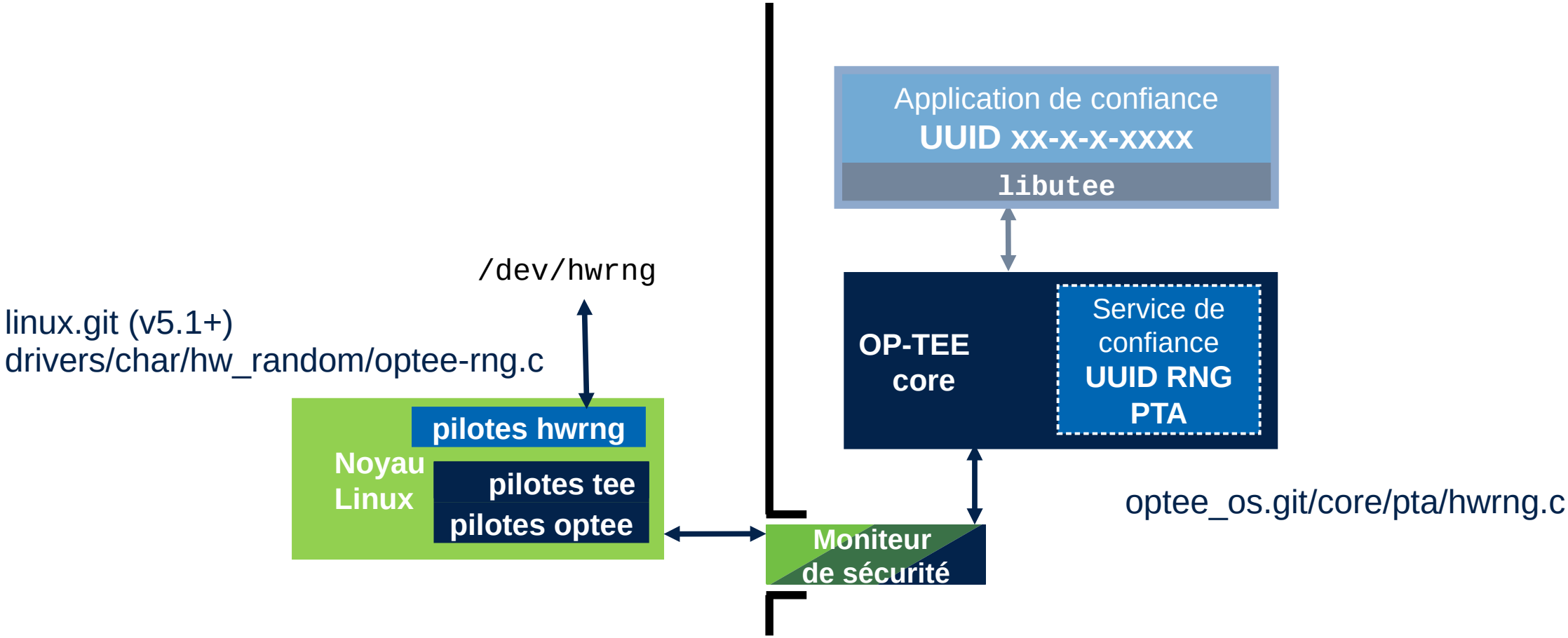


Intégration dans OP-TEE entre les livraisons 3.9.0 et 3.15.0

optee_os.git: Android Verified Boot (AVB)



Enumération de services exemple: hwrng



Découverte des services par énumération d'UUID de TA/PTA par OP-TEE