



Développement d'une carte électronique sécurisée

{LAM,LCR,LRP,LSC}@ANSSI

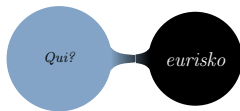
prenom.nom@ssi.gouv.fr

Symposium sur la
Sécurité des
Technologies de
l'Information et des
Communications

1^{er} juin 2016







labos
@ANSSI

Qui?

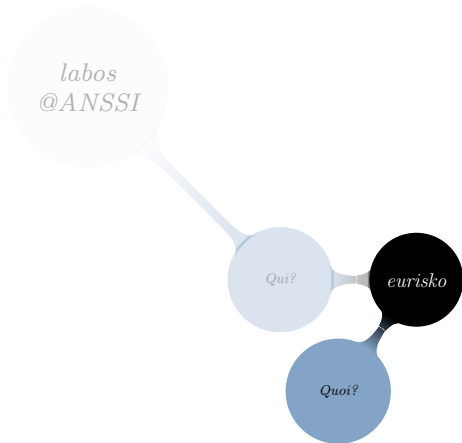
eurisko

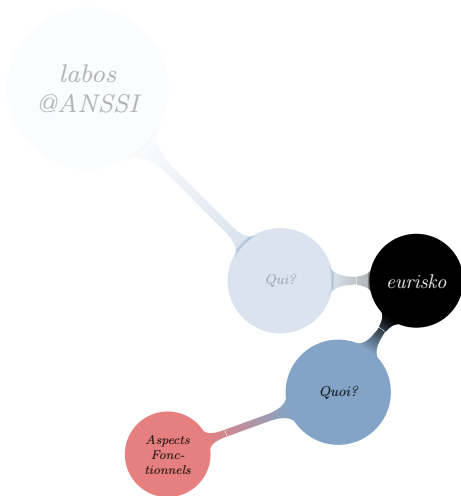
labos
@ANSSI

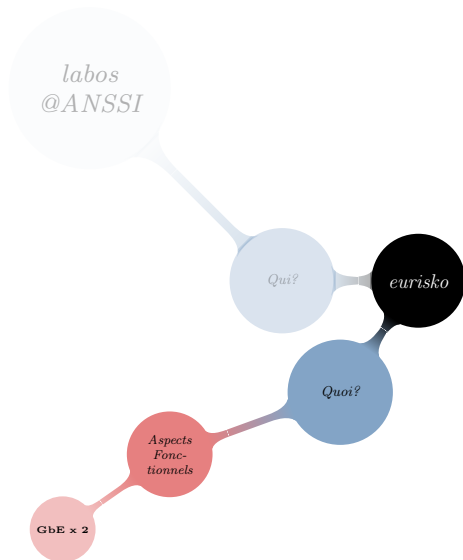
Qui?

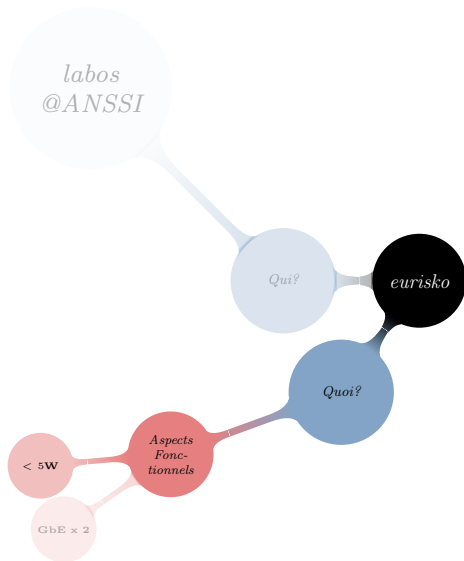
eurisko

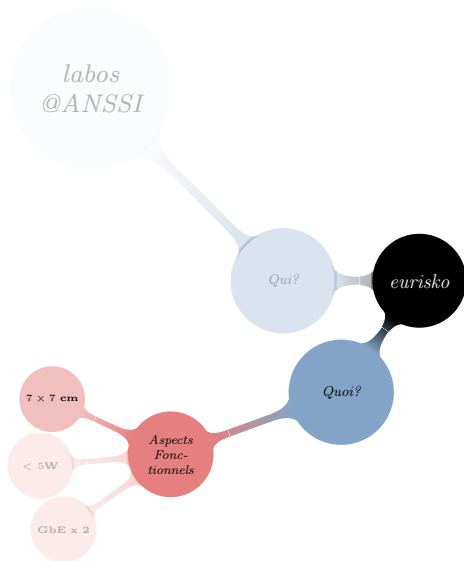
Quoi?

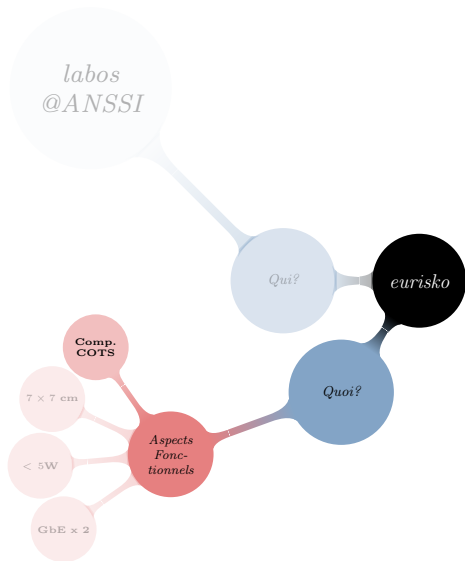


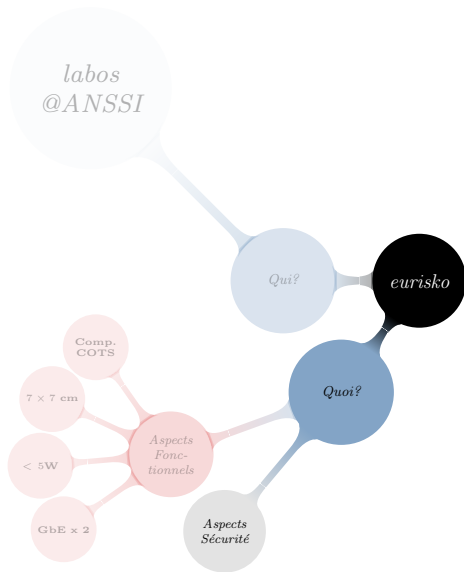


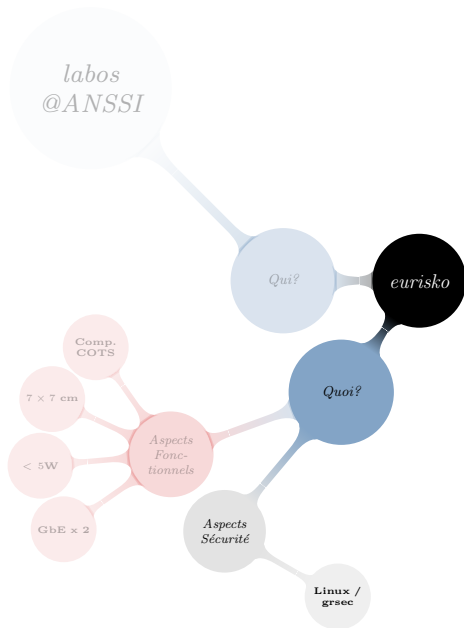


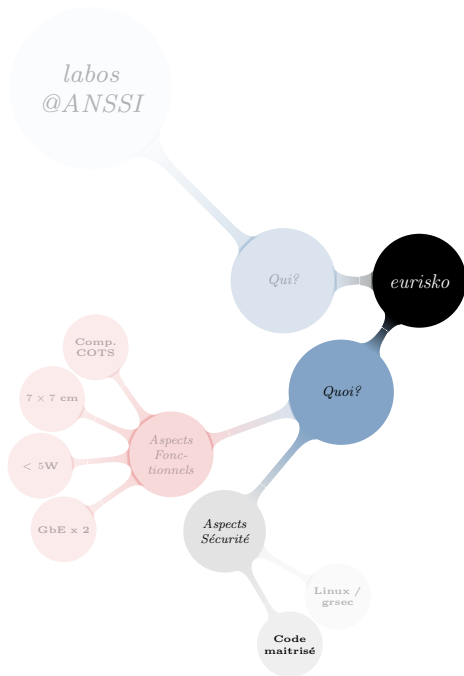


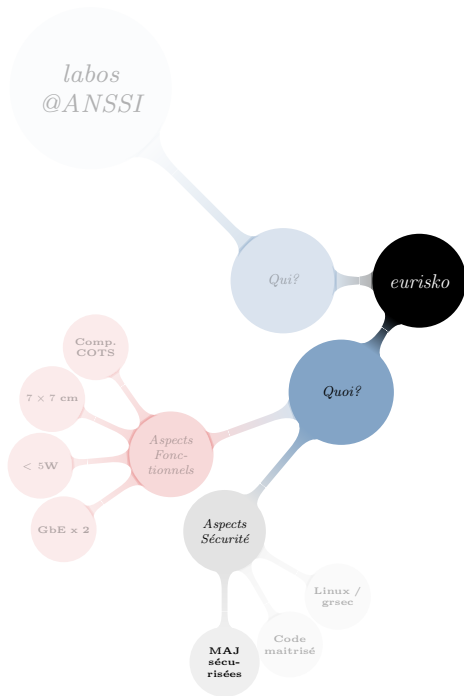


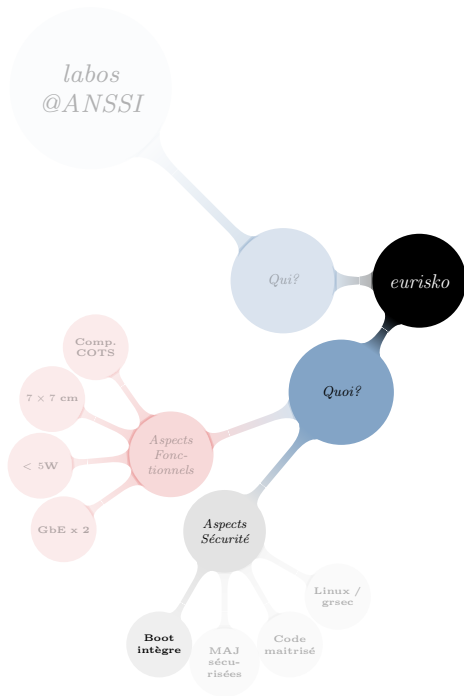


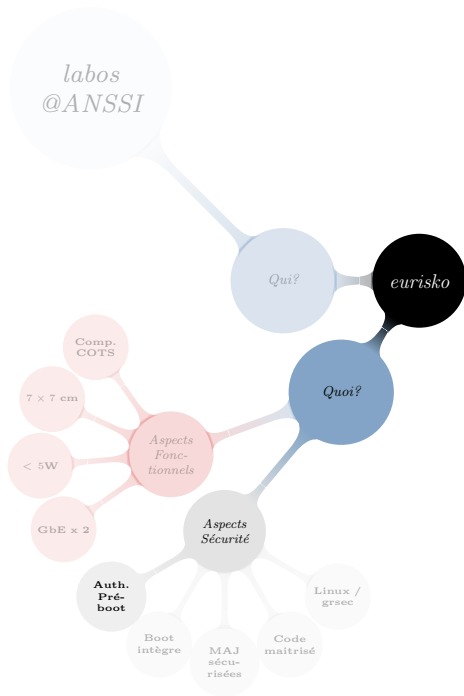


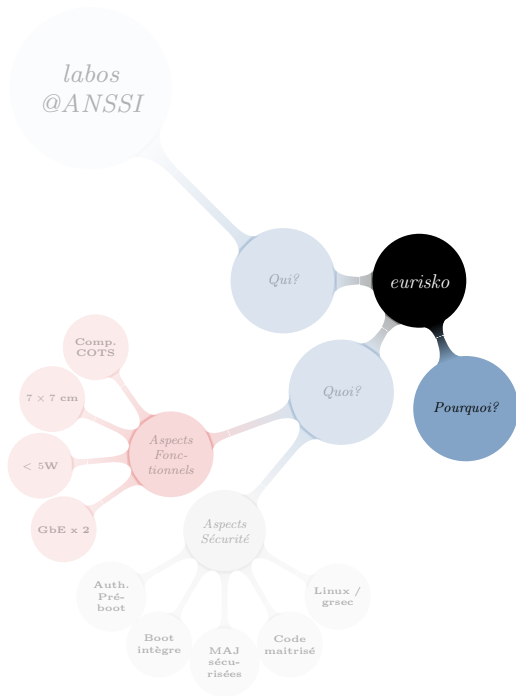


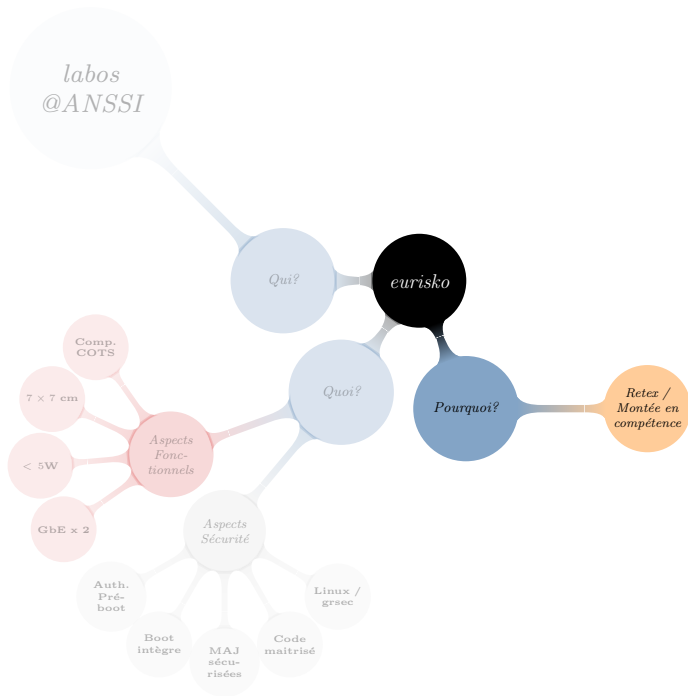


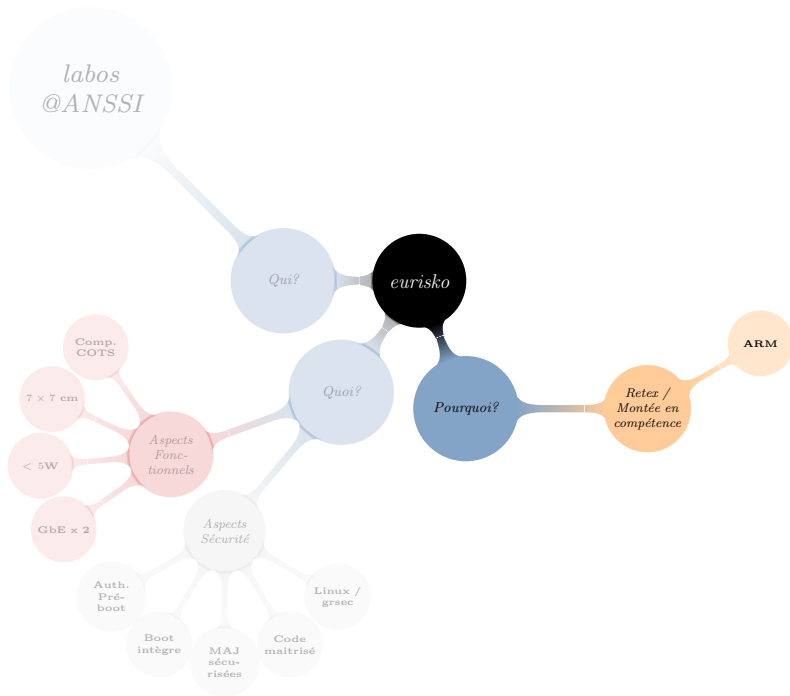


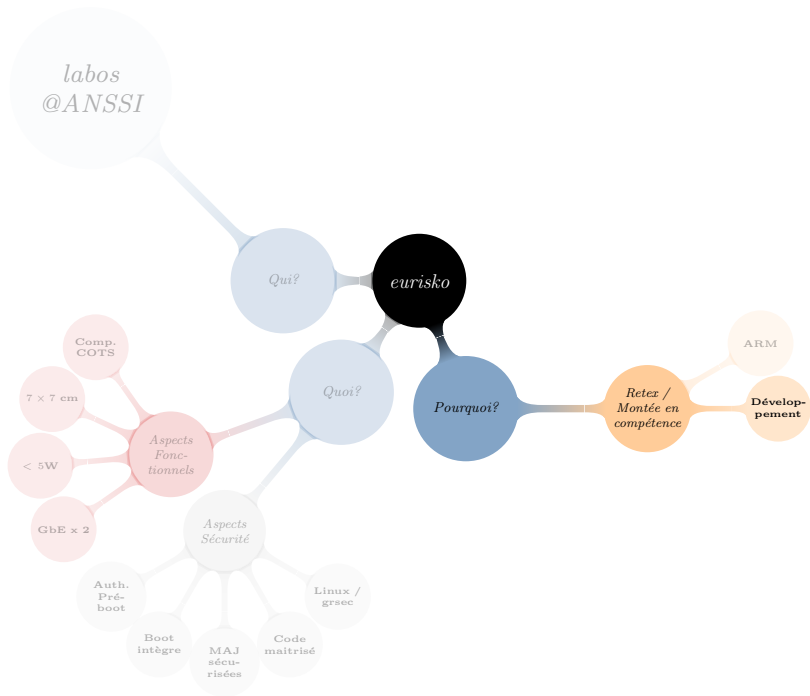


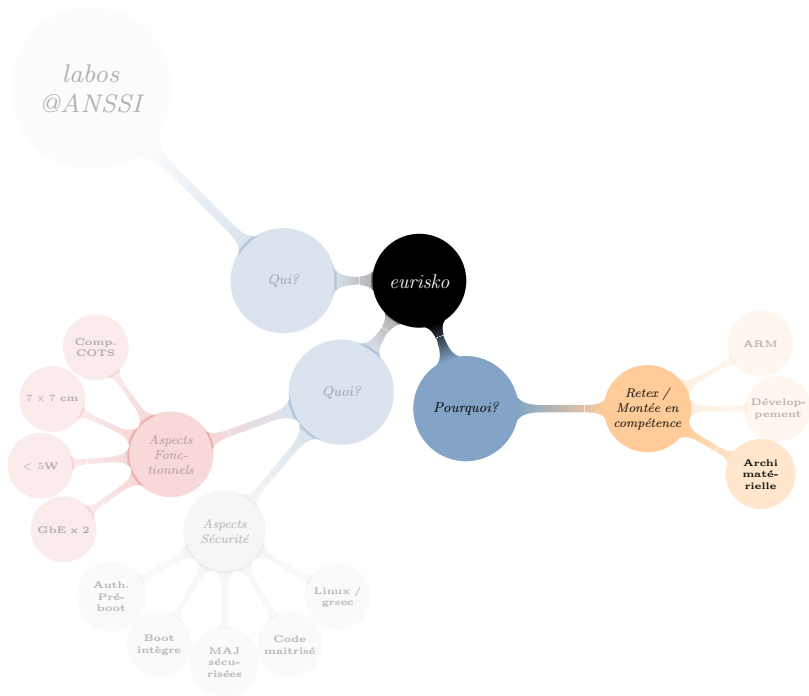


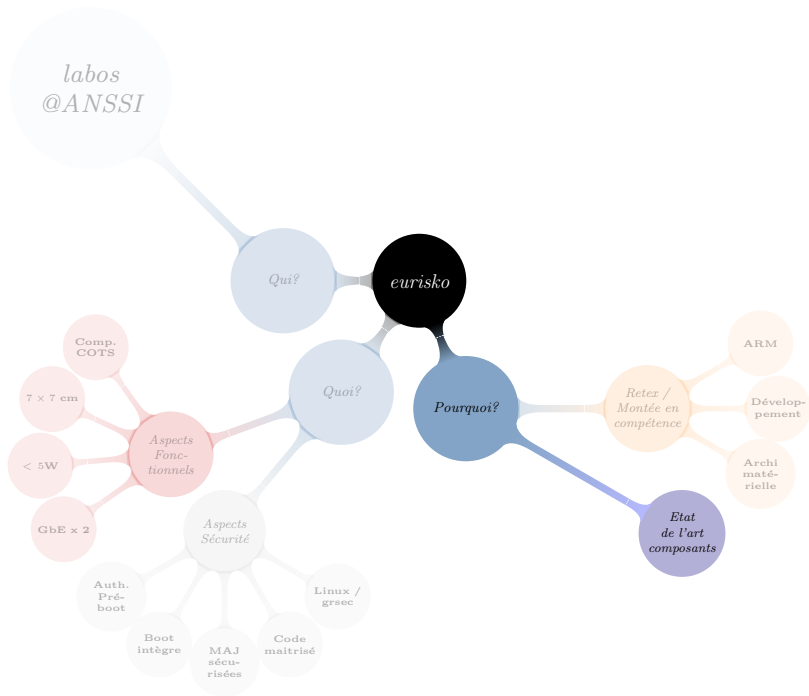


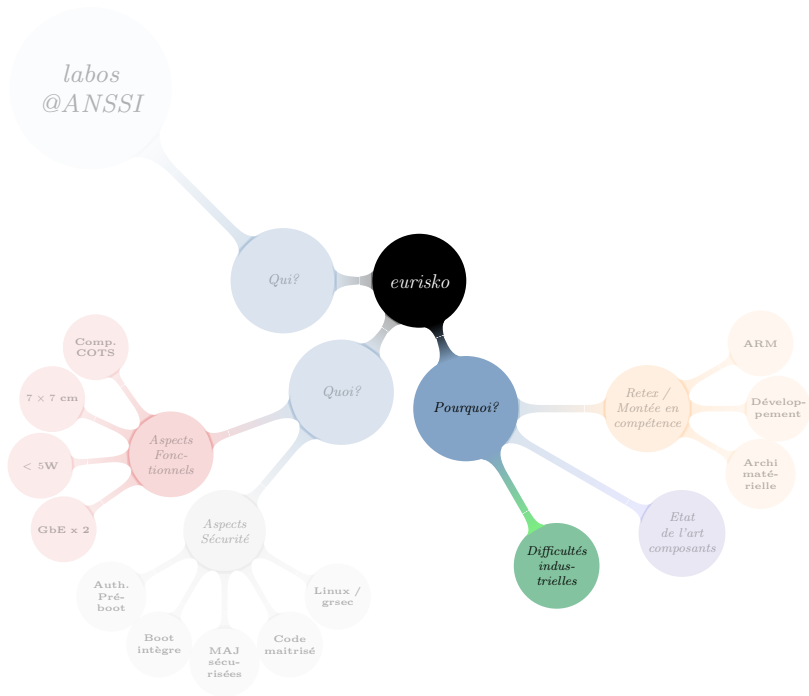


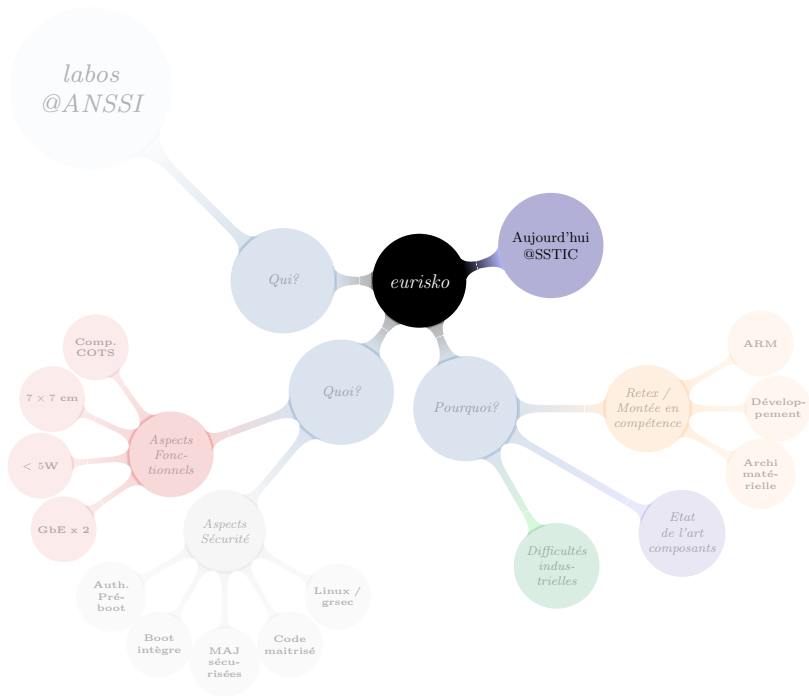


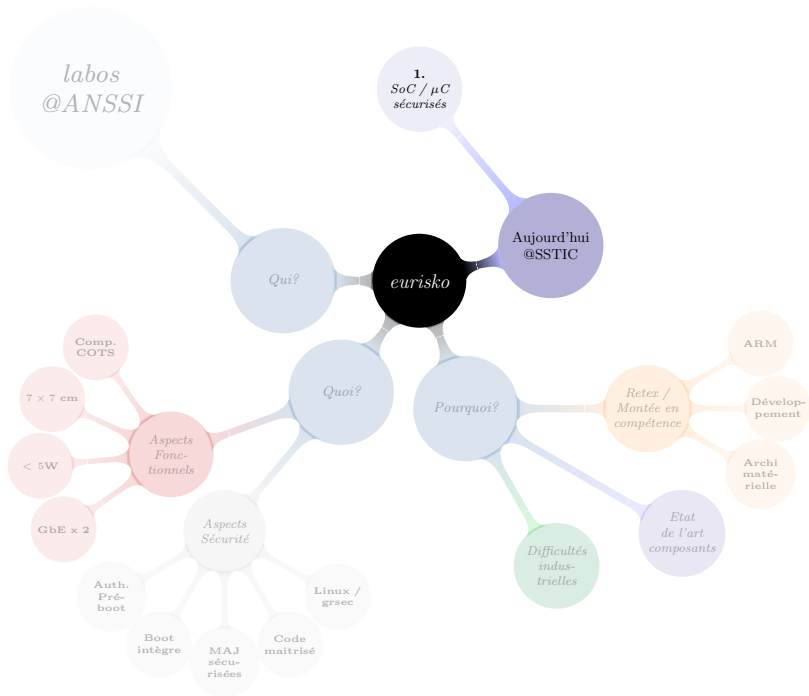


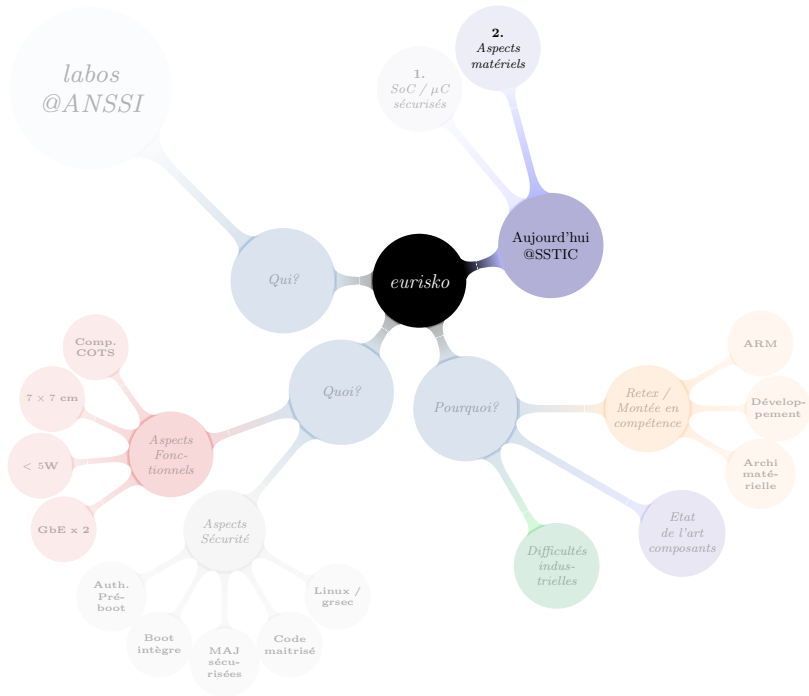


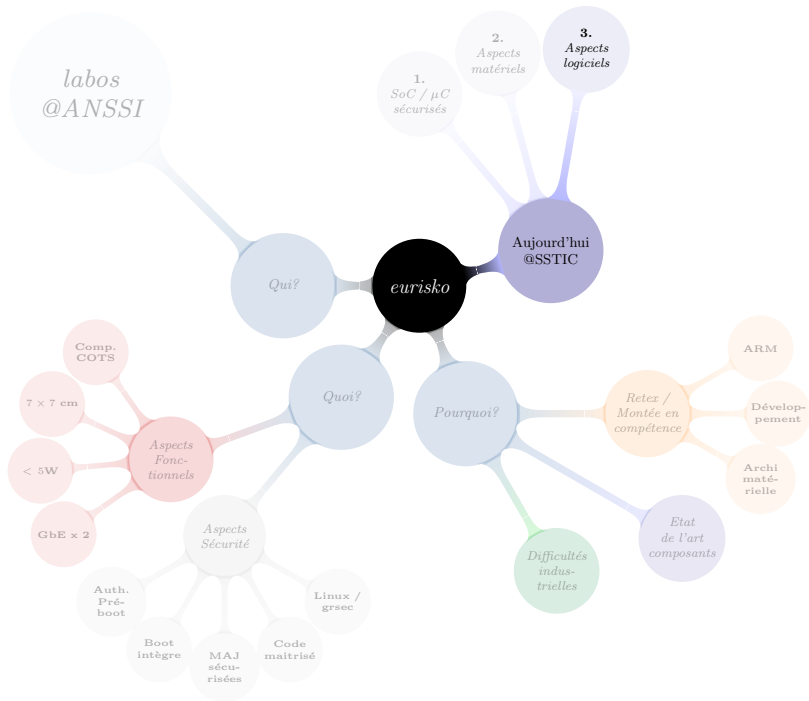


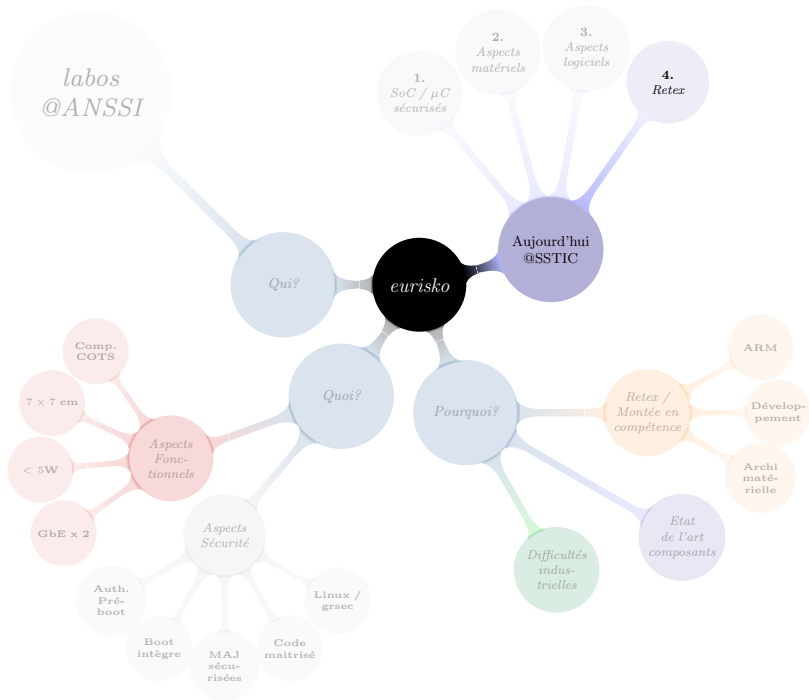


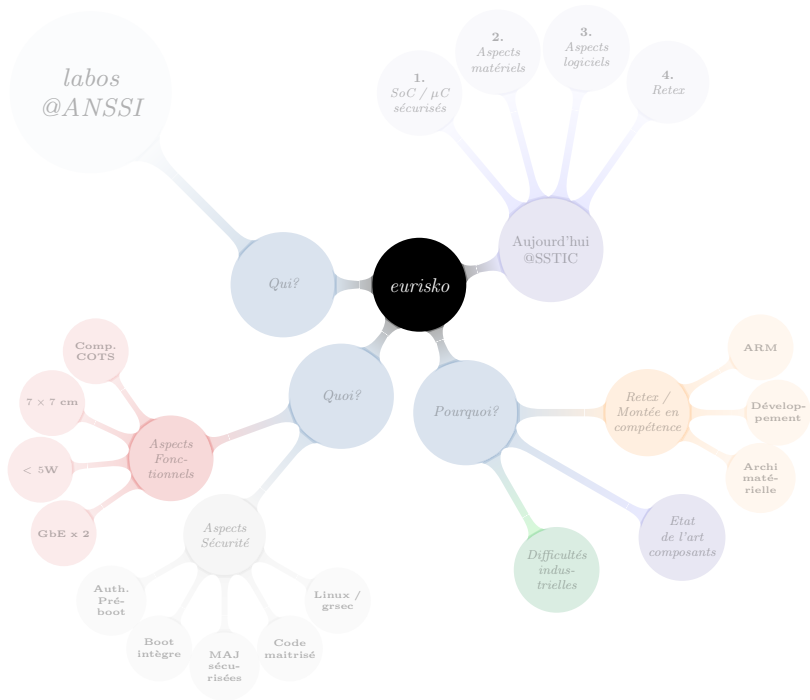


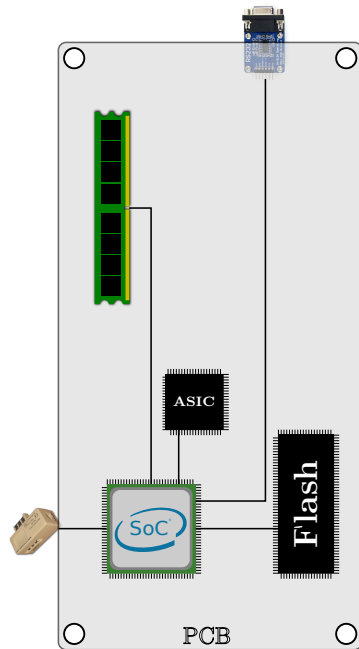


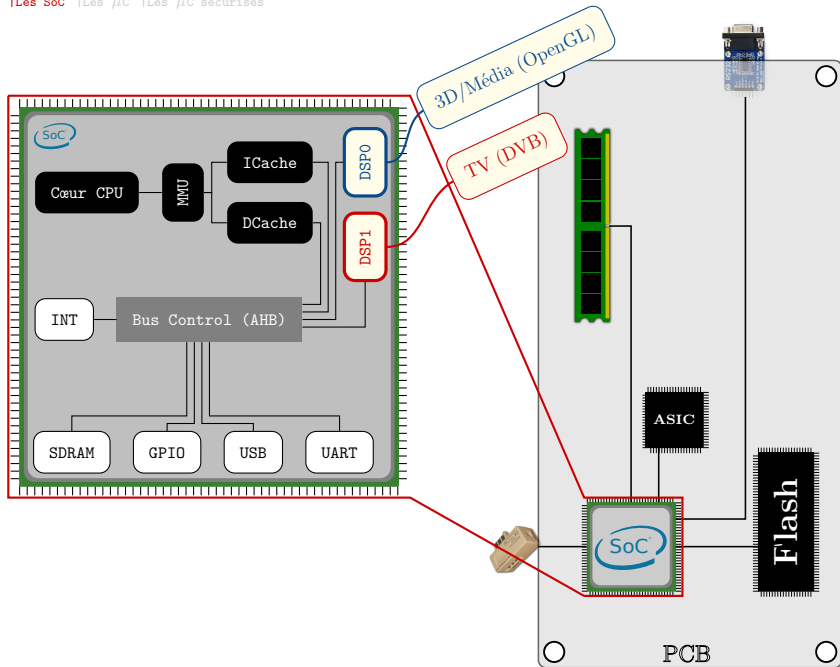


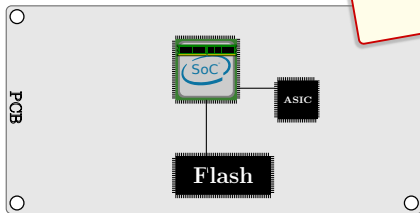




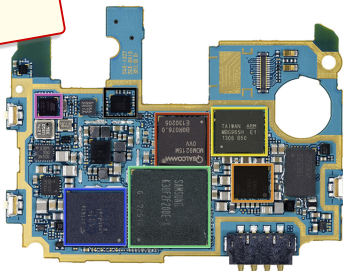


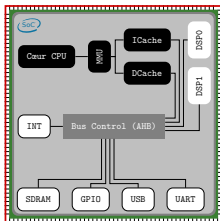




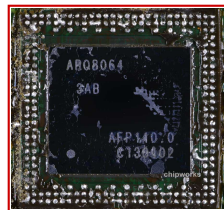


OEM
Samsung
Galaxy S4

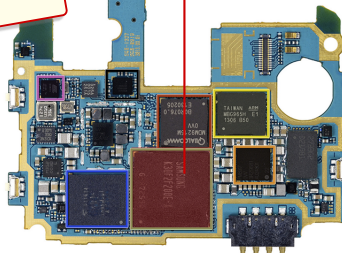
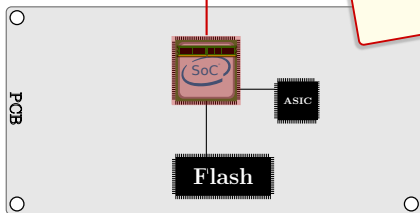


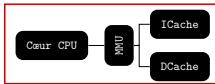


Fabriqueur SoC
Qualcomm
Snapdragon 600

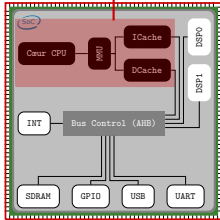


OEM
Samsung
Galaxy S4

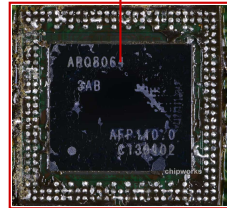




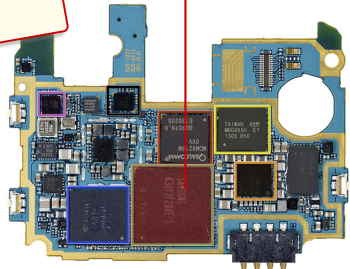
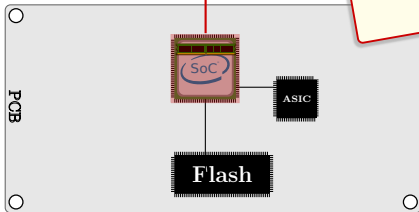
IP Cœur
ARM/Qualcomm
Krait 300



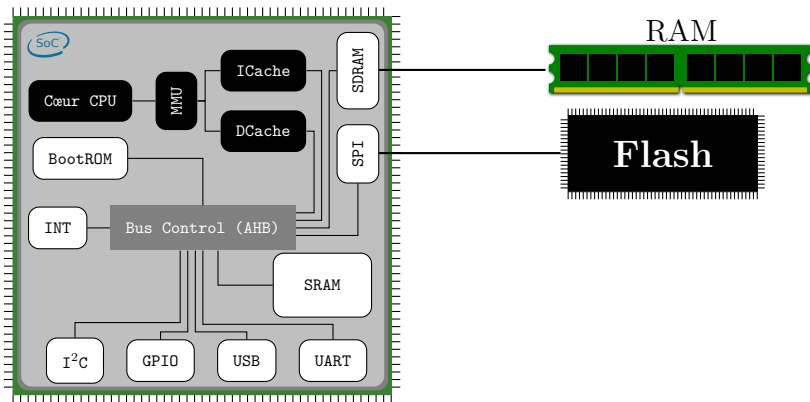
Fabriqueur SoC
Qualcomm
Snapdragon 600



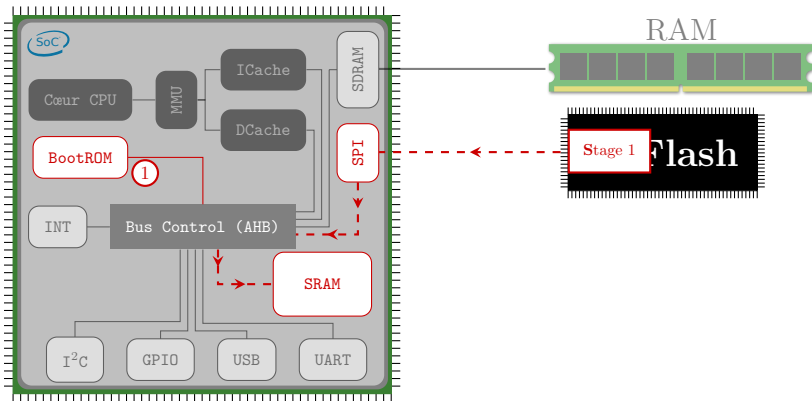
OEM
Samsung
Galaxy S4



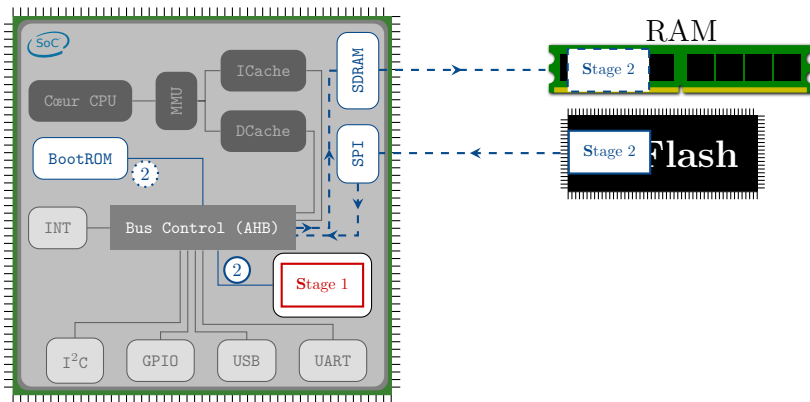
Chaîne de démarrage des SoC



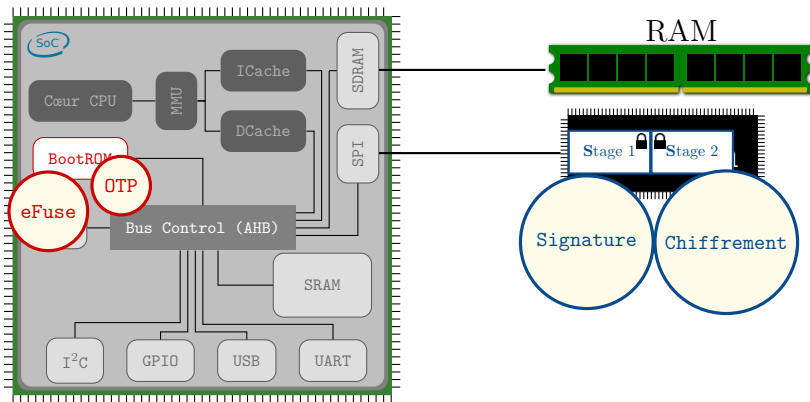
Chaîne de démarrage des SoC



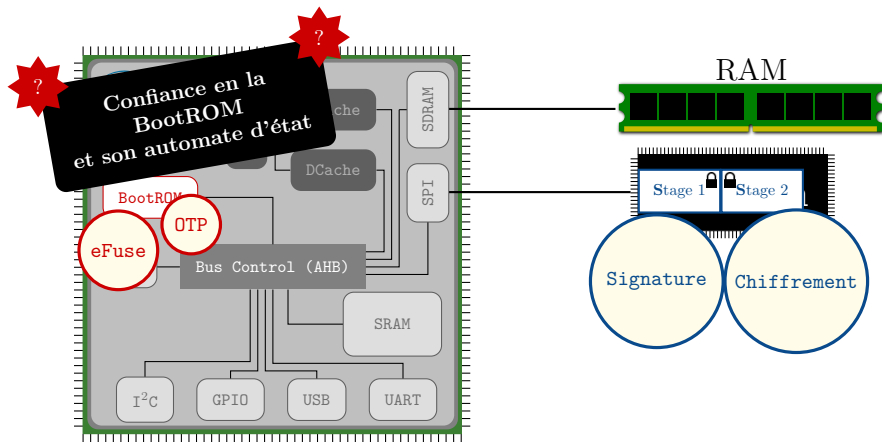
Chaîne de démarrage des SoC



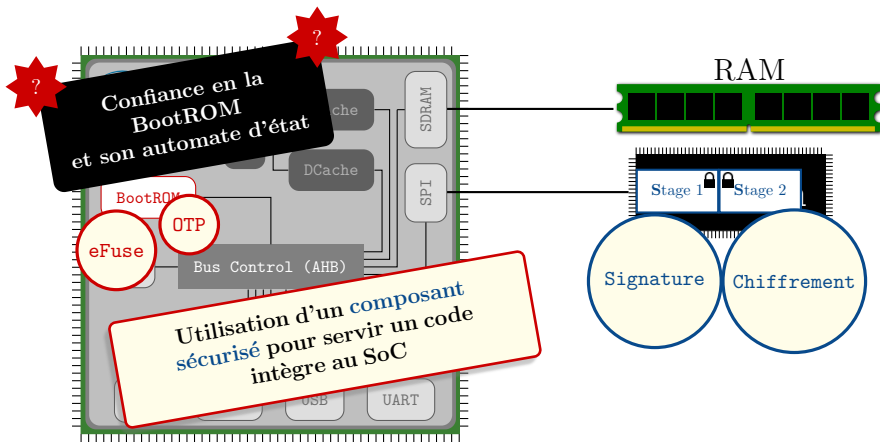
Chaîne de démarrage des SoC

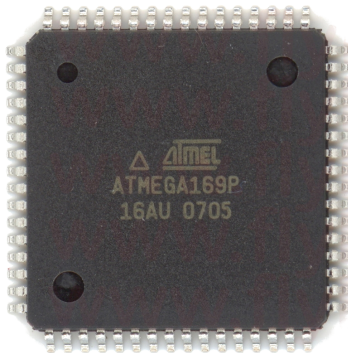


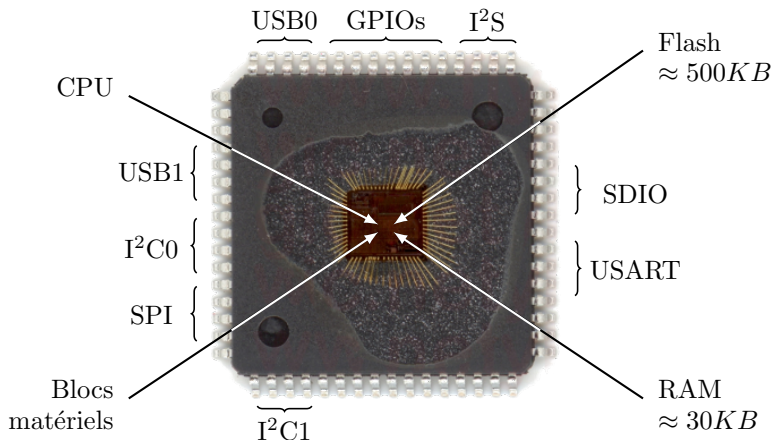
Chaîne de démarrage des SoC

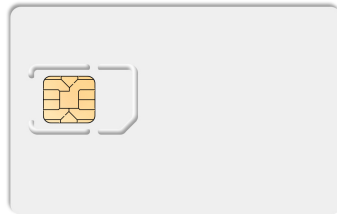


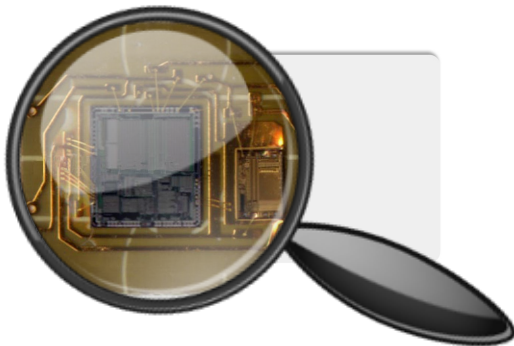
Chaîne de démarrage des SoC

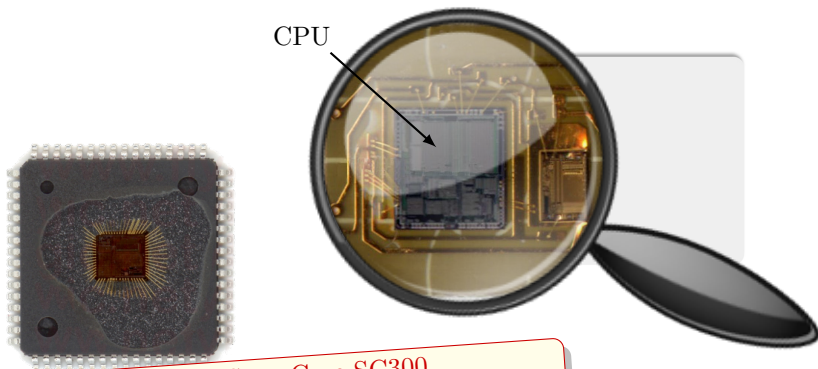




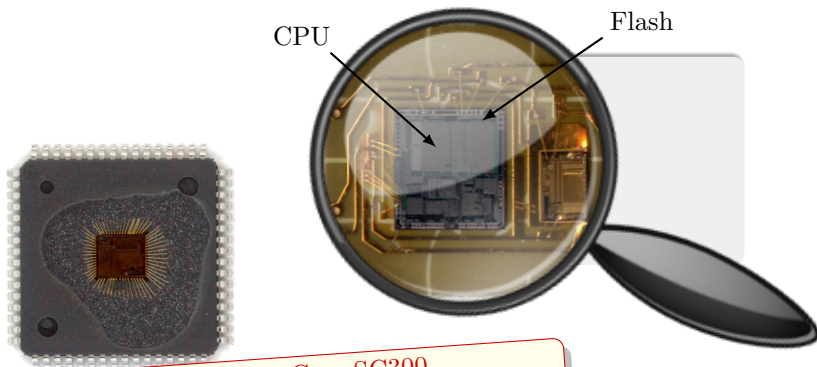




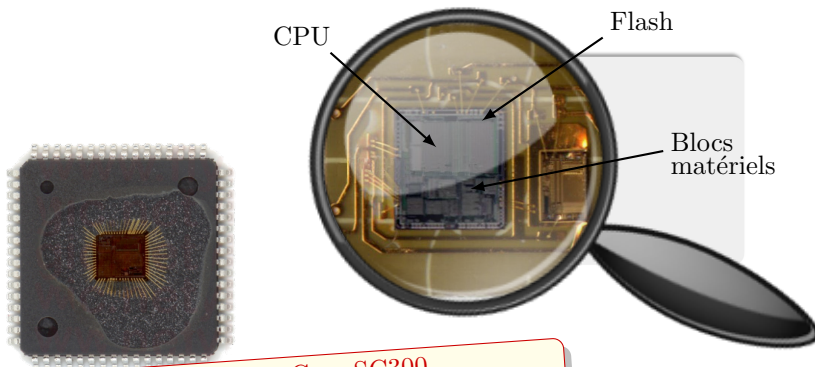




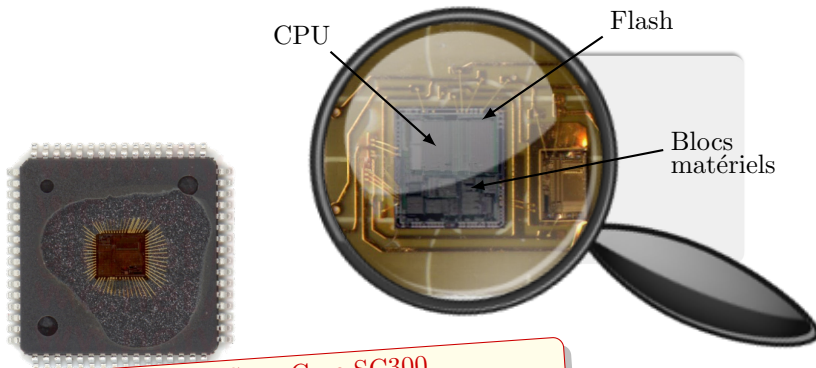
ARM SecurCore SC300



ARM SecurCore SC300
Flash protégée en confidentialité et intégrité



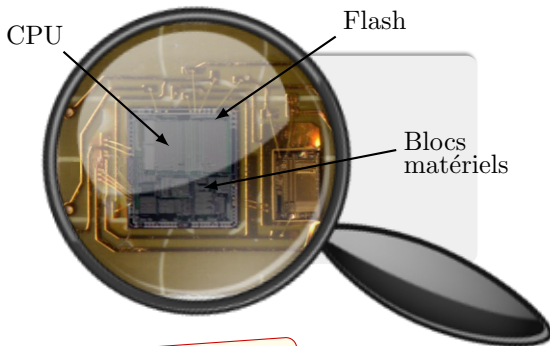
ARM SecurCore SC300
Flash protégée en confidentialité et intégrité
Générateur de nombres aléatoires
Accélérateurs cryptographiques



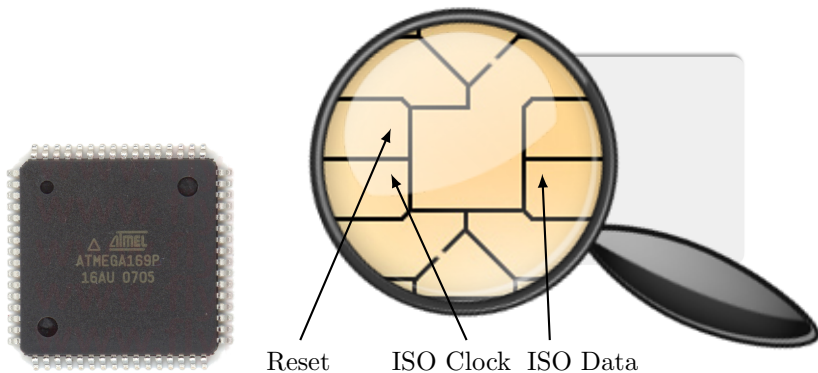
ARM **SecurCore SC300**
Flash protégée en **confidentialité** et **intégrité**
Générateur de **nombres aléatoires**
Accélérateurs cryptographiques
Détection des attaques par **probing** ou
modification de signaux sur le circuit
Protection contre les fautes (**anti-DPA**)

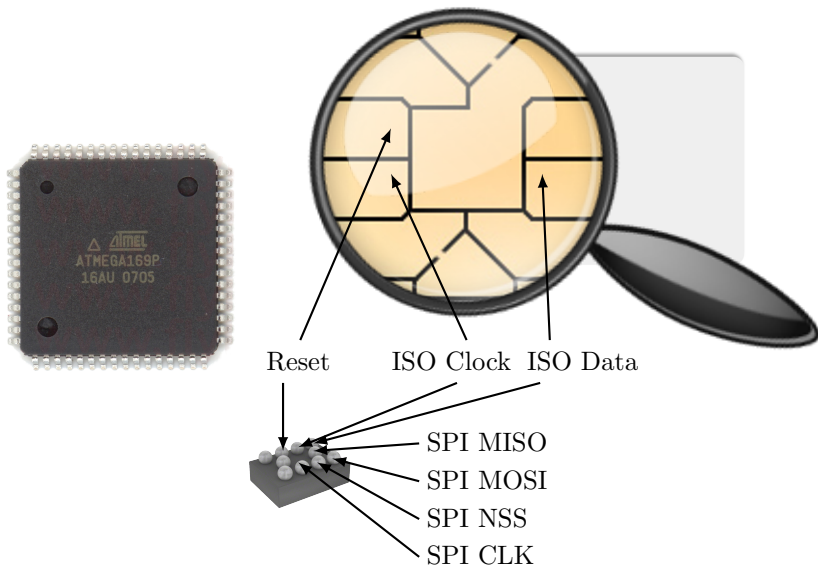


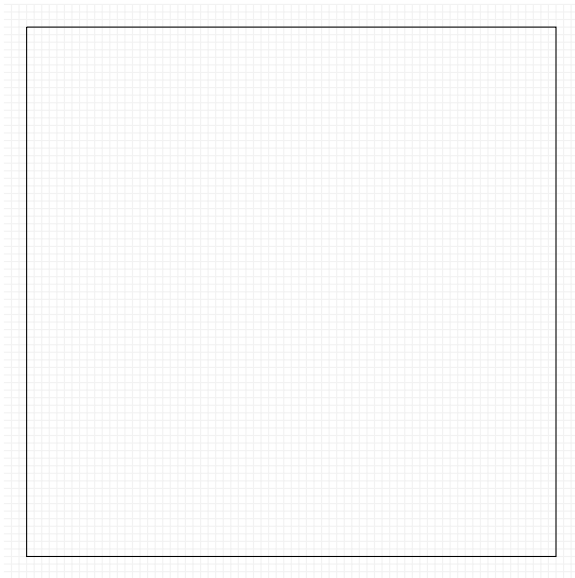
COMMON CRITERIA
CERTIFIED
EAL 5+

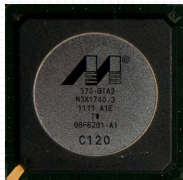


ARM SecurCore SC300
Flash protégée en confidentialité et intégrité
Générateur de nombres aléatoires
Accélérateurs cryptographiques
Détection des attaques par probing ou
modification de signaux sur le circuit
Protection contre les fautes (anti-DPA)



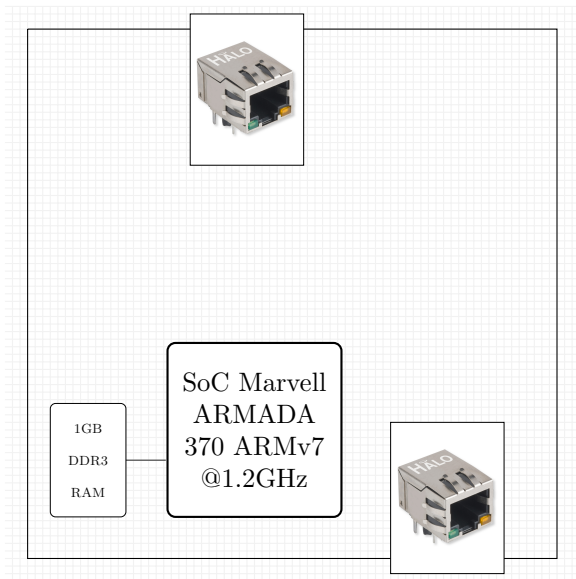


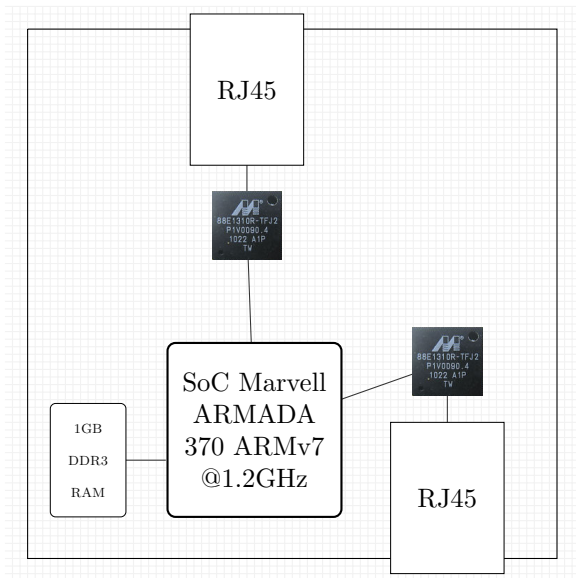


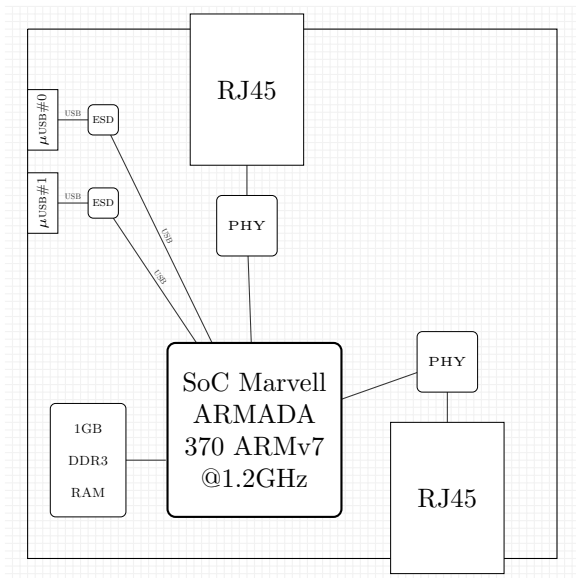


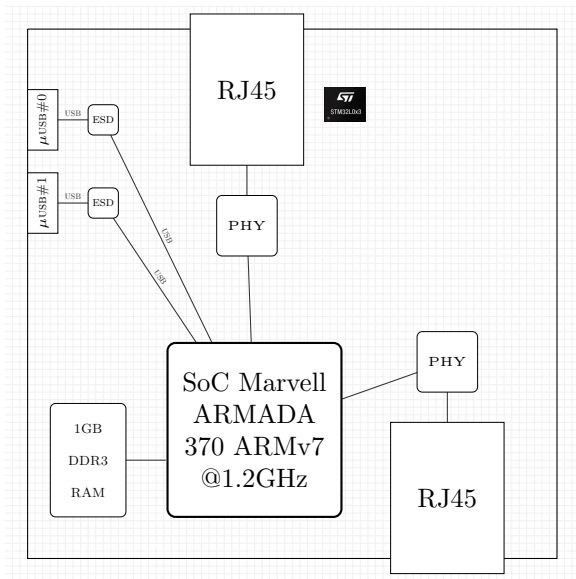


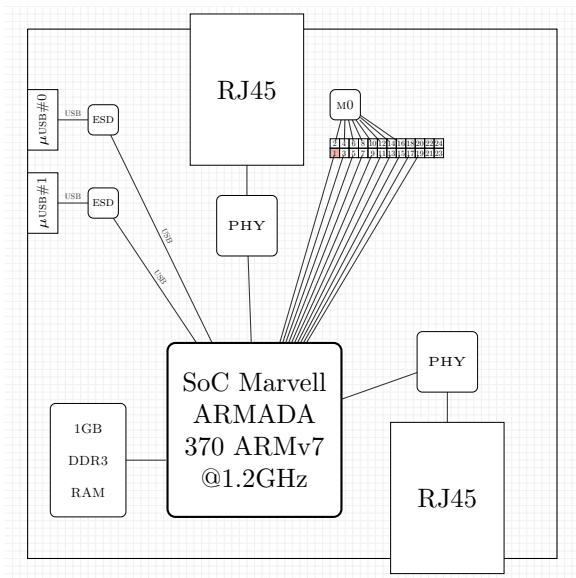
SoC Marvell
ARMADA
370 ARMv7
@1.2GHz

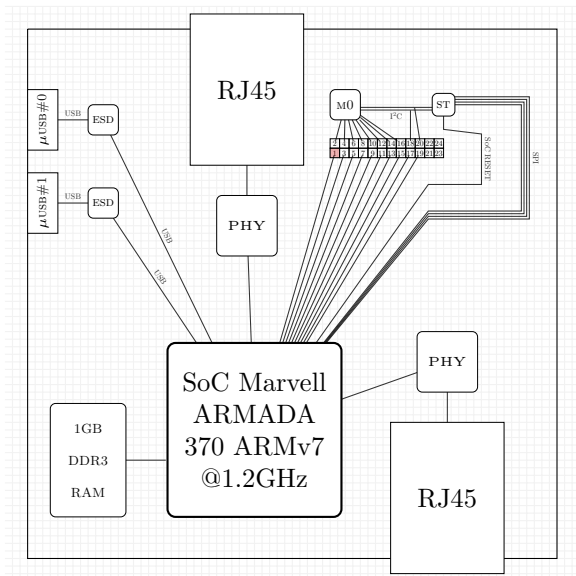


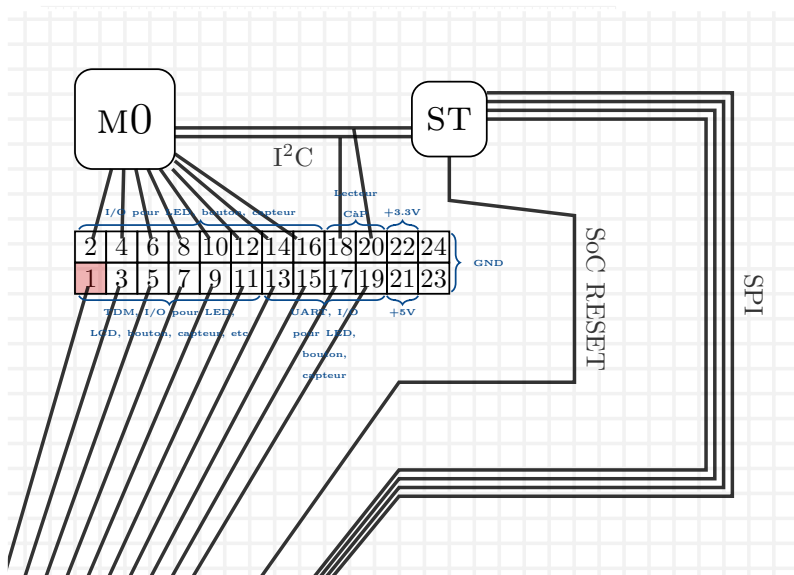


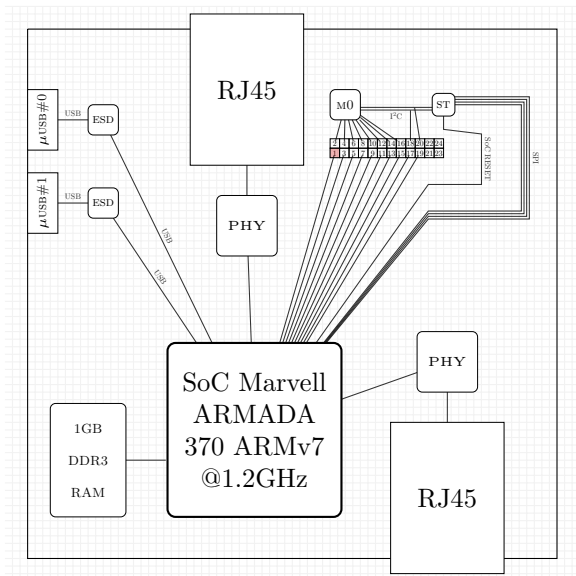


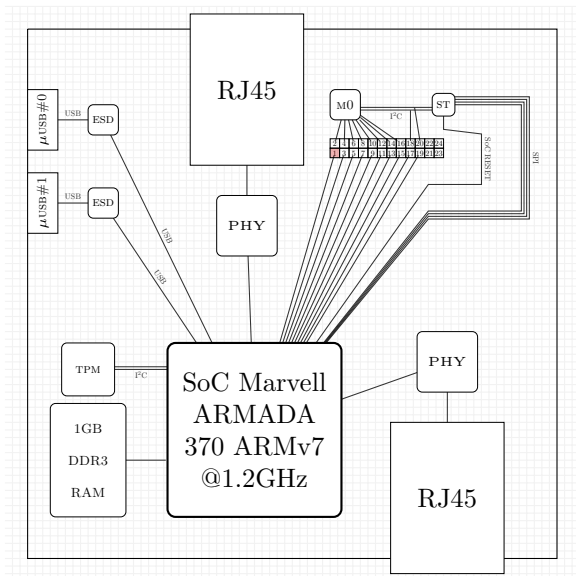


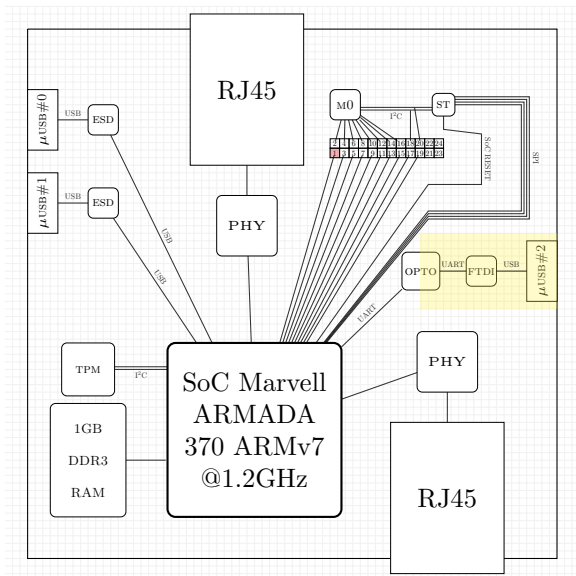


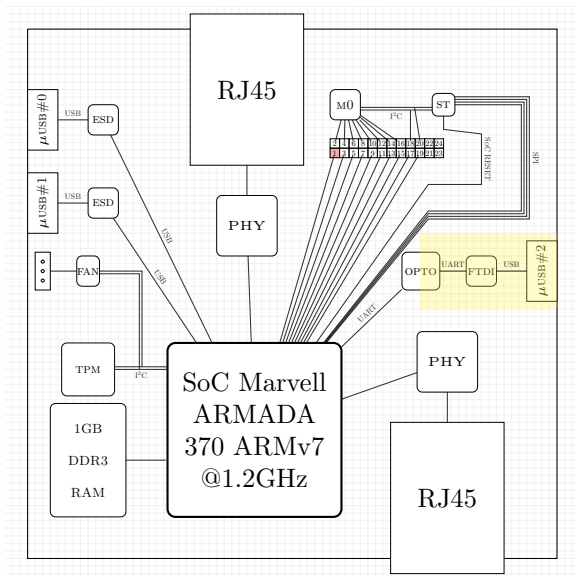


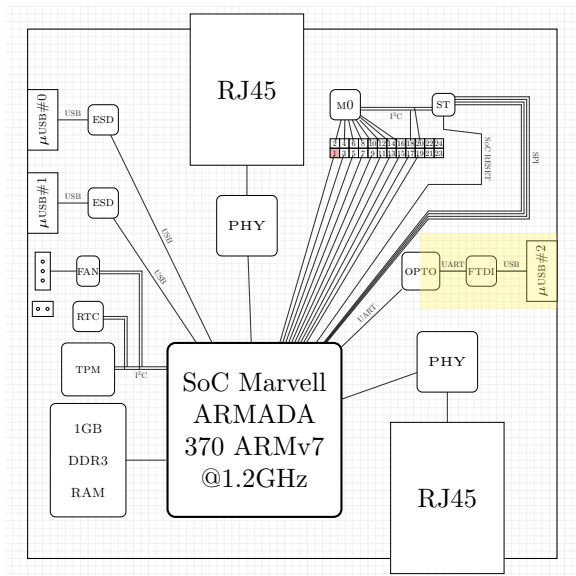






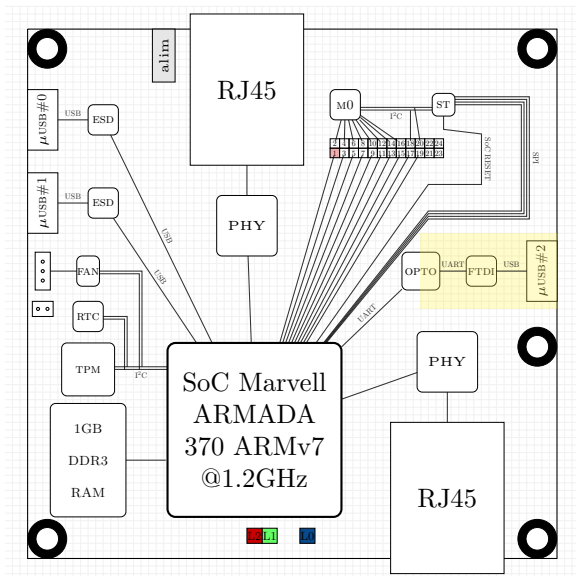


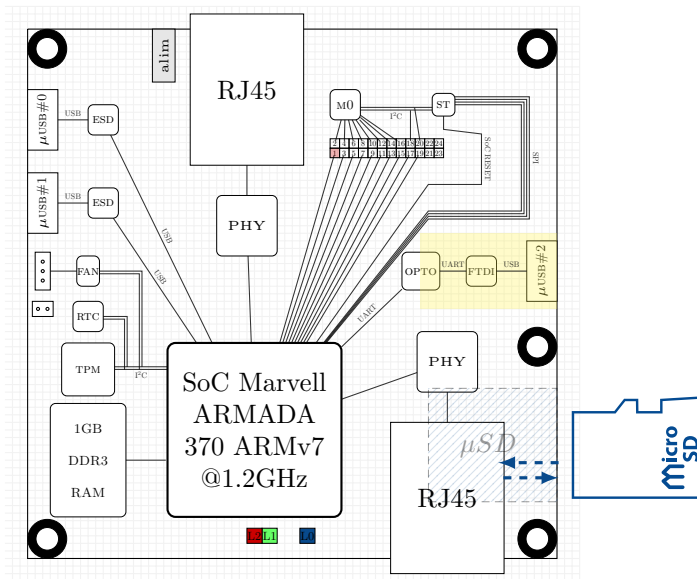


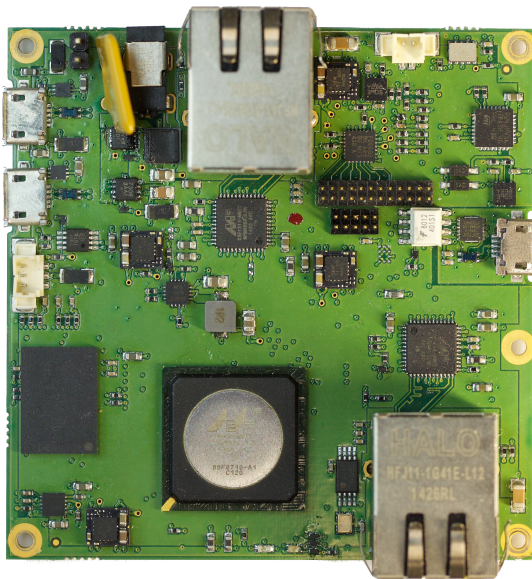


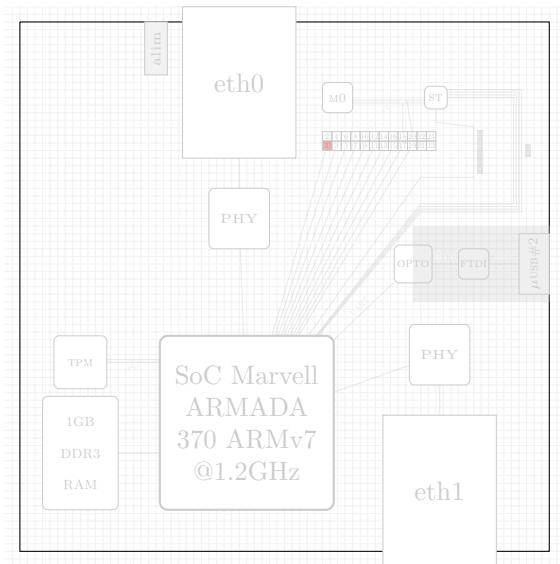


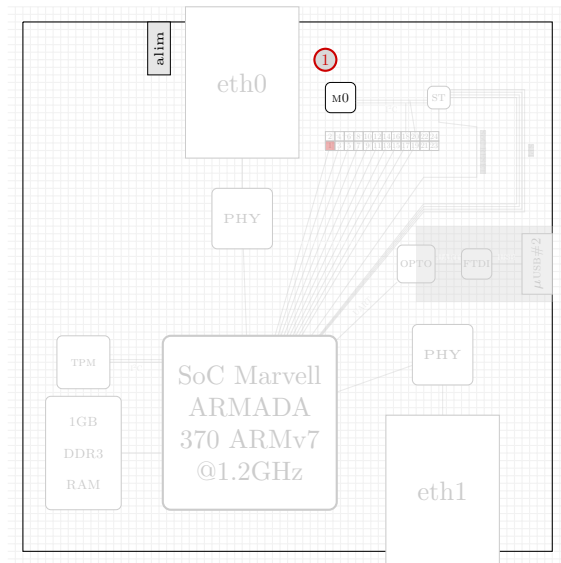






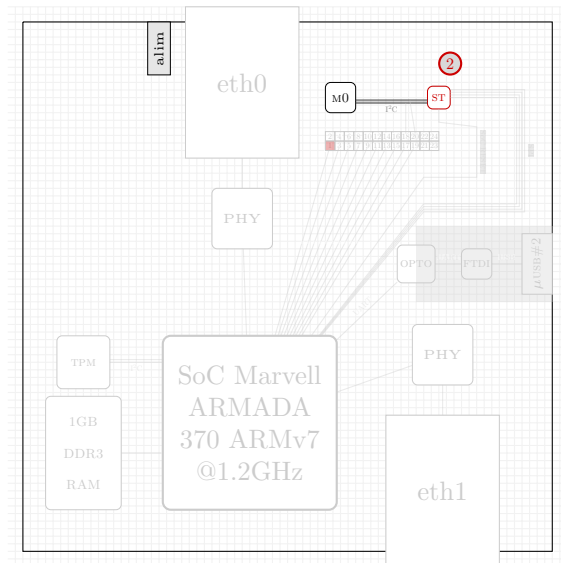






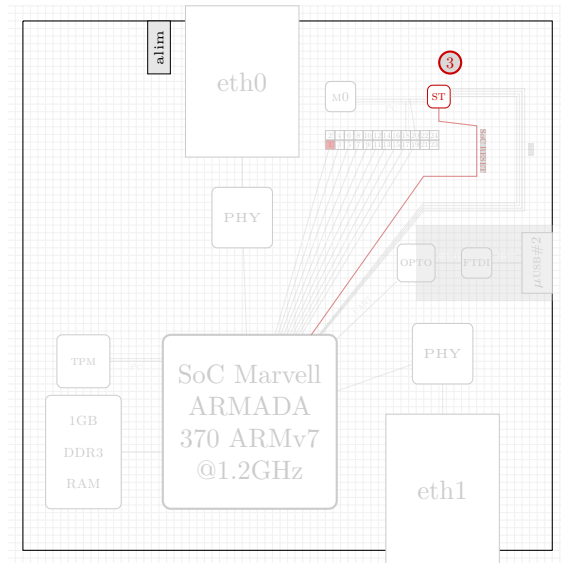
Logique de boot

- ① Alimentation/démarrage du M0



Logique de boot

- ① Alimentation/démarrage du M0
- ② Alimentation/démarrage du ST33



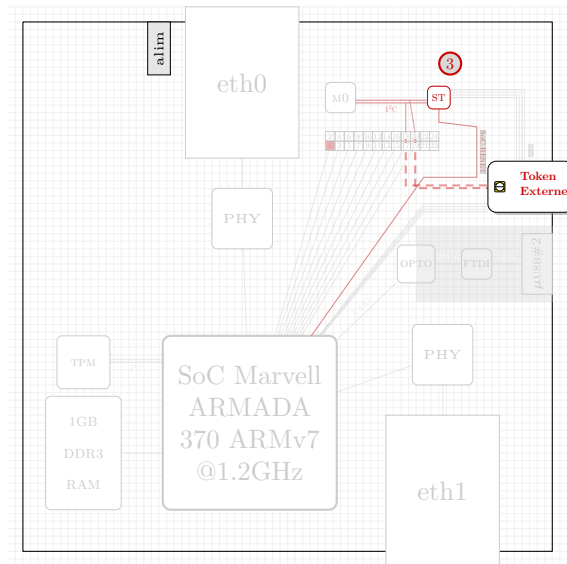
Logique de boot

① Alimentation/démarrage du M0

② Alimentation/démarrage du ST33

③

- SoC maintenu sous *reset* par le ST33
- Vérifications par le ST33 avant levée du *reset* : intégrité

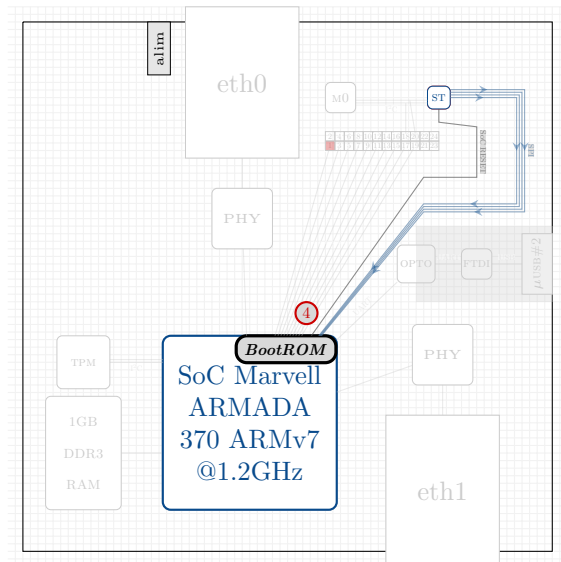


Logique de boot

① Alimentation/démarrage du M0

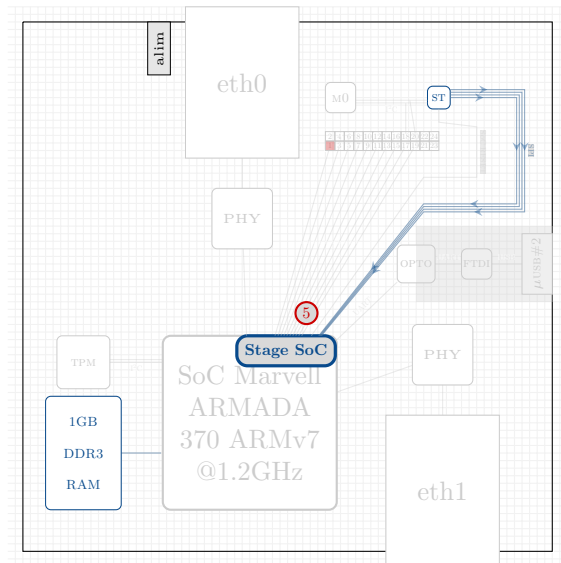
② Alimentation/démarrage du ST33

- ③
- SoC maintenu sous *reset* par le ST33
 - Vérifications par le ST33 avant levée du *reset* : intégrité, *authentification pré-boot*



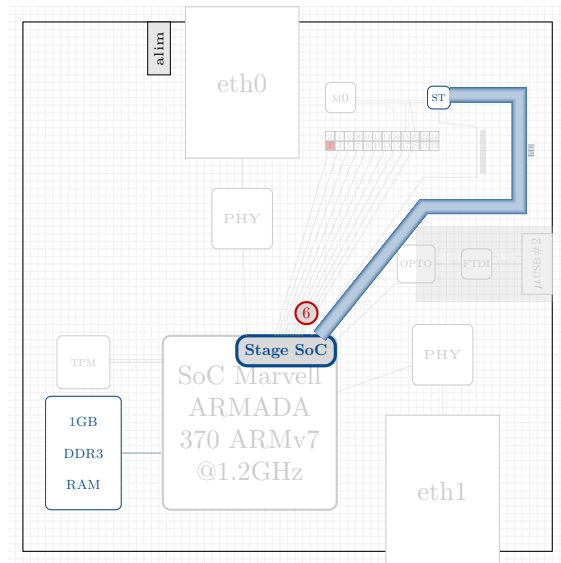
Logique de boot

- ① Alimentation/démarrage du M0
- ② Alimentation/démarrage du ST33
- ③
 - SoC maintenu sous *reset* par le ST33
 - Vérifications par le ST33 avant levée du *reset* : intégrité, *authentification pré-boot*
- ④ Envoi *Stage SoC* via le bus SPI



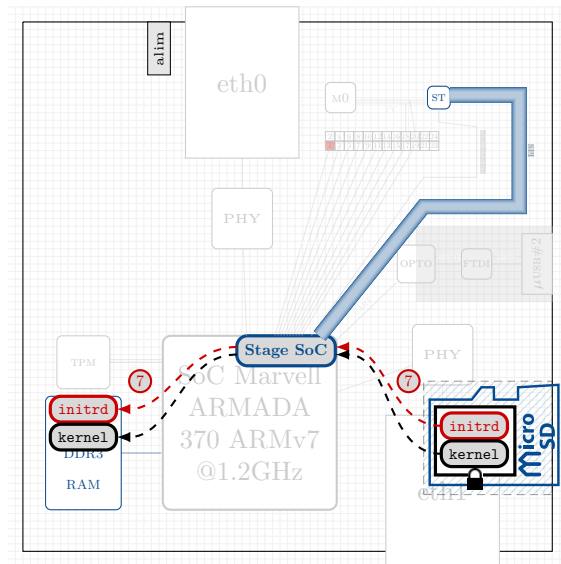
Logique de boot

- ① Alimentation/démarrage du M0
- ② Alimentation/démarrage du ST33
- ③
 - SoC maintenu sous *reset* par le ST33
 - Vérifications par le ST33 avant levée du *reset* : intégrité, *authentification pré-boot*
- ④ Envoi *Stage SoC* via le bus SPI
- ⑤ Exécution par la *bootROM*



Logique de boot

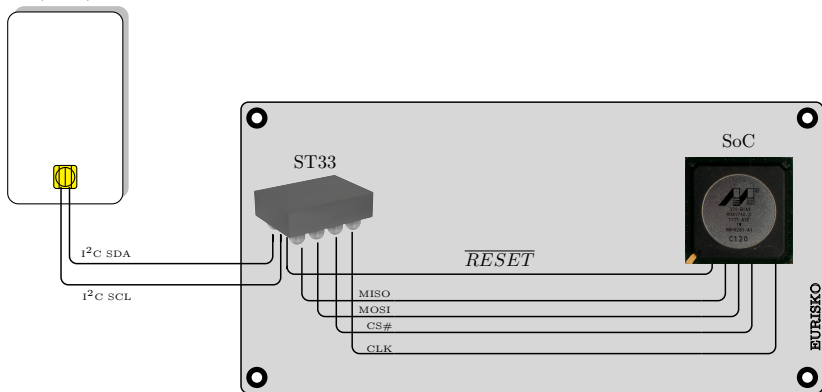
- ① Alimentation/démarrage du M0
- ② Alimentation/démarrage du ST33
- ③
 - SoC maintenu sous *reset* par le ST33
 - Vérifications par le ST33 avant levée du *reset* : intégrité, *authentification* *pré-boot*
- ④ Envoi *Stage SoC* via le bus SPI
- ⑤ Exécution par la *bootROM*
- ⑥ Négociation *canal sécurisé* ST33↔SoC



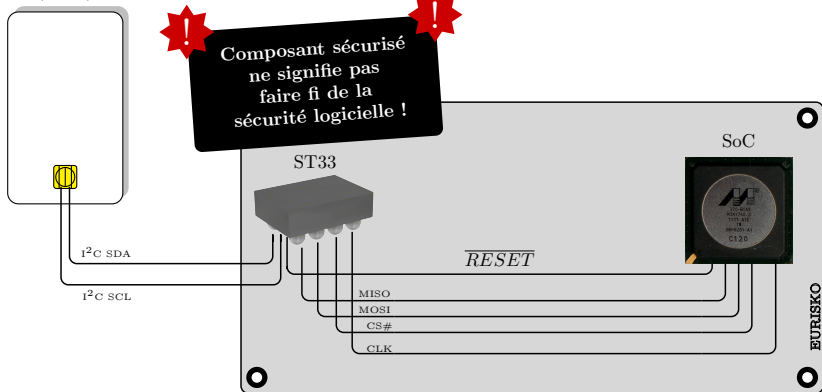
Logique de boot

- ① Alimentation/démarrage du M0
- ② Alimentation/démarrage du ST33
- ③
 - SoC maintenu sous *reset* par le ST33
 - Vérifications par le ST33 avant levée du *reset* : intégrité, *authentification* *pré-boot*
- ④ Envoi *Stage SoC* via le bus SPI
- ⑤ Exécution par la *bootROM*
- ⑥ Négociation *canal sécurisé* ST33↔SoC
- ⑦ Déverrouillage de la plateforme :
 - Reconstruction de la *clé de déchiffrement* du second *stage/noyau linux*
 - Déchiffrement en RAM depuis la carte SD et exécution

token extractible
(ST33)



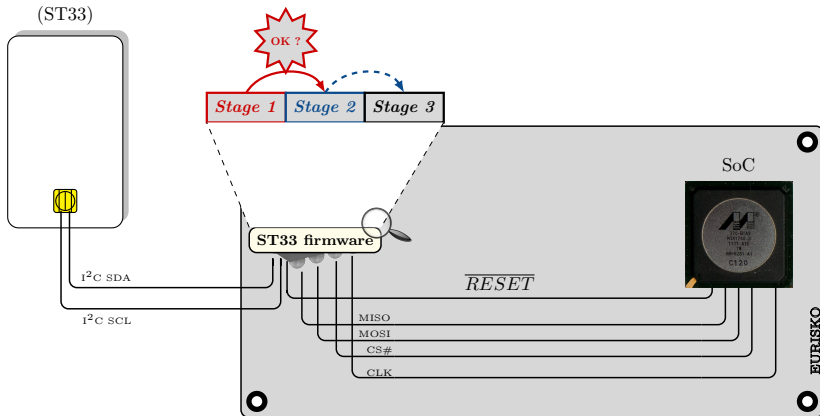
token extractible
(ST33)



Défense en profondeur OS

ST33 : démarrage *bare metal* en **plusieurs stages** qui se vérifient les uns les autres

token extractible
(ST33)

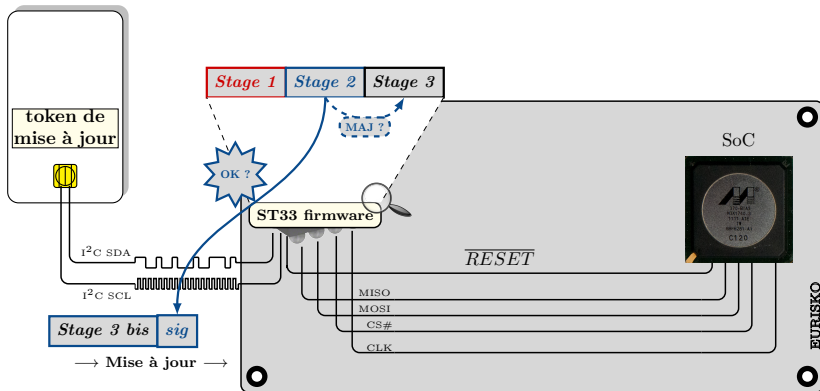


Défense en profondeur OS

ST33 : démarrage *bare metal* en **plusieurs stages** qui se vérifient les uns les autres

ST33 : mises à jour signées et gérées par des *stages* dédiés

token extractible
(ST33)



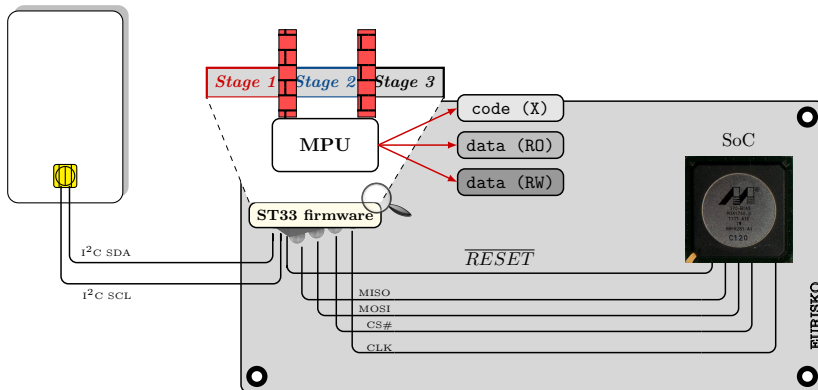
Défense en profondeur OS

ST33 : démarrage *bare metal* en **plusieurs stages** qui se vérifient les uns les autres

ST33 : mises à jour signées et gérées par des *stages* dédiés

ST33 : utilisation de la MPU du SC300 pour faire du $W \oplus X$ et protéger les *stages*

token extractible
(ST33)



Défense en profondeur OS

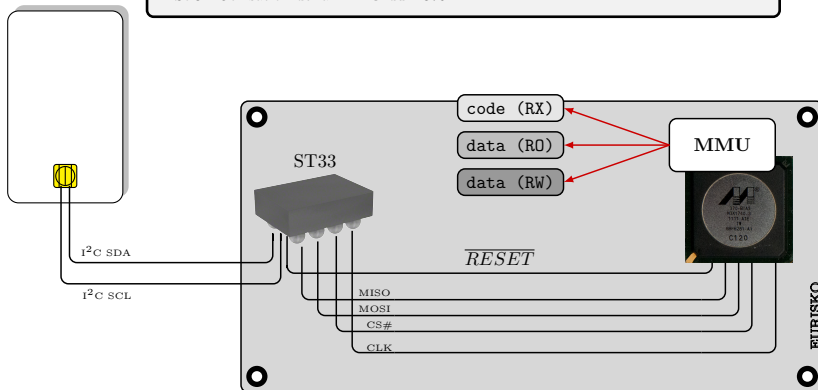
ST33 : démarrage *bare metal* en **plusieurs stages** qui se vérifient les uns les autres

ST33 : mises à jour signées et gérées par des *stages* dédiés

ST33 : utilisation de la **MPU** du SC300 pour faire du **W $\oplus$ X** et protéger les *stages*

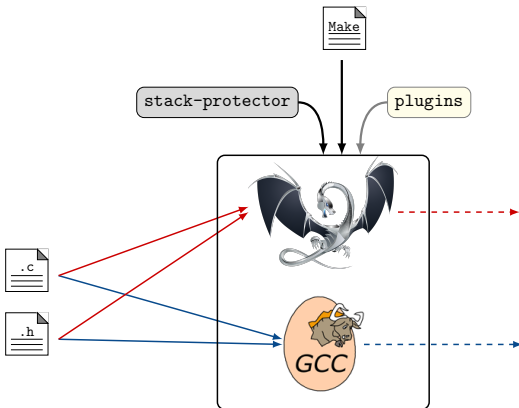
SoC : Utilisation de la **MMU** du A370

token extractible
(ST33)



Maîtrise et protection du code

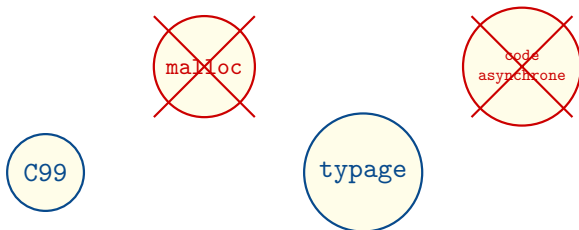
Utilisation de **toolchains** diverses et maîtrisées (gcc, clang, ...)



Maîtrise et protection du code

Utilisation de **toolchains** diverses et maîtrisées (gcc, clang, ...)

Règles de code : C99, pas d'allocation dynamique, **typage** du mieux que possible, code **linéaire/synchrone**, ...

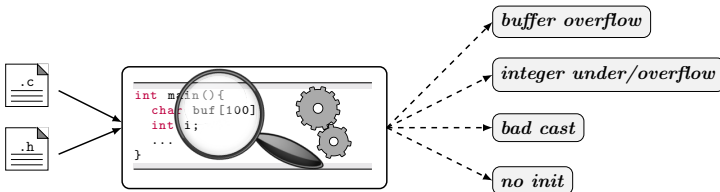


Maîtrise et protection du code

Utilisation de **toolchains** diverses et maîtrisées (gcc, clang, ...)

Règles de code : C99, pas d'allocation dynamique, **typage** du mieux que possible, code **linéaire/synchrone**, ...

Utilisation d'outils d'analyse statique de code



Montage de canaux sécurisés

Authentification mutuelle sur I²C et SPI

token extractible
 (ST33)

