OpenPGP et GnuPG avancés

Damien GOUTTE-GATTAT

Ateliers CLI – L@Bx

9 février 2016

PGP, OpenPGP, GnuPG

PGP

Pretty Good Privacy, écrit en 1991 par Phil Zimmerman (aujourd'hui *Symantec PGP* ou *Symantec Encryption Desktop*)

OpenPGP

Standard décrivant le fonctionnement de PGP, écrit en 1998 (RFC 2440), raffiné en 2007 (RFC 4880)

GnuPG

Gnu Privacy Guard (gpg), implémentation libre d'OpenPGP, écrit à partir de 1999 par Werner Koch

Sommaire

Versions de GnuPG

- GnuPG 1.4
  - Pour les vieux systèmes et les serveurs
  - Pour la compatibilité avec PGP 2.6 et les clefs V3
- GnuPG 2.0
  - Ancienne branche principale pour les bureaux
  - Bientôt abandonnée
- GnuPG 2.1
  - Branche principale pour les bureaux modernes
  - Toutes les nouvelles fonctionnalités sont dans cette branche :
    - ★ Cryptographie ECC
    - ★ Modèle de confiance TOFU
    - ★ Support de DANE

Vous devriez utiliser GnuPG 2.1.

Quelle version utilisez-vous habituellement ?

```
$ gpg --version
gpg (GnuPG) 1.4.19
Copyright (C) 2015 Free Software Foundation, Inc.

$ gpg2 --version
gpg (GnuPG) 2.1.10
libgcrypt 1.6.4
Copyright (C) 2015 Free Software Foundation, Inc.
```

Variations entre les distributions

- Debian, Fedora, Slackware...
  - `gpg` est GnuPG 1.4
  - `gpg2` est GnuPG 2.x
- OpenSuSE
  - `gpg1` est GnuPG 1.4
  - `gpg` est GnuPG 2.x
- Archlinux
  - `gpg` et `gpg2` sont GnuPG 2.x

Environnement de test

Dossier de travail de GnuPG

```
$ gpg --version
[...]
Home: ~/.gnupg

$ ls ~/.gnupg
S.gpg-agent  private-keys-v1.d  random_seed  trustdb.gpg
gpg.conf     pubring.kbx        tofu.d
```

Utiliser un autre dossier de travail

- Option `--homedir`
- Variable d'environnement `GNUPGHOME`

```
$ mkdir /tmp/test
$ chmod 0700 /tmp/test
$ export GNUPGHOME=/tmp/test
$ gpg [...]
```

Sommaire

Le problème

Anatomie d'une clef OpenPGP

- Une clef publique brute (*key material*)
- Une ou plusieurs *identité(s)*
  de la forme "Nom (Commentaire) `<adresse email>`"

- Génération de clefs complètement décentralisée
- N'importe qui peut générer une clef avec n'importe quelle identité

La cryptographie à clef publique remplace le problème de l'échange sécurisé des clefs secrètes par celui de l'*authentification* des clefs publiques.

Notion de validité

Définition

- Degré de certitude qu'une clef publique et une identité sont associées
- a.k.a *trust*, *calculated trust*

Valeurs possibles

- Validité inconnue (*unknown*)
- Validité marginale (*marginal*)
- Validité complète (*full*)
- Validité ultime (*ultimate*)

Pas d'informations sur l'appartenance de la clef

Notion de validité

Définition

- Degré de certitude qu'une clef publique et une identité sont associées
- a.k.a *trust*, *calculated trust*

Valeurs possibles

- Validité inconnue (*unknown*)
- Validité marginale (*marginal*)
- Validité complète (*full*)
- Validité ultime (*ultimate*)

Quelques raisons de penser que la clef appartient bien à son propriétaire proclamé

Notion de validité

Définition

- Degré de certitude qu'une clef publique et une identité sont associées
- a.k.a *trust*, *calculated trust*

Valeurs possibles

- Validité inconnue (*unknown*)
- Validité marginale (*marginal*)
- Validité complète (*full*)
- Validité ultime (*ultimate*)

Certitude que la clef appartient bien à son propriétaire proclamé

Notion de validité

Définition

- Degré de certitude qu'une clef publique et une identité sont associées
- a.k.a *trust*, *calculated trust*

Valeurs possibles

- Validité inconnue (*unknown*)
- Validité marginale (*marginal*)
- Validité complète (*full*)
- Validité ultime (*ultimate*)

Valeur spéciale réservée à la propre clef de l'utilisateur

Notion de validité

```
$ gpg2 -k microcheap
pub   rsa4096/BC0A9DD1 2014-08-22 [SC]
uid          [ unknown] MicroCheapFx <fx@microcheap.info>
uid          [  full  ] MicroCheapFx <microcheapfx@microcheap.info>
uid          [marginal] François-Xavier Lesaffre <fxlesaffre@ovh.fr>
uid          [ unknown] [jpeg image of size 15587]
sub   rsa4096/206367B1 2014-08-22 [E]
```

(GnuPG < 2.1 : `--list-options show-uid-validity`)

Notion de validité

```
$ gpg2 -k microcheap
pub   rsa4096/BC0A9DD1 2014-08-22 [SC]
uid         [ unknown] MicroCheapFx <fx@microcheap.info>
uid         [  full  ] MicroCheapFx <microcheapfx@microcheap.info>
uid         [marginal] François-Xavier Lesaffre <fxlesaffre@ovh.fr>
uid         [ unknown] [jpeg image of size 15587]
sub   rsa4096/206367B1 2014-08-22 [E]
```

Identités non-validées

Notion de validité

```
$ gpg2 -k microcheap
pub   rsa4096/BC0A9DD1 2014-08-22 [SC]
uid          [ unknown] MicroCheapFx <fx@microcheap.info>
uid          [  full  ] MicroCheapFx <microcheapfx@microcheap.info>
uid          [marginal] François-Xavier Lesaffre <fxlesaffre@ovh.fr>
uid          [ unknown] [jpeg image of size 15587]
sub   rsa4096/206367B1 2014-08-22 [E]
```

Identité marginalement valide

Notion de validité

```
$ gpg2 -k microcheap
pub   rsa4096/BC0A9DD1 2014-08-22 [SC]
uid          [ unknown] MicroCheapFx <fx@microcheap.info>
uid          [  full  ] MicroCheapFx <microcheapfx@microcheap.info>
uid          [marginal] François-Xavier Lesaffre <fxlesaffre@ovh.fr>
uid          [ unknown] [jpeg image of size 15587]
sub   rsa4096/206367B1 2014-08-22 [E]
```

Identité pleinement valide

Notion de validité

```
$ gpg2 -k microcheap
pub   rsa4096/BC0A9DD1 2014-08-22 [SC]
uid          [ unknown] MicroCheapFx <fx@microcheap.info>
uid          [  full  ] MicroCheapFx <microcheapfx@microcheap.info>
uid          [marginal] François-Xavier Lesaffre <fxlesaffre@ovh.fr>
uid          [ unknown] [jpeg image of size 15587]
sub   rsa4096/206367B1 2014-08-22 [E]
```

La validité est

- relative à un utilisateur donné
- *calculée automatiquement* par le *modèle de confiance*

Modèle de confiance

Définition

Règles déterminant la validité d'un couple {clef publique, identité}

Plusieurs modèles possibles

- Aucun modèle imposé par le standard
- Choix du modèle laissé à chaque utilisateur
- Modèles possibles avec GnuPG :
  - Direct (`--trust-model direct`)
  - Externe (`--trust-model always`)
  - Toile de confiance (`--trust-model classic`)
  - Toile de confiance étendue (`--trust-model pgp`)
  - *Trust-on-first-use* (`--trust-model tofu`)
  - Combinaison des deux précédents (`--trust-model tofu+pgp`)

Modèle de confiance

Définition

Règles déterminant la validité d'un couple {clef publique, identité}

Plusieurs modèles possibles

- Aucun modèle imposé par le standard
- Choix du modèle laissé à chaque utilisateur
- Modèles possibles avec GnuPG :
  - Direct (`--trust-model direct`)
  - Externe (`--trust-model always`)
  - Toile de confiance (`--trust-model classic`)
  - Toile de confiance étendue (`--trust-model pgp`)
  - *Trust-on-first-use* (`--trust-model tofu`)
  - Combinaison des deux précédents (`--trust-model tofu+pgp`)

À partir de GnuPG 2.1.10

Modèle de confiance

Définition

Règles déterminant la validité d'un couple {clef publique, identité}

Plusieurs modèles possibles

- Aucun modèle imposé par le standard
- Choix du modèle laissé à chaque utilisateur
- Modèles possibles avec GnuPG :
  - Direct (`--trust-model direct`)
  - Externe (`--trust-model always`)
  - Toile de confiance (`--trust-model classic`)
  - Toile de confiance étendue (`--trust-model pgp`)
  - *Trust-on-first-use* (`--trust-model tofu`)
  - Combinaison des deux précédents (`--trust-model tofu+pgp`)

Modèle de confiance par défaut

Notion de certification

Attestation qu'une clef et une identité données sont associées

Exemple : certification de la clef de Bob par Alice

1. Concaténer :
   - la clef publique de Bob (ID 0xB4902A74)
   - son identité (e.g., « Bob <bob@example.com> »)
   - l'identifiant de la clef d'Alice (0x2EADF7D4)
   - la date de signature
2. Condenser le tout
3. Signer le condensat avec la clef privée d'Alice

« Je soussigné, Alice, *certifie* que la clef 0xB4902A74 appartient bien à Bob <bob@example.com>.
Fait le 1er janvier 1970,

(signature) »

La confiance (*ownertrust*)

Définition

- Valeur attribuée *par l'utilisateur* à chaque clef
- Définit le crédit accordé aux certifications émises par cette clef

Valeurs possibles

- Confiance indéfinie (*unknown*)
- Aucune confiance (*never*)
- Confiance marginale (*marginal*)
- Confiance complète (*full*)
- Confiance ultime (*ultimate*)

La confiance (*ownertrust*)

Définition

- Valeur attribuée *par l'utilisateur* à chaque clef
- Définit le crédit accordé aux certifications émises par cette clef

Valeurs possibles

- Confiance indéfinie (*unknown*)
- Aucune confiance (*never*)
- Confiance marginale (*marginal*)
- Confiance complète (*full*)
- Confiance ultime (*ultimate*)

Valeur par défaut

La confiance (*ownertrust*)

Définition

- Valeur attribuée *par l'utilisateur* à chaque clef
- Définit le crédit accordé aux certifications émises par cette clef

Valeurs possibles

- Confiance indéfinie (*unknown*)
- Aucune confiance (*never*)
- Confiance marginale (*marginal*)
- Confiance complète (*full*)
- Confiance ultime (*ultimate*)

Valeur spéciale réservée à la propre clef de l'utilisateur

Règles de la toile de confiance

Un couple {clef, identité} est

- *complètement valide* s'il est certifié par
  - une clef à confiance ultime, ou
  - au moins n clefs à confiance complète, ou
  - au moins m clefs à confiance marginale
- *marginalement valide* s'il est certifié par
  - entre 1 et $n-1$ clefs à confiance complète, ou
  - entre 1 et $m-1$ clefs à confiance marginale
- *à validité inconnue* dans les autres cas

Paramètres par défaut

- $n = 1$ (modifiable avec `--completes-needed`)
- $m = 3$ (modifiable avec `--marginals-needed`)

Règles de la toile de confiance

Un couple {clef, identité} est

- *complètement valide* s'il est certifié par
  - une clef à confiance ultime, ou
  - au moins n clefs à confiance complète, ou
  - au moins m clefs à confiance marginale
- *marginalement valide* s'il est certifié par
  - entre 1 et $n-1$ clefs à confiance complète, ou
  - entre 1 et $m-1$ clefs à confiance marginale
- *à validité inconnue* dans les autres cas

Paramètres par défaut

- $n = 1$ (modifiable avec `--completes-needed`)
- $m = 3$ (modifiable avec `--marginals-needed`)

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Liste et vérifie les certifications

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Auto-certifications

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Seulement une auto-certification : validité inconnue

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Certification par ma propre clef

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Une certification par une clef à confiance ultime : validité complète

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Certification par une clef tierce

Règles de la toile de confiance

```
$ gpg2 --edit-key gregory@polymorphisme.fr

pub  rsa4096/93CCE46B
     created: 2014-10-10  expires: never       usage: C
     trust: marginal      validity: full
sub  rsa4096/5B8BCFCE
     created: 2014-10-10  expires: never       usage: S
sub  rsa4096/E7E32274
     created: 2014-10-10  expires: never       usage: E
sub  rsa4096/BE4A24CF
     created: 2014-10-10  expires: never       usage: A
[  full  ] (1). Grégory Roche <gregory@polymorphisme.fr>

gpg> check
uid  Grégory Roche <gregory@polymorphisme.fr>
sig!3        93CCE46B 2014-10-10  [self-signature]
sig!         E25FBABB 2014-10-25  Damien Goutte-Gattat <dgouttegattat@incenp.org>
[...]
4 signatures not checked due to missing keys
```

Règles de la toile de confiance

```
$ gpg2 --edit-key gregory@polymorphisme.fr

pub  rsa4096/93CCE46B
     created: 2014-10-10  expires: never       usage: C
     trust: marginal      validity: full
sub  rsa4096/5B8BCFCE
     created: 2014-10-10  expires: never       usage: S
sub  rsa4096/E7E32274
     created: 2014-10-10  expires: never       usage: E
sub  rsa4096/BE4A24CF
     created: 2014-10-10  expires: never       usage: A
[  full  ] (1). Grégory Roche <gregory@polymorphisme.fr>

gpg> check
uid  Grégory Roche <gregory@polymorphisme.fr>
sig!3        93CCE46B 2014-10-10  [self-signature]
sig!         E25FBABB 2014-10-25  Damien Goutte-Gattat <dgouttegattat@incenp.org>
[...]
4 signatures not checked due to missing keys
```

Certification par ma propre clef

Règles de la toile de confiance

```
$ gpg2 --edit-key gregory@polymorphisme.fr

pub  rsa4096/93CCE46B
     created: 2014-10-10  expires: never       usage: C
     trust: marginal      validity: full
sub  rsa4096/5B8BCFCE
     created: 2014-10-10  expires: never       usage: S
sub  rsa4096/E7E32274
     created: 2014-10-10  expires: never       usage: E
sub  rsa4096/BE4A24CF
     created: 2014-10-10  expires: never       usage: A
[  full  ] (1). Grégory Roche <gregory@polymorphisme.fr>

gpg> check
uid  Grégory Roche <gregory@polymorphisme.fr>
sig!3        93CCE46B 2014-10-10  [self-signature]
sig!         E25FBABB 2014-10-25  Damien Goutte-Gattat <dgouttegattat@incenp.org>
[...]
4 signatures not checked due to missing keys
```

Donc identité complètement valide

Règles de la toile de confiance

```
$ gpg2 --edit-key gregory@polymorphisme.fr

pub  rsa4096/93CCE46B
     created: 2014-10-10  expires: never       usage: C
     trust: marginal      validity: full
sub  rsa4096/5B8BCFCE
     created: 2014-10-10  expires: never       usage: S
sub  rsa4096/E7E32274
     created: 2014-10-10  expires: never       usage: E
sub  rsa4096/BE4A24CF
     created: 2014-10-10  expires: never       usage: A
[  full  ] (1). Grégory Roche <gregory@polymorphisme.fr>

gpg> check
uid  Grégory Roche <gregory@polymorphisme.fr>
sig!3        93CCE46B 2014-10-10  [self-signature]
sig!         E25FBABB 2014-10-25  Damien Goutte-Gattat <dgouttegattat@incenp.org>
[...]
4 signatures not checked due to missing keys
```

Confiance assignée à cette clef : marginale

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Une certification par une clef à confiance marginale

Règles de la toile de confiance

```
$ gpg2 --edit-key microcheap

pub  rsa4096/BC0A9DD1
     created: 2014-08-22  expires: never       usage: SC
     trust: marginal      validity: full
sub  rsa4096/206367B1
     created: 2014-08-22  expires: never       usage: E
[ unknown] (1). MicroCheapFx <fx@microcheap.info>
[  full  ] (2)  MicroCheapFx <microcheapfx@microcheap.info>
[marginal] (3)  François-Xavier Lesaffre <fxlesaffre@ovh.fr>

gpg> check
uid  MicroCheapFx <fx@microcheap.info>
sig!3        BC0A9DD1 2015-02-15  [self-signature]
uid  MicroCheapFx <microcheapfx@microcheap.info>
sig!3        BC0A9DD1 2014-08-22  [self-signature]
sig!         E25FBABB 2014-08-23  Damien Goutte-Gattat <dgouttegattat@incenp.org>
uid  François-Xavier Lesaffre <fxlesaffre@ovh.fr>
sig!         93CCE46B 2015-03-23  Grégory Roche <gregory@polymorphisme.fr>
sig!3        BC0A9DD1 2014-09-06  [self-signature]
```

Donc identité marginalement valide

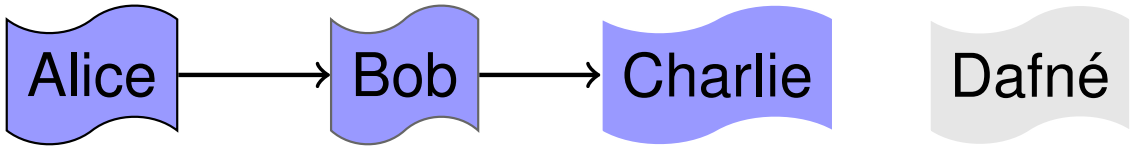
Non-transitivité de la confiance

- Alice certifie la clef de Bob et lui assigne une confiance complète
- Bob certifie la clef de Charlie
- Charlie certifie la clef de Dafné

Non-transitivité de la confiance

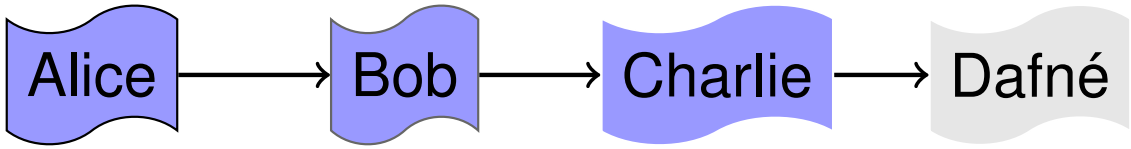
- Alice certifie la clef de Bob et lui assigne une confiance complète
- Bob certifie la clef de Charlie
- Charlie certifie la clef de Dafné

Non-transitivité de la confiance



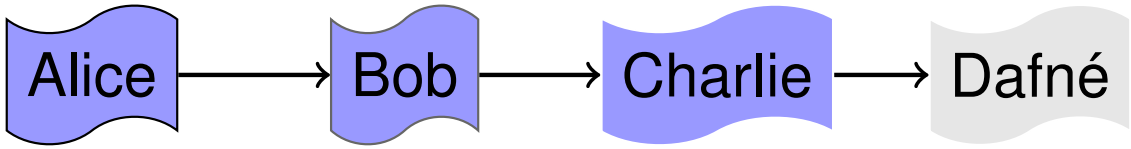
- Alice certifie la clef de Bob et lui assigne une confiance complète
- Bob certifie la clef de Charlie
- Charlie certifie la clef de Dafné

Non-transitivité de la confiance



- Alice certifie la clef de Bob et lui assigne une confiance complète
- Bob certifie la clef de Charlie
- Charlie certifie la clef de Dafné

Non-transitivité de la confiance



- Alice certifie la clef de Bob et lui assigne une confiance complète
- Bob certifie la clef de Charlie
- Charlie certifie la clef de Dafné

- La confiance est un attribut *privé*
- Alice ne sait pas quelle confiance Bob accorde à Charlie
- Seule Alice décide de la confiance qu'*elle* accorde à Charlie
- La confiance est indéfinie par défaut, donc la clef de Dafné est de validité inconnue

Niveaux de certification

4 niveaux de certification

- Niveau 0 (*generic certification*)
- Niveau 1 (*persona certification*)
- Niveau 2 (*casual certification*)
- Niveau 3 (*positive certification*)

Certifications acceptées pour l'évaluation de la validité

- Certifications de niveau 0
- Certifications de niveau supérieur ou égal au paramètre `--min-cert-level` (par défaut, 2)

Choix du niveau de certification

- Niveau par défaut : 0 (ou `--default-cert-level`)
- Sélectionnable au cas par cas (`--ask-cert-level`)

Options de certification

Politique de certification

Option `--cert-policy-url http://example.org/cert-policy.txt`

Certifications non-révocables

Commande `nrsign`

☞ Toute révocation ultérieure sera ignorée.

Certifications locales

Commande `lsign`

☞ Certification non-exportée.

Certifications de confiance (*trust signatures*)

Toile de confiance classique (`--trust-model classic`)

- Les certifications déterminent la validité
- La confiance est locale et privée
- Les *trust signatures* sont ignorées (traitées comme des certifications normales)

Toile de confiance étendue (`--trust-model pgp`)

- Toile de confiance classique + prise en compte des *trust signatures*
- Une *trust signature* affirme simultanément la *validité* d'une identité et la *confiance* à lui accorder

Caractéristiques d'une *trust signature*

Confiance

- Valeur comprise entre 0 et 255
- < 120 : confiance marginale
- ≥ 120 : confiance complète

Profondeur

- Valeur comprise entre 0 et 255
- Une clef certifiée avec une profondeur n peut émettre à son tour des *trust signatures* de niveau $n-1$
- Profondeur 0 : équivalent à une certification normale

Expression rationnelle (optionnelle)

La clef certifiée ne peut émettre de *trust signatures* que sur des identités correspondant à l'expression rationnelle.

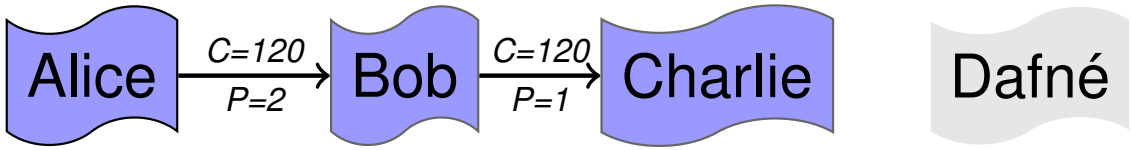
Toile de confiance étendue

- Alice certifie la clef de Bob avec une *trust signature* de confiance complète (C=120) et de profondeur 2
- Bob certifie la clef de Charlie avec une *trust signature* de profondeur 1
- Pour Alice, la clef de Charlie est *valide* et *à confiance complète*
- Charlie certifie la clef de Dafné
- Pour Alice, la clef de Dafné est *valide* et *à confiance inconnue*

Toile de confiance étendue

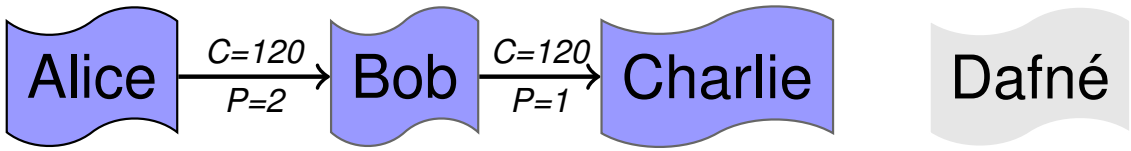
- Alice certifie la clef de Bob avec une *trust signature* de confiance complète (C=120) et de profondeur 2
- Bob certifie la clef de Charlie avec une *trust signature* de profondeur 1
- Pour Alice, la clef de Charlie est *valide* et *à confiance complète*
- Charlie certifie la clef de Dafné
- Pour Alice, la clef de Dafné est *valide* et *à confiance inconnue*

Toile de confiance étendue



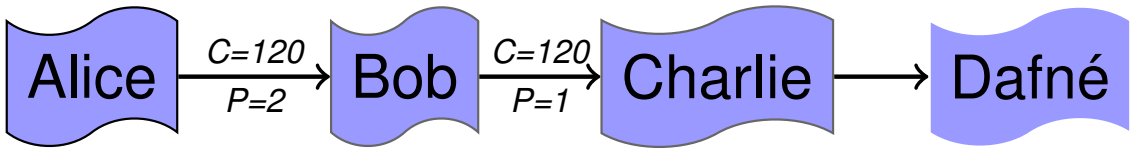
- Alice certifie la clef de Bob avec une *trust signature* de confiance complète (C=120) et de profondeur 2
- Bob certifie la clef de Charlie avec une *trust signature* de profondeur 1
- Pour Alice, la clef de Charlie est *valide* et *à confiance complète*
- Charlie certifie la clef de Dafné
- Pour Alice, la clef de Dafné est *valide* et *à confiance inconnue*

Toile de confiance étendue



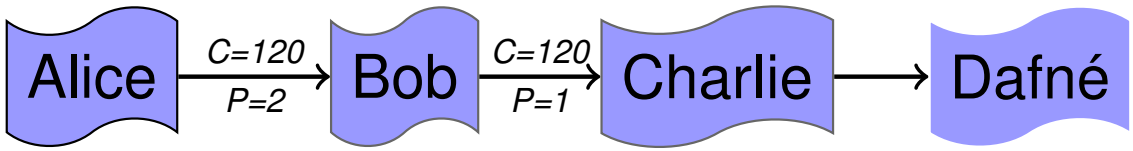
- Alice certifie la clef de Bob avec une *trust signature* de confiance complète (C=120) et de profondeur 2
- Bob certifie la clef de Charlie avec une *trust signature* de profondeur 1
- Pour Alice, la clef de Charlie est *valide* et *à confiance complète*
- Charlie certifie la clef de Dafné
- Pour Alice, la clef de Dafné est *valide* et *à confiance inconnue*

Toile de confiance étendue



- Alice certifie la clef de Bob avec une *trust signature* de confiance complète (C=120) et de profondeur 2
- Bob certifie la clef de Charlie avec une *trust signature* de profondeur 1
- Pour Alice, la clef de Charlie est *valide* et *à confiance complète*
- Charlie certifie la clef de Dafné
- Pour Alice, la clef de Dafné est *valide* et *à confiance inconnue*

Toile de confiance étendue



- Alice certifie la clef de Bob avec une *trust signature* de confiance complète (C=120) et de profondeur 2
- Bob certifie la clef de Charlie avec une *trust signature* de profondeur 1
- Pour Alice, la clef de Charlie est *valide* et *à confiance complète*
- Charlie certifie la clef de Dafné
- Pour Alice, la clef de Dafné est *valide* et *à confiance inconnue*

Sommaire

Pourquoi un nouveau modèle de confiance ?

Complexité de la toile de confiance

- Beaucoup d'utilisateurs ne comprennent pas réellement la toile de confiance...
- et donc ne l'utilisent pas correctement
- Modèle trop pénible à utiliser pour beaucoup

Trust-On-First-Use (TOFU)

- Moins puissant que la toile de confiance
- Vulnérable à une tentative d'usurpation lors du premier contact
- Mais plus facile à appréhender et à utiliser
- "Better-than-nothing security"

Principe des modèles TOFU

Base de données TOFU

- GnuPG garde une trace de toutes les associations (email, clef) qu'il rencontre
- et leur associe une *politique* ("policy")

Politiques TOFU

unknown	Clef à validité inconnue
auto	Clef marginalement valide
good	Clef complètement valide
bad	Clef invalide
ask	demander à l'utilisateur

Principe des modèles TOFU

Premier message venant de `alice@example.org`

- Ajouter l'association (`alice@example.org`, 0x2EADF7D4)
- Avec la politique par défaut (*auto*)
- Politique modifiable à tout moment *a posteriori* avec `--tofu-policy`

Nouveau message venant de `alice@example.org`

- Vérifier si la clef est la même que précédemment
- Si oui, utiliser la politique associée
- Si non, alerter l'utilisateur d'un conflit

Choix de la politique par défaut

`--tofu-default-policy good`

- Politique « optimiste », ou TOFU proprement dit
- Toute nouvelle clef est implicitement considérée complètement valide

`--tofu-default-policy unknown`

- Politique « pessimiste » ou « paranoïaque »
- Aucune confiance implicite
- Toute nouvelle clef est à validité inconnue

`--tofu-default-policy auto`

- Politique « intermédiaire »
- Confiance implicite limitée
- Toute nouvelle clef est considérée marginalement valide

Le modèle TOFU+PGP

Interrogation successive des deux modèles

La clef est globalement valide si

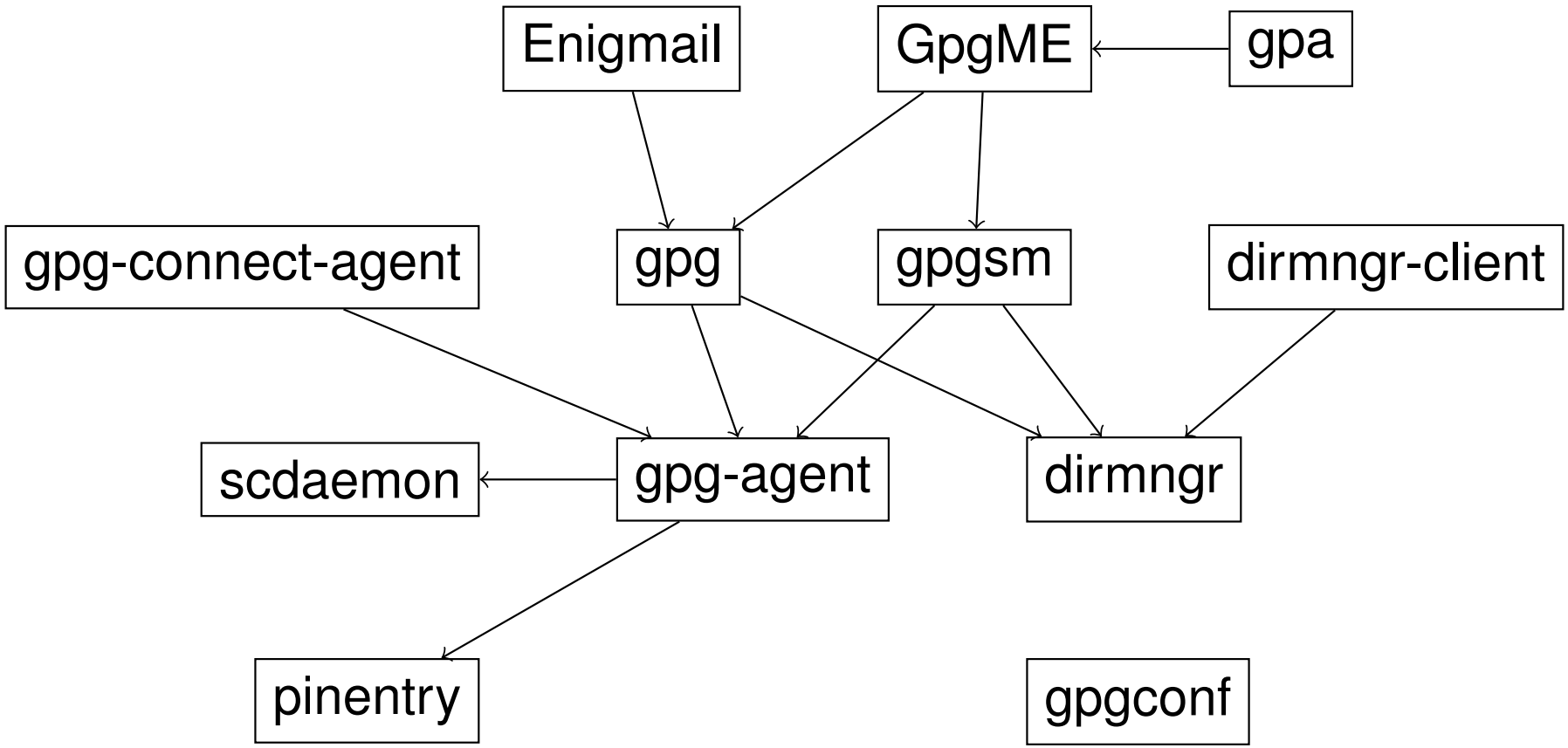
- elle est valide dans au moins un modèle
- elle n'est invalide dans aucun modèle

Avec `--tofu-default-policy unknown`

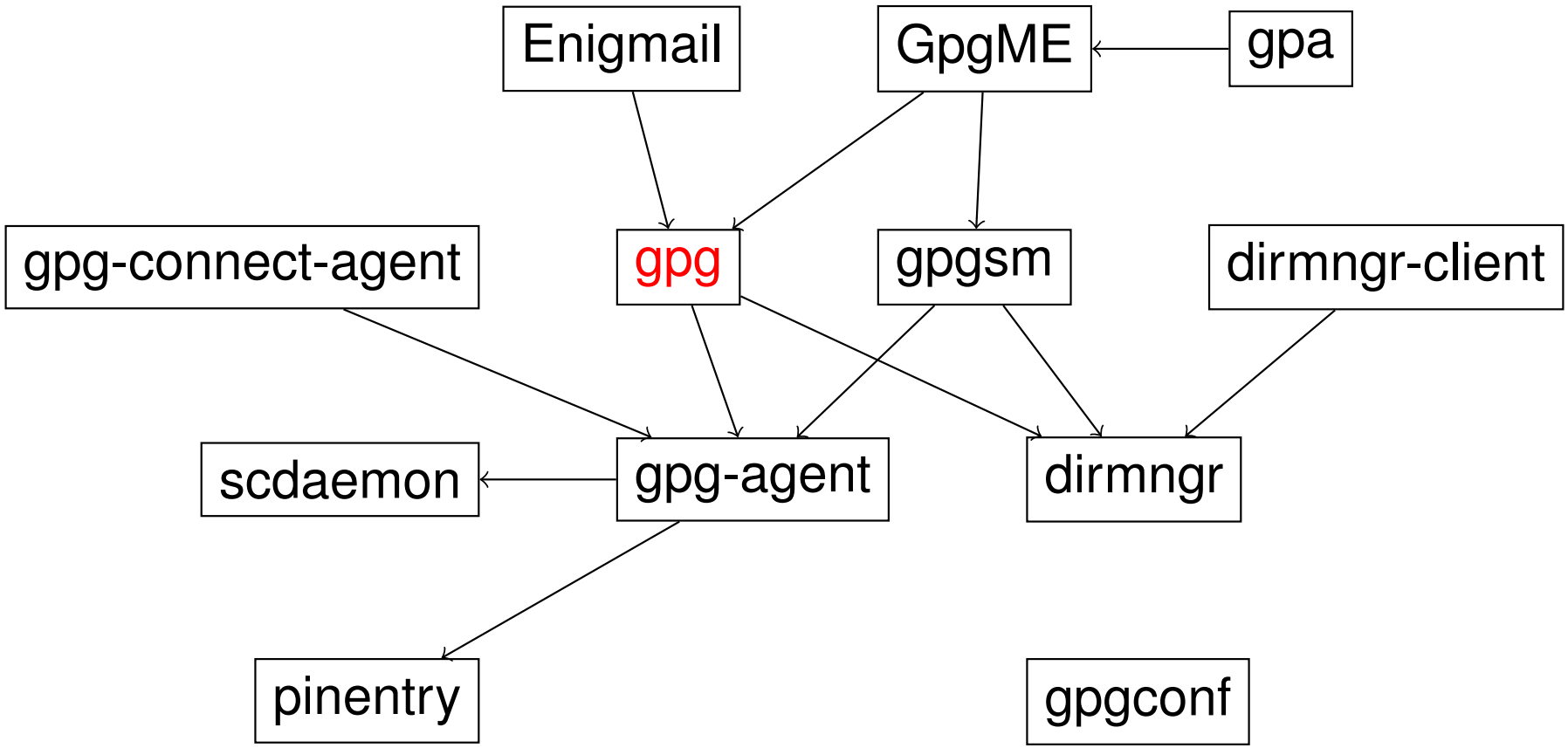
- Seul la toile de confiance assigne une confiance positive
- Le modèle TOFU ne sert alors qu'à détecter les conflits

Sommaire

Architecture de GnuPG 2.1

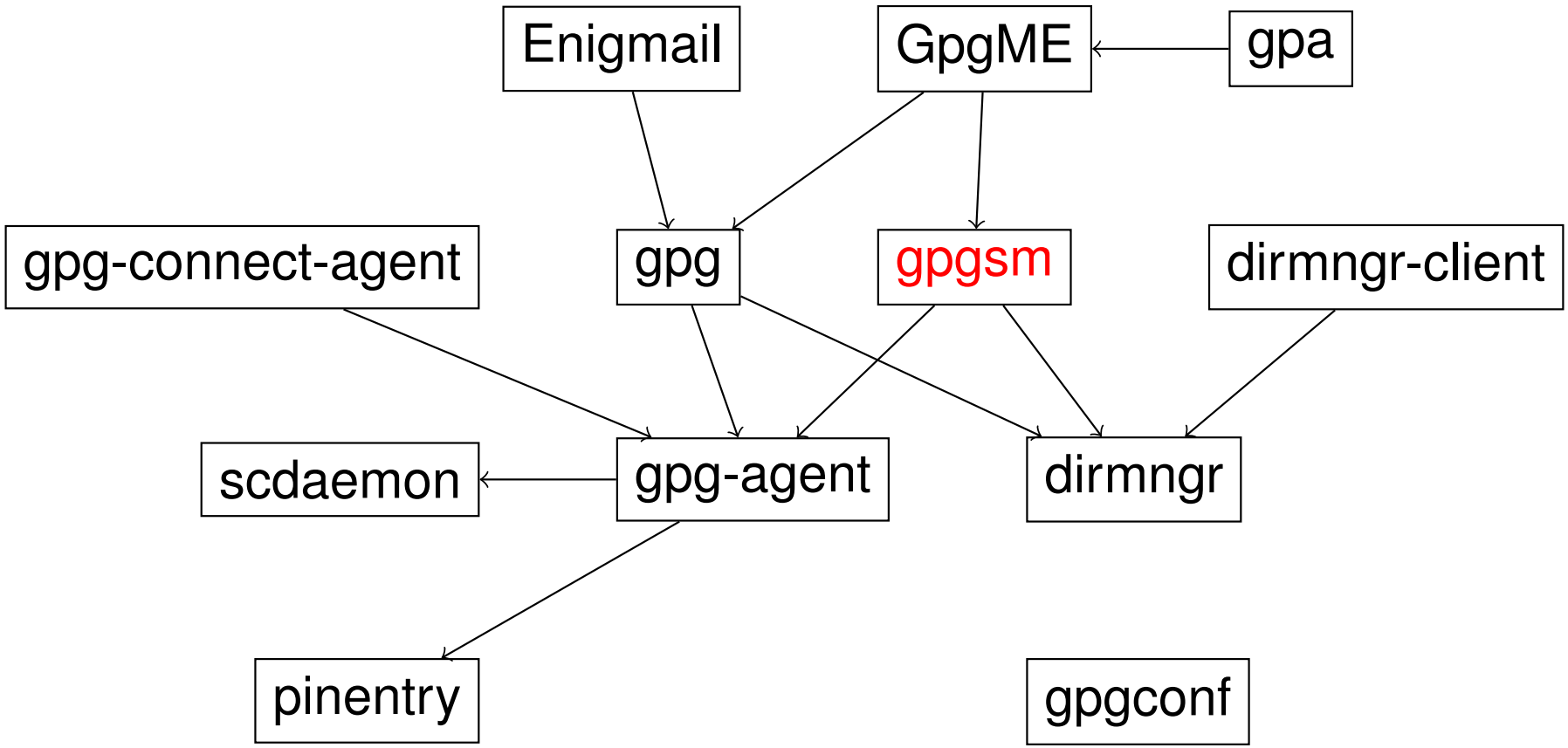


Architecture de GnuPG 2.1



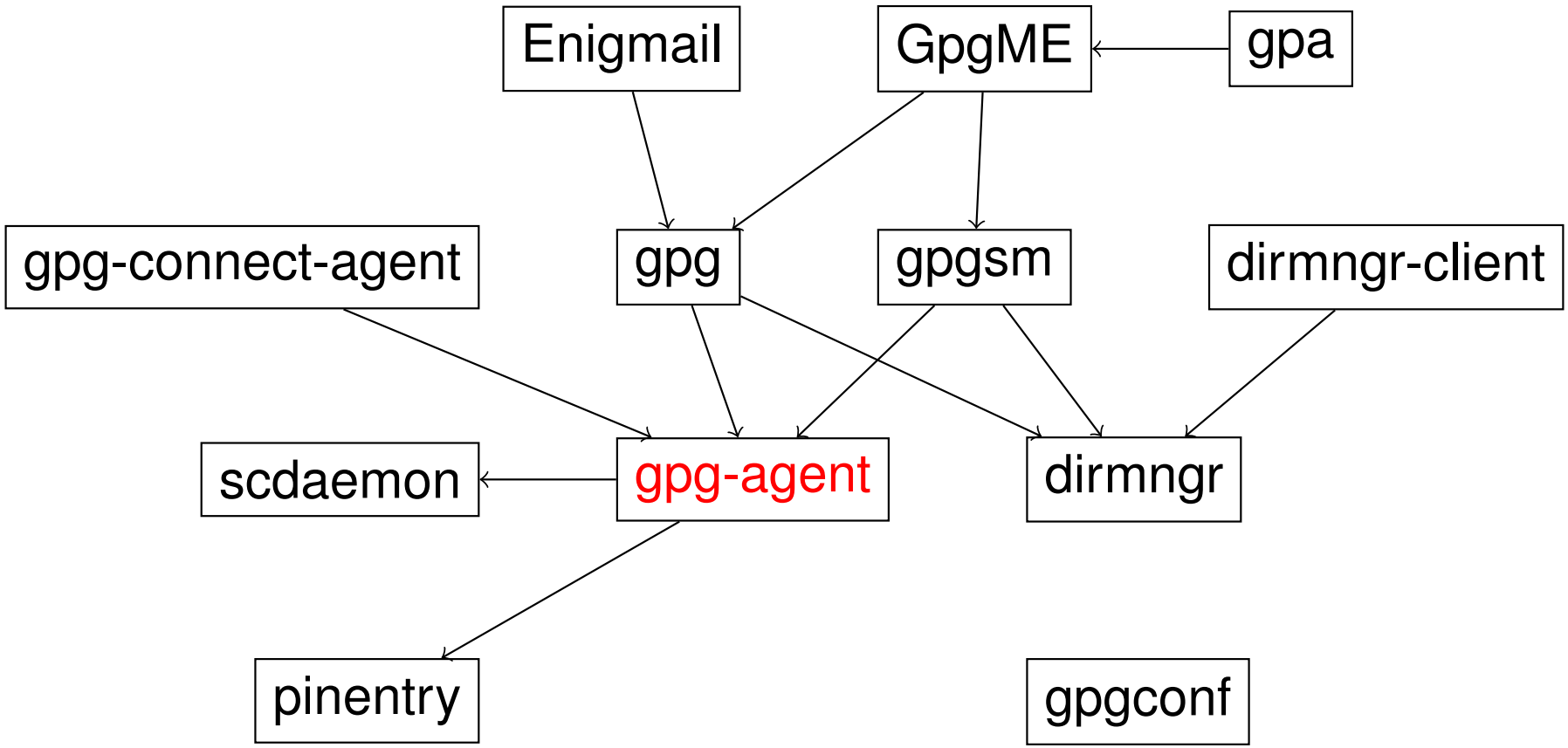
Programme principal pour OpenPGP

Architecture de GnuPG 2.1



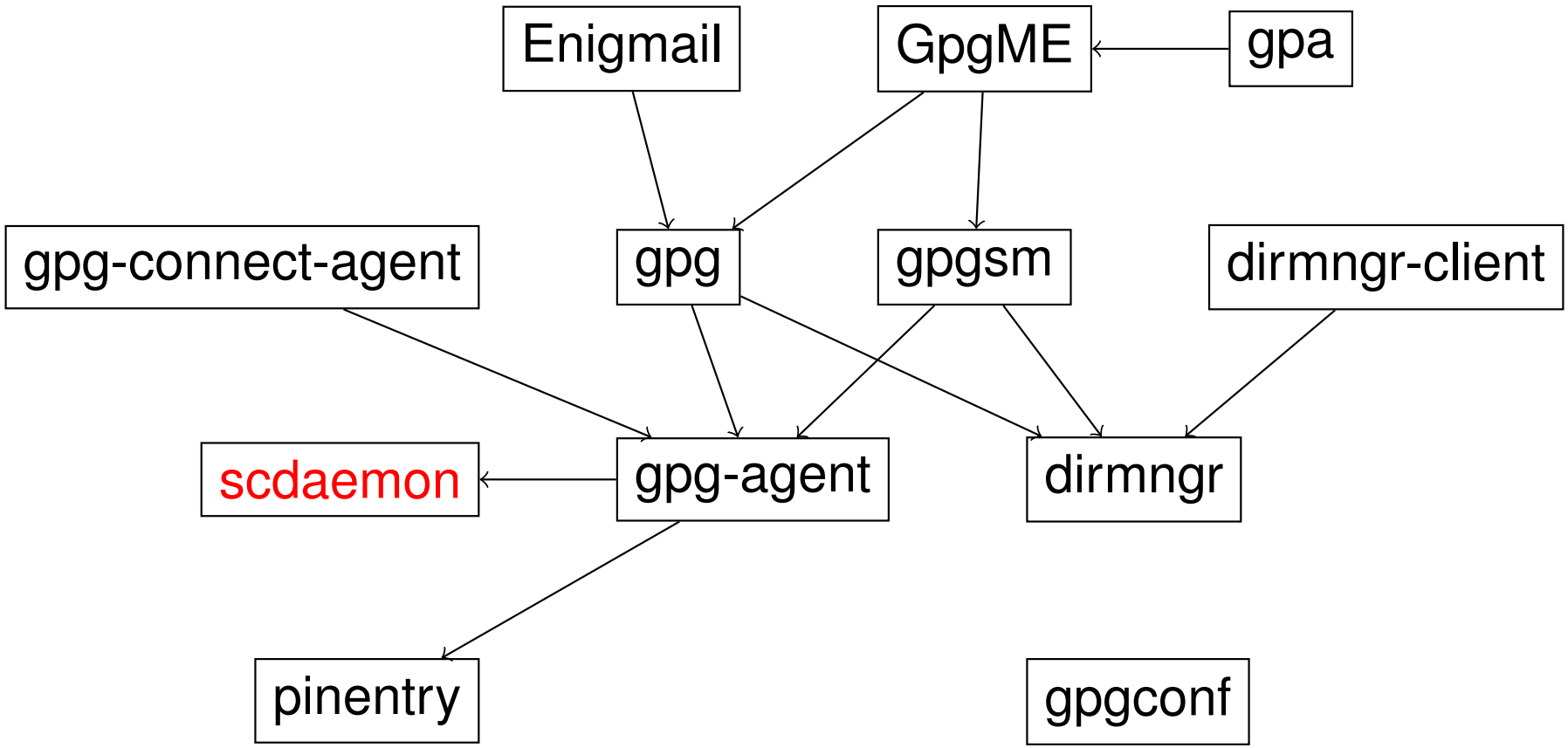
Programme principal pour X.509 et S/MIME

Architecture de GnuPG 2.1



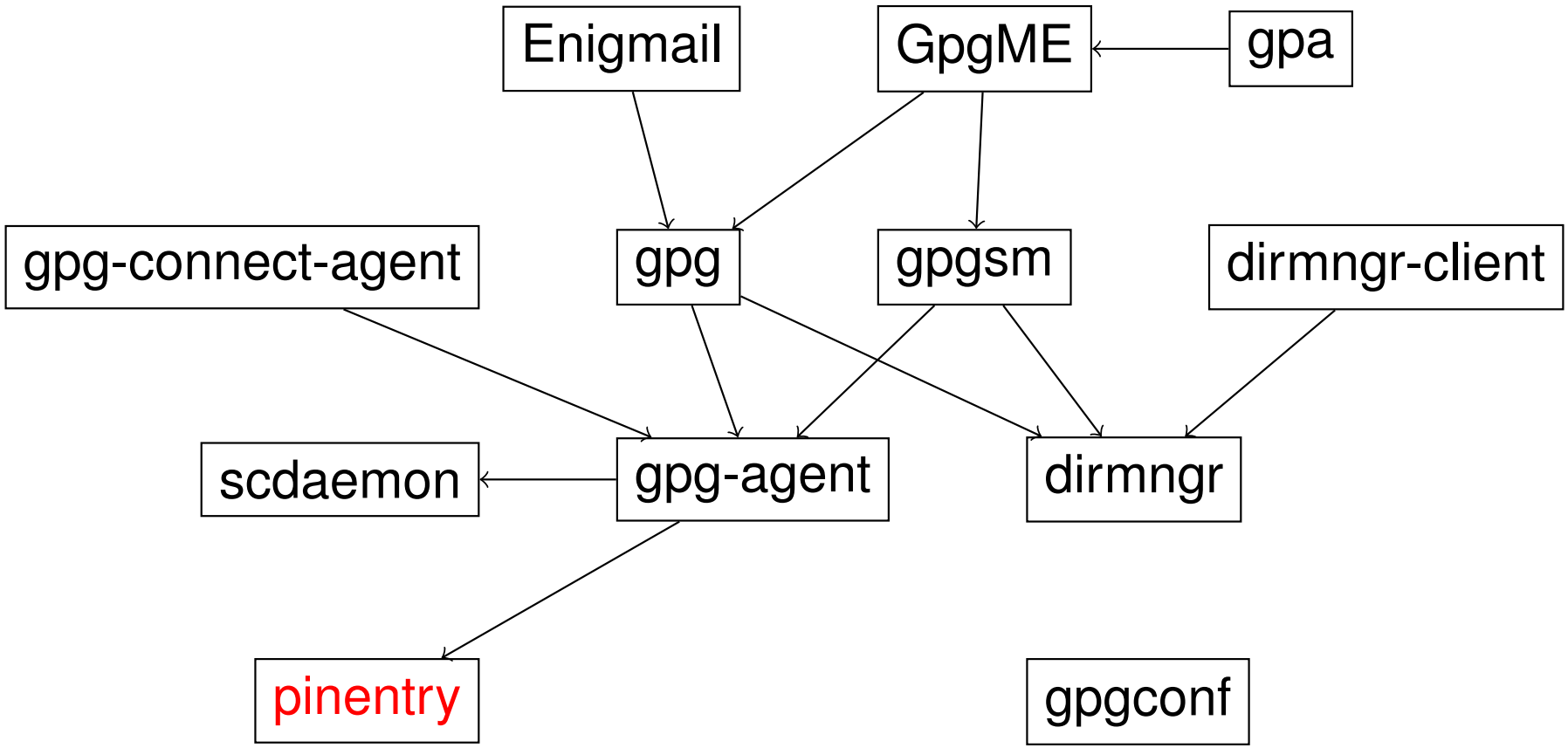
Agent de gestion des clefs privées

Architecture de GnuPG 2.1



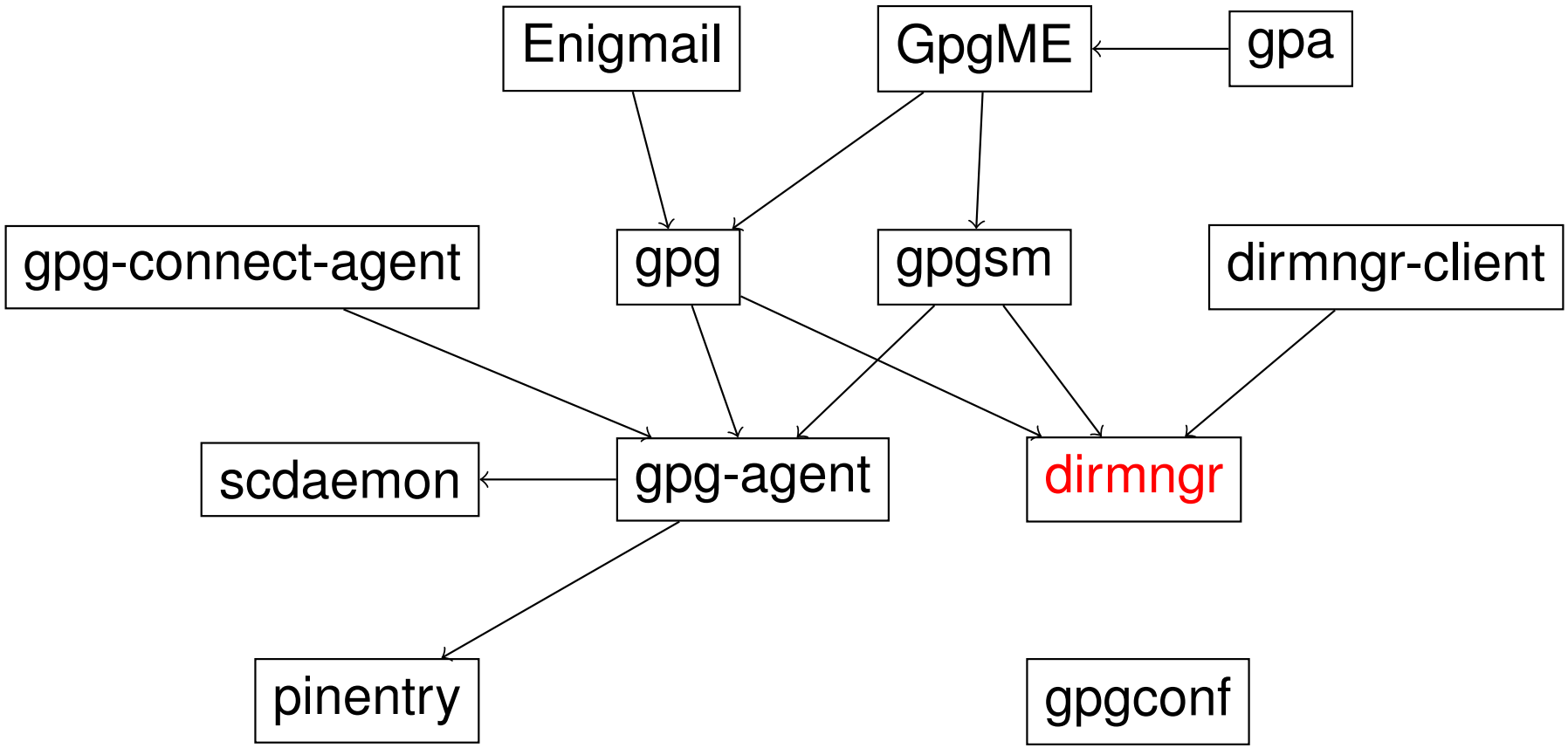
Démon d'accès aux cartes à puces

Architecture de GnuPG 2.1



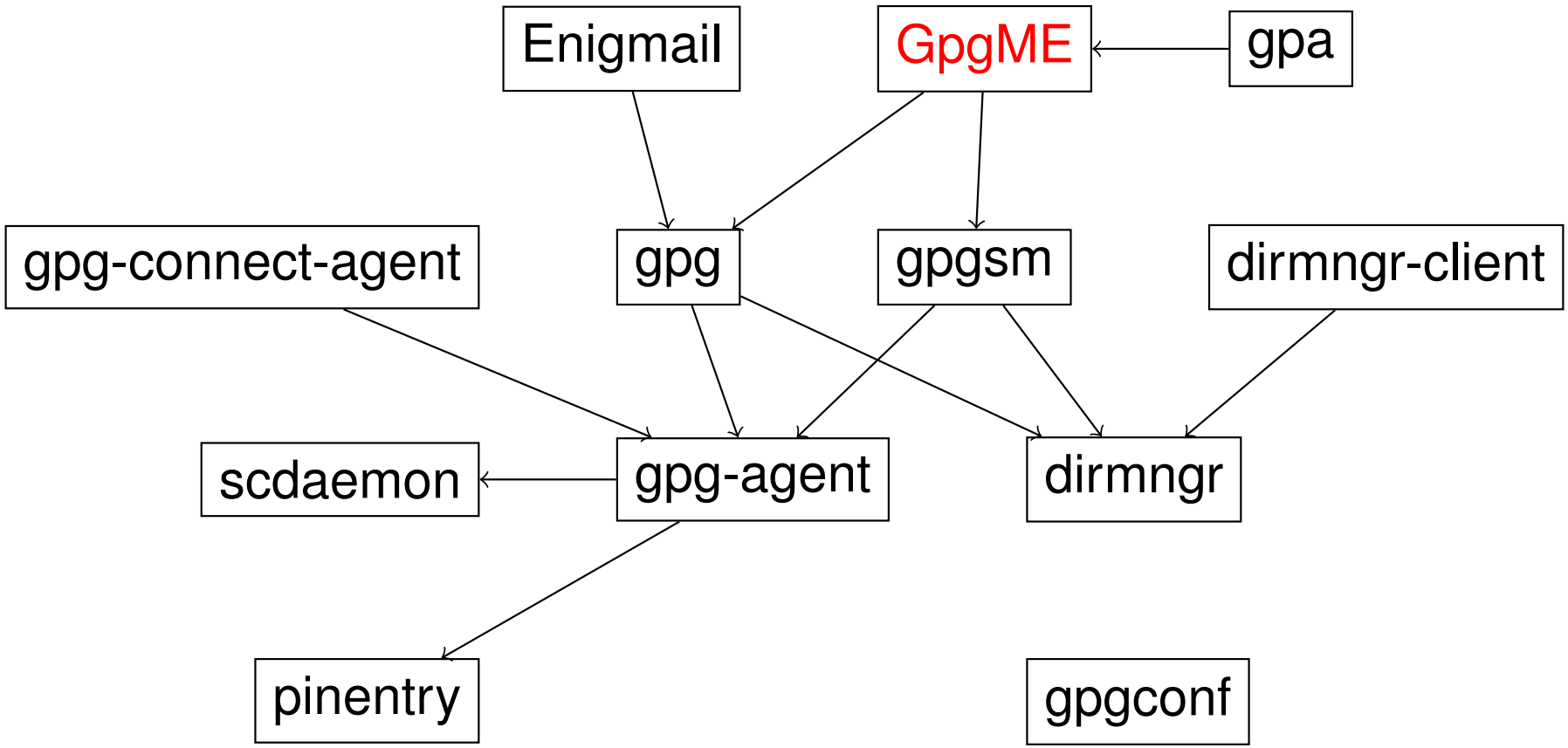
Interface de saisie des phrases de passe

Architecture de GnuPG 2.1



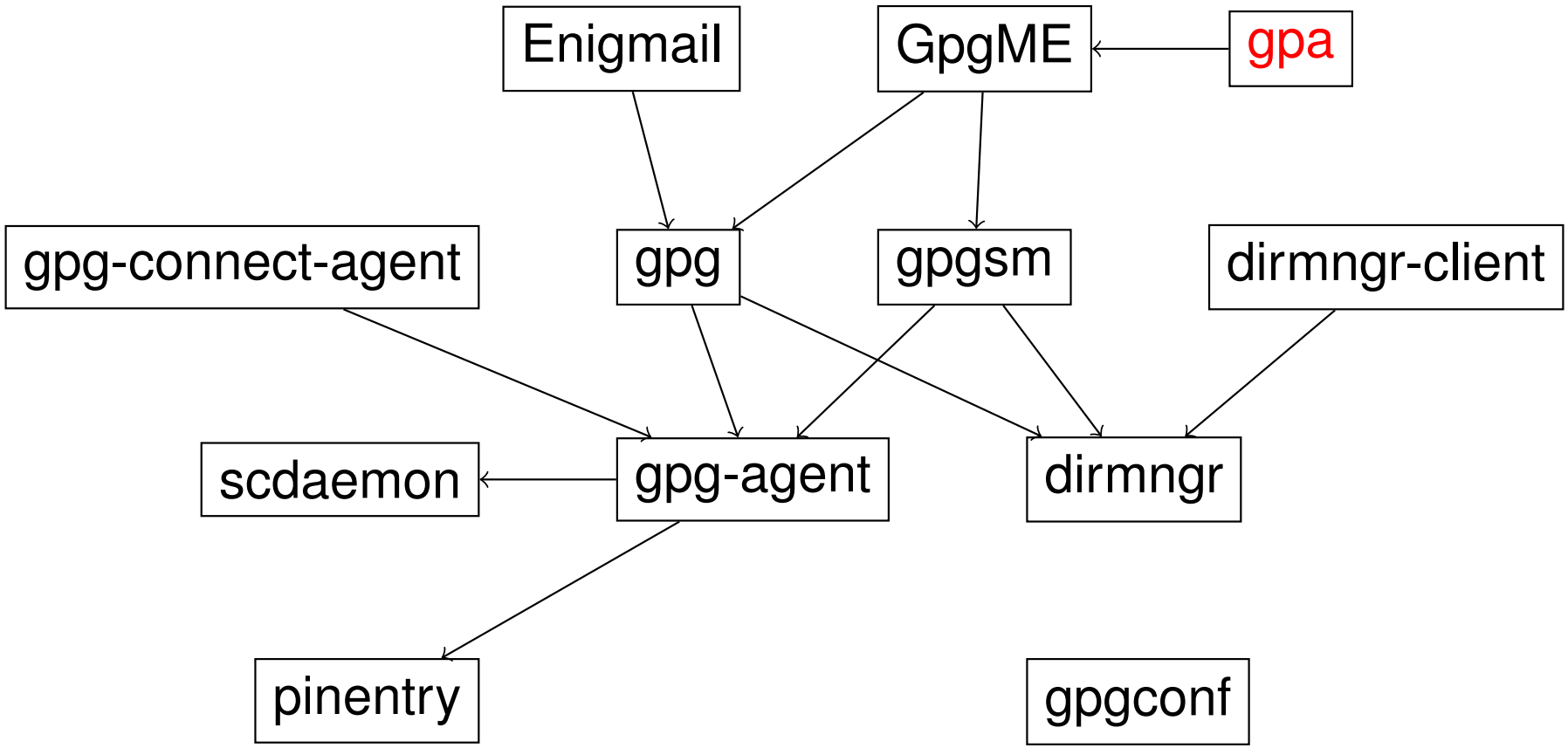
Démon réseau (accès aux serveurs de clefs)

Architecture de GnuPG 2.1



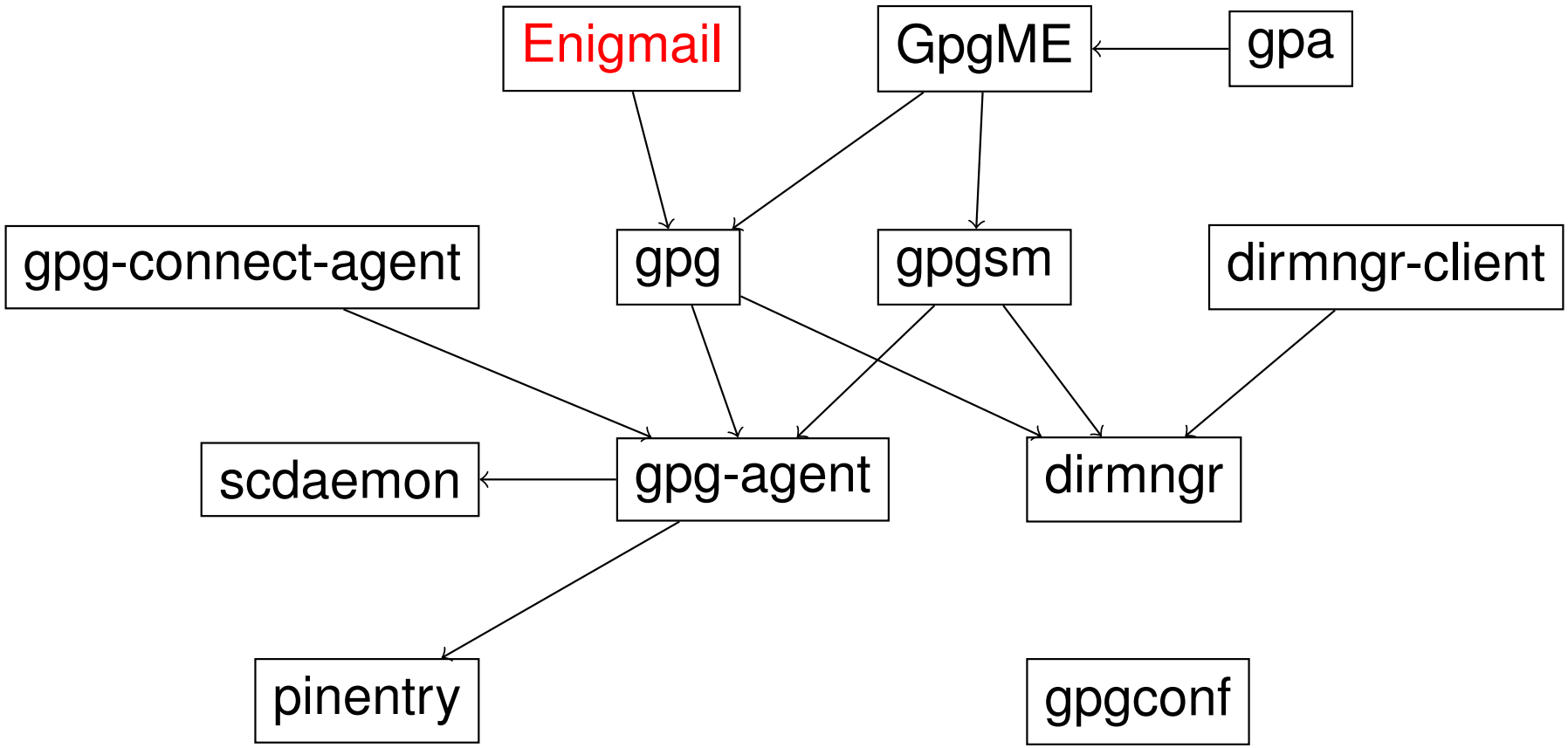
Bibliothèque pour applications tierces

Architecture de GnuPG 2.1



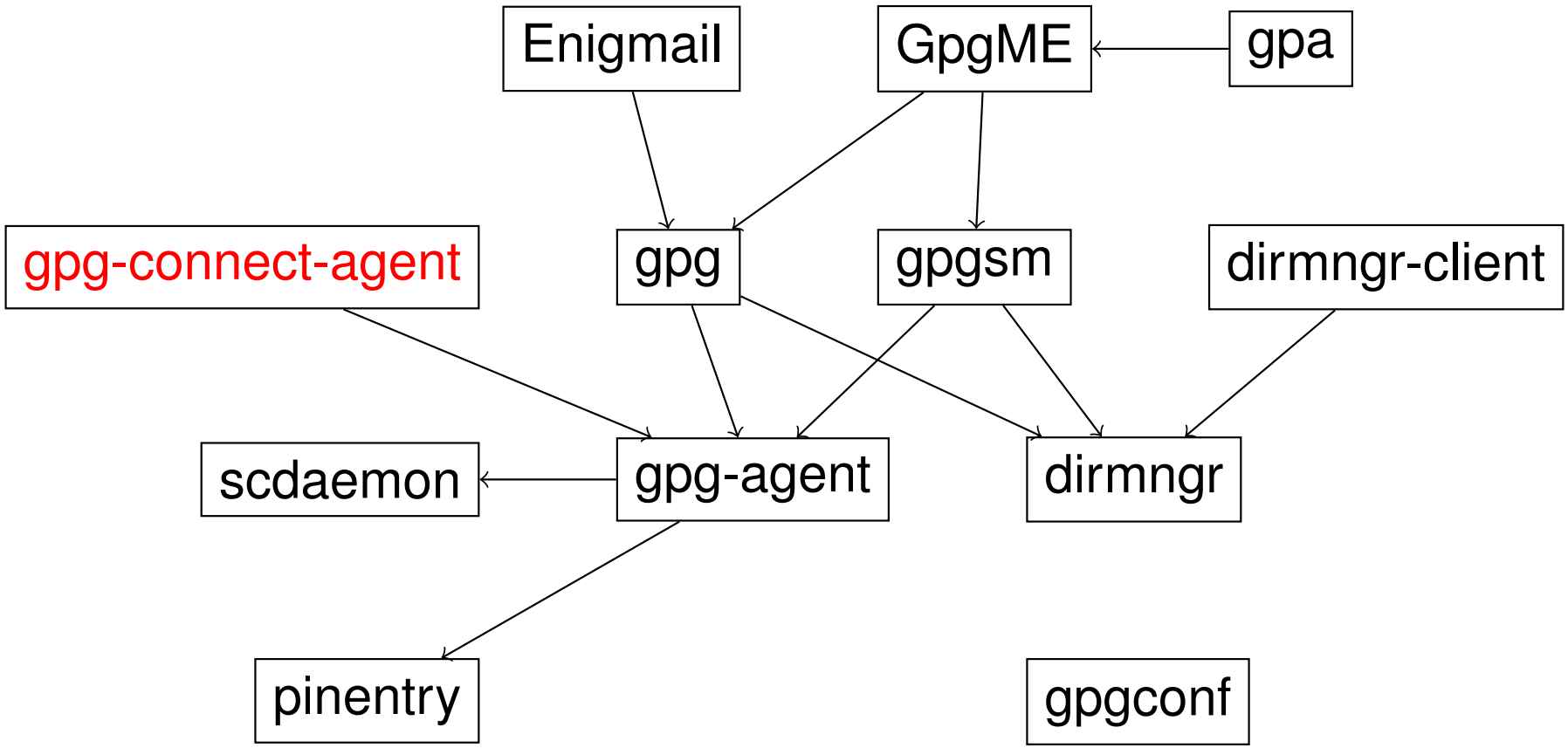
frontend graphique officiel

Architecture de GnuPG 2.1



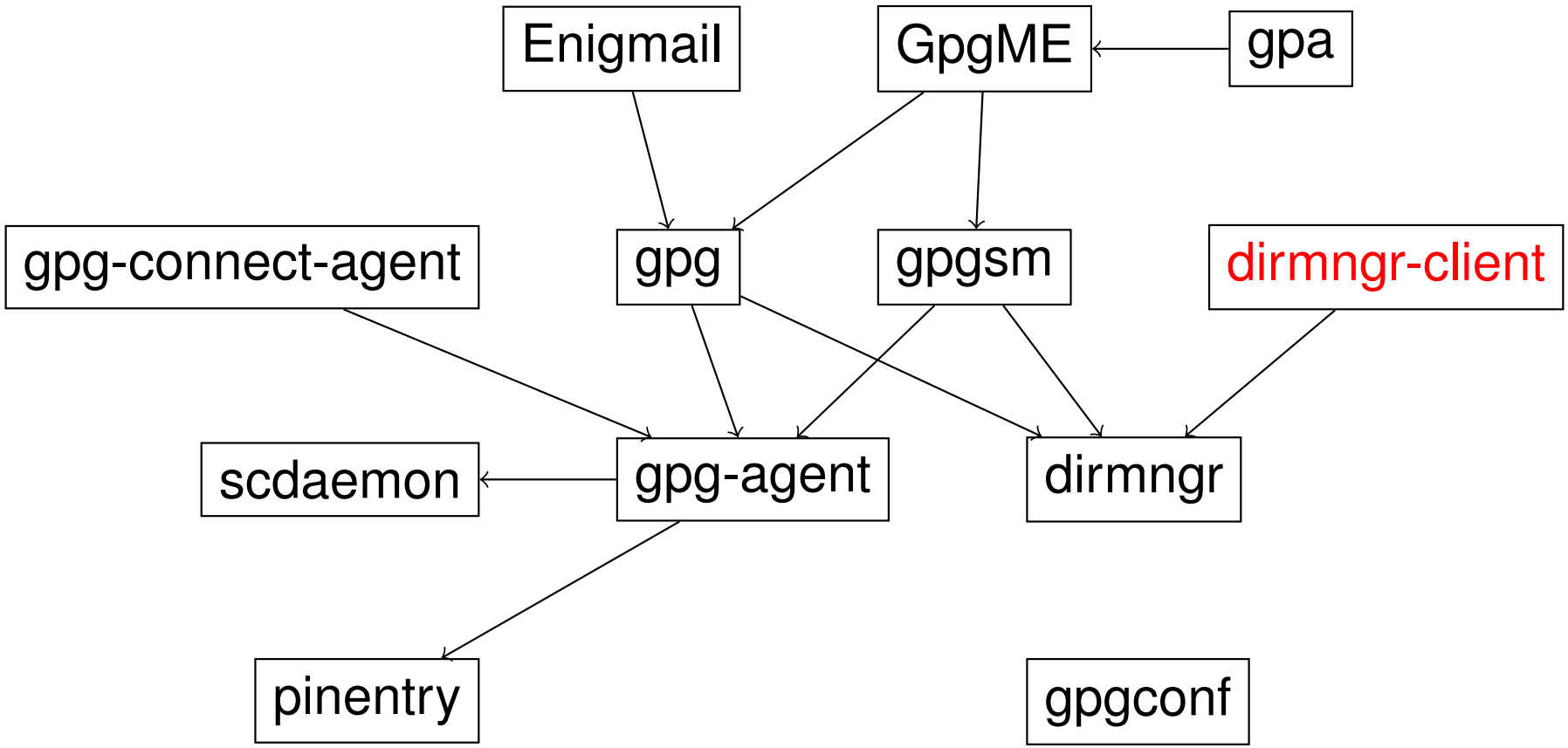
Greffon pour Mozilla Thunderbird

Architecture de GnuPG 2.1



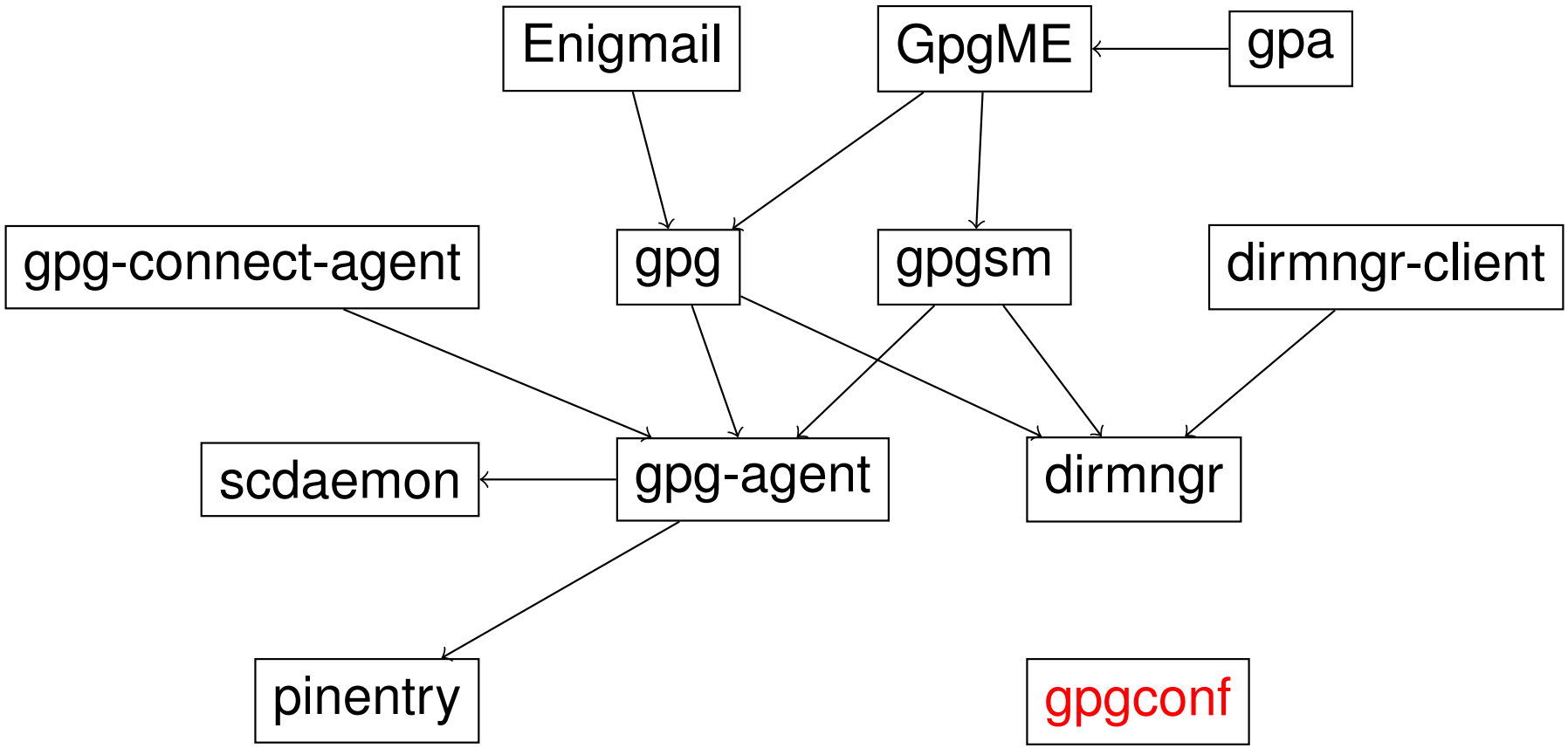
Outil d'interaction directe avec l'agent

Architecture de GnuPG 2.1



Outil d'interaction directe avec le démon réseau

Architecture de GnuPG 2.1



Outil de configuration

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx       sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed       tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status   trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event         trustlist.txt
```

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx      sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed      tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status  trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event        trustlist.txt
```

Sockets de communication avec les programmes auxiliaires

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx       sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed       tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status   trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event         trustlist.txt
```

Cache du démon réseau

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx       sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed       tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status   trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event         trustlist.txt
```

Fichiers de configuration des différents composants

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx      sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed      tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status  trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event        trustlist.txt
```

Fichiers auxiliaires pour X.509 et S/MIME

- policies.txt Politiques de certification acceptables
- trustlist.txt Certificats racines de confiance

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx       sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed       tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status   trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event         trustlist.txt
```

Clefs privées gérées par l'agent GnuPG

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx       sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed       tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status   trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event         trustlist.txt
```

Clefs publiques

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx      sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed      tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status  trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event        trustlist.txt
```

Bases de données des modèles de confiance

trustdb.gpg Pour tous les modèles

tofu.db Données spécifiques aux modèles TOFU

Données modérément sensibles

Permet de tisser un graphe précis de vos contacts, incluant la confiance que vous accordez à chacun

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx      sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed      tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status  trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event        trustlist.txt
```

Certificats de révocation (générés automatiquement lors de la création de nouvelles paires de clefs)

Données sensibles

Permet de révoquer inconditionnellement vos clefs

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx      sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed      tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status  trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event        trustlist.txt
```

État du générateur de nombres aléatoires

Très sensible

Permet de déduire les prochaines clefs de session !
À ne pas sauvegarder !

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent       dirmngr.conf    gpgsm.conf          pubring.kbx       sshcontrol
S.gpg-agent.ssh   gpa.conf        openpgp-revocs.d    random_seed       tofu.db
S.scdaemon        gpg-agent.conf  policies.txt        reader_0.status   trustdb.gpg
clrs.d            gpg.conf        private-keys-v1.d   scd-event         trustlist.txt
```

Utilisation d'un lecteur de cartes à puce

reader_0.status État du lecteur 0

scd-event Script appelé à chaque changement d'état du lecteur

Contenu de GNUPGHOME

```
$ ls ~/.gnupg
S.gpg-agent      dirmngr.conf    gpgsm.conf          pubring.kbx      sshcontrol
S.gpg-agent.ssh  gpa.conf        openpgp-revocs.d    random_seed      tofu.db
S.scdaemon       gpg-agent.conf  policies.txt        reader_0.status  trustdb.gpg
clrs.d           gpg.conf        private-keys-v1.d   scd-event        trustlist.txt
```

Clefs privées utilisables par les client SSH

À propos de cette présentation

Contact

`dgouttegattat@incenp.org`