Un langage de composition des techniques de sécurité pour préserver la vie privée dans le nuage

Présenté par

Ronan-Alexandre Cherrueau

Sous la direction de

M. Mario Südholt, Professeur, Mines Nantes

École Doctorale Sciences et technologies de l'information, et mathématiques

Mines Nantes, Inria, Lina

MINES Nantes

Inria INVENTEURS DU MONDE NUMÉRIQUE

lina

UNIVERSITE BRETAGNE LOIRE

Données personnelles

Données qui permettent d'identifier un individu :

- État civil : Alice Martin, 10/10/1988
- NIR : 2 88 10 54 129 245 59
- GPS : `[(48.8583700, 2.2944813), …]`
- Recherche : 

Constitue la vie privée d'un individu

Agenda personnel en ligne

Aujourd'hui – Vendredi 18 Novembre

`09:00` IRC OpenStack `#OS-performance`

`14:00` Soutenance de thèse, B218

Inférence :

- *Profession*
- *Position géographique*

Agenda personnel en ligne

Aujourd'hui – Vendredi 18 Novembre

`09:00` IRC OpenStack `#OS-performance`

`14:00` Soutenance de thèse, B218

Inférence :

- *Profession*
- *Position géographique*

Inférence ignorée par Alice
$\Rightarrow$ Violation vie privée [Solove, 2006]

Violation de la vie privée [Solove, 2006]

Inférence Analyse à l'insu

Collecte Hébergement à l'insu

Diffusion Mise à disposition d'un tiers

Interférence Impossibilité de rectifier/supprimer

Violation de la vie privée [Solove, 2006]

Inférence Analyse à l'insu

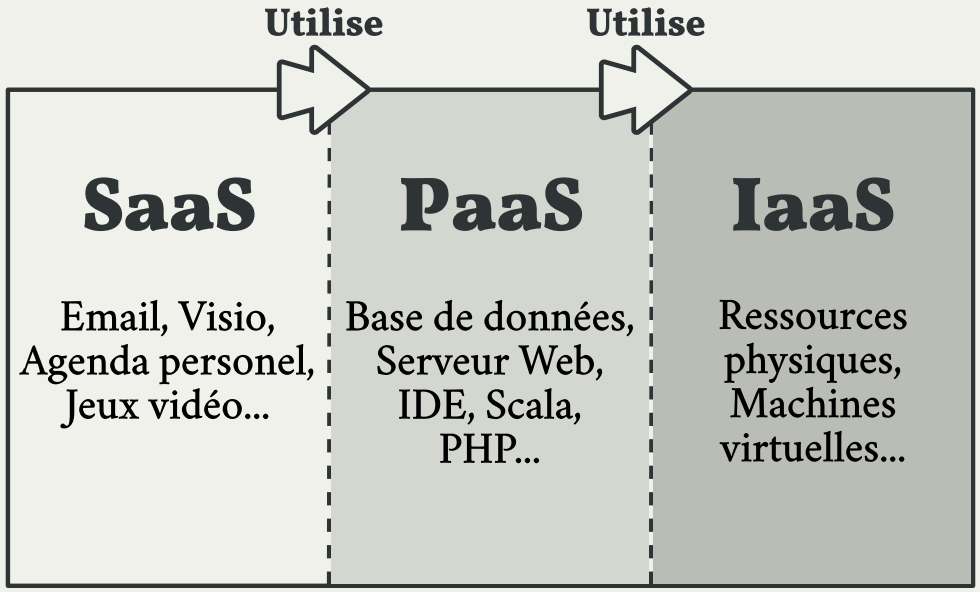
Collecte Hébergement à l'insu

Diffusion Mise à disposition d'un tiers

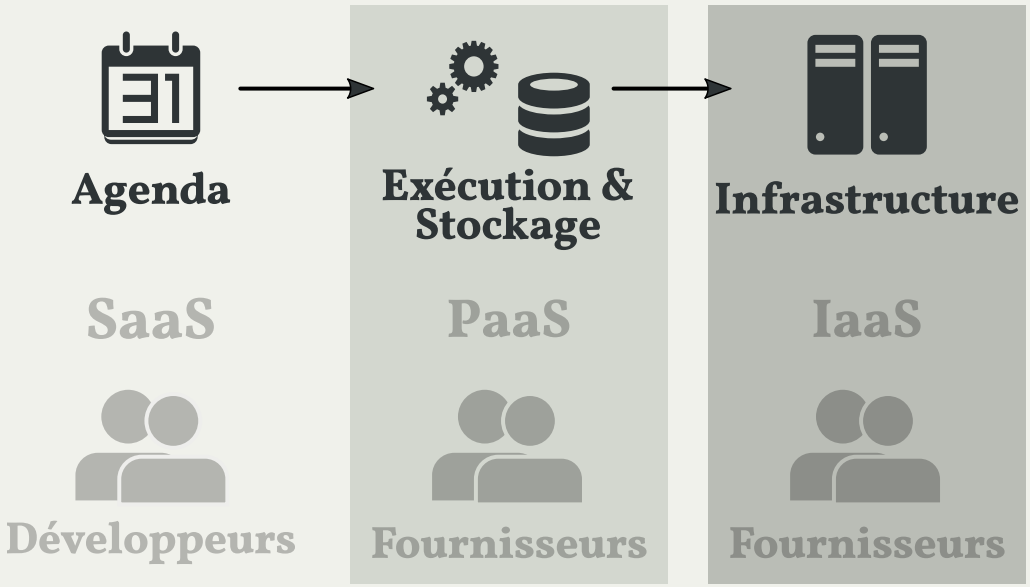
Interférence Impossibilité de rectifier/supprimer

Développeur : Est-ce que je peux développer une application du nuage qui prévient de ces violations ?

Informatique en nuage (I)



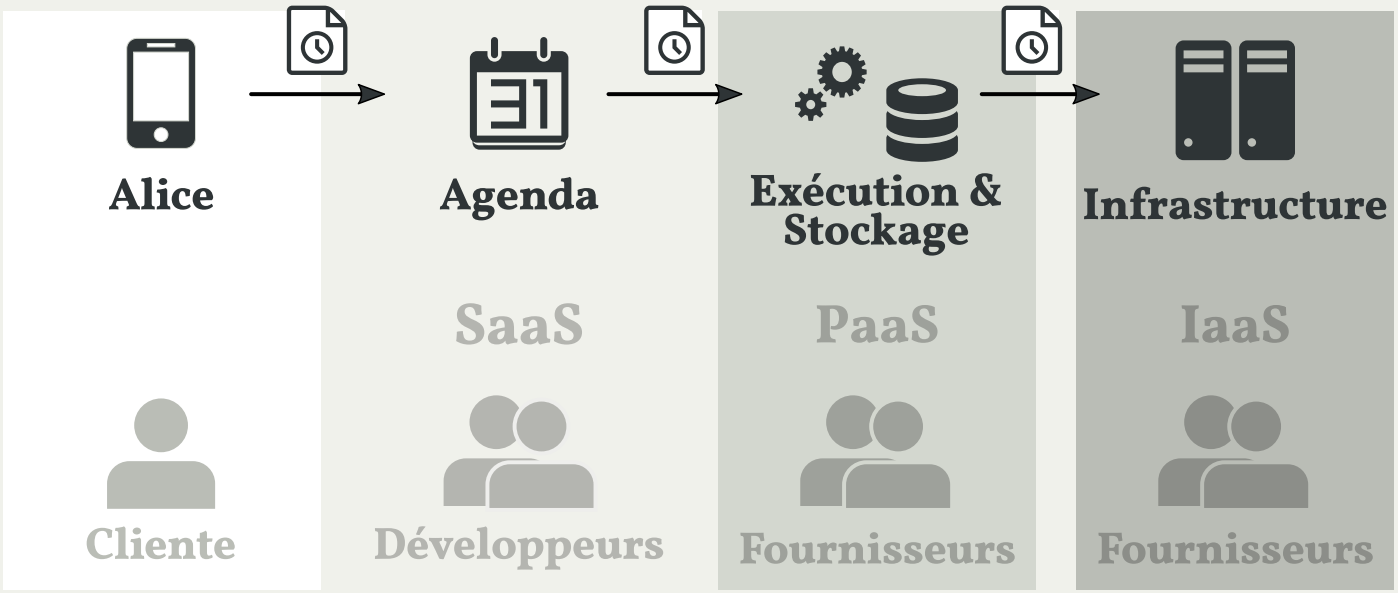
Informatique en nuage (2)



Avantage pour le développeur :

- Cycle de développement externalisé
- Réplication des rendez-vous externalisée
- Accélère le développement
- Limite les coûts

Informatique en nuage & Vie privée



- Rendez-vous transmis à SaaS, puis PaaS, puis IaaS

$\Rightarrow$ Chaque acteur peut collecter/inférer/diffuser/interférer

Techniques de sécurité & Vie privée

Applications du nuage qui préservent la vie privée :

- Applications de messagerie en ligne
- Cryptocat, Telegram Messenger, WhatsApp

Techniques de sécurité [Van Blarkom et al., 2003] :

- Donnée personnelle inintelligible pour tous sauf la propriétaire
- Ex : chiffrement point-à-point des communications

Techniques rendent l'application respectueuse de la vie privée

Cas de l'agenda personnel

Agenda :

- *rendezvous* : (*date*, *nom*, *adresse*)
- Contrainte de vie privée 1 : *nom*
- Contrainte de vie privée 2 : (*date, adresse*)
- Requête `[adresse]` : Liste des adresses et contacts visités par la Cliente la semaine dernière.

Techniques de sécurité :

- Chiffrement symétrique [Menezes et al., 1996]
- Calculs côté client [Fournet et al., 2013]
- Fragmentation verticale [Aggarwal et al., 2005]

Objectifs :

- Sécurité
- Utilisation du nuage
- Performance

Chiffrement sym. (AES) [Menezes et al., 1996]

Alice chiffre ses rendez-vous avant de les sauvegarder dans la BD PaaS.

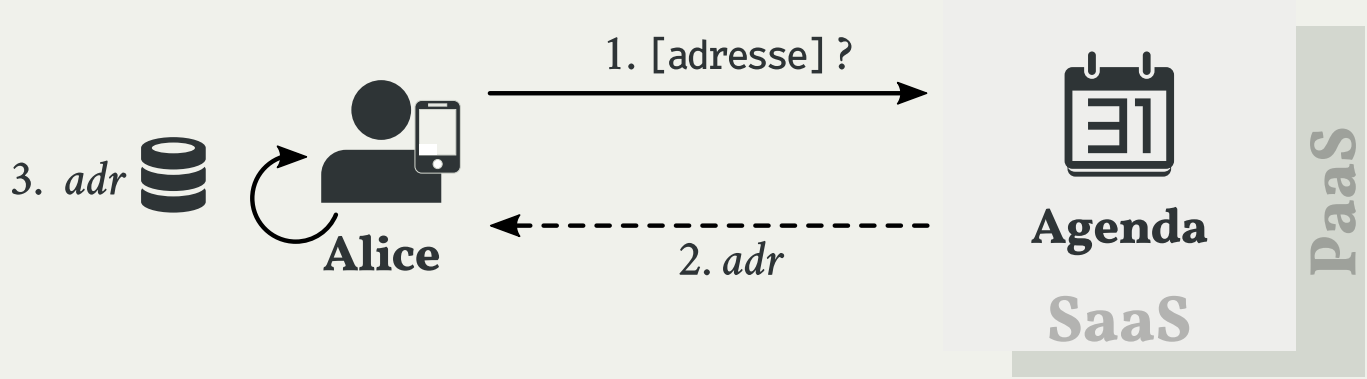
\+ **Sécurité** : *nom, (date,adresse)*

\+ **Nuage** : Serveur PaaS, BD PaaS

\− **Performance** : Problématiques

Calcul côté client [Fournet et al., 2013]

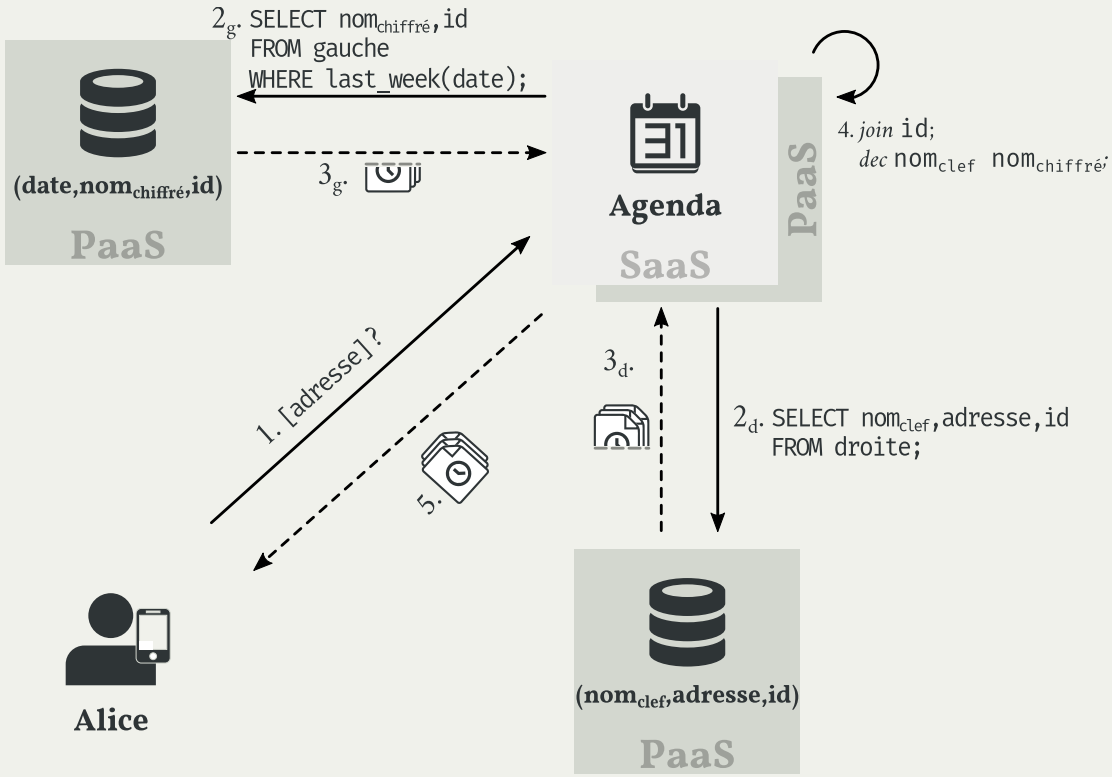
Alice conserve ses rendez-vous et fait les calculs sur son terminal.



+ **Sécurité** : *nom, (date,adresse)*
− **Nuage** : Serveur PaaS, ~~BD PaaS~~
+ **Performance** : Bonne

Fragmentation verticale [Aggarwal et al., 2005]

Agenda divise les rendez-vous d'Alice sur la contrainte associative *(date,adresse)*.

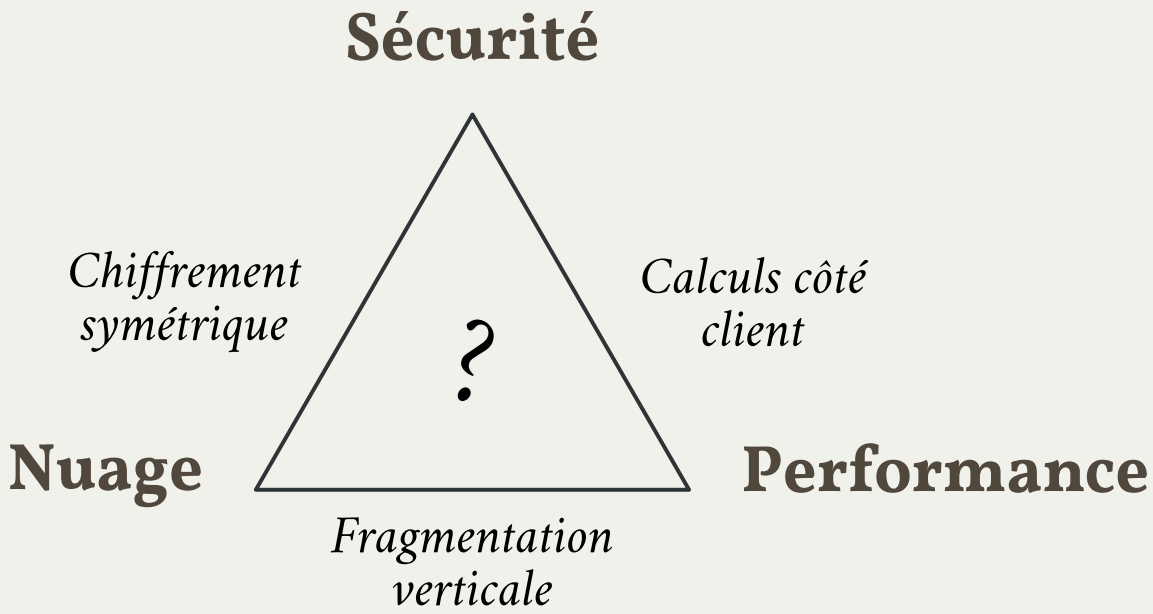


– **Sécurité** : ~~*nom*~~, ~~*(date,adresse)*~~

\+ **Nuage** : Serveur PaaS, BD PaaS

\+ **Performance** : Bonne

Défis

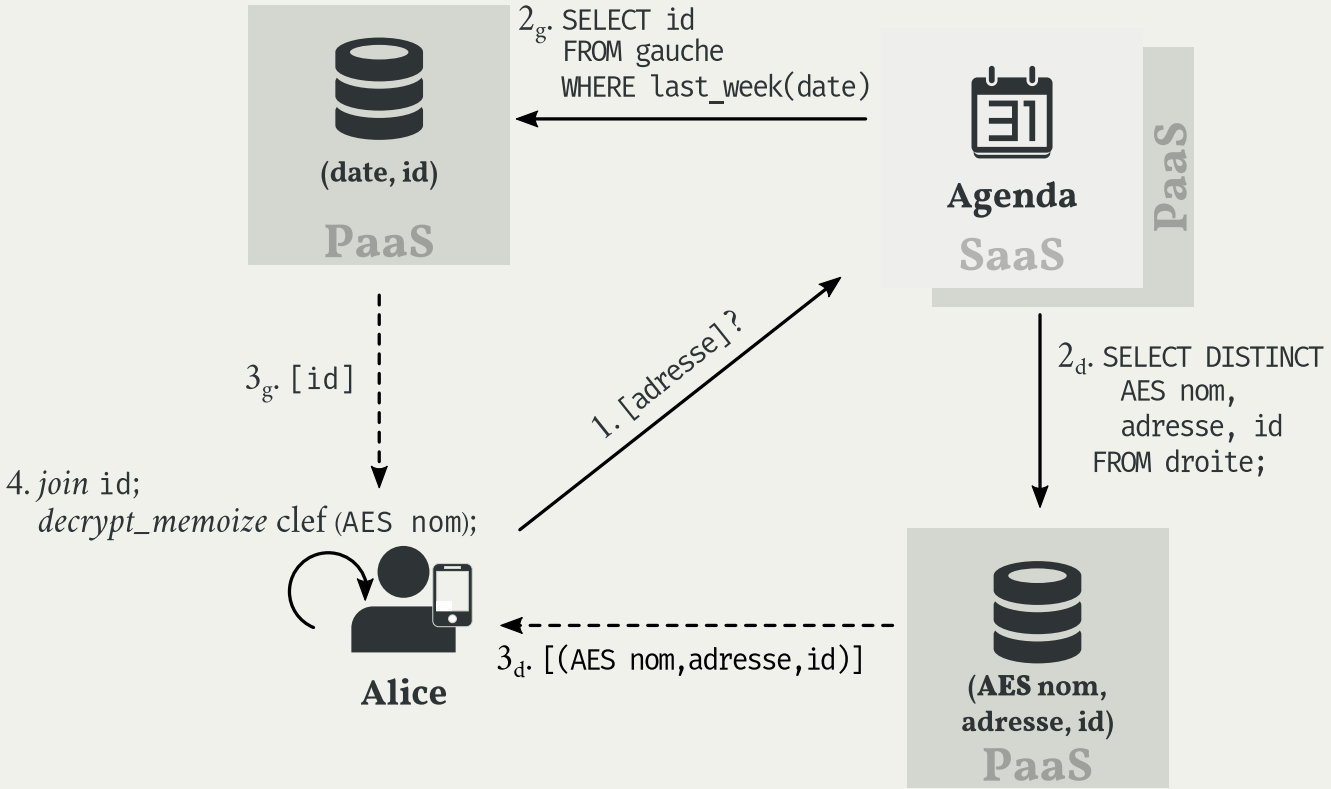


État de l'art :

- **Nombreuses techniques de sécurité** [Gentry, 2009 ; di Vimercati et al., 2013 ; Dwork and Roth, 2014 ; Erlingsson et al., 2014] …
- **Limitées sur l'un des trois objectifs : Sécurité, Nuage, Performance**.

Solution : composer les techniques de sécurité

Composer : Chiffrement + Calcul côté client + Fragmentation verticale



+ **Sécurité** : *nom, (date,adresse)*
+ **Nuage** : Serveur PaaS, BD PaaS
+ **Performance** : Bonne

Composition des techniques de sécurité

Composition des techniques de sécurité **nécessaire** pour application :

- Complexe (manipule des données personnelles)
- Préserve la vie privée
- Profite du nuage

Problèmes :

- Contexte d'utilisation et garanties spécifiques par technique
- Difficile de composer / produire une application correcte

État de l'art sur la composition :

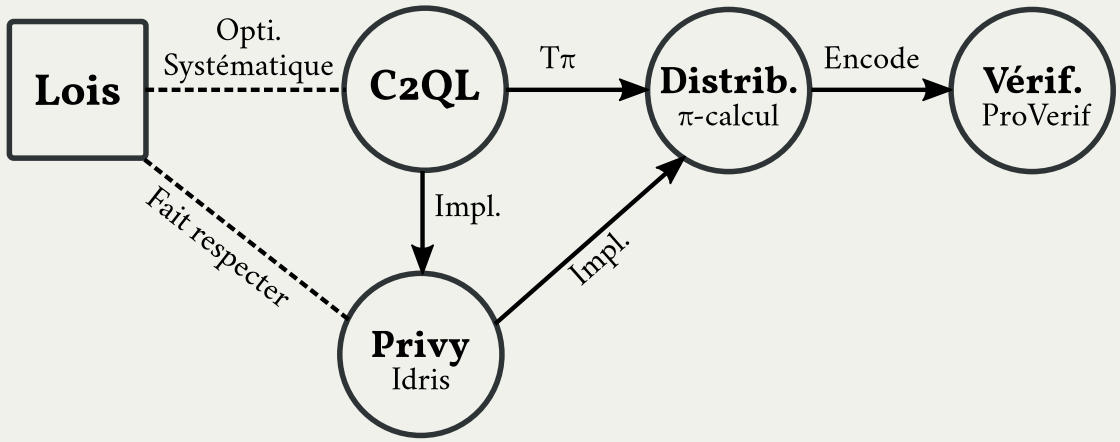
- Préservation vie privée partielle : CryptoDB [Popa et al., 2011]
- Préservation vie privée holistique : Antignac [2015]

Contributions pour la composition

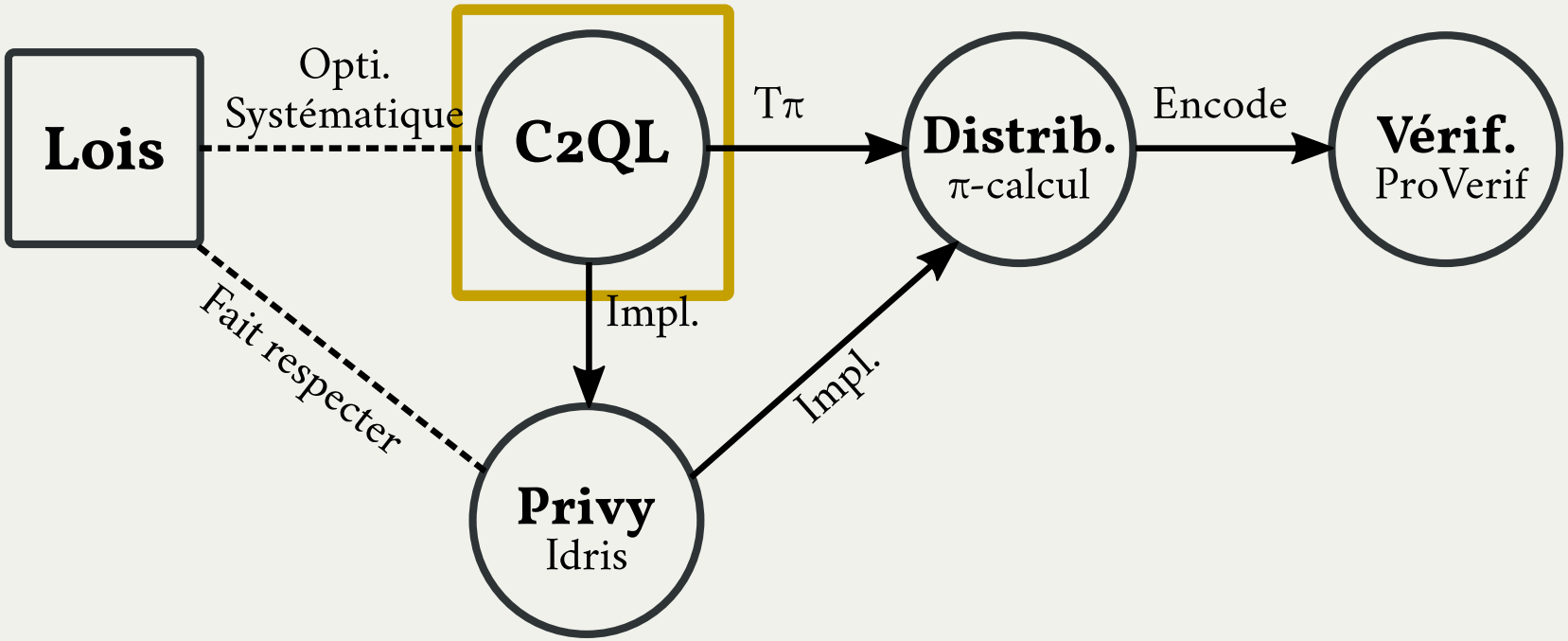
- Programmer
- Optimiser
- Vérifier

Contributions pour la composition

- Programmer
- Optimiser
- Vérifier



Programmer : Le langage C2QL



Objectifs :

- Décrire une application du nuage
- Composer les techniques de sécurités

L'algèbre relationnelle

Basé sur l'algèbre relationnelle :

$\pi_{a_1,\cdots,a_n}$ Projection, `SELECT a1, ..., an`

$\sigma_{p_{a_1,\cdots,a_n}}$ Sélection, `WHERE p(a1, ..., an)`

Requête `[adresse]` :

- Liste des adresses et contacts visités par Alice la semaine dernière.
- $\pi_{nom,adresse} \circ \sigma_{\texttt{last\_week}(date)}$ *rendezvous*
- `SELECT nom, adresse FROM rendezvous WHERE last_week(date);`

Les techniques de sécurité

Spécificateurs de sécurité :

$crypt_{a,\mathbf{c}}$ Chiffrement

$frag_{a_1,\cdots,a_n}$ Fragmentation

Destructeurs de sécurité :

$decrypt_{a,\mathbf{c}}$ Déchiffrement

$defrag_{a_1,\cdots,a_n}(id, id)$ Défragmentation

Requête `[adresse]` :

$$decrypt_{nom,\texttt{AES}} \circ defrag_{date}(\pi_{\emptyset} \circ \sigma_{\texttt{last\_week(}date\texttt{)}}, \pi_{nom,adresse}) \circ frag_{date} \circ crypt_{nom,\texttt{AES}} \quad rendezvous$$

Langage sans tiers

Spécificateurs de sécurité (*crypt*/*frag*) :

- Données inintelligibles
- Calcul sur la BD PaaS

Destructeurs de sécurité (*decrypt*/*defrag*) :

- Données lisibles
- Calcul chez la Cliente

Application SaaS : Intermédiaire entre BD PaaS et la Cliente

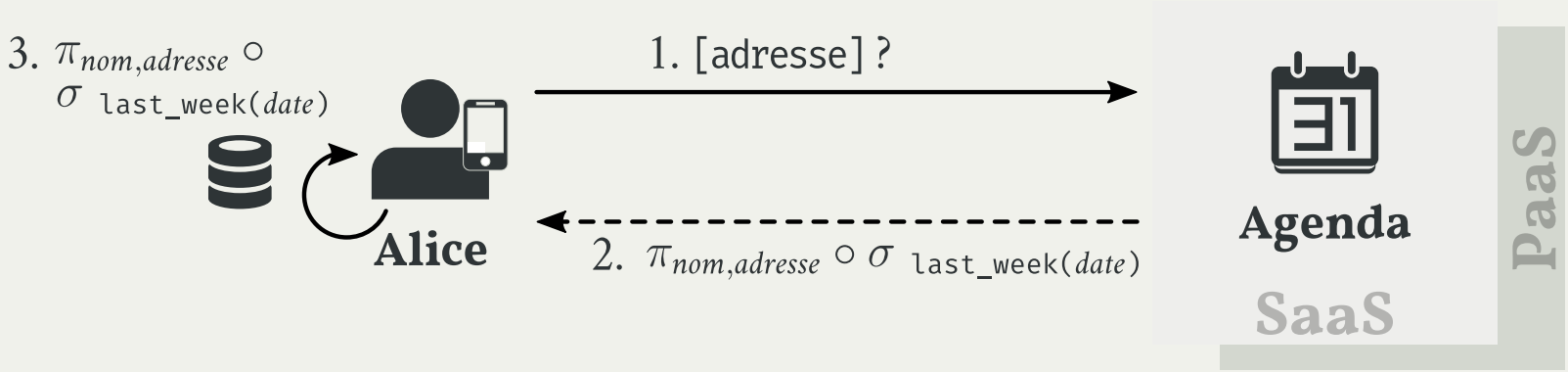
Ex. [adresse] sans sécurité

Absence de spécificateur de sécurité (*crypt*/*frag*) :

- Les rendez-vous ne sont pas protégés.
- Risque pour la vie privée.

$\Rightarrow$ Les rendez-vous sont sauvegardés chez la cliente.

$\pi_{nom,adresse} \circ \sigma_{\texttt{last\_week}(date)}$ *rendezvous*



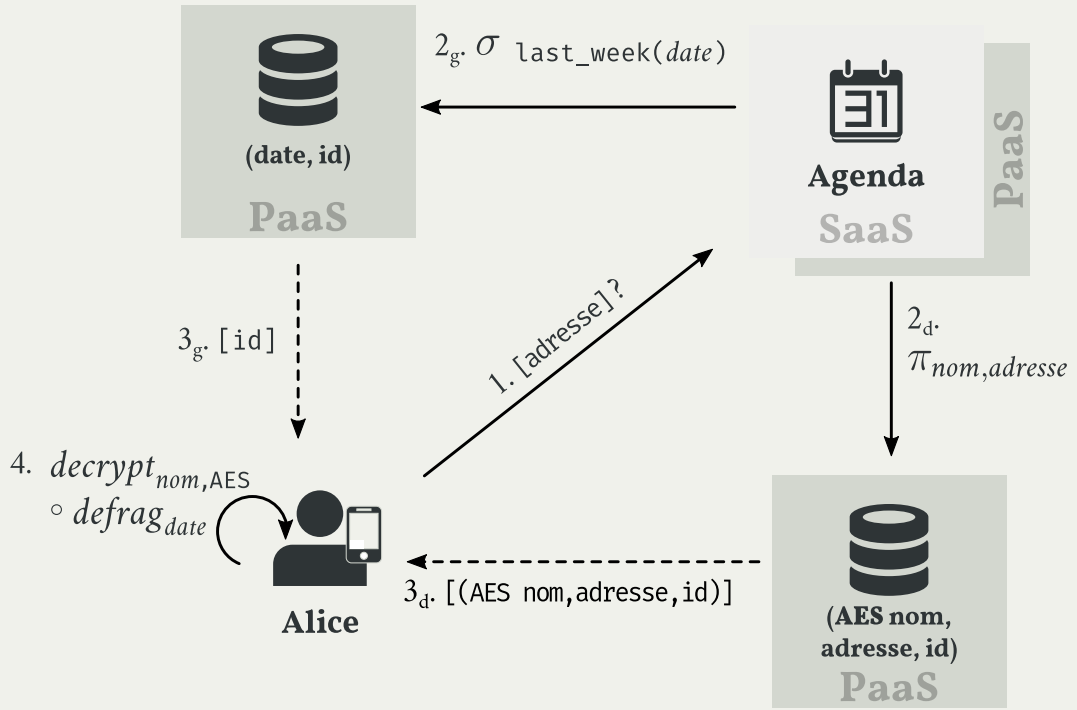
Ex. [adresse] avec techniques

Composition des techniques de sécurité :

- Les rendez-vous sont protégés.
- Pas de risque pour la vie privée

$\Rightarrow$ Les rendez-vous sont sauvegardés dans une BD PaaS et rapatriées chez la Cliente au moment d'un destructeur.

$$decrypt_{nom,\texttt{AES}} \circ defrag_{date}\ (\pi_{\emptyset} \circ \sigma_{\texttt{last\_week}(date)},\ \pi_{nom,adresse}) \circ frag_{date} \circ crypt_{nom,\texttt{AES}}\ \ rendezvous$$



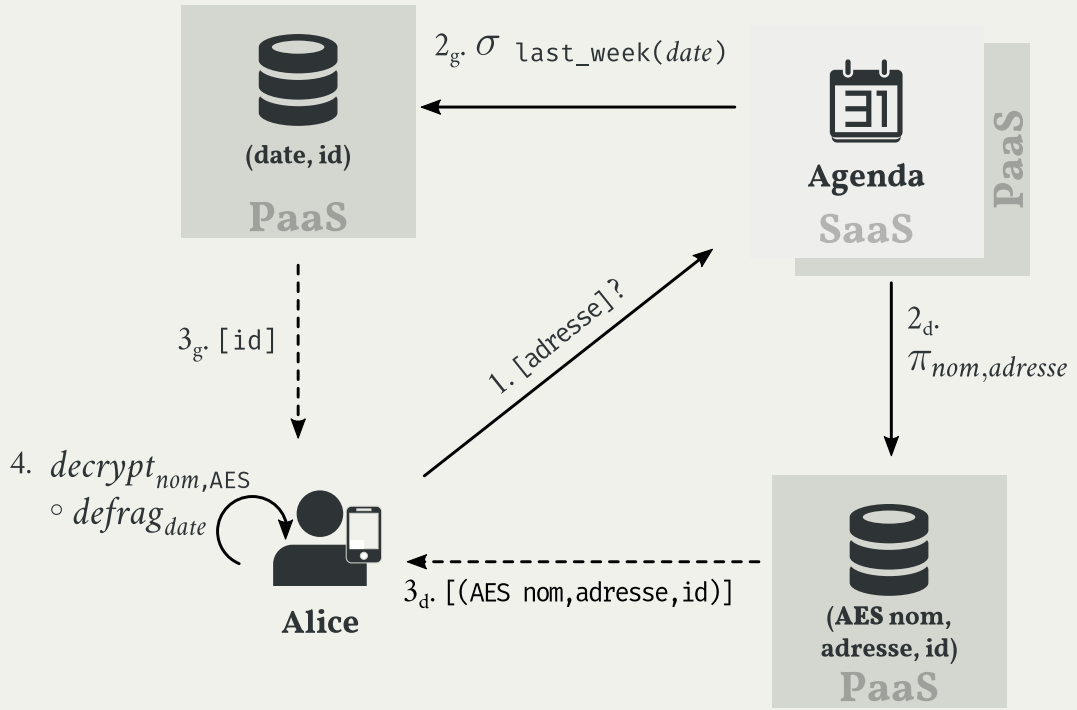
Ex. [adresse] avec techniques

Composition des techniques de sécurité :

- Les rendez-vous sont protégés.
- Pas de risque pour la vie privée

$\Rightarrow$ Les rendez-vous sont sauvegardés dans une BD PaaS et rapatriées chez la Cliente au moment d'un destructeur.

$$
\begin{aligned}
&\mathit{decrypt}_{\mathit{nom},\mathtt{AES}} \\
&\circ\ \mathit{defrag}_{\mathit{date}}\ (\pi_{\emptyset} \circ \sigma_{\ \mathtt{last\_week}(\mathit{date})}\ , \\
&\qquad\qquad \pi_{\mathit{nom},\mathit{adresse}}) \\
&\circ\ \mathit{frag}_{\mathit{date}} \circ \mathit{crypt}_{\mathit{nom},\mathtt{AES}} \qquad \mathit{rendezvous}
\end{aligned}
$$



Maximiser l'utilisation du nuage : Méthodologie

Absence de techniques de sécurité (*crypt*/*frag*) :

+ Requête facile à écrire (algèbre relationnelle)
− Ne profite pas des avantages du nuage (requête locale)

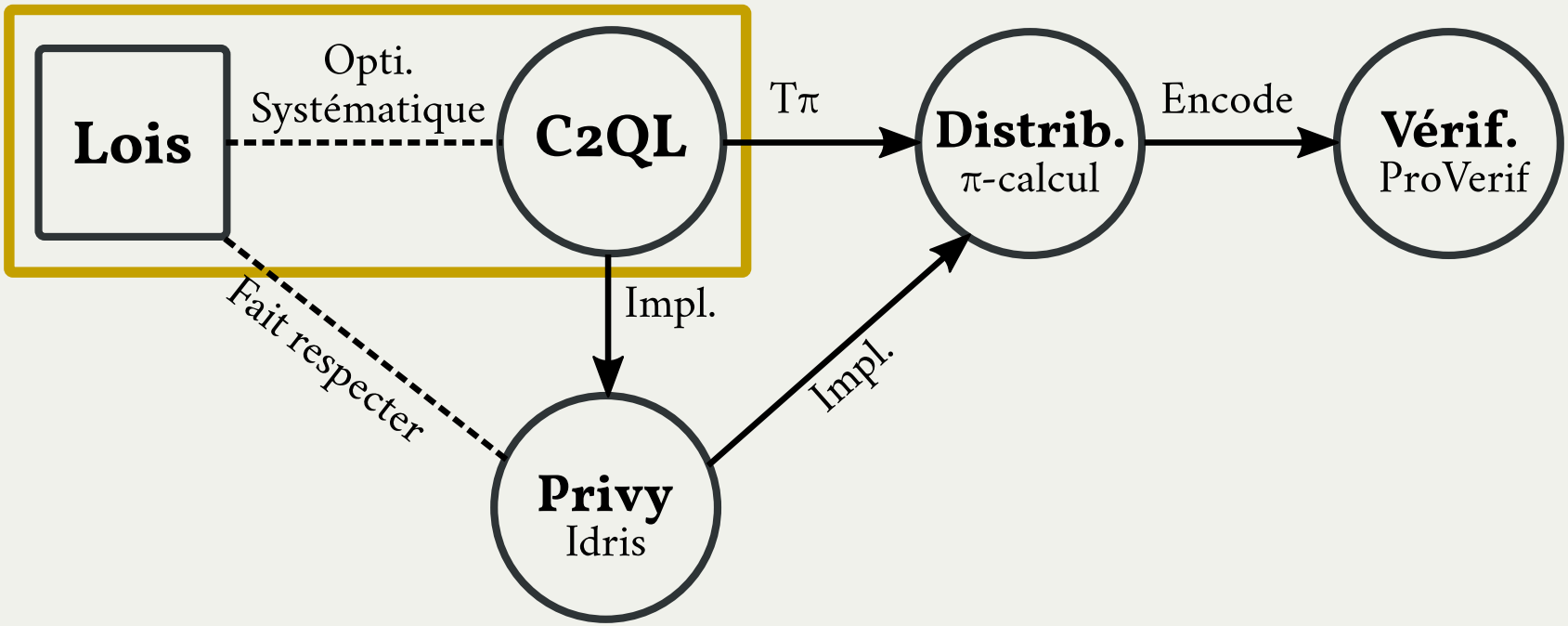
Avec composition des techniques de sécurité :

− Requête difficile à écrire
+ Profite des avantages du nuage

Méthodologie

1. Introduire les techniques dans la requête sans techniques.
2. Pousser les destructeurs à gauche de l'expression pour maximiser l'utilisation du nuage.

Optimiser : Lois algébriques



Objectifs :

- Équivalence entre deux requêtes ($r \equiv s$)
- Réécriture d'un programme pour maximiser l'utilisation du nuage

Équivalence de deux requêtes

Équivalences observationnelles :

- Équivalence de l'algèbre relationnelle [Ullman, 1982]
- r et s produisent le même résultat
- Ex : $\sigma_{p_\delta} \circ defrag_{\delta'} \equiv defrag_{\delta'}\ (\sigma_{p_\delta}, id)$ avec $\delta \subseteq \delta'$

Équivalences de confidentialités :

- r et s ne présentent pas de risques pour la confidentialité
- Ex : $\pi_\delta \circ decrypt_{a,\mathsf{c}} \equiv \pi_\delta$ si $a \notin \delta$

Vue d'ensemble (22 lois) :

- 2 : Introduction techniques sécurité
- 12 : Commutativité algèbre relationnel/destructeurs de sécurité
- 8 : Commutativité constructeur/destructeurs de sécurité

Maximiser l'utilisation du nuage de [adresse]

$$Q_1 \equiv \pi_{nom,adresse} \circ \sigma_{\texttt{last\_week}(date)}$$

Lois d'introduction

$$\equiv \pi_{nom,adresse} \circ \sigma_{\texttt{last\_week}(date)} \circ defrag_{date} \circ frag_{date} \circ decrypt_{nom,\texttt{AES}} \circ crypt_{nom,\texttt{AES}}$$

Lois de sélection

$$\equiv \pi_{nom,adresse} \circ defrag_{date}(\sigma_{\texttt{last\_week}(date)}, id) \circ frag_{date} \circ decrypt_{nom,\texttt{AES}} \circ crypt_{nom,\texttt{AES}}$$

$$\vdots$$

Lois de composition

$$\equiv decrypt_{nom,\texttt{AES}} \circ defrag_{date}(\pi_{\emptyset} \circ \sigma_{\texttt{last\_week}(date)}, \pi_{nom,adresse}) \circ frag_{date} \circ crypt_{nom,\texttt{AES}}$$

Mauvais choix de techniques de sécurité ?

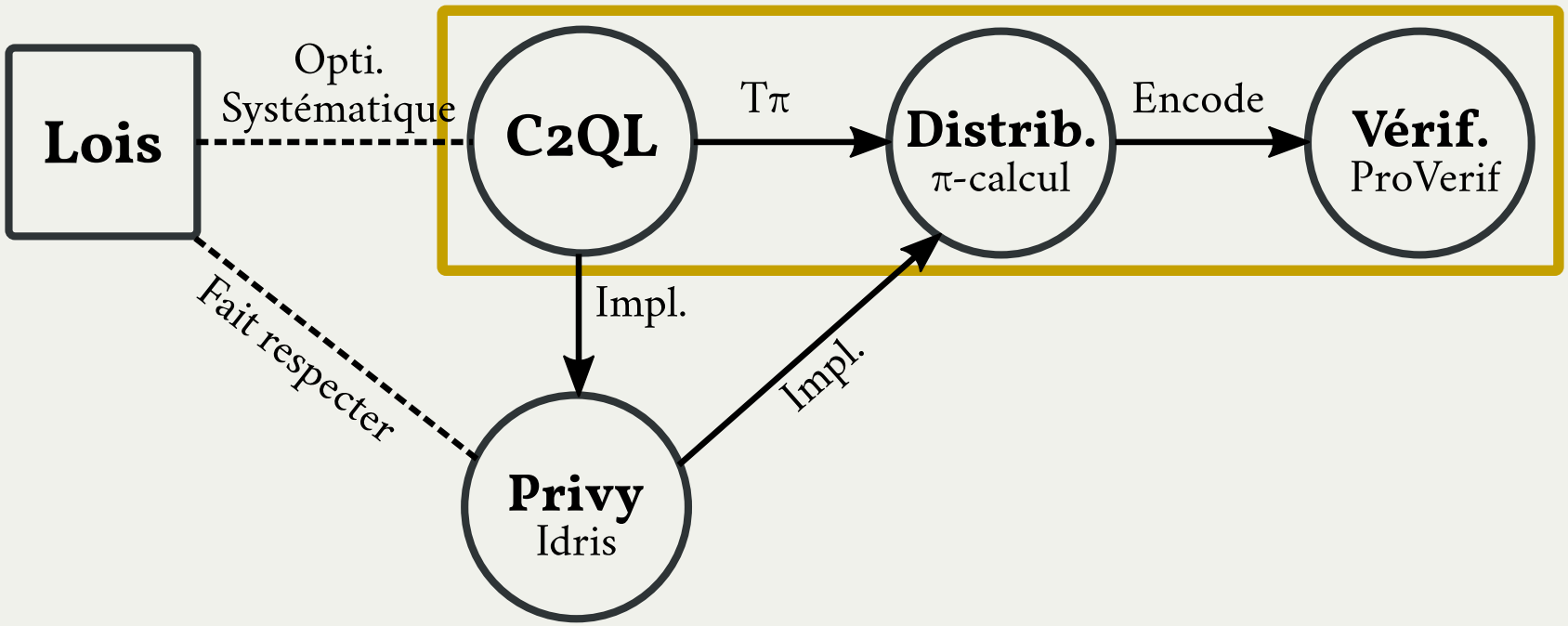
$$\pi_{nom,adresse} \circ \sigma_{\texttt{last\_week}(date)} \circ defrag_{nom} \circ frag_{nom}$$

Fragmentation sur *nom* :

- Fragment gauche (nom, id)
- Fragment droite $(date, adresse, id)$

$\Rightarrow$ Violation des contraintes *nom* et *(date, adresse)*

Vérifier : Garantir la préservation de la vie privée



Objectifs :

- Vérifier le choix des techniques de sécurité
- Garantir la préservation des contraintes de vie privée

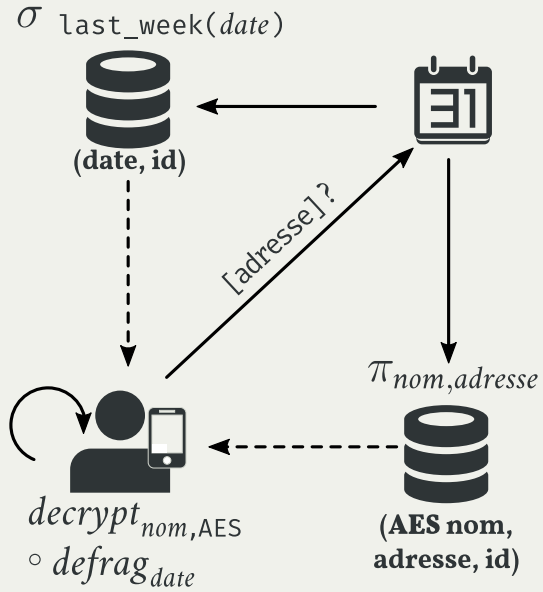
Étape 1 : Traduction en π-calcul

π-calcul [Milner, 1999] :

- Modélisation de systèmes concurrents
- $\overline{x}\langle m\rangle.0 \mid x(m).0$

Tπ :

- Modélisation d'une requête C2QL sur le nuage
- Traduction automatique (récursion sur C2QL) vers le π-calcul
- Stratégie langage sans tiers
- $Agenda \equiv app(url).[url = \texttt{"[adresse]"}](\overline{db_0}\langle url, client\rangle.0 \mid \overline{db_1}\langle url, client\rangle.0)$
- $Frag_d \equiv (\rho r : (\texttt{AES}\ nom, adresse, id))db_1(url, k).[url = \texttt{"[adresse]"}]let\ s = \pi_{nom,adresse}\ r\ in\ \overline{k}\langle s\rangle.0$



Étape 2 : Encodage en ProVerif

ProVerif [Blanchet et al., 2014] :

- Vérificateur de modèles
- Modélise les protocoles de sécurité en π-calcul
- Modélise les techniques de sécurité par réécriture d'un terme
- Analyse l'atteignablité d'un secret [Dolev and Yao, 1983]

Application à C2QL :

- Analyse l'atteignablité des contraintes de vie privée, ex : *nom* et *(date,adresse)*
- Contraintes de vie privée atteignables $\Rightarrow$ Techniques de sécurité inappropriées.

Encodage

Contraintes de vie privée *(date,adresse)* :

```
query attacker(cc_da)
reduc forall an: attribut;
confidentiel_da((brut(d),an,brut(a))) = cc_da.
```

Spécificateur/Destructeur de sécurité (*crypt*/*decrypt*) :

```
fun senc(key, attribut): attribut.
reduc forall a:attribut, k:key; sdec(k, senc(k,a)) = a.
```

Algèbre relationnelle ($\pi_{nom,adresse}$) :

```
forall ad: attribut, an: attribut, aa: attribut;
proj((n,a), (ad,an,aa)) = (unit,an,aa);
```

Application :

$Frag_d \equiv (\rho r : (\mathsf{AES}\ nom, adresse, id))db_1(url, k).[url = \texttt{"[adresse]"}]$

$$let\ s = \pi_{nom,adresse}\ r\ in\ \overline{k}\langle s\rangle.0$$

```
let FragD =
  in (db1, to: channel);
  out(to, proj((n,a), (unit, senc(k, n), brut(a)).
```

Une expression incorrecte ?

$$crypt_{foo,\mathsf{AES}} \quad rendezvous$$

Chiffrement Symétrique des valeurs de *foo* :

- Attribut *foo* n'est pas un élément de *rendezvous*

$\Rightarrow$ Erreur de construction de sécurité

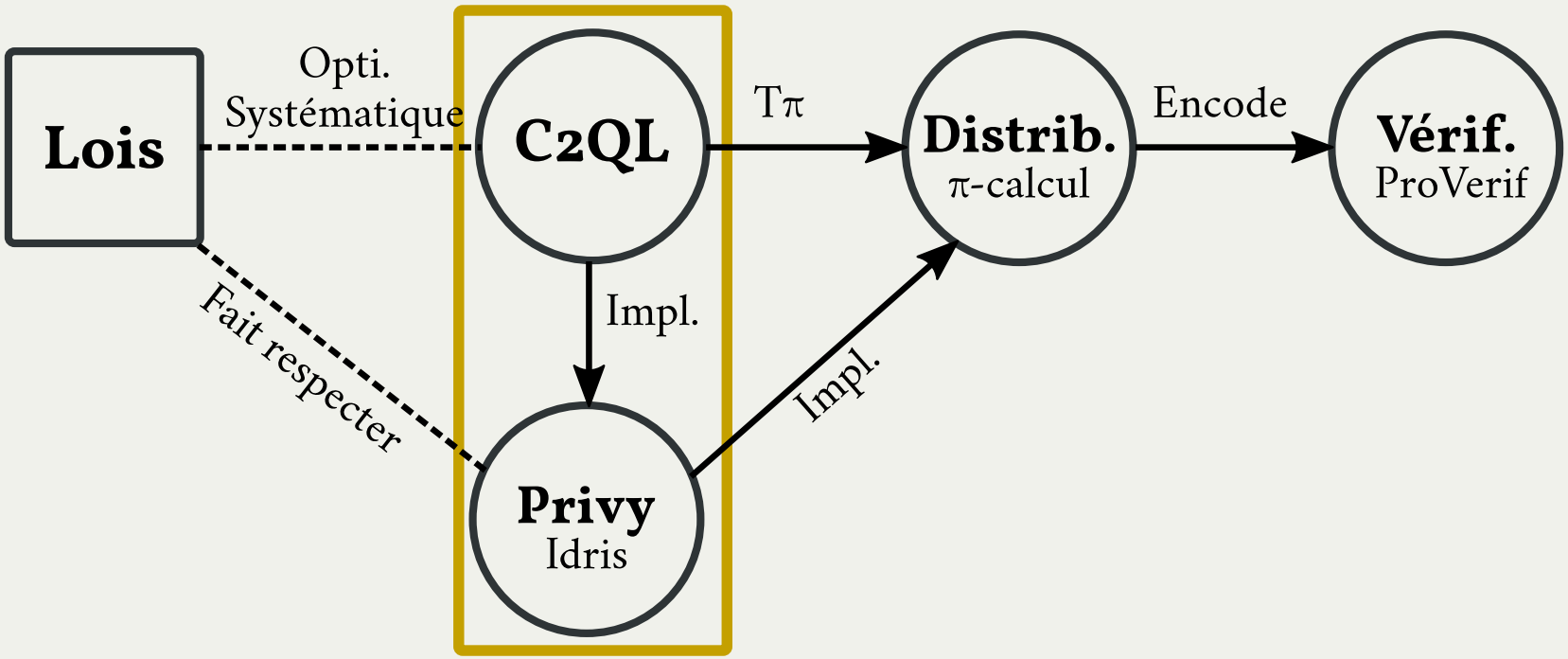
$$defrag_{date}(\pi_{nom}, \pi_{adresse}) \circ frag_{date} \quad rendezvous$$

Projection sur les *nom* dans le fragment de gauche :

- Attribut *nom* n'est pas un élément de *rendezvous*

$\Rightarrow$ Erreur de manipulation des n-uplet

Implémentation



Objectifs :

- Représenter une expression C2QL
- Produire une expression correcte

EDSL de C2QL en Idris

Le langage Idris [Brady, 2013] :

- Langage fonctionnel
- Types dépendants

Avantage d'un EDSL en Idris :

- Exploiter l'analyse syntaxique
- Profiter des types dépendants pour prévenir des erreurs

Exemple : EDSL avec constructeur de projection

```
data Query : Schema → Type where
  -- Appliquer la fonction πδ sur une relation
  -- R de type Δ (avec δ ⊆ Δ), produit une
  -- nouvelle relation R′ de type δ, Ullman, 1982
  π : (δ: Schema) → Query Δ → {auto p: Include δ Δ} → Query δ
```

Conclusion

Constat :

- Développer une application complexe
- Préserver la vie privée
- Profiter du nuage

$\Rightarrow$ Composition des techniques de sécurité **indispensable**

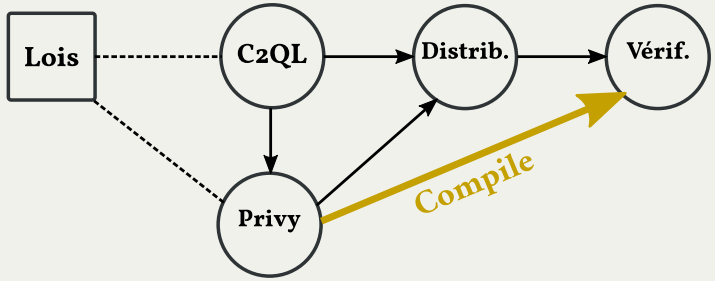
Proposition : Approche complète pour le développeur

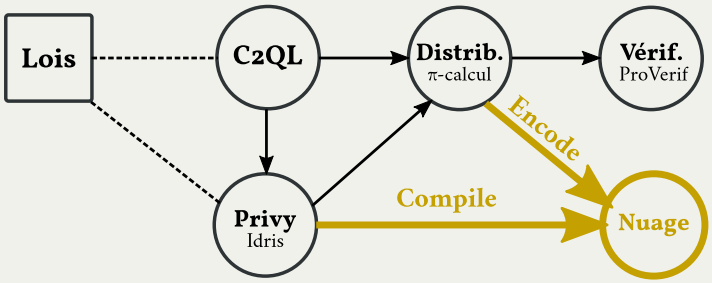
- Programmer : C2QL
- Optimiser : Lois algébriques
- Vérifier : Transformation π-calcul + Encodage en ProVerif
- Implémenter : EDSL Idris

Perspectives à court terme : compléter l'approche

Vérification automatique des erreurs de vie privée :

- Génération de l'encodage en ProVerif
- EDSL Privy → Vérificateur ProVerif





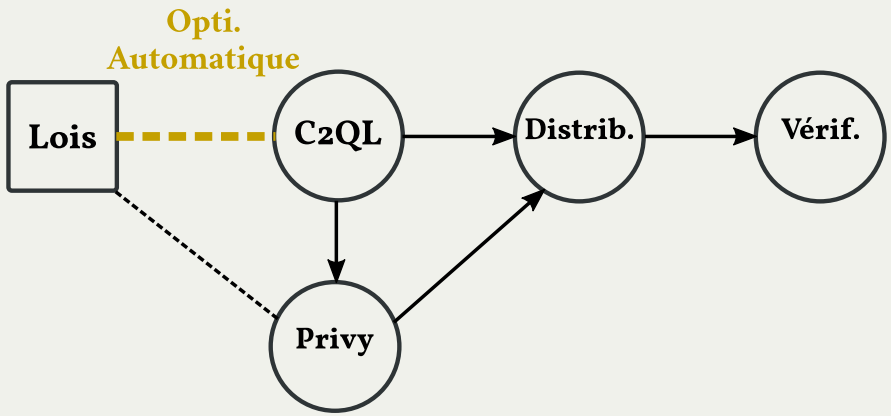
Production d'un programme exécutable sur le nuage :

- Rendre C2QL opérationnel
- Faire adopter l'outil
- EDSL Privy → Compilateur JavaScript

Autre perspective

Optimiser automatiquement la requêtes :

- Développeur n'a pas à faire le développement équationnel
- Définir des stratégies d'application des lois



Publications

Plusieurs idées et figures présentées dans ici ont fait l'objet des publications suivantes :

A Language for the Composition of Privacy-Enforcement Techniques Ronan-Alexandre Cherrueau, Rémi Douence et Mario Südholt dans *RATSP, pages 1037–1044, Helsinki, Finland, August 2015*. IEEE.

Enforcing Expressive Accountability Policies Ronan-Alexandre Cherrueau et Mario Südholt dans *WETICE, pages 333–338, Parma, Italie, Jun 2014*. IEEE Computer Society.

Adapting Workflows Using Generic Schemas : Application to the Security of Business Processes Ronan-Alexandre Cherrueau, Mario Südholt et Omar Chebaro dans *CloudCom, pages 519–524, Bristol, Royaume-Uni, Decembre 2013*. IEEE Computer Society.

Reference Monitors for Security and Interoperability in OAuth 2.0 Ronan-Alexandre Cherrueau, Rémi Douence, Jean-Claude Royer, Mario Südholt, Anderson Santana de Oliveira, Yves Roudier et Matteo Dell'Amico dans *Workshop, SETOP 2013, pages 235–249, Egham, UK, September 2013*. Springer.

Flexible Aspect-Based Service Adaptation for Accountability Properties in the Cloud Ronan-Alexandre Cherrueau, Omar Chebaro et Mario Südholt dans *VariComp'13, pages 13–18, Fukuoka, Japan, March 2013*. ACM.

Merci !

Questions

References (1)

Aggarwal, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Motwani, R., Srivastava, U., Thomas, D., and Xu, Y. (2005). Two can keep A secret : A distributed architecture for secure database services. In *CIDR*, pages 186–199.

Antignac, T. and Métayer, D. L. (2015). Trust driven strategies for privacy by design. In *Trust Management IX - 9th IFIP WG 11.11 International Conference, IFIPTM 2015, Hamburg, Germany, May 26-28, 2015, Proceedings*, pages 60–75.

Blanchet, B., Smyth, B., and Cheval, V. (2014). Proverif 1.88 : Automatic cryptographic protocol verifier, user manual and tutorial.

Brady, E. (2013). Idris, a general-purpose dependently typed programming language : Design and implementation. *Journal of Functional Programming*, 23:552–593.

di Vimercati, S. D. C., Erbacher, R. F., Foresti, S., Jajodia, S., Livraga, G., and Samarati, P. (2013). Encryption and fragmentation for data confidentiality in the cloud. In *Foundations of Security Analysis and Design VII - FOSAD 2012/2013 Tutorial Lectures*, pages 212–243.

References (2)

Dolev, D. and Yao, A. C. (1983). On the security of public key protocols. *IEEE Trans. Information Theory*, 29(2):198–207.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Erlingsson, Ú., Pihur, V., and Korolova, A. (2014). RAPPOR : randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 1054–1067.

Fournet, C., Kohlweiss, M., Danezis, G., and Luo, Z. (2013). ZQL : A compiler for privacy-preserving data processing. In *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013*, pages 163–178.

Gentry, C. (2009). *A fully homomorphic encryption scheme*. PhD thesis, Stanford University. crypto.stanford.edu/craig.

Menezes, A., van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. CRC Press.

References (3)

Milner, R. (1999). *Communicating and mobile systems - the Pi-calculus*. Cambridge University Press.

Popa, R. A., Redfield, C. M. S., Zeldovich, N., and Balakrishnan, H. (2011). Cryptdb : protecting confidentiality with encrypted query processing. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles 2011, SOSP 2011, Cascais, Portugal, October 23-26, 2011*, pages 85–100.

Solove, D. J. (2006). A taxonomy of privacy. *University of Pennsylvania law review*, pages 477–564.

Ullman, J. D. (1982). *Principles of Database Systems, 2nd Edition*. Computer Science Press.

Van Blarkom, G., Borking, J., and Olk, J. (2003). Handbook of privacy and privacy-enhancing technologies. *Privacy Incorporated Software Agent (PISA) Consortium, The Hague*.

Distribution π-calcul

L	$::=$	a, b, c	Nom
	$\mid$	r, s, t	Relation
P	$::=$	$a(L_1, \ldots, L_n).P$	Réception
	$\mid$	$\overline{a}\langle L_1, \ldots, L_n \rangle.P$	Émission
	$\mid$	$P_1 \mid P_2$	Composition parallèle
	$\mid$	$(\nu a)P$	Restriction
	$\mid$	$!P$	Réplication
	$\mid$	$[a = b]P$	Garde
	$\mid$	$(\rho r : \delta)P$	Localisation
	$\mid$	$let\ s = (Q\ L)\ in\ P$	Application C2QL
	$\mid$	0	Processus terminé

Distribution : Ex avec [adresse]

$$Agenda \equiv app(url).[url = \texttt{"[adresse]"}]$$
$$(\overline{db_0}\langle url, client\rangle.0 \mid \overline{db_1}\langle url, client\rangle.0)$$

$$Frag_g \equiv (\rho rdv_g : (date, id))db_0(url, k).[url = \texttt{"[adresse]"}]$$
$$let\ s = \pi_{\emptyset}\ rdv_g\ in\ \overline{k}\langle s\rangle.0$$

$$Frag_d \equiv (\rho rdv_d : (\texttt{AES}\ nom, adresse, id))db_1(url, k).[url = \texttt{"[adresse]"}]$$
$$let\ s = \pi_{nom,adresse}\ rdv_d\ in\ \overline{k}\langle s\rangle.0$$

$$Alice \equiv \overline{app}\langle\texttt{"[adresse]"}\rangle.client(r_1).client(r_2).$$
$$let\ s = defrag_{date}\ r_1\ r_2\ in$$
$$let\ t = decrypt_{nom,\texttt{AES}}\ s\ in$$
$$let\ u = \sigma_{nom\ \texttt{LIKE}\ `C*'}\ t\ in\ 0$$

$$\texttt{[adresse]} \equiv (\nu app)(\nu db_0)(\nu db_1)(\nu client)$$
$$!Agenda\ |!Frag_g\ |!Frag_d \mid Alice$$

Vérifier la préservation des contraintes de vie privée

Modèle d'attaquant :

- Observe les n-uplets qui transitent sur les canaux
- Utilise les destructeurs pour atteindre une contrainte de vie privée
- Ex : Applique *defrag* sur toutes les combinaisons de n-uplets pour reconstruire la contraintes *(date,adress)*

Représentation des n-uplets :

- Table (*rendezvous*) modélisée par son type (*date,nom,adresse*)
- Raisonnement globale

Approche :

- Lister les types qui sont une contrainte de vie privée
- Spécifier comment un spécificateur modifie le type
- Spécifier comment l'algèbre relationnelle modifie le type

L'ADT Privy

```
qadr : Privy RendezVousEnv SafeRendezVousEnv [N,A]
qadr = do
   -- description de l'environnement :
   crypt N AES
   frag [D]
   -- description de la requête :
   r0 ← query 0 ( π [ ] . σ (lastWeek D) )
   r1 ← query 1 ( π [N,A] )
   return ( decrypt N (aes "the-key") $ defrag r0 r1 )
```