Projet FUI (18e appel)
SecurOCaml

Réunion de clôture
Vendredi 14 septembre 2018, Paris

Participants

- Département de l'Essonne : Estelle SBORDONE

PMEs

- OCamlPro : Fabrice LE FESSANT, Vincent LAVIRON, Thomas BLANC, Muriel SHAN SEI FAN
- Lexifi : Alain FRISCH
- Trust-In-Soft : Benjamin MONATE, David MAISON
- Safe River : Véronique DELEBARRE

Laboratoires

- CEA – Centre de Saclay : Virgile PREVOSTO
- ENSTA / INRIA : Michel MAUNY

Le projet SecurOCaml

Conception d'un environnement pour développer des applications de sécurité en OCaml

- Projet porté par OCamlPro SAS
- Labellisé par le Groupe thématique Logiciel Libre (GTLL) du pôle de compétitivité Systematic
- Financé par le Fonds Unique Interministériel (18e appel FUI)
  - 844 k€ d'aide sur 1,6 M€ de dépenses
  - CG91 (OCamlPro) et BPIFrance
- Début du projet : 16 janvier 2015
- Durée du projet : 36 mois (*+ 12 mois de délai complémentaire accordé par BPIFrance*)
- *Clôture septembre 2018, avec dépenses à prendre en compte jusque fin 2018*

SecurOCaml

- **Objectif** :

  Fournir un **langage de programmation** combinant ***sûreté de fonctionnement*** et ***sécurité,*** avec une **trousse à outils d'audit de code**.

OCaml

- Langage développé à l'INRIA depuis 30 ans
- Le ”typage statique fort” d'OCaml:
  - La rapidité de développement (peu de bugs/tests)
  - La performance à l'exécution
  - La portabilité (y compris vers JavaScript)
  - La maintenabilité du code dans le temps
- Communauté en pleine croissance
  - OCamlPro, OCamlLabs, ocaml.org, etc.
  - OPAM : de 200 à 750 contributions depuis 2013

Secure-OCaml

- L'**ANSSI** a commandé deux études de trois ans sur la sécurité des langages :
  - L'étude **JavaSec** ⟶ quasi-impossible de sécuriser Java
  - L'étude **LaFoSec** ⟶ OCaml surclasse en sécurité les autres langages fonctionnels, mais :
    - Recommandations d'évolution pour améliorer encore le langage
    - Besoin d'outillage pour l'audit de code pour la certification (CESTI)

Pourquoi s'intéresser à la sécurité des applications en OCaml ?

Réunion de clôture du projet SecurOCaml (FUI18)
Vendredi 14 septembre 2018, Paris

OCaml

- Cloud: Citrix XenServer $\longrightarrow$15% du marché de la virtualisation (Amazon). Le logiciel qui contrôle chaque machine Xen est écrit en OCaml.
- Finance: Jane Street Capital, l'une des 10 plus grandes firmes de "proprietary trading" avec 10 Milliard$ de transaction par jour. Toute leur infrastructure est écrite en OCaml.
- Logiciel: Microsoft diffuse Terminator, outil d'analyse des drivers sous Windows. Terminator est écrit en OCaml.

OCaml dans l'avionique

- Scade 6 chez Esterel-Technologies :
  - Compilateur KCG migré de C à OCaml
  - Un des seuls compilateurs certifié au niveau A de la norme DO-178B
- Frama-C et Astrée, utilisés chez Airbus
  - Frama-C: framework d'analyse de code C développé au CEA, appliqué à prouver la correctoin fonctionnelle de 10k lignes de C
  - Astrée: détecteur d'erreurs à runtime, développé à l'ENS, appliqué à 1M lignes de C

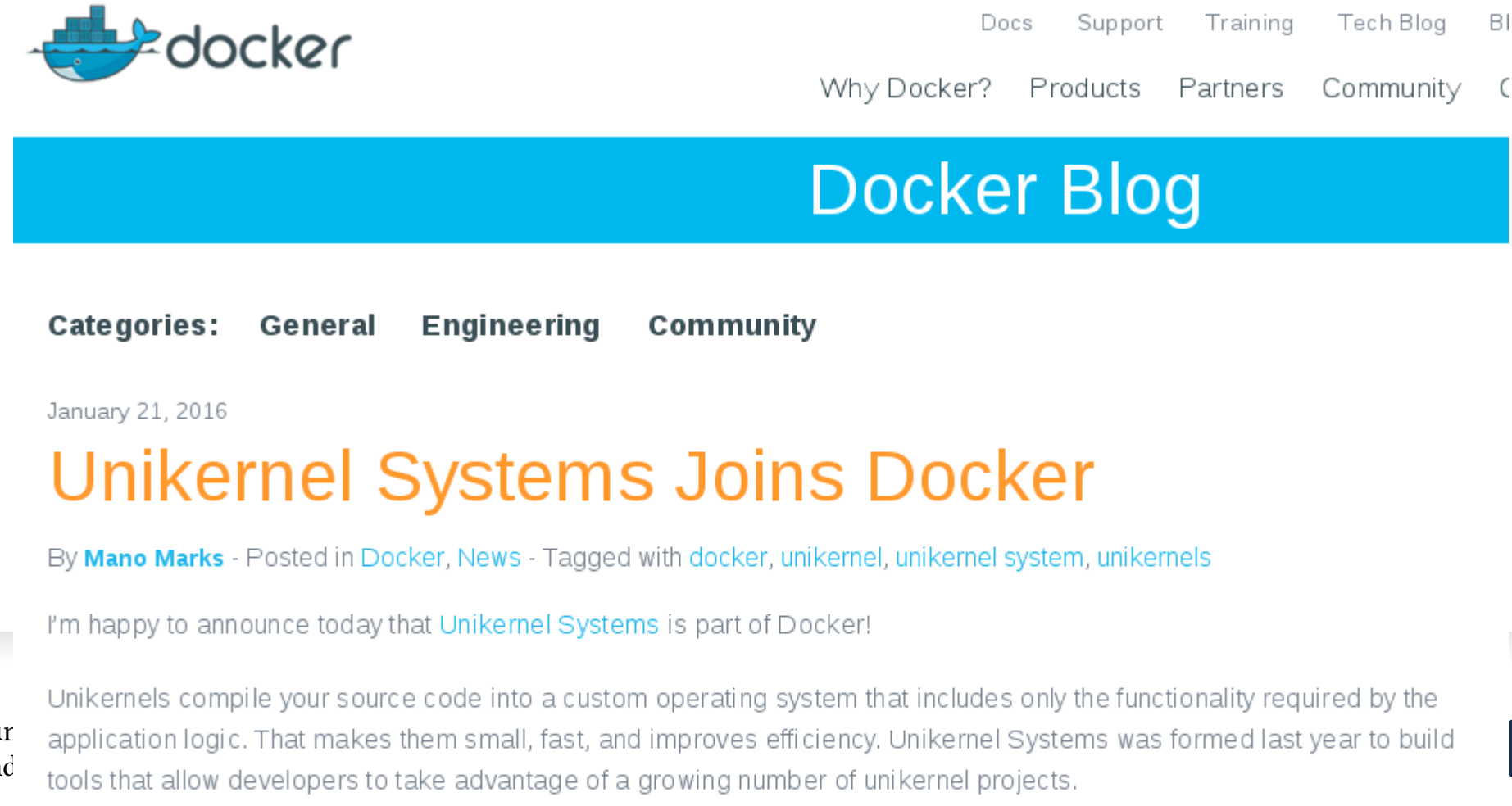
OCaml chez Facebook

- 20 mars 2014: Facebook annonce qu'ils dévelopent un nouveau langage, Hack, écrit en OCaml.
  Rejoint par Flow le 18 nov. 2014

Another long-term design goal -- more implied by the language's construction than explicitly stated by Facebook -- is making Hack safer with fewer inherent vulnerabilities than PHP. In addition to the type-checking constraints now available in the language, there's also the way Hack was written; rather than being coded in C, as PHP was, Hack was designed using the OCaml language -- the same platform used by the Xen Project folks to develop its secure-by-design Mirage OS. PHP has had its share of low-level vulnerabilities over the years, not all of them obvious (or specific to that platform), so it makes sense to adopt a new design methodology that makes casual exploits more difficult to pull off.

OCaml chez Docker

21 janvier 2016 : Docker rachète les dévelopeurs de micro-conteneurs en OCaml



Réur
Vend

PRO

Tour de table des partenaires

Quatre PME

- **OCamlPro** (porteur du projet ⟶ outillage OCaml)
- **LexiFi** (gestion de contrats financiers en OCaml, Bloomberg, SimCorp, contributeur OCaml)
- **SafeRiver** (audit en sécurité, porteur LaFoSec)
- **TrustInSoft** (outils de Cyber-Sécurité en OCaml)

Trois académiques

- INRIA (maintenance OCaml)
- CEA List (utilisateur important d'OCaml)
- ENSTA (contributeur important à OCaml)

2 « observateurs »

- *TrustedLabs (sécurité et cartes à puce)*
- *ANSSI (sécurité des infrastructures informatiques)*

OCamlPro - porteur du projet

- Bureau d'études et éditeur logiciel avec 4 domaines d'activités :

  Prototypage de pointe
  Méthodes formelles
  Outillage OCaml
  Blockchains (Tezos / TzScan / Liquidity)

- Clients principalement à l'international :
  - Finance : JaneStreet Capital
  - Blockchain : Fondation Tezos, Dynamic Ledger Solutions
  - Éditeurs : Citrix, Facebook, Fondation OCaml

Effectifs

- Janvier 2015 : 6 CDI + 4 CDD
- Novembre 2018 : 17 CDI + 3 CDD
- Recrutements liés directement au projet :
  - Michaël Laporte : CDD [Analyseur de style]
    ⟶ embauché en CDI fin 2016 [Ocp-lint]
  - Vincent Laviron : CDD [Analyseur d'exceptions (~10 HM restant)]
    ⟶ embauché en CDI mi-2017
  - Thomas Blanc : thèse CIFRE ⟶ INRIA
    ⟶ embauché en CDI en octobre 2017
  - Pierrick Couderc : thèse CIFRE ENSTA
    ⟶ embauché en CDI en décembre 2017

Bilan à T0+36

- Derniers livrables OCamlPro
  - (2.2) Définition d'un sous-langage sécurisé : janvier 2018
  - Analyseur statique / analyses modulaires : décembre 2017 (https://github.com/OCamlPro/ocp-analyzer)
  - Ocp-lint : vérification automatique des recommandations, tests par les partenaires mi-2018 (https://github.com/OCamlPro/typerex-lint)
- Dépenses : ~60 HM prévus ⟶ +60 HM consommés

Perspectives

- Applications à la blockchain :
  - Logiciel Tezos (blockchain http://tezos.com/), prototype développé chez OCamlPro
    - Contraintes de sécurité importantes : système ouvert permettant de transférer de la cryptomonnaie
    - Contrats avec la Fondation Tezos (Suisse)
    - Spinoff « Nomadic Software » (2 anciens d'OCamlPro ⟶ 20 recrutements en IDF en 2018-2019)
  - Projet FUI24 MoneyTrack pour appliquer Tezos au marché de la consommation dirigée (WiziPay, Fintech Enterprise) – label des Pôles Finance-Innovation et Systematic (GTSI)

LexiFi

- Éditeur de logiciels pour la gestion de produits financiers dérivés et structurés
- Effectifs : 18 salariés (+38% depuis janvier 2015)
- Utilisateur d'OCaml depuis 2001; contributeur direct depuis 2007
- Clients majoritairement internationaux (voir planche suivante)

LexiFi

Environs 30 clients, largement internationaux.

- Clients directs (banques, asset managers, brokers etc.)
  Barclays (UK), HSBC (UK),
  Deutsche Bank (CH, LUX, BE), ABN Amro (NL),
  Natixis (FR), Credit Mutuel CIC (FR)...
- Intégrateurs de notre technologie :
  Bloomberg (USA), SimCorp (DK), Fact(DE), etc.

LexiFi

Retombées du projet 1/3

- La sécurité du langage d'implémentation a levé un frein à la vente pour les éditeurs tiers qui intègrent les composants de LexiFi et les grosses institutions financières (audits et questionnaires de sécurité).
- Augmentation récente du poids des questions de sécurité dans les gros appels d'offres bancaires.
- Aspect critique pour intégrateurs de techno.

LexiFi

Retombées du projet 2/3

- La participation à des projets collaboratifs a amélioré la compétitivité de LexiFi sur le marché du travail : recrutements OCaml facilités.

- Un recrutement sénior qui se consacre de manière significative aux extensions et à la maintenance d'OCaml. Deux stages encadrés autour du thème du projet collaboratif.

LexiFi

Retombées du projet 3/3

- Utilisation directe de certains livrables : ppx , analyseur code mort, nouvelles versions d'OCaml, etc.
- Amélioration de la productivité.

LexiFi

Contributions 1/2

- 2.2 (ss langage sécurisé): discussion et validation. Proposition d'extension (en cours) d'OCaml pour permettre de définir des sous-langages via des annotations et avertissements dans la compilateur.
- 3.1: nombreuses contributions au projet open-source OCaml (4.03, 4.04, 4.05, mais aussi 4.06, 4.07). 2 ingénieurs LexiFi membres de l'équipe de développement d'OCaml.

LexiFi

Contributions 2/2

- 3.2: mise en place techno PPX et outils open-source, gros succès dans la communauté.
- 4.2: détection code mort (1 stage; utilisation intensive en interne, qq utilisateurs extérieurs).
- 5.2: vérification données structurées (1 stage en cours; projet d'open-sourcer une partie de notre extension « types runtime »).

TRUST IN SOFT

FUI Secure-OCaml

Réunion de clôture

14 Septembre 2018

bpifrance

Essonne LE DÉPARTEMENT — TERRE D'AVENIRS —

OCaml PRO

Présentation de TrustInSoft

- société française créée en 2013 issue du CEA
- éditeur de logiciel basé à Paris
- acteur de la sécurité et de la sûreté des logiciels
- la moitié du chiffre d'affaire aux États-Unis
- 15 employés à temps plein (8 en 2015)

Principal produit: TrustInSoft Analyzer

- analyseur statique de code source C et C++

TRUST IN SOFT

OCaml chez TrustInSoft

- apporte des garanties mathématiques sur le code de TrustInSoft Analyzer
- est le langage principal:
  - pour TrustInSoft Analyser (= 335kloc[1])
  - pour des outils en interne (= 25kloc[1])
  - pour remplacer des scripts bash compliqués

[1] `git ls-files '*.ml*' | xargs ocamlwc -c`

TRUST IN SOFT

Intérêts de SOCaml chez TrustInSoft

- augmenter la qualité de la base de code
    - pour nous, développeurs
- augmenter la sécurité de TrustInSoft Analyzer, fournissant ainsi une chaîne complète de certification sécuritaire
    - pour vous, clients

Participation de TrustInSoft

- gestion du projet (T1.1)
- a participé à la définition du sous-langage SecurOCaml (T2)
- rôle d'utilisateur des outils développés dans le projet
- responsable du sous-projet 6 : cas d'étude et retours d'expérience (T6)[2]

[2] livrable 6.1.1 et 6.1.2 ont été fusionnés en un seul livrable

Perspective d'avenir

- pouvoir utiliser en production les outils d'audit de qualité (ocp-lint, dead-code-analyzer, ocp-analyzer, ...)
- continuer de participer aux retours d'expériences pour améliorer ces outils

Cette catégorie d'outils est indispensable pour un langage de programmation et il n'existe que peu (voire pas) d'outils maintenus pour OCaml.

SafeRiver

- Safety and Cyber Security for Embedded Systems
  - Formal Verification Tools
  - Static Analysis Tools
    - Adhoc
    - Generic
  - Les outils sont développés en OCaml
- 16 personnes
  - Recrutements sur la période [2015-2017] : 3 dont 1 directement alloué au projet
- Clients : Alstom, Ansaldo, ANSSI, RATP, THALES, Veoneer

SAFE RIVER
Safety & Security Forge

Contributions

- Référentiel des Règles (langages, outils) pour la sécurité (issu de LaFoSec)
    - Impact sur la définition d'un sous-langage sécurisé
- Mise en œuvre des règles sur un développement interne
- Profils d'application des règles en regard des exigences normatives et des guides de bonnes pratiques pour les systèmes critiques (e.g. ISO 26262, Misra)
- Justification du sous langage sécurisé en regard des exigences normatives
- Mise à jour du référentiel

Bilan sur les livrables à T0+36

- L2.1.1 Mise à jour des recommandations
    - Livré en 01/2016 – relatif à la versions Ocaml 4.02.3
- L2.1.2 Etude de l'applicabilité des recommandations
    - Livré en 02/2017
- L2.1.3 Préparation à la définition d'un sous langage sécurisé
    - Livré en 10/2017
- - L2.1.2 (v2) Mise à jour des recommandations et traçabilité avec les référentiels normatifs

Impact

- Interactions avec les partenaires au travers des outils OCaml
- Durant la période, plusieurs outils (SR) ont été développés en Ocaml et diffusés vers des clients finaux mais pas d'applicatifs
- Liens entre utilisation de Ocaml pour les outils et les processus de qualification des outils pour les référentiels normatifs en cours d'étude.
    - Apport du langage sur la fourniture des evidences de preuve

SecurOCaml: Bilan CEA

Réunion de clôture | Virgile Prevosto

CEA Tech List

- Institut de recherche
  - Laboratoire de Sûreté et Sécurité des Logiciels : outils d'analyse de code et méthodes formelles, développés en OCaml : Frama-C et BinSec.
- List : 700 personnes, LSL : 30 personnes (stable)
- Principaux partenaires : Airbus, Dassault Aviation, Thales, Bureau Veritas
- Partenariat issu du projet : OCamlPro (projet sur les Blockchains)
- Le renforcement de la robustesse du code de Frama-C à l'aide des outils et méthodes issues de SecurOCaml rend la plateforme plus attractive pour une utilisation dans des contextes fortement normés (aéronautique, critères communs,...)

Objectifs initiaux

- 14 h.mois
- Participation T2.2 (Définition sous-langage sécurisé)
- Participation T4.1 (Analyseur statique d'exception)
- Participation SP6 (Cas d'étude), rédaction d'un rapport d'évaluation avec TrustInSoft

Recommandations de codage dans Frama-C

- Évaluation de la pertinence et faisabilité de la mise en œuvre de chacune des règles de codage du livrable 2.1 dans le développement de Frama-C
- Amélioration d'un certain nombre de points du cycle de développement sur cette base

Outil dead_code_analyzer

- But: détection de code mort
- Bonnes performances (20 secondes sur l'analyse de Frama-C)
- informations intéressantes pour un refactoring de la plateforme
- ne semble fonctionner qu'avec OCaml 4.03
- manque de configurabilité (liste blanche de fonctions exportées pour liaison dynamique)

Outil ocp-analyzer

- But: détection de problèmes dans la gestion des exceptions
- problème de passage à l'échelle

Outil ocp-lint

- But: détection de problèmes de style de codage
- Bonnes performances (90 secondes sur l'analyse de Frama-C)
- Configuration des règles assez facile, bonnes pistes d'amélioration de la base de code
- tests effectués avec OCaml 4.02.3 (plus vieille version avec laquelle Frama-C est compatible)
- 3 bugs mineurs rapportés lors de l'évaluation de l'outil

Outils extérieurs au projet

- `Crowbar`: générateur de tests aléatoires (fuzzing)
  - expériences concluantes
  - utilisation dans les tests d'intégration de Frama-C en cours
- `ocamllint`: détection de problèmes de style de codage
  - moins flexible qu'`ocp-lint`
  - n'offre pas de diagnostics supplémentaires pertinents vis-à-vis de Frama-C

Bilan du projet

- amélioration significative du processus de développement de Frama-C
- une mise à niveau des outils vis-à-vis des versions récentes d'OCaml serait utile
- SecurOCaml a contribué à l'attractivité de Frama-C dans des contextes industriels critiques

ENSTA-ParisTech

- **École généraliste d'ingénieurs**
  - sous la tutelle du Ministère de la Défense
  - école d'application de l'École Polytechnique
- **Partenariats industriels**
  - industrie de la défense
  - partenariats stratégiques en 2017 : Naval Group, Safran, EY, AKKA Technologies, EDF, Gendarmerie Nationale, Air Liquide
- **6 laboratoires de recherche**
  - 77 enseignants-chercheurs, ~ 110 doctorants, ~ 880 étudiants
  - **U2IS** : Unité d'Informatique et d'Ingéniérie des Systèmes (12 EC, 23 doctorants)
- **Participants:**
  - **Michel Mauny** : détachement ENSTA-PT -> 31/07/2016, Inria-Paris depuis
  - **Florent Balestrieri (postdoc)** : embauché au 01/01/2016, démission en 10/2017, suite à des difficultés avec l'ENSTA-ParisTech

Sous-projet 5

- 5.1 : Vérificateur de typage du bytecode OCaml (ENSTA)
  - prototype réalisé, mais non intégrable dans un environnement de développement ou d'exécution
- 5.2 : Vérificateur de typage de données sérialisées (ENSTA)
  - bibliothèque de programmation générique (*présentation OCaml 2016, publication en cours de soumission*)
  - algorithme de vérification de compatibilité de données sérialisées avec le type attendu

Partenariats issus du projet

- OCamlPro, LexiFi, Inria, CEA List sont membres du Consortium Caml, et bientôt de la Fondation OCaml, montée par Michel Mauny
- Trust-in-Soft et CEA List seront bientôt membres du Club Alt-Ergo, initié et animé par OCamlPro
- La Fondation OCaml et OCamlPro travaillent sur le projet Learn-OCaml, financé par la Fondation Tezos

Bilan sur les livrables à T0+36

Sous-Projet 1
Gestion de projet et Diffusion

- 1.1 Gestion de Projet (OCamlPro)
  - Accord de Consortium signé
  - Réunion annuelle précédente : 18 février 2016
- 1.2 Diffusion et valorisation (OCamlPro)
  - Site web du projet
    - http://ocamlpro.github.io/SecurOCaml/
  - Forge du projet
    - https://github.com/OCamlPro/SecurOCaml

Sous-projet 2
Recommandations et Sous-Langage

- 2.1 Mise à jour des recommandations (SafeRiver)
  - Réunions Safe-River/Inria en 2015/2016
  - Livrable dans le dépôt
  - Mises à jours jusqu'à la version 4.02.3
- 2.2.1 Définition du sous-langage (OCamlPro & SafeRiver)
- 2.2.1bis Notion d'applicabilité pour la définition du sous-langage : *prolongement sur fin 2018*

Sous-projet 3

- 3.1 Évolution de la distribution OCaml (Inria)
  - Nouvelles versions : 4.03.0 (avril 2016) et 4.04.0 (nov 2016)

- 3.2 Évolution des outils (LexiFi)
  - Ajout de préprocesseurs (ppx)
  - Points d'extensions (annotations)
  - Intégrés dans la version courante d'OCaml

Sous-projet 4

- 4.1 Outils de vérification statique (OCamlPro)
  - Détecteur statique d'exceptions
    - Transformation d'un programme OCaml complet en hyper-graphe, pour supprimer l'ordre supérieur
    - Problèmes à résoudre : passage à l'échelle des environnements, précision des domaines abstraits
  - Analyseur de style (ocp-lint)
    - Détection de l'application des recommandations
    - Source sur Github, présenté à OCaml'2016

Sous-projet 4 (suite)

- 4.1 Outils de vérification statique (OCamlPro)
  - Vérificateur de cohérence de l'inférence :
    - Formalisation d'un grand sous-ensemble du système de types d'OCaml
    - Détection de programmes mal inférés et potentiellement non sûr.
  - Sources sur Github, thèse CIFRE

Sous-projet 4 (suite)

- 4.2 Détection de code inutilisé (LexiFi)
  - Détecteur de code mort, releasé sur Github

- 4.3 Interprète alternatif (Inria)
  - Ré-orientation vers une sémantique opérationnelle exécutable
  - Recrutement de l'ingénieur

Sous-projet 5

- 5.1 : Vérificateur de typage du bytecode OCaml (ENSTA)
  - prototype réalisé, mais non intégrable dans un environnement de développement ou d'exécution
- 5.2 : Vérificateur de typage de données sérialisées (ENSTA)
  - bibliothèque de programmation générique (*présentation OCaml 2016, publication en cours de soumission*)
  - algorithme de vérification de compatibilité de données sérialisées avec le type attendu

Sous-projet 6
Cas d'étude

- Trust-In-Soft (+ CEA + TrustedLabs)
  - Test des recommandations sur TIS Analyzer et Frama-C

.

Livrables

N	Intitulé	Responsable	Date de livraison prévue	Date de livraison effective	Commentaires
1.1.1	Kick-off et accord de consortium	OCamlPro	15/7/2015	30/6/2016	ok
1.1.3	Rapport annuel d'activité	OCamlPro	15/01/2017	23/01/2017	ok
1.2.1	Mise en place de sites web et forges	OCamlPro	15/07/2015	11/12/2015	ok
2.1.1	Mise à jour des recommandations pour 4.00.0	SafeRiver	15/01/2016	17/01/2016	ok
2.1.2	Mise à jour des recommandations pour 4.02.0	SafeRiver	15/07/2017	17/01/2016	En avance
2.2.1	Définition du sous-langage SecurOCaml v1	OCamlPro	15/07/2016	03/04/2018	ok
2.2.1 bis	Définition du sous-langage SecurOCaml v1 - applicabilité	SafeRiver	15/07/2016	03/12/2018	ok

Livrables

N	Intitulé	Responsable	Date de livraison prévue	Date de livraison effective	Commentaires
3.1.1	Release d'OCaml 4.03	Inria	15/01/2016	25/04/2016	ok
3.1.2	Release d'OCaml 4.04	Inria	15/01/2017	04/11/2016	ok
3.2.1	Préprocesseurs et points d'extension	LexiFi	15/07/2016	15/01/2016	ok
3.2.2	OPAM et ocp-build sécurisés	OCamlPro	15/01/2017	15/01/2017	OPAM 2.0 uniquement
4.1.1	Analyseur d'exception, version initiale	OCamlPro	15/01/2016	15/01/2017	ok
4.1.2	Analyseur d'exception, version SecurOCaml v1	OCamlPro	15/01/2017		ok

Livrables

N	Intitulé	Responsable	Date de livraison prévue	Date de livraison effective	Commentaires
4.2.1	Détecteur de code mort, version initiale	LexiFi	15/01/2016	15/01/2016	ok
4.2.2	Détecteur de code mort, version SecurOCaml v1 *(version existante adaptée couche OO non pertinente pour sous-langage)*	OCamlPro	15/01/2017	3/4/2018	ok
4.3.1	Interprète alternatif, version initiale	Inria	15/07/2016		-
4.3.2	Interprète alternatif, version SecurOCaml v1	Inria	15/01/2017		-

Livrables (fin)

N	Intitulé	Responsable	Date de livraison prévue	Date de livraison effective	Commentaires
5.1.1	Prototype d'interprète typé	ENSTA	15/01/2017		ok
5.2.1	Prototype de vérificateur de données	ENSTA	15/01/2017	20/12/2016	ok
6.1.1.	Retours d'évaluation de SecurOCaml v1	CEA / TrustInSoft	15/01/2017	13/04/2018	ok