



Générer les tests de régression par apprentissage sur les traces d'exécution

Fabrice Bouquet – Julien Botella – Frédéric Tamagnan



- Introduction
 - Motivations
 - Jalons (fait, future)
 - Exemple file rouge
- PHILAE
 - L'IA dans PHILAE
 - Process global
 - Outillage PHILAE
 - Clustering
 - Visualisation et sélection
 - Apprentissage
 - Génération des scénarios de tests
- Etudes de cas, résultats et perspectives

- Introduction
 - Motivations
 - Jalons (fait, future)
 - Exemple file rouge
- PHILAE
 - L'IA dans PHILAE
 - Process global
 - Outillage PHILAE
 - Clustering
 - Visualisation et sélection
 - Apprentissage
 - Génération des scénarios de tests
- Etudes de cas, résultats et perspectives

- La transformation Agile renforce les besoins en tests de régression :
 - Les changements sur le logiciel sont plus fréquents
 - La non-qualité n'est pas une option
 - L'automatisation de ces tests est requise
 - Il s'agit de prévenir les risques en production dans l'usage réel des utilisateurs

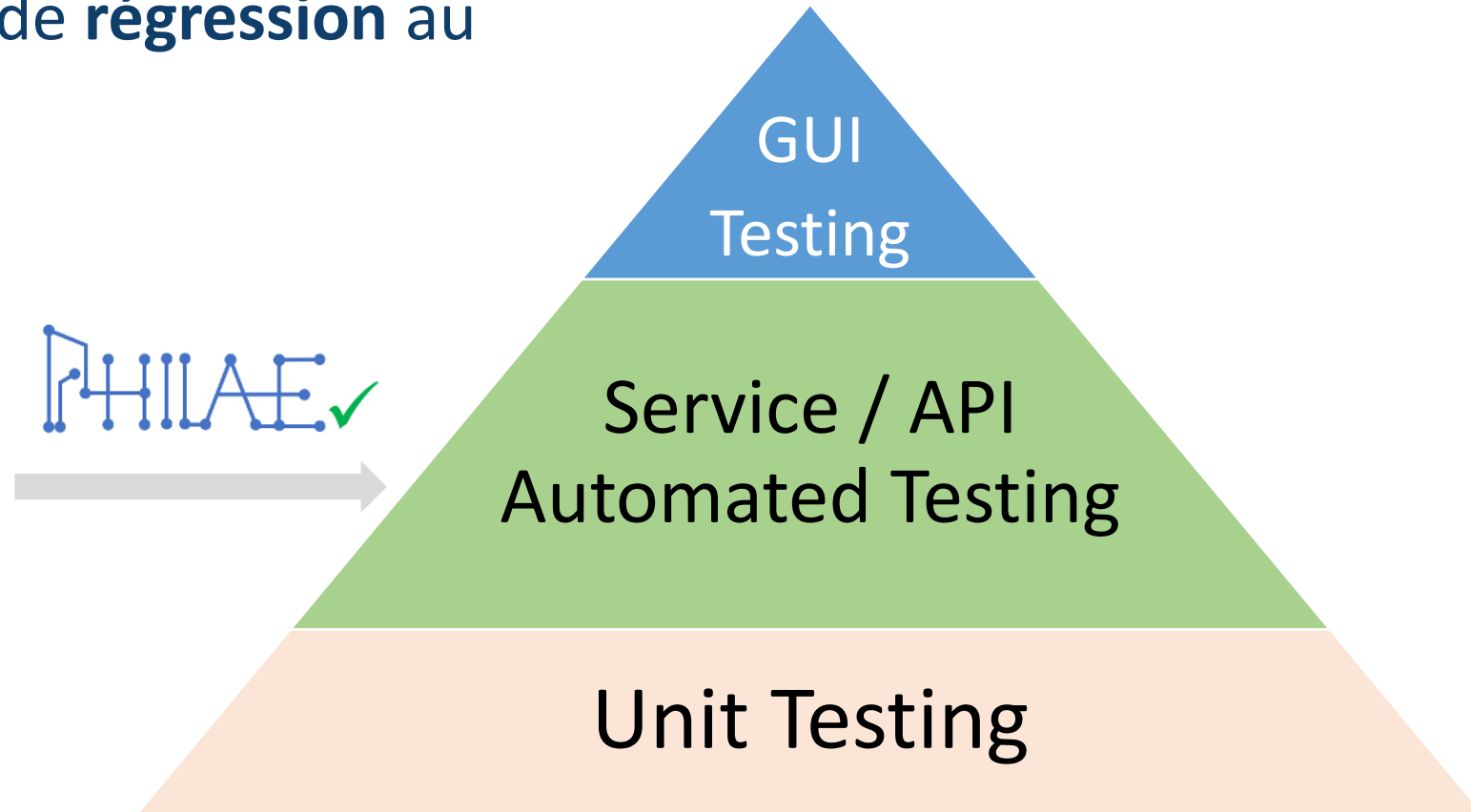
PHILAE est un projet de recherche qui vise une automatisation complète, par l'apprentissage automatique, de la création et de la maintenance des tests automatisés de régression.

Projet ANR PHILAE : Octobre 2018 – Septembre 2022

Deux jalons :

- 1. Apprendre les cas de test de régression à partir des traces d'exécution et produire les scripts automatisés abstraits manquants**
→ Ce que nous vous présentons
- 2. Implémenter automatiquement les scripts automatisés concrets et leurs données de test**
→ Ce sur quoi nous travaillons actuellement

PHILAE vise les tests **fonctionnels automatisés de régression** au niveau **API**



Scanette

- Développement de l'Université de Franche-Comte
- Application web simulant l'utilisation de scanette
- Tests fonctionnels existants

→ Mise au point des outils PHILAE



Données de log des traces d'exécution : [4818 traces](#) avec plus de 65 000 appels au total

Actions utilisateur (scanette) :

- 'unlock'
- 'scan'
- 'delete'
- 'add'
- 'transmit'
- 'openSession'
- 'closeSession'
- 'pay'
- 'abandon'

```
# timestamp, sessID, object, action, inputs, output
1570573649196, 41, scan3, abandon, [], 0
1570573649191, 42, scan1, transmit, [checkout0], 0
1570573649197, 42, scan1, abandon, [], 0
1570573649355, 43, scan1, unlock, [], 0
1570573649358, 42, checkout0, openSession, [], 0
1570573649996, 43, scan1, scan, [5410188006711], 0
1570573650366, 43, scan1, scan, [5410188006711], 0
1570573650366, 42, checkout0, add, [3570590109324], 0
1570573650389, 44, scan2, scan, [3046920010856], 0
1570573651369, 42, checkout0, closeSession, [], 0
1570573652376, 42, checkout0, pay, [68.27], 0
1570573655132, 40, scan0, scan, [7640164630021], -2
1570573656245, 44, scan2, scan, [3270190022534], 0
1570573656633, 43, scan1, scan, [3474377910724], 0
...
```

- Introduction
 - Motivations
 - Jalons (fait, future)
 - Exemple file rouge
- PHILAE
 - L'IA dans PHILAE
 - Process global
 - Outillage PHILAE
 - Clustering
 - Visualisation et sélection
 - Apprentissage
 - Génération des scénarios de tests
- Etudes de cas, résultats et perspectives

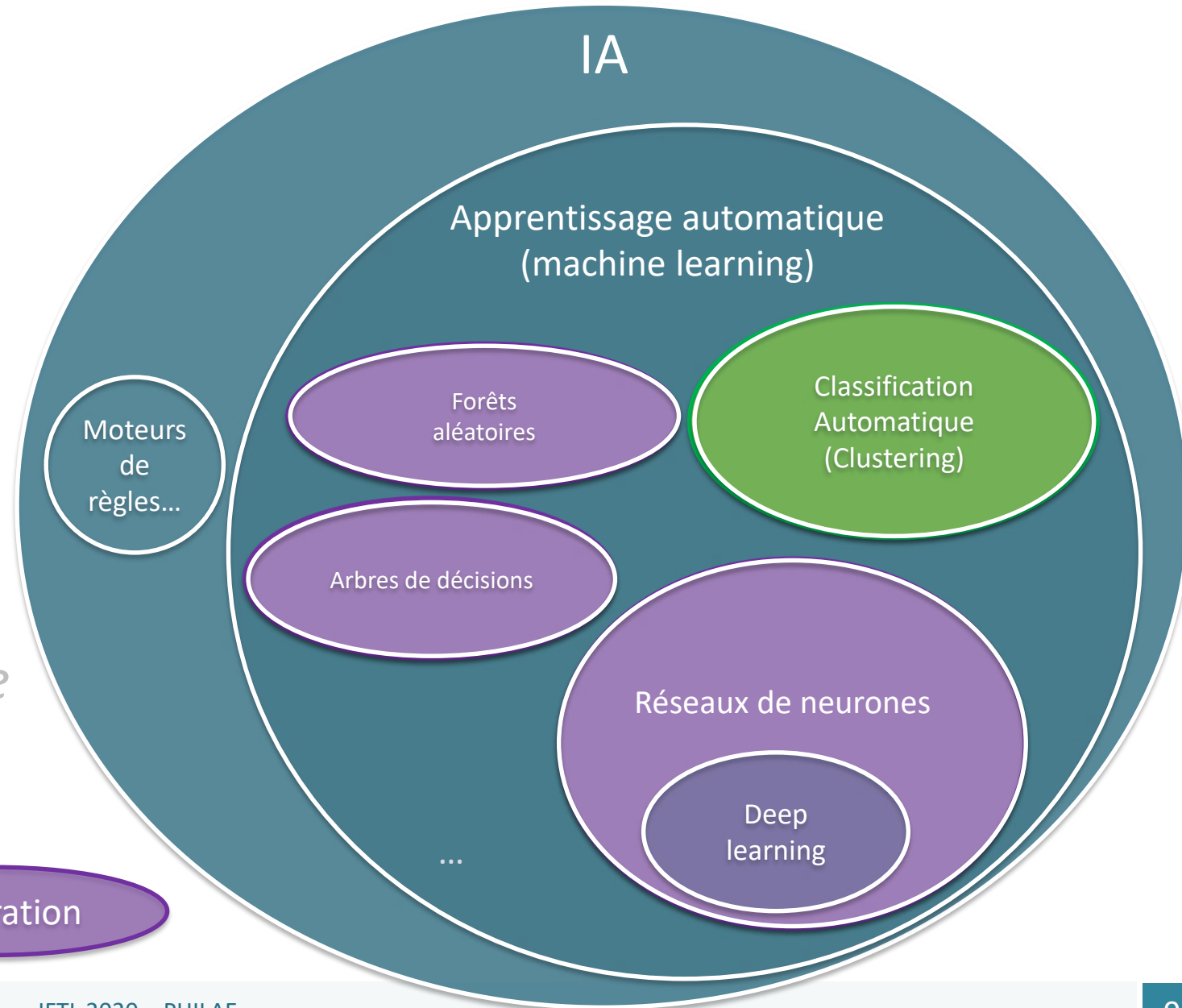
L'intelligence artificielle (IA) est :

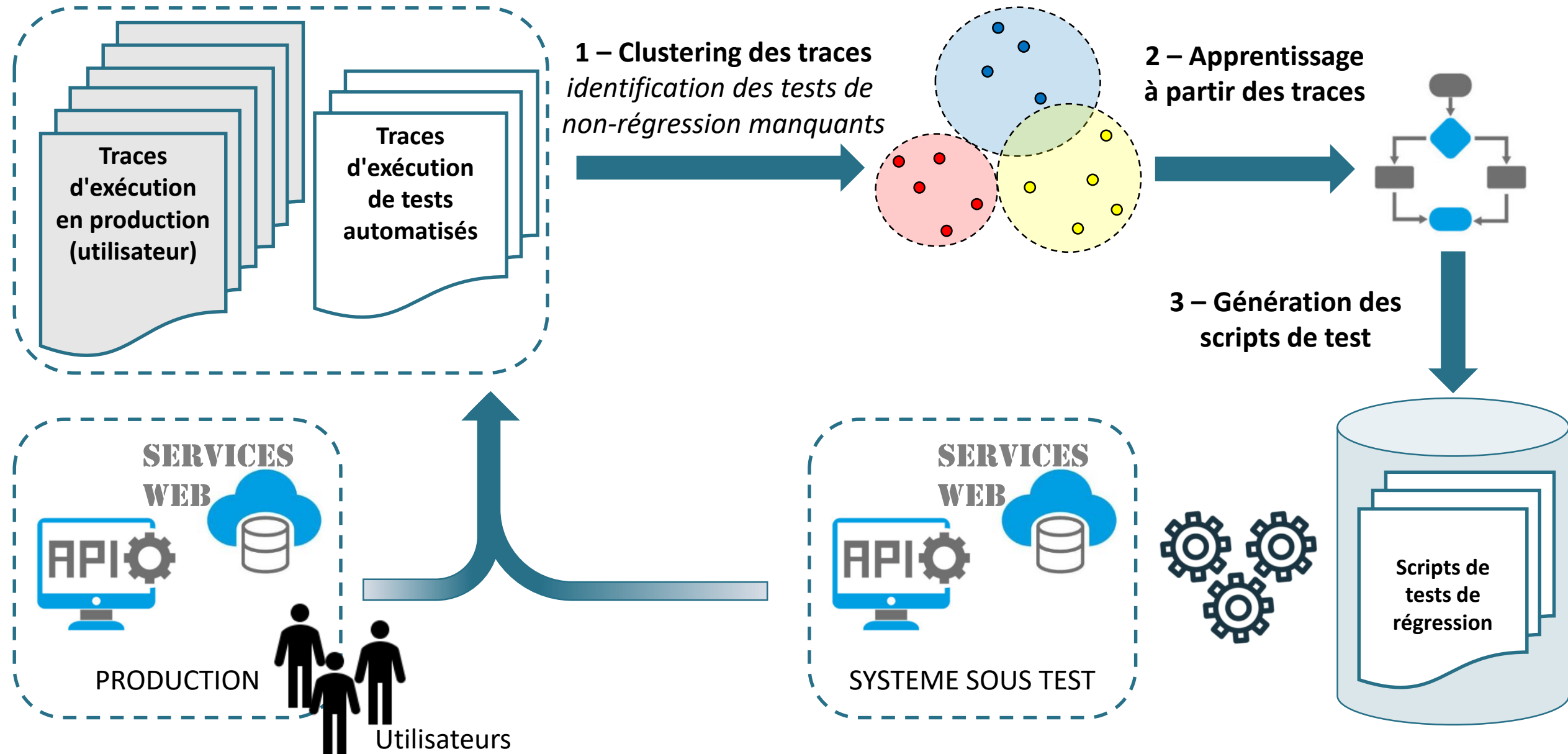
« l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence »

Encyclopédie Larousse

Selection

Génération



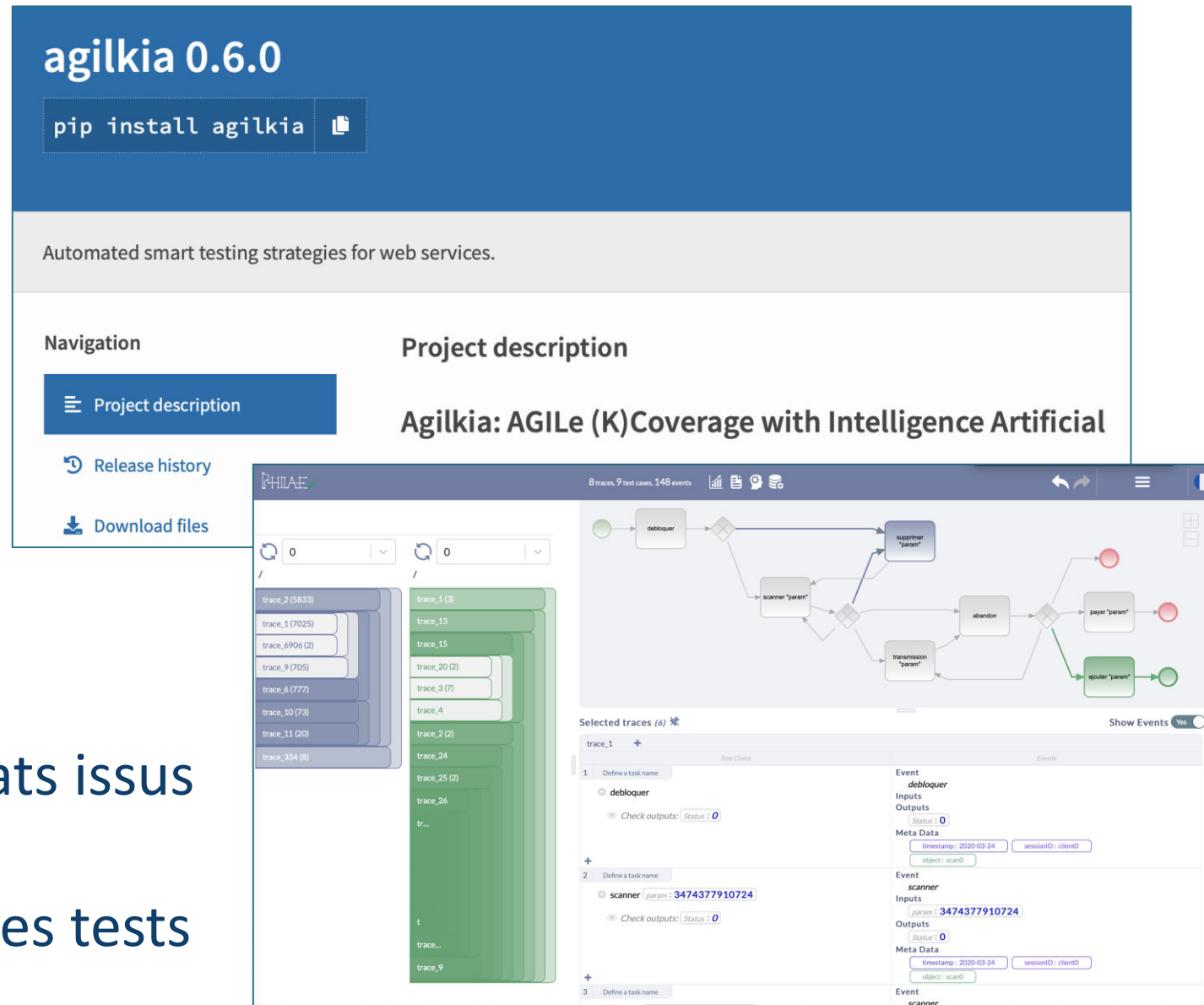


Agilkia

Boîte à outils (python) pour la génération de tests à partir de l'apprentissage automatique sur les logs

Viewer Philae

- application web
- affichage des résultats issus d'Agilkia
- Aide à la sélection des tests



The screenshot displays the Agilkia 0.6.0 web interface and the Philae viewer. The top section of the Agilkia interface shows the version and a command to install it: `pip install agilkia`. Below this, a navigation menu includes 'Project description', 'Release history', and 'Download files'. The main content area is titled 'Project description' and 'Agilkia: AGILe (K)Coverage with Intelligence Artificial'. The Philae viewer, overlaid on the bottom right, shows a flowchart of test cases and a table of selected traces. The table has columns for 'Test Cases' and 'Events'. The 'Test Cases' column lists tasks like 'debloquer', 'scanner', and 'supprimer'. The 'Events' column shows the corresponding events and their outputs. The table also includes a 'Show Events' toggle.

agilkia 0.6.0

`pip install agilkia`

Automated smart testing strategies for web services.

Navigation

- Project description
- Release history
- Download files

Project description

Agilkia: AGILe (K)Coverage with Intelligence Artificial

PhilAE

8 traces, 9 test cases, 148 events

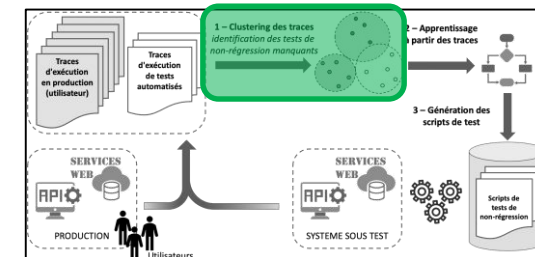
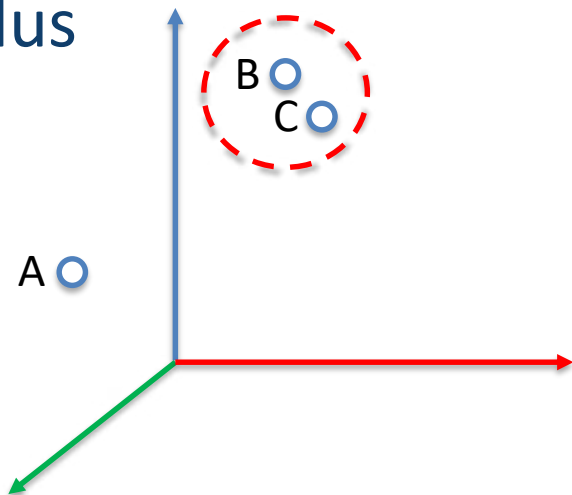
Selected traces (4)

Test Cases	Events
1 Define a task name	Event debloquer
2 Define a task name	Event scanner
3 Define a task name	Event scanner

Clustering : « méthode d'apprentissage permettant de grouper des individus d'un ensemble, basé sur des caractéristiques »

Pourrions nous utiliser le clustering pour grouper les traces par similarité ?

Le clustering nécessite une représentation de chaque individu par un vecteur, et se base sur ces vecteurs pour calculer la distance entre les individus

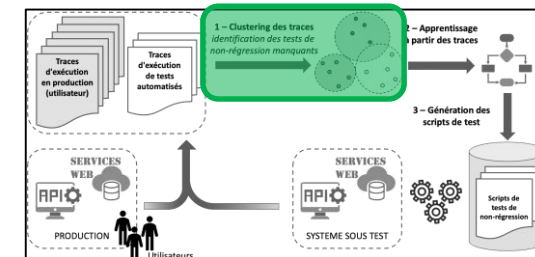


Préparation des données

1. Parsing les logs
2. Identification des sessions utilisateurs et sessions de test (individus)
 - Besoin de connaissance métier (*numéro de session, début/fin, ...*)
3. Représentation des sessions sous forme de vecteurs pour les algorithmes de machine learning

Comment créer des composantes de vecteurs pour représenter correctement un session ?

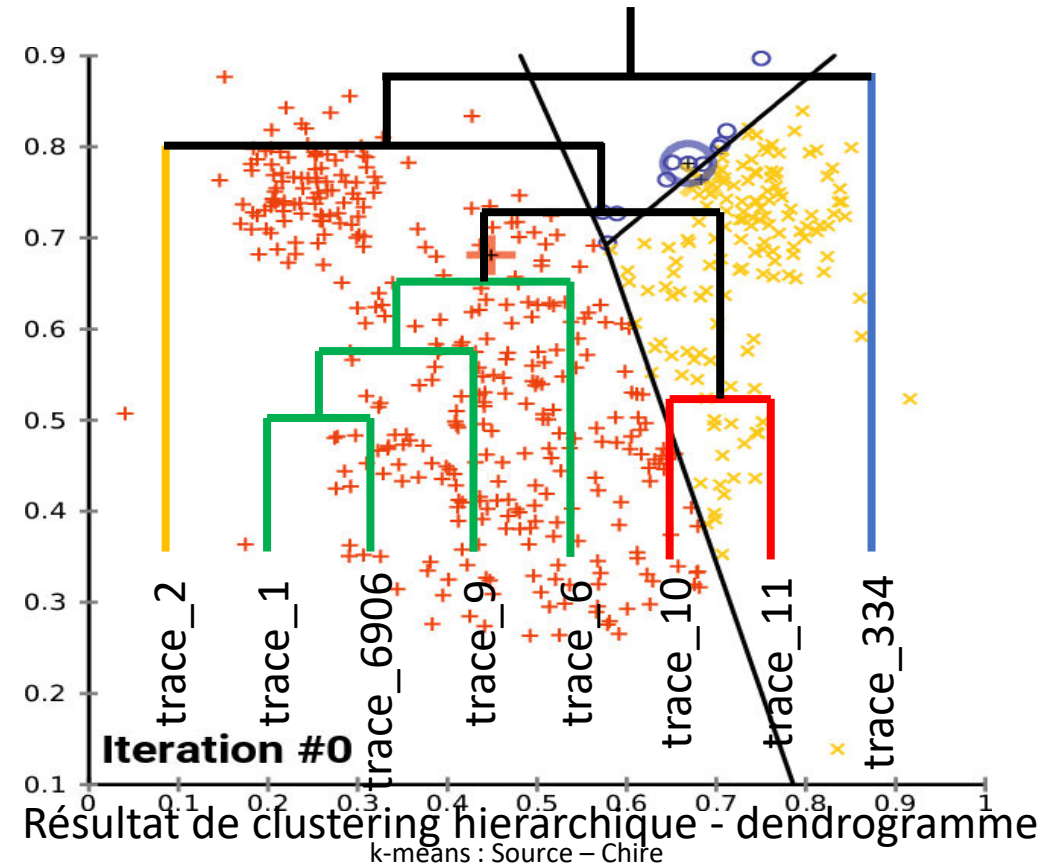
- Nom des actions utilisateurs
- Valeurs de paramètres
- Valeurs de retour
- Meta-données
- Ordre des appels (*seq2vec, ...*)



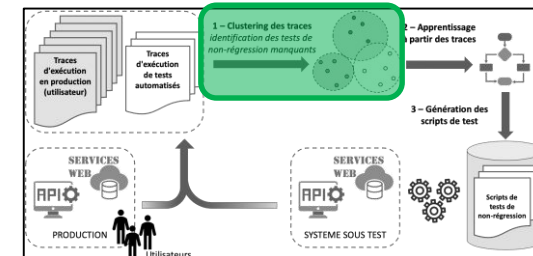
Clustering

Algorithmes

1. k-means
produit k groupes
→ *nécessite de donner k à priori*
2. mean-shift
produit un nombre de groupes optimal
→ *les utilisateurs veulent parfois assembler, ou diviser un groupe*
3. Hierarchical Clustering
produit des groupes hiérarchisés



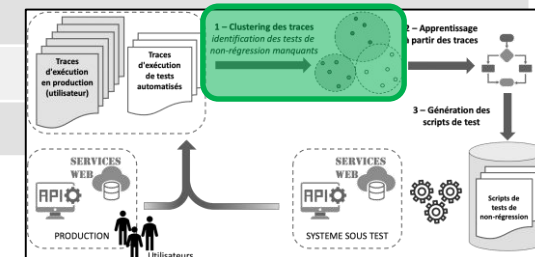
https://commons.wikimedia.org/wiki/File:K-means_convergence.gif



Pertinence du clustering : exemple avec **MeanShift** sur **Scanette**

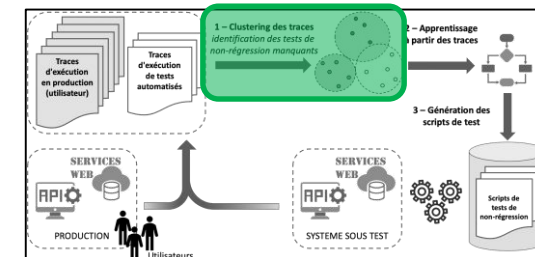
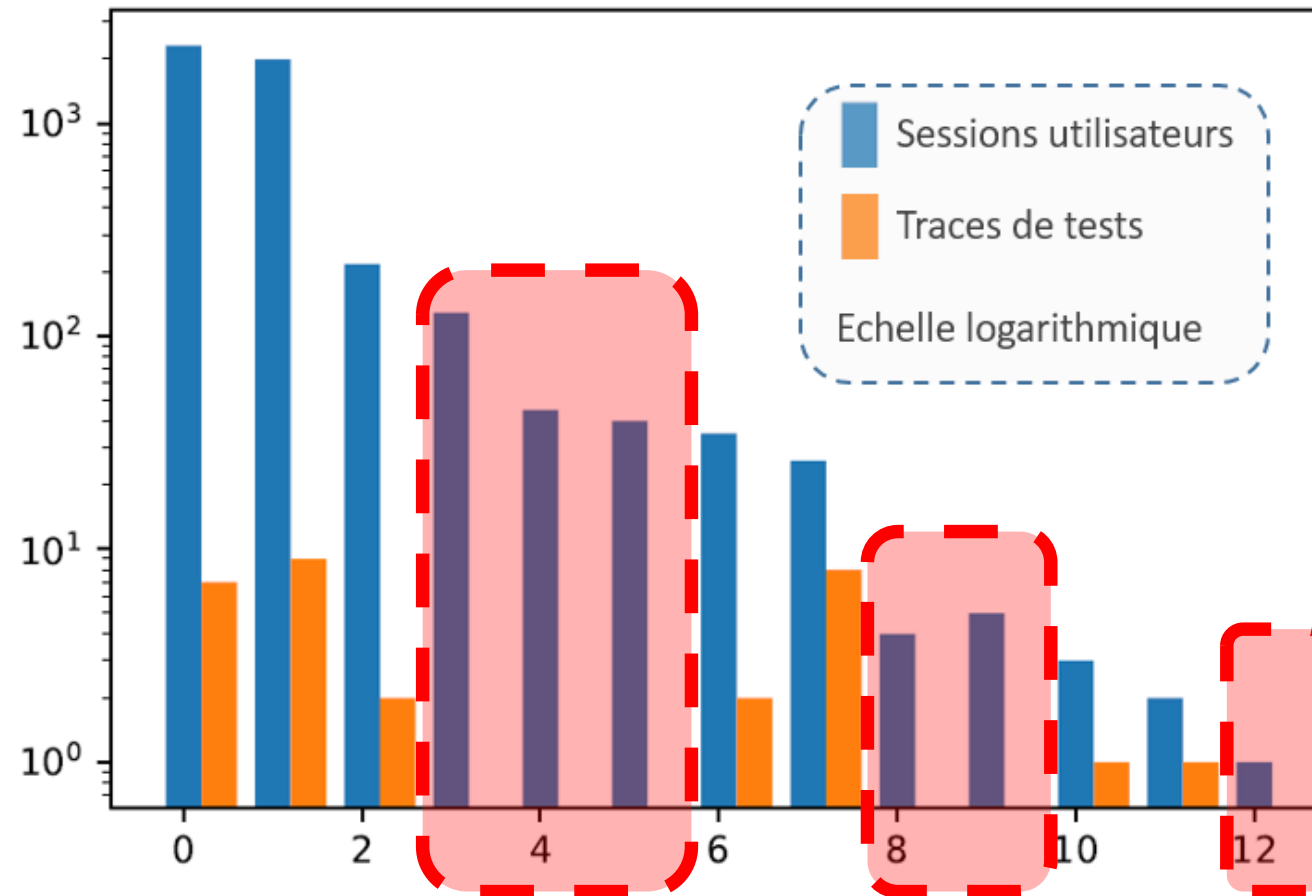
Sur les traces d'usage, **13 clusters** (regroupement de traces) sont calculés

N° cluster	Nombre traces	Description
0	2314	Utilisation classique de la scanette (pas de contrôle, paiement direct)
1	1996	Des achats avec des références inconnues qui sont ajoutées ensuite par la caissière avant le paiement
2	218	Les achats sont suivis d'un contrôle du caddie par le caissier puis paiement
3	129	Similaire à 2 avec l'ajout de produits non référencés
4	45	Similaire à 3 mais avec des séquences plus longues (plus de produits à scanner et à contrôler)
5	40	Similaire à 2 mais avec des séquences plus longues (plus de produits à scanner et à contrôler)
6	35	Identique à 1 mais avec des produits enlevés lors des achats
7	26	Les achats sont suivis d'un contrôle qui échoue (en détectant un produit pas été scanné auparavant)
8	5	Les achats sont suivis d'un ajout manuel d'un produit par le caissier
9	4	Les sessions ne sont pas finalisées (interrompues dans le fichier log)
10	3	Des achats avec retrait et paiement direct sans contrôle
11	2	Des achats avec retrait et des ajouts manuels de produits
12	1	Séquence complète (achats, contrôle, ajout de produits)

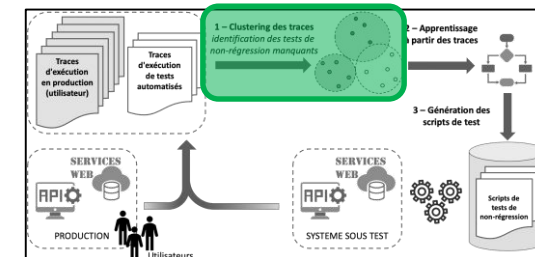
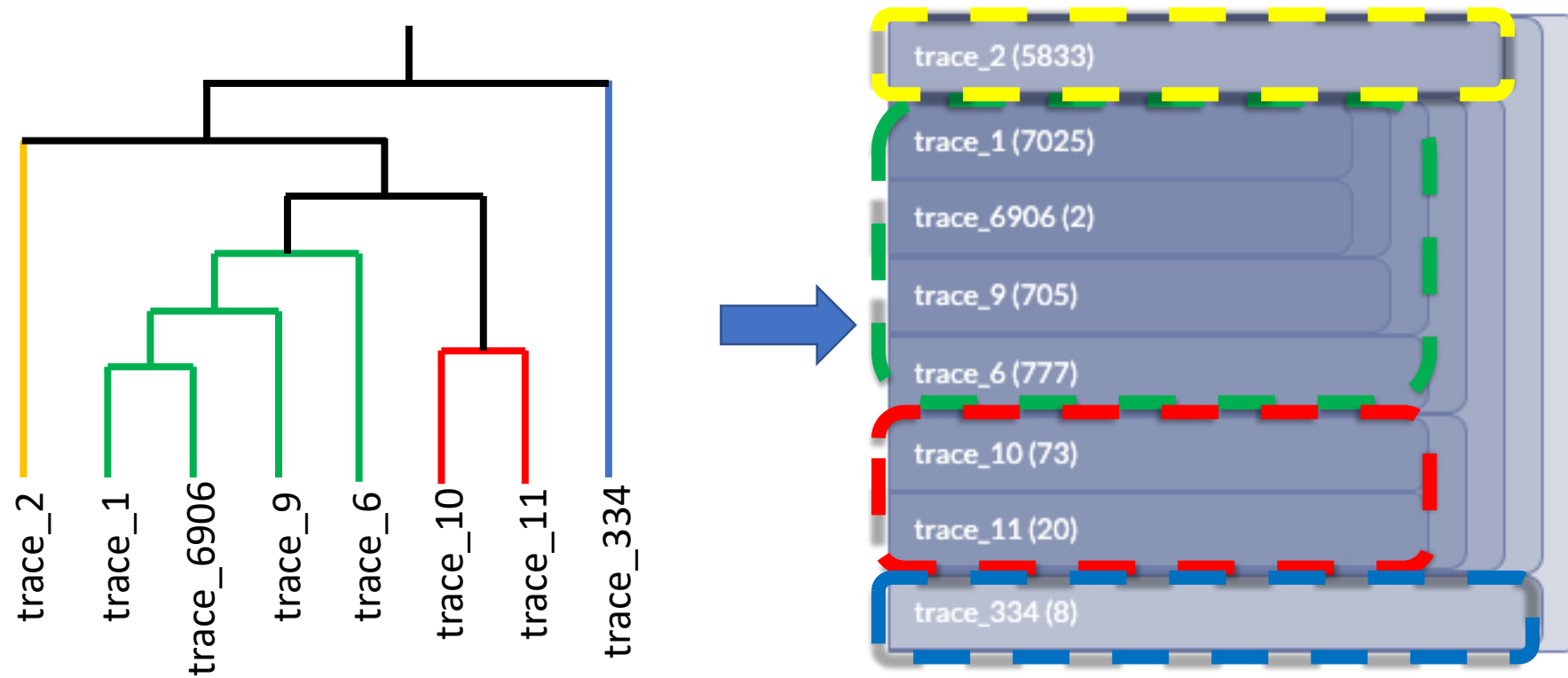


Comparaison entre traces issues de l'usage, et des tests :

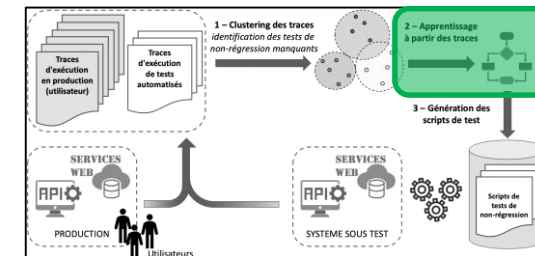
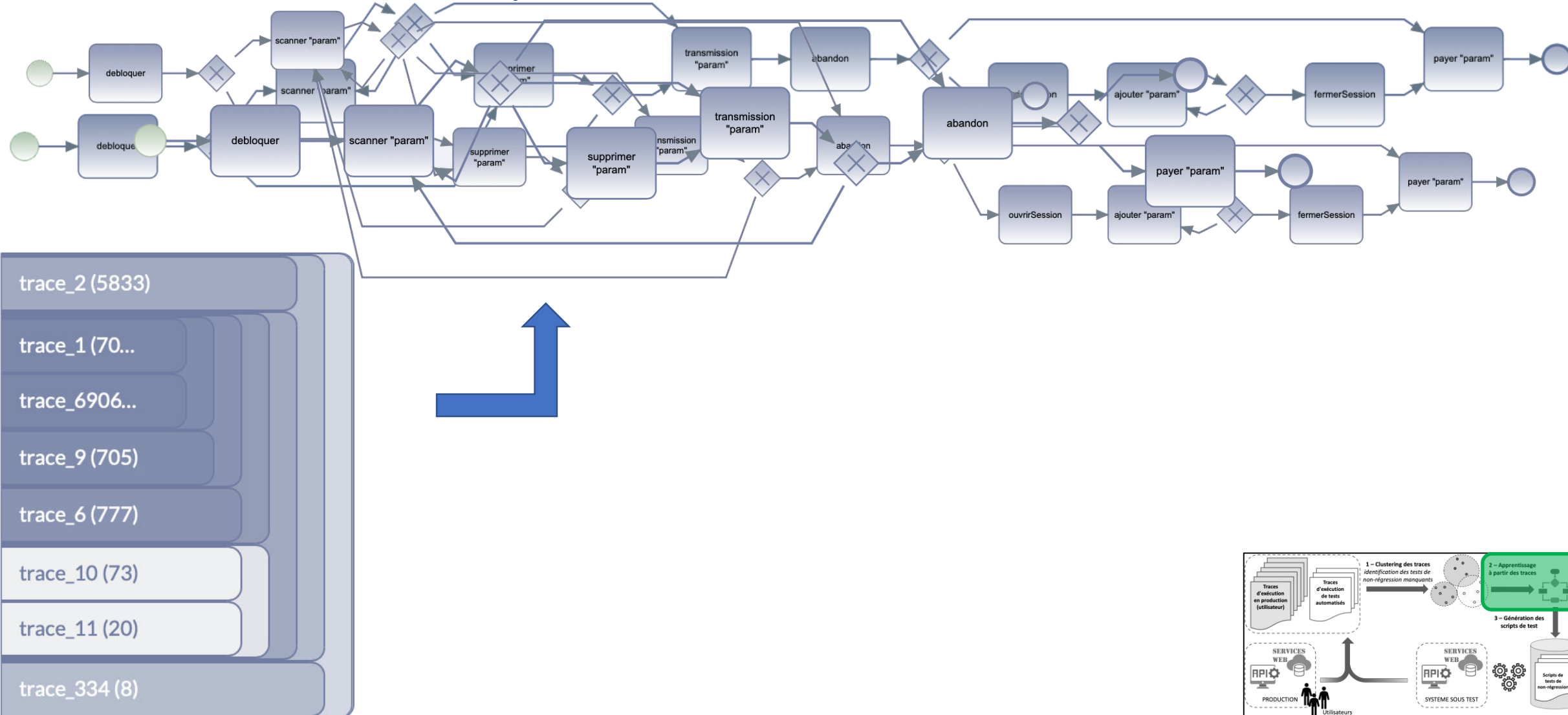
→ patterns d'usage peu ou pas testés



Représentation manipulable du dendrogramme



Inférence d'un modèle pour la visualisation



Visualisation de la couverture des traces issues de l'usage par les tests

The screenshot displays the PHILAE interface for trace coverage visualization. The top bar indicates '8 traces, 9 test cases, 148 events'. The left sidebar shows a list of traces, including trace_2 (5833), trace_1 (7025), trace_6906 (2), trace_9 (705), trace_6 (777), trace_10 (73), trace_11 (20), trace_334 (8), trace_1 (3), trace_13, trace_15, trace_20 (2), trace_3 (7), trace_4, trace_2 (2), trace_24, trace_25 (2), trace_26, and trace_9. The central area shows a flowchart with nodes like 'debloquer', 'scanner "param"', 'supprimer "param"', 'abandon', 'payer "param"', 'ajouter "param"', and 'transmission "param"'. The bottom panel shows details for selected traces, including 'trace_1' and 'trace_2', with fields for 'Define a task name', 'Check outputs: Status: 0', and 'param: 3474377910724'. The right side of the bottom panel shows 'Event' details for 'debloquer' and 'scanner', including inputs, outputs, and meta data.

PHILAE✓ 8 traces, 9 test cases, 148 events

Selected traces (6)

trace_1 +

Test Cases

1 Define a task name

debloquer

Check outputs: Status: 0

2 Define a task name

scanner param: 3474377910724

Check outputs: Status: 0

3 Define a task name

Events

Event debloquer

Inputs

Outputs Status: 0

Meta Data

timestamp: 2020-03-24 sessionID: client0

object: scan0

Event scanner

Inputs param: 3474377910724

Outputs Status: 0

Meta Data

timestamp: 2020-03-24 sessionID: client0

object: scan0

Event scanner

1 - Clustering des traces

2 - Apprentissage à partir des traces

3 - Génération des scripts de test

Traces d'exécution en production (utilisateur)

Traces d'exécution de tests automatisés

Services WEB PRODUCTION

Services WEB SYSTEME SOUS TEST

Scripts de tests de non-régression

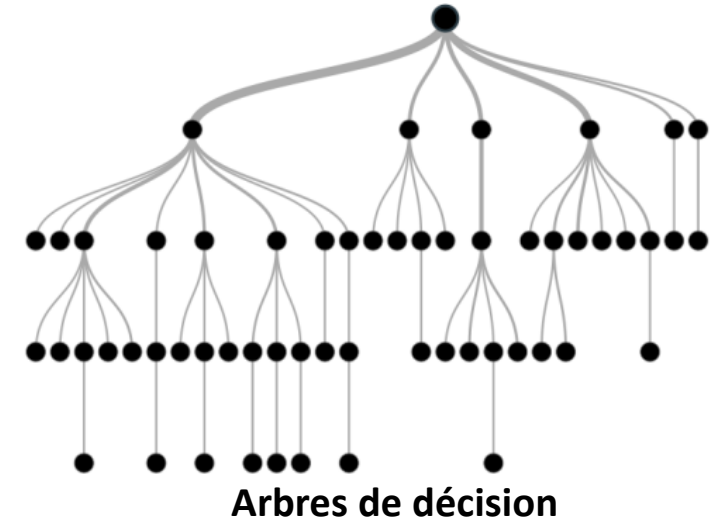
Utilisateurs

Apprentissage de modèles pour la génération

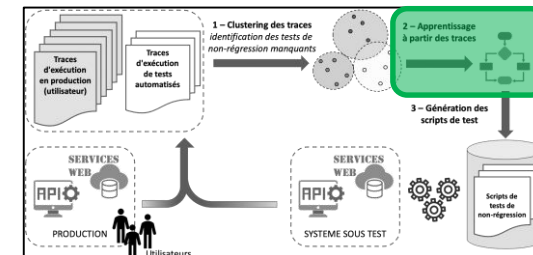
Algorithmes d'apprentissage automatique « prédictif » : utilisation de la probabilité de la prochaine action utilisateur

Différents types d'algorithme IA peuvent être utilisés :

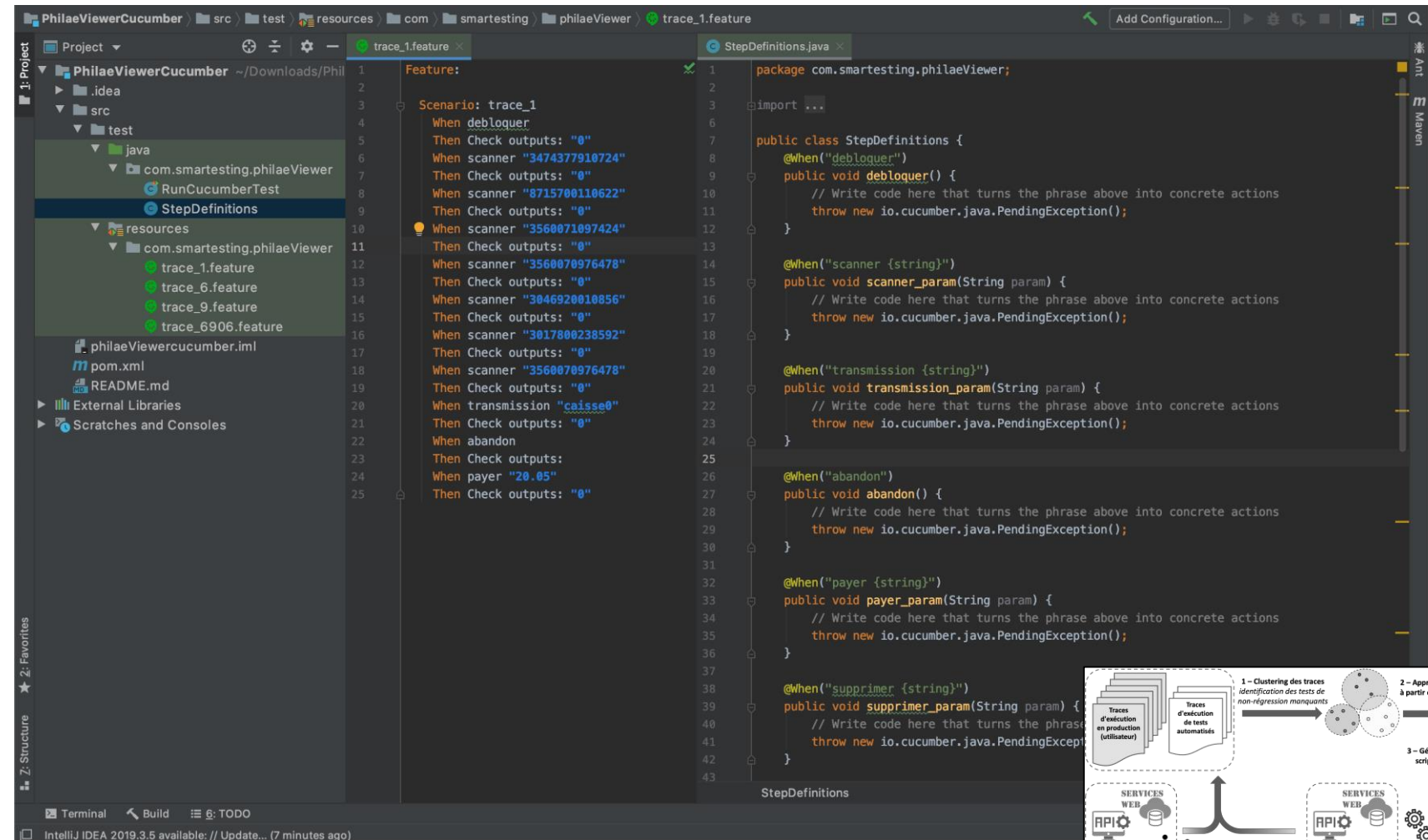
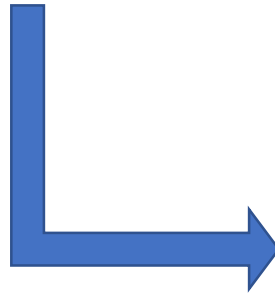
- arbres de décisions
- réseaux de neurones
- forêts aléatoires
- ...



Source –
<https://blog.bigml.com/2012/01/23/beautiful-decisions-inside-bigmls-decision-trees/>



Génération des scripts de tests



```
package com.smartesting.philaeViewer;

import ...

public class StepDefinitions {

    @When("debloquer")
    public void debloquer() {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }

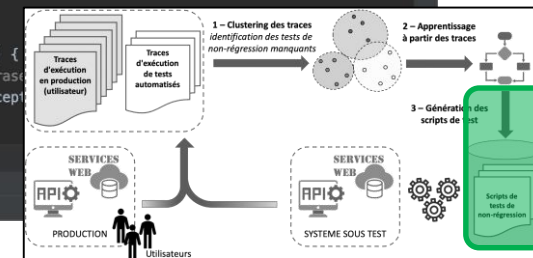
    @When("scanner {string}")
    public void scanner_param(String param) {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }

    @When("transmission {string}")
    public void transmission_param(String param) {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }

    @When("abandon")
    public void abandon() {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }

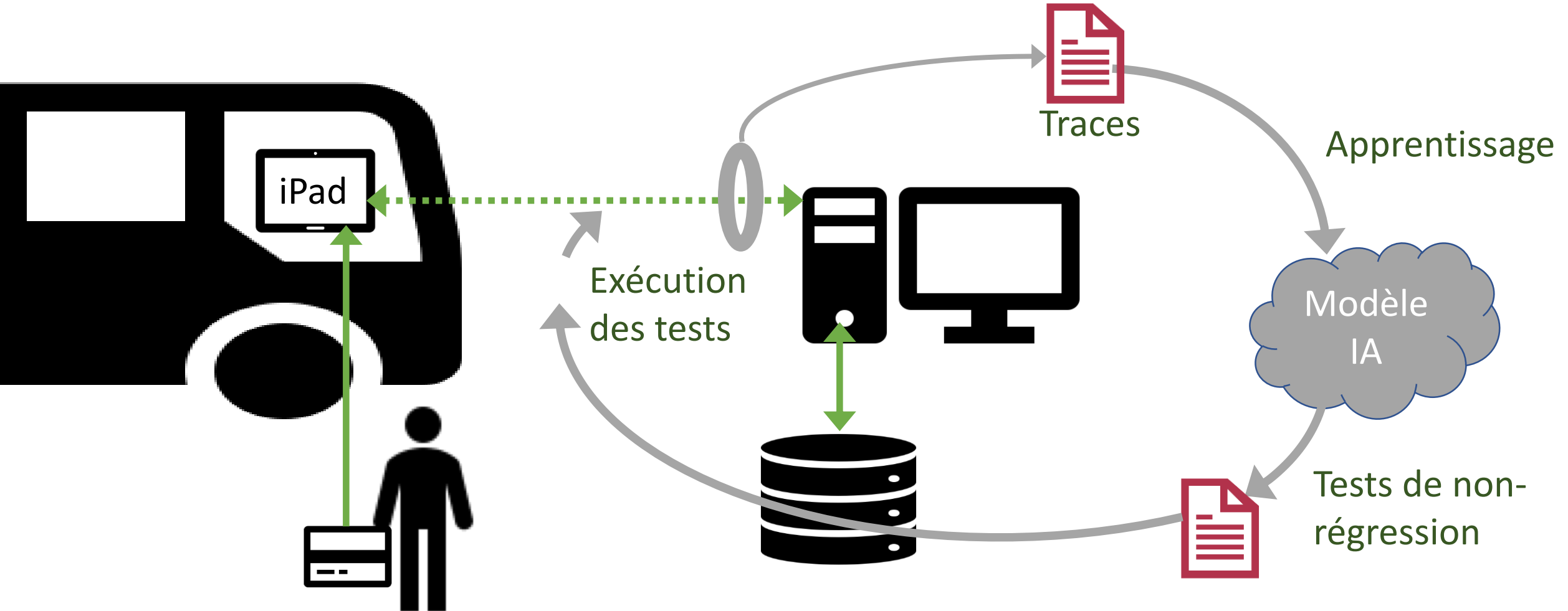
    @When("payer {string}")
    public void payer_param(String param) {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }

    @When("supprimer {string}")
    public void supprimer_param(String param) {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }
}
```



- Introduction
 - Motivations
 - Jalons (fait, future)
 - Exemple file rouge
- PHILAE
 - L'IA dans PHILAE
 - Process global
 - Outillage PHILAE
 - Clustering
 - Visualisation et sélection
 - Apprentissage
 - Génération des scénarios de tests
- Etudes de cas, résultats et perspectives

- Application de **gestion des transports scolaires** en Australie (Brisbane)
- Architecture **Cloud / Web services**



- Les regroupements de traces d'usage (5 clusters) correspondent à des patterns d'usage différents, et utiles pour les tests de non-régression
- Les tests générés automatiquement sont pertinents et assurent une bonne couverture des usages de la plateforme

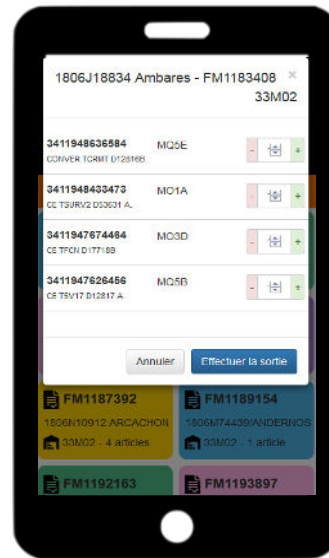
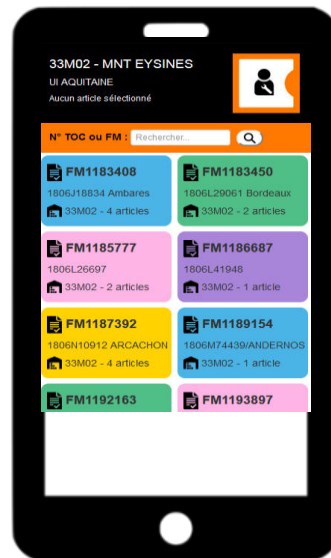
```
5 clusters found
Cluster 0:
.LPMA.A.i.i.ii.i.i.i.i.O.LPMA.I.o.o.o.o.o.o.o.o.
LPMAAA.A.i.i.i.i.i.A.i.i.i.iiO.LPMA.AA.I.oo.oo.o.o.o.o.o.
LPMAA.A.i.i.i.i.i.i.i.i.O.LPMAAA.A.A.I.o.o.o.o.o.
LPMAAA.A.i.i.i.i.i.i.O.LPM.AAAA.I.o.o.o.o.o.
LPM.A.i.i.i.i.i.Ai.i.i.i.O.LPMAA.AI.oo.o.o.o.o.o.o.
LPMAA.i.i.i.i.i.i.i.O.LPMAAA.I.o.o.o.o.o.o.
LPMA.A.i.AAA.ii.ii.i.i.i.i.i.O.LPM.A.AI.oo.o.o.o.o.o.o.o.o
LPM.i.i.i.i.i.i.O.LPM.I.o.o.o.o.o.o.o.
LPMA.i.iii.i.i.i.i.i.i.i.i.O.LPM.I.o.o.o.o.o.o.o.o.ooo.o.
LPM.A.i.i.i.i.i.i.i.i.O.LPM.I.o.o.o.o.o.o.
LPM.iiiAAi.AAA.O.LPMAA.I.oo.ooooo.
Cluster 1:
LPMA.C.i.i.i.i.i.i.i.i.i.O.LPMA.I.o.o.o.o.o.o.o.o.o.C.o.
```

Extrait des clusters calculés avec PHILAE

- Pas de tests automatisés initiaux (les tests de régression réalisés en mode exploratoire avant chaque release).
 - PHILAE a permis :
 1. D'identifier les patterns d'usage à couvrir par les tests de régression
 2. De générer automatiquement des cas de test représentatif pour chaque pattern d'usage
 3. De générer les scripts Java d'automatisation (avec implémentation manuelle des keywords)
- ➔ 27 tests automatisés de régression créés et implémentés avec le support PHILAE

Résultats obtenus

- Application web utilisée sur mobile par plusieurs dizaines de milliers de techniciens de maintenance
- But de l'application : avoir une trace des entrées et des sorties du matériel dans les magasins de maintenance et permettre un meilleur management des stocks.



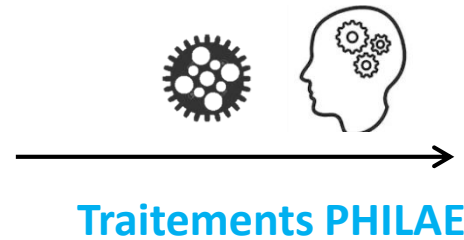
Interface de l'application

```
</RESULT>
</SOAP-ENV:Body></SOAP-ENV:Envelope></reponse><temps_reponse>
message_erreur</row>
<row><id>68181677</id><utilisateur_id>188909</utilisateur_id><code_partenaire>
code_partenaire<requete>'<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ifsworld-com:schemas=inbound_customer_order_get_list_customer_order">
<SOAP-ENV:Body>
<ns1:INPUT_VIEW>
<ns1:SITE>95M01</ns1:SITE>
<ns1:START_DATE>1970-01-01</ns1:START_DATE>
<ns1:END_DATE>2019-12-30</ns1:END_DATE>
<ns1:CUSTOMER_REFERENCE>FM1378483</ns1:CUSTOMER_REFERENCE>
<ns1:MODE>0</ns1:MODE>
</ns1:INPUT_VIEW>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
'</requete><reponse>'<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ifsworld-com:schemas=inbound_customer_order_get_list_customer_order">
<RESULT_CODE>000</RESULT_CODE>
<RESULT_DESCRIPTION>OK</RESULT_DESCRIPTION>
<CUSTOMER_ORDER_LISTS>
<CUSTOMER_ORDER_LIST>
<CUSTOMER_ORDER_NO>I6886153</CUSTOMER_ORDER_NO>
<BUSINESS_PROCESS xsi:nil="1"/>
<WORKING_ORDER_REFERENCE_NO>FM1378483</WORKING_ORDER_REFERENCE_NO>
<CUSTOMER_ORDER_STATUS>Reserved</CUSTOMER_ORDER_STATUS>
<EXTERNAL_ORDER_REFERENCE_NO>1378483/1</EXTERNAL_ORDER_REFERENCE_NO>
</CUSTOMER_ORDER_LIST>
</CUSTOMER_ORDER_LISTS>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
'</reponse></reponse>
```

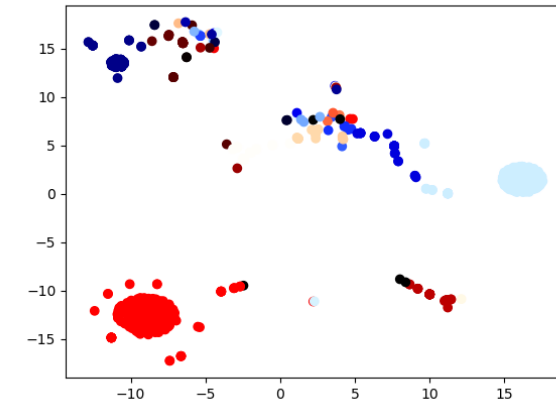
Exemple de logs

Actions / Séquences d'actions

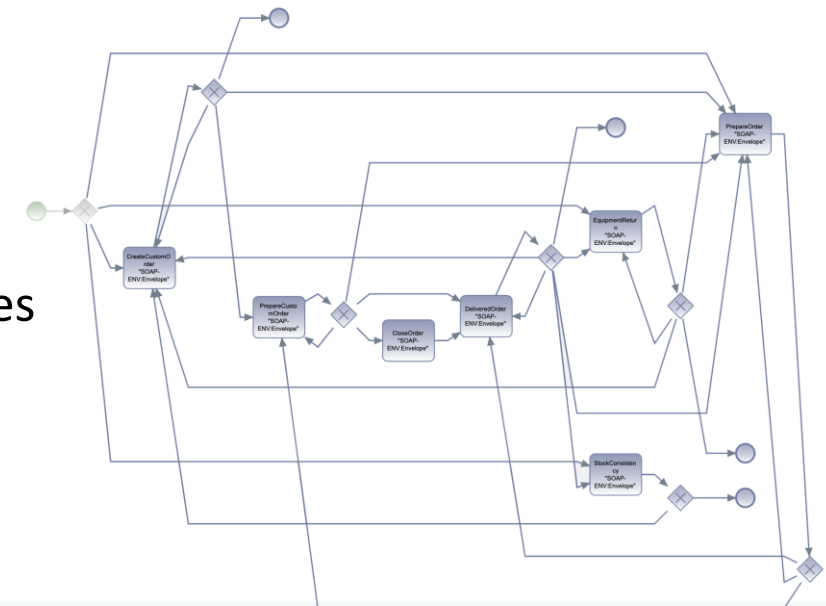
Actions	s	#	
PrepareOrder	P	374	</RESULT> <temps_reponse> <utilisateur_id><e ncoding="UTF-8"> as.xmlsoap.org/s _get_list_custom
CancelOrder	X	2	
CloseOrder	V	5	
Equipment Return	R	227	_REFERENCE> ng="UTF-8"> as.xmlsoap.org/s s="urn:ifsworld-
DeliveredOrder	D	720	
StockConsistency	-	681	R_NO> KING ORDER REFER ORDER_STATUS> TERNAL_ORDER_REF
CreateCustomOrder	N	421	
PrepareCustomOrder	p	468	



Visualisation des clusters



Visualisation des traces



Fait

- Analyse des logs et visualisation des sessions d'usage
- Interprétation des comportements utilisateurs à couvrir
- Analyse des champs de données et identification des données

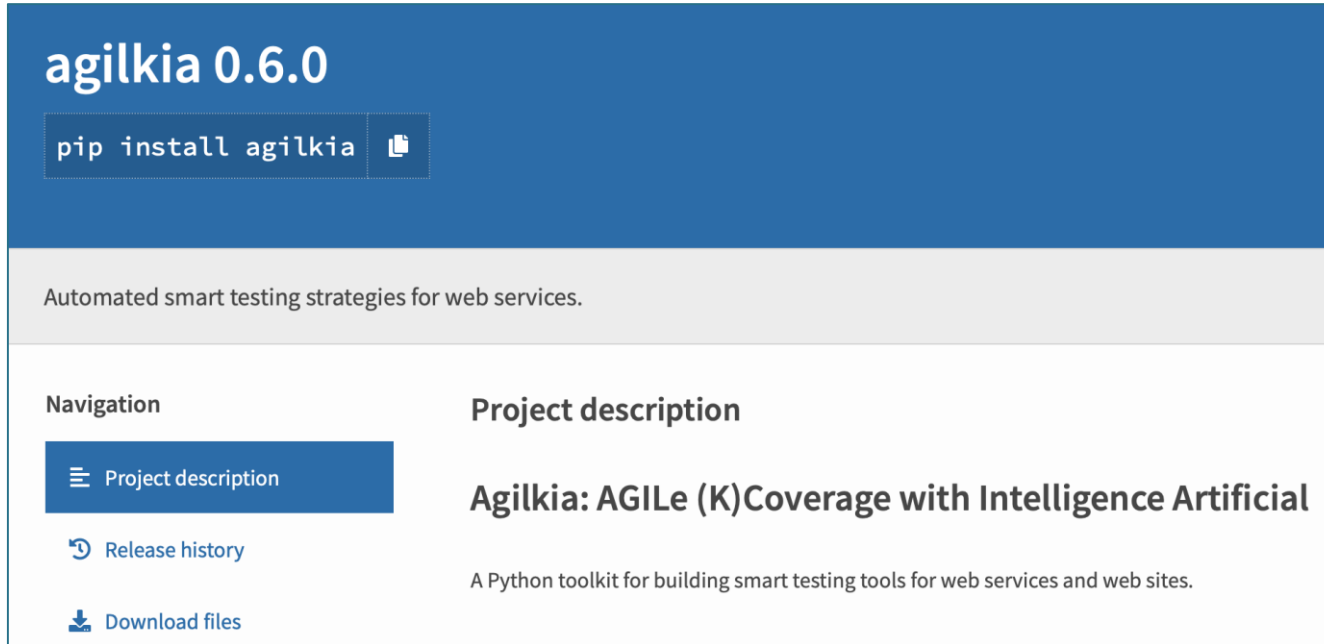
En cours

- Calcul des classes d'équivalence sur le données
- Génération des scripts SOAP-UI

Conclusion et perspectives

PHILAE – Une boîte à outils open-source pour la génération de tests à partir de l'apprentissage automatique sur les logs

Outillage PHILAE **Agilkia** librement disponible :



The screenshot shows the PyPI page for the 'agilkia' package. At the top, it displays 'agilkia 0.6.0' and a button to 'pip install agilkia'. Below this, a description reads 'Automated smart testing strategies for web services.' The page is divided into a 'Navigation' sidebar and a 'Project description' main area. The sidebar includes links for 'Project description' (selected), 'Release history', and 'Download files'. The main area contains the title 'Agilkia: AGILe (K)Coverage with Intelligence Artificial' and a subtitle 'A Python toolkit for building smart testing tools for web services and web sites.'

<https://pypi.org/project/agilkia/>

Viewer PHILAE librement disponible :

<https://smarttestingphilaevviewer.herokuapp.com>

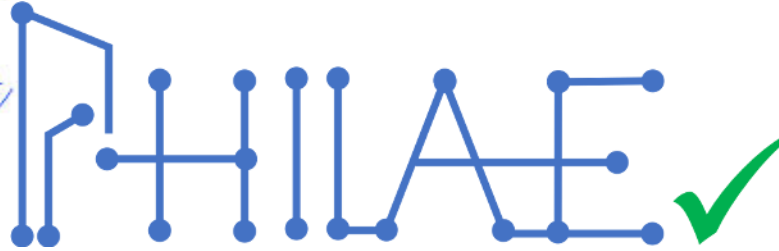
- **Analyse des logs d'exécution en production et en test**
 - Extraction des sessions utilisateurs et formatage des données
 - Anonymisation des données clients
- **Identification des patterns d'usage du logiciel**
 - Calcul et visualisation des clusters
 - Visualisation des graphes d'appel / sessions d'usage
- **Sélection des traces pour assurer la couverture de tests à partir des patterns d'usage**
 - Visualisation comparée
 - Différents algorithmes de génération de tests à partir du modèle d'apprentissage
- **Génération des scripts automatisés sous forme d'appel APIs / web services**
 - Le squelette des scripts à ce stade du projet

- **Les algorithmes IA utilisés dans PHILAE continuent à être améliorés**, en particulier au niveau du clustering et de la génération des tests, et l'apprentissage sur les données de test
 - **La génération des scripts de test** sera disponible pour différents **Framework d'automatisation cible**
 - **Le calcul des classes d'équivalence sur les données**, permettra la génération des données de tests en combinant apprentissage sur les traces d'exécution de production et de test
- ➔ **Expérimentation et validation de la technologie PHILAE** dans différents contextes.

Appel à manifestation d'intérêt

- L'approche vous semble pertinente dans votre contexte (application web, tests de régression sur API à renforcer / compléter / créer)
- Vous disposez de logs à l'exécution en opération et/ou en test
- Vous souhaitez expérimenter l'apport de l'outillage PHILAE et vous avez un peu de temps pour cela

Contactez nous : exp-philae@univ-fcomte.fr



Générer les tests de régression par apprentissage sur les traces d'exécution

Fabrice Bouquet – Julien Botella – Frédéric Tamagnan

Contact : exp-philae@univ-fcomte.fr

