Conseil en technologies / Service numérique / Ingénierie informatique

CNES - SII

KOSMOS Day - Présentation génération COM-TU

24/09/2026

© Groupe SII | Usage interne SII

Document propriété exclusive de SII, communication, reproduction, utilisation interdite sauf autorisation écrite préalable de SII



1

Cadre de l'étude

© Groupe SII | Usage interne SII

1 Cadre de l'étude

2 problématiques principales :

- Comment générer des commentaires et des tests unitaires sur du code ?

L'étude se base sur les deux partitions IOS et MMDL de KOSMOS et fait l'hypothèse que le LLM utilisé n'impacte pas les résultats.

On a fait le choix d'utiliser Qwen3-4B pour toute l'étude :

- Petit modèle
- Local
- Sécurisé

Qwen



© Groupe SII | Usage interne SII



2

Base commune

© Groupe SII | Usage interne SII

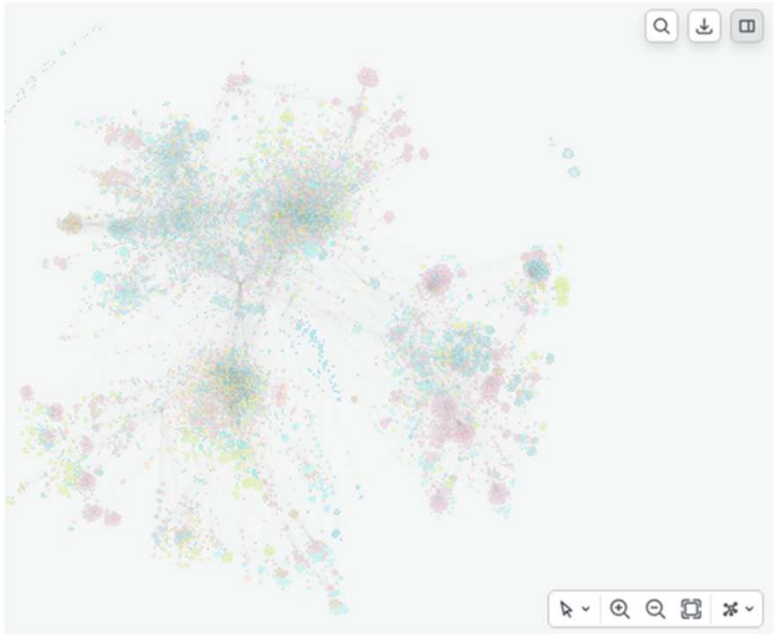
2 Base commune – Graph

- Utilisation d'un graph comme base de données :

neo4j



- Le parsage du code est réutilisé pour les 2 composants principaux



© Groupe SII | Usage interne SII

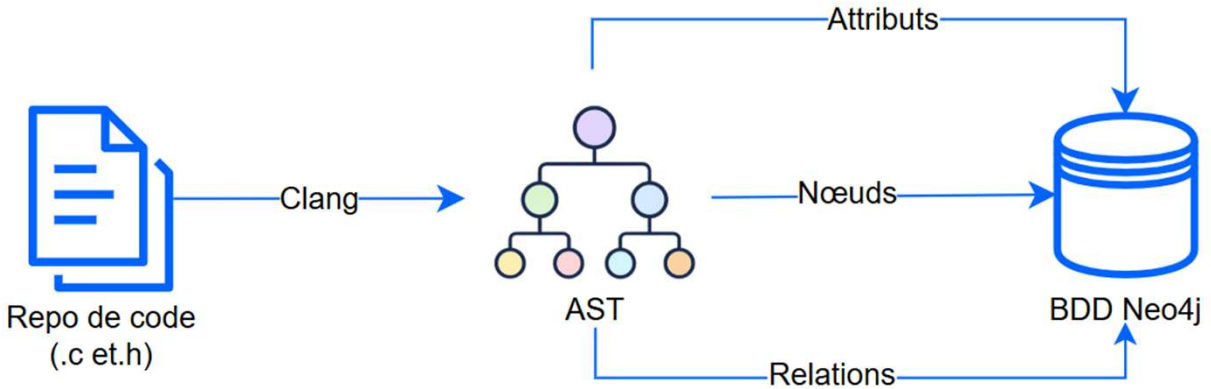
2 Base commune – Parsage

- Construction du graphe

```c
int add(int a, int b)
{
  int result = a + b;
  printf("%d\n", result);
  return result;
}
```

Attributs :
- Name : add
- Type : Function
- Body : …

Relation :
- CALLS : printf
- INPUT : int a, int b
- OUTPUT : int



© Groupe SII | Usage interne SII

2 Base commune – Nœuds

Define
file
name
body
values

Enum
file
name
body
values

Function
file
name
body
signature

GlobalVariable
file
name
body
type
value

Include
file
name
body
values

Struct
file
name
body
fields

Union
file
name
body
fields

Typedef
file
name
body



© Groupe SII | Usage interne SII

2 Base commune – Nœuds

StructField
name
struct

Type
name
type

UnionField
name
union

EnumArgument
name
enum

File
name
path

Folder
name
path



© Groupe SII | Usage interne SII

2 Base commune – Relations

HAS_FIELDS
Struct, Union
StructField, UnionField

INCLUDES
File
Include

CALLS
Function
Function, Define

CONTAINS
Folder, File
Define, Enum, File, Folder, Function, GlobalVariable, Struct, Typedef, Union

INPUT
Type
Function

OUTPUT
Function
Type

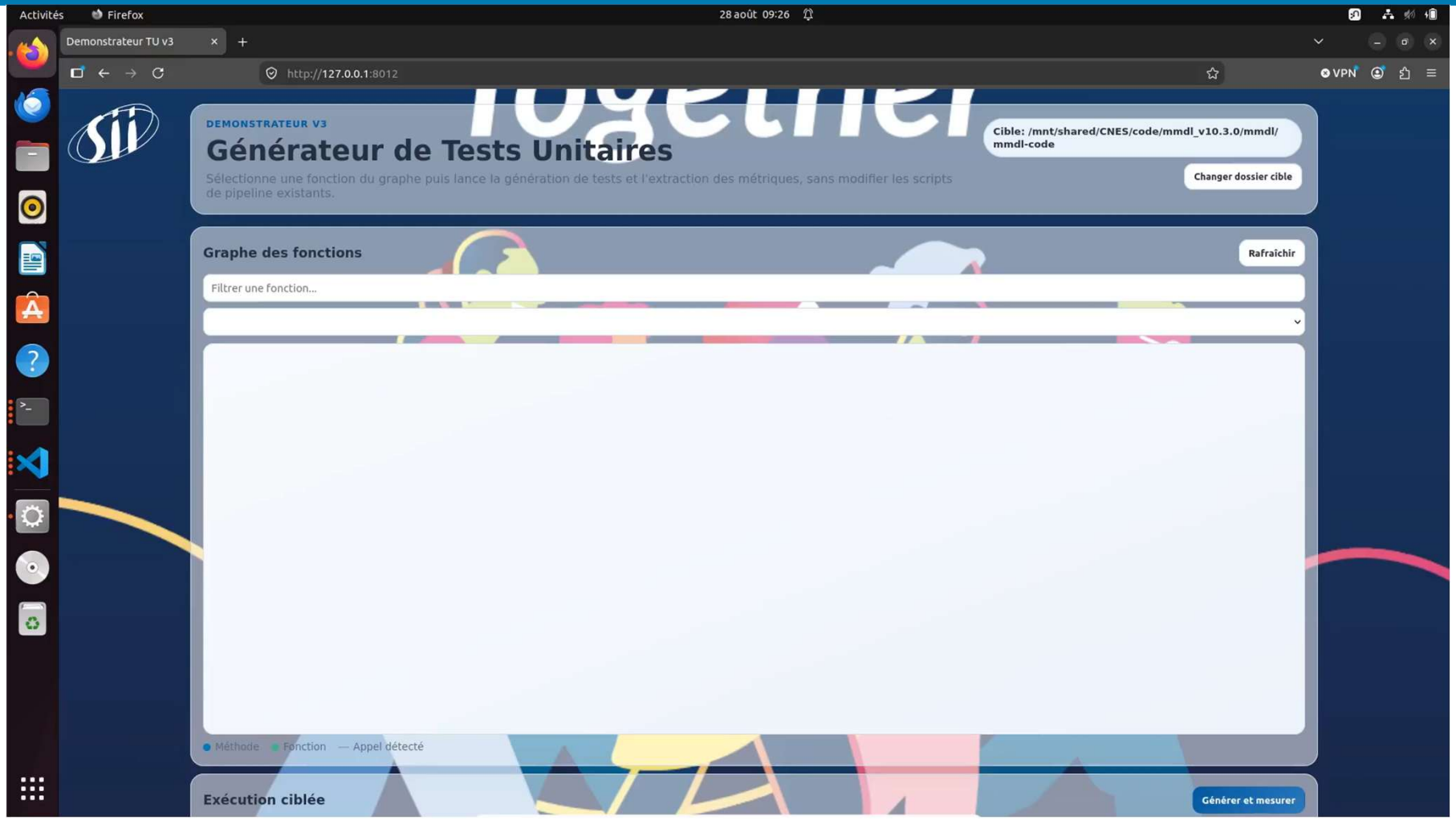
HAS_ARGUMENT
Enum
EnumArgument

HAS_DEPENDENCY
Function, Enum, Struct, Union, GlobalVariable, Define, Type, Typedef
Function, Enum, Struct, Union, GlobalVariable, Define, Type, Typedef, EnumArgument



© Groupe SII | Usage interne SII

2 Démo - Génération du graph



© Groupe SII | Usage interne SII



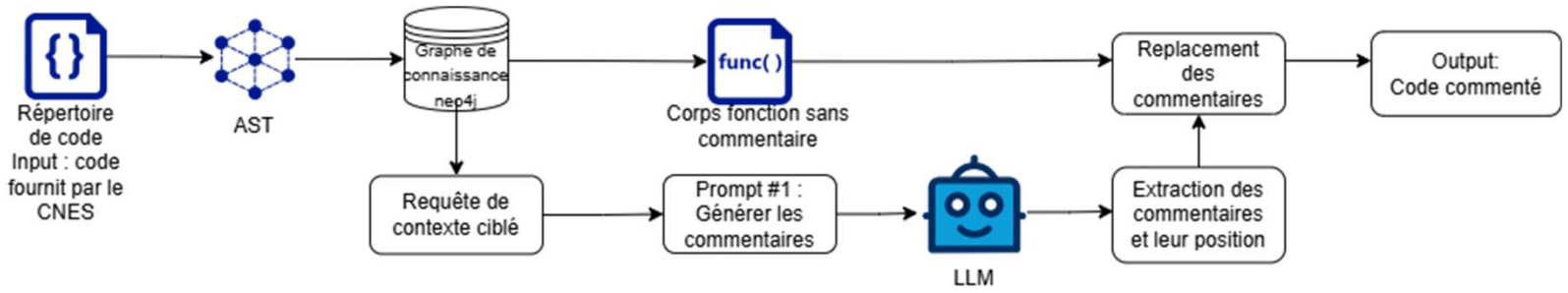
3

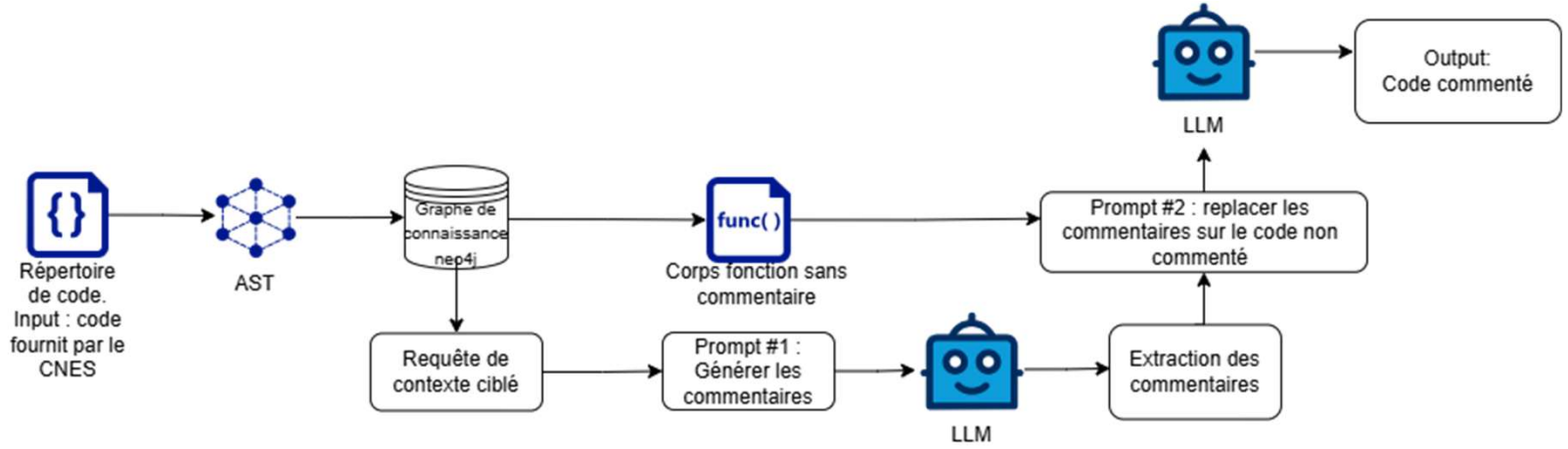
Génération des commentaires

© Groupe SII | Usage interne SII

3 Commentaires - Architecture initiale

- 2 architectures testées : script & LLM





© Groupe SII | Usage interne SII

3 Commentaires – Comparatif architectures

```
================================================================================
                    COMPARAISON DES DEUX METHODES
================================================================================
                          Code Integrity
--------------------------------------------------------------------------------
similarity_score_sequence_matcher | Script =   1.0000 | LLM =   0.9941
bert_scores                       | Script =   1.0000 | LLM =   0.9384
rouge_score-R1                    | Script =   1.0000 | LLM =   1.0000
rouge_score-R2                    | Script =   1.0000 | LLM =   1.0000
rouge_score-RL                    | Script =   1.0000 | LLM =   1.0000
bleu_score_value                  | Script =   1.0000 | LLM =   0.9418
--------------------------------------------------------------------------------
                        Comment Quality
--------------------------------------------------------------------------------
average_meteor_score              | Script =   0.1213 | LLM =   0.1348
bert_comment_scores               | Script =   0.0178 | LLM =   0.0178
rouge_comment_score-R1            | Script =   0.2722 | LLM =   0.2760
rouge_comment_score-R2            | Script =   0.0665 | LLM =   0.0683
rouge_comment_score-RL            | Script =   0.1344 | LLM =   0.1397
bleu_comment_score                | Script =   0.0138 | LLM =   0.0151
--------------------------------------------------------------------------------
                          Other Metrics
--------------------------------------------------------------------------------
score_comparison_llm              | Script =   9.6000 | LLM =   9.6000
score_alone_llm                   | Script =  10.0000 | LLM =  10.0000
comment_percentage                | Script =  67.6829 | LLM =  67.0807
time_taken                        | Script =  63.9972 | LLM = 126.6734
nb_code_lines                     | Script =  87.0000 | LLM =  87.0000
================================================================================
```

Script → Pas de modification du code

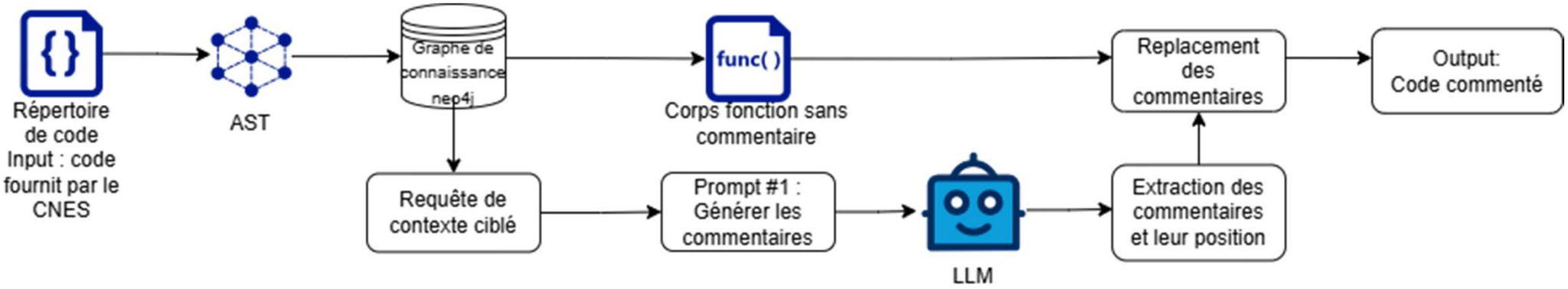
LLM → Pas d'élément non commenté

Solution retenue : Script



© Groupe SII | Usage interne SII

3 Commentaires - Architecture finale



- Identique à l'architecture testée initialement
- Placement par script en détectant la ligne de code suivante lors de la génération
  → Potentiellement des commentaires mis de côté



© Groupe SII | Usage interne SII

3 Commentaires – Résultats

Métrique	driver_dispatcher		init	
	Short	Long	Short	Long
Intégrité code	1	1	1	1
METEOR score	0.2941	0.2035	0.2544	0.2069
BERTScore	-0.1413	-0.1413	-0.1474	-0.1474
Rouge-1	0.5897	0.5211	0.4256	0.3335
Rouge-2	0.2395	0.1714	0.1360	0.1016
Rouge-L	0.2847	0.2022	0.1724	0.1322
Bleu	0.0595	0.0265	0.0448	0.0180
LLM – comparaison	8.1	8.1	7.3	7.3
LLM - seul	8.1	8.1	8.3	8.1
% of comments	48.6	59.25	37.21	58.29%

Les + :

- 30% de commentaires facilement atteints
- Possibilité de régler la taille des commentaires

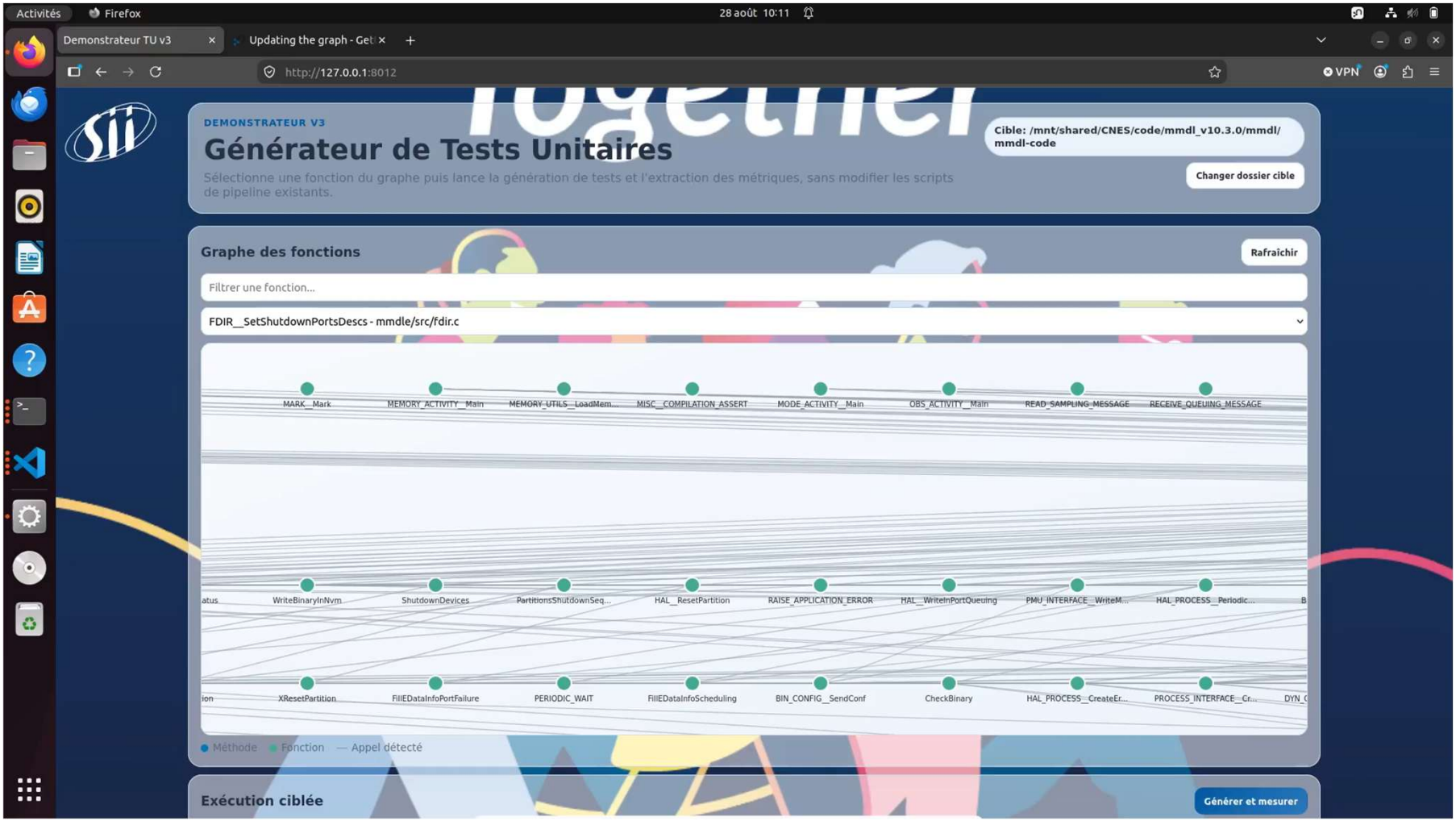
Les - :

- Assez verbeux
- Manque d'informations fonctionnelles --> Pas de spec en entrée



© Groupe SII | Usage interne SII

3 Démo - Génération des commentaires



© Groupe SII | Usage interne SII

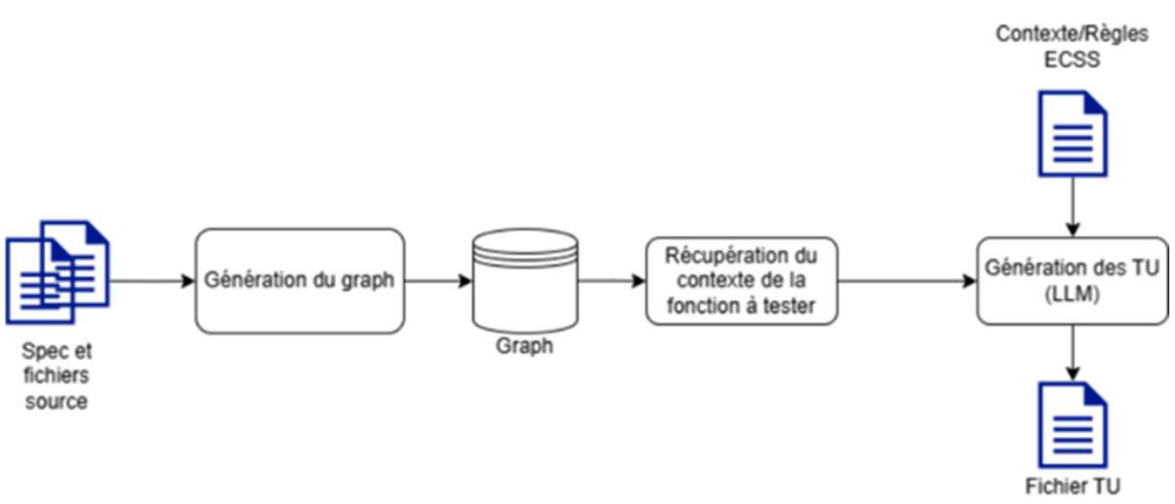


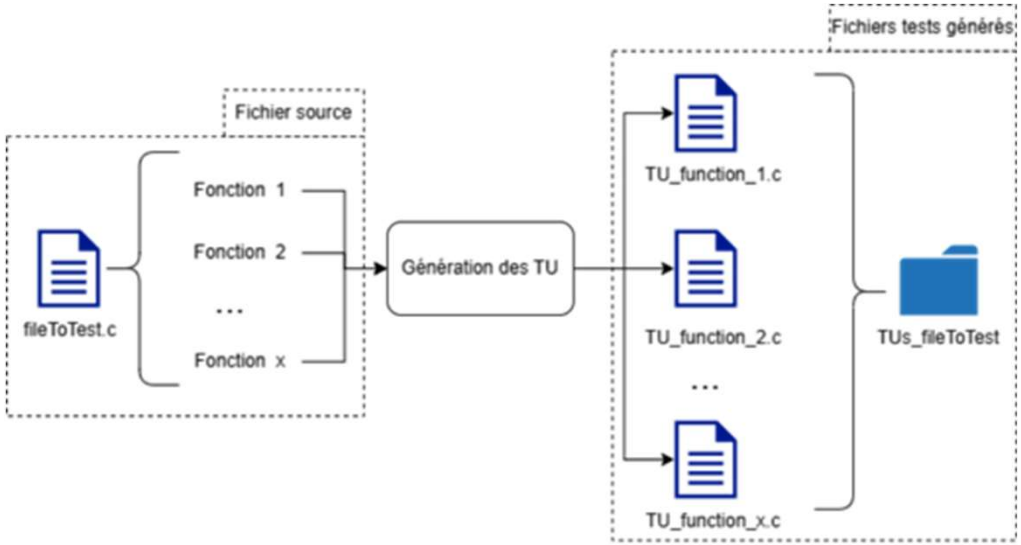
4

Génération des TU

© Groupe SII | Usage interne SII

4 TU – Architecture initiale

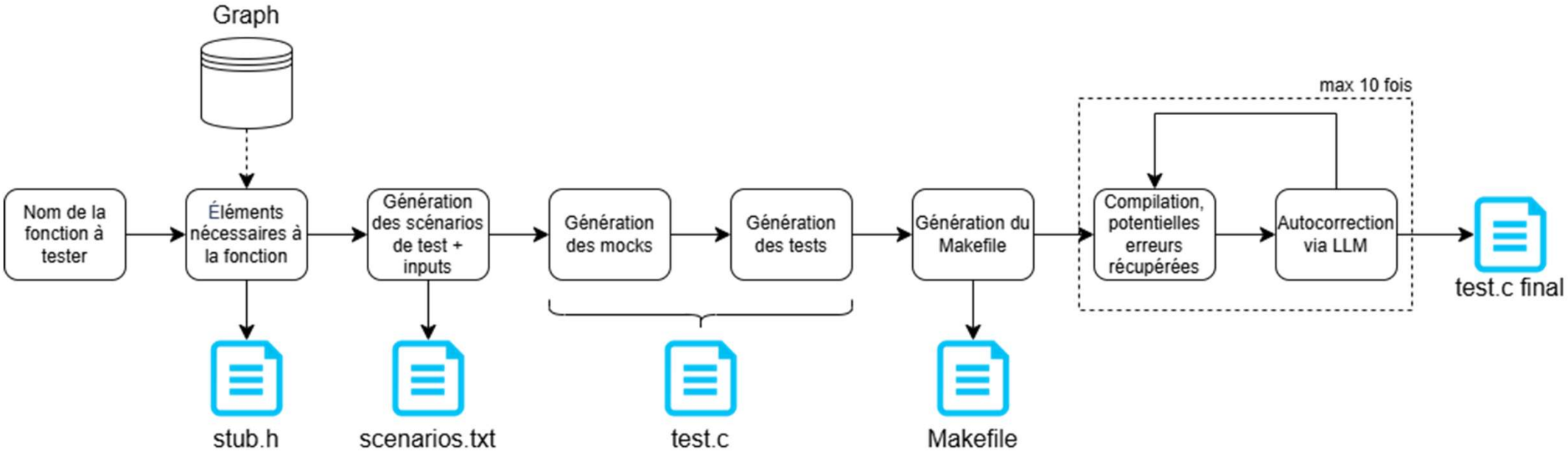




- Problème : générer tout le fichier de TU en une fois est trop ambitieux
- Solution : découper en sous-étapes

© Groupe SII | Usage interne SII

4 TU – Architecture finale



- Nouvelle architecture
  - Même logique
  - Plus d'étapes moins complexes
  - Création de fichiers intermédiaires

© Groupe SII | Usage interne SII

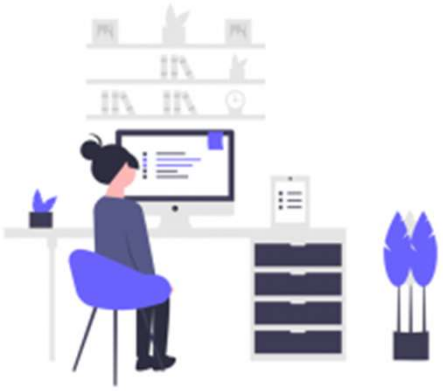
4 TU - Résultats

3 Critères pour 6 Fonctions :

- Prises aléatoirement
- Tailles variables
- Complexités variables

• Métriques:

- Mutation
- Coverage
- Taux de modification
- Nombre de tests



© Groupe SII | Usage interne SII

4 TU - Résultats

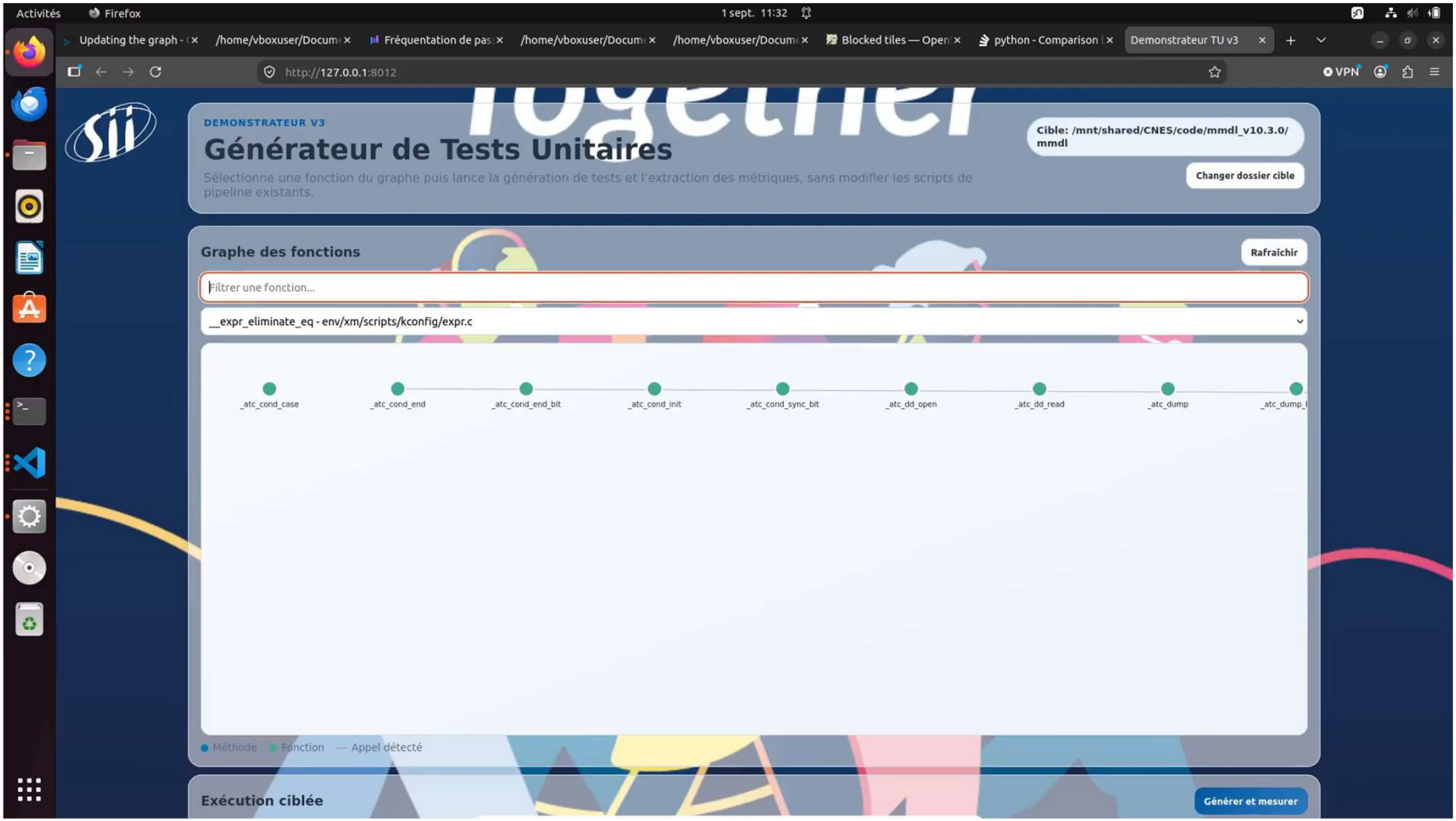
	ProbeMemAbortPortRequest	ComputeSanction	ProbeMemInPortRequest	ProbeShutdownRequest	PROCESS_INTERFACE__ Read
Moyenne Mutation	**57.8%**	**41%**	**58%**	**17.75%**	**34.6%**
Moyenne Coverage	**100%**	**82.74%**	**86.67%**	**76.66%**	**65.46%**
Moyenne Modification	**2.18% (2)**	**1.90% (5)**	**5.63% (3)**	**0**	**0**
Tests	**49/57**	**23/45**	**16/35**	**8/40**	**45/64**
Ecart Type Mutation	**16.33**	**20.1**	**6.93**	**26.99**	**12.54**
Ecart Type Coverage	**0**	**10.12**	**5.77**	**26.64**	**14.94**
Ecart Type Modification	**0.83**	**0.57**	**4.82**	**0**	**0**

- Résultats très positifs, très bonne base pour faire les tests --> scénarios
- Lorsque le test ne passe pas, quelques modifications à faire
- Résultats assez variables d'une fonction à l'autre



© Groupe SII | Usage interne SII

4 Démo - Génération TU



© Groupe SII | Usage interne SII



5 Conclusion

© Groupe SII | Usage interne SII

5 Conclusion - Résumé

Commentaires	Tests Unitaires
• Génération des commentaires sur un fichier entier • 30% de commentaires largement atteints • Possibilité d'adapter la taille des commentaires à partir du prompt • Manque d'informations fonctionnelles (normal vis-à-vis de l'architecture) • Redondance avec le code • Cohérence des commentaires au sein du même fichier à améliorer → Rajout d'une mémoire au LLM	• TU générés à partir du code • Quelques modifications nécessaires dans certains cas • Actuellement : donne une bonne base pour la création de TU, pas encore autonome • Performances obtenues sur un 'petit' modèle : Qwen3-4B • Manque aussi d'informations fonctionnelles, TU obtenus uniquement en se basant sur le code

- Ce prototype montre la faisabilité d'un assistant au développement sur les deux problématiques en question.
- Cependant cet outil n'est pas encore utilisable en temps que tel et nécessite plus de développement.



© Groupe SII | Usage interne SII

5 Conclusion

Pistes d'amélioration non explorées :

- Commentaires :
  - Rajout de spec fonctionnelle sur la génération de commentaires (RAG doc en parallèle du RAG code)
  - Rajout d'une mémoire au LLM lors de la génération → homogénéité des commentaires
- Tests unitaires :
  - Tests avec LLM different
  - TU spec → Besoin de specs bas niveau style LLR (Low Level Requirements)
  - Mieux sélectionner les informations données au LLM afin d'améliorer les résultats
  - Rajout d'une mémoire des corrections effectuées



© Groupe SII | Usage interne SII

Questions ?

© Groupe SII | Usage interne SII



Développer **ensemble** un monde **numérique** et **durable**

© Groupe SII | Usage interne SII

Document propriété exclusive de SII, communication, reproduction, utilisation interdite sauf autorisation écrite préalable de SII