

Trusted Application LLM × 静的テイント解析

日本大学 理工学部
吉永 達哉

自己紹介

- 名前：吉永達哉
- 所属：日本大学 理工学部 (B4)
- 興味：TEE, SBOM
- 趣味：CTF, 競プロ



きっかけ

- Trusted Execution Environment(TEE)に魅力
→コンピュータの「信頼性の基盤」で誤りが発生するのは致命的
- LLMがどこまで理解？
→LLMの文脈力とTEEへの知識で脆弱性を検知できるか

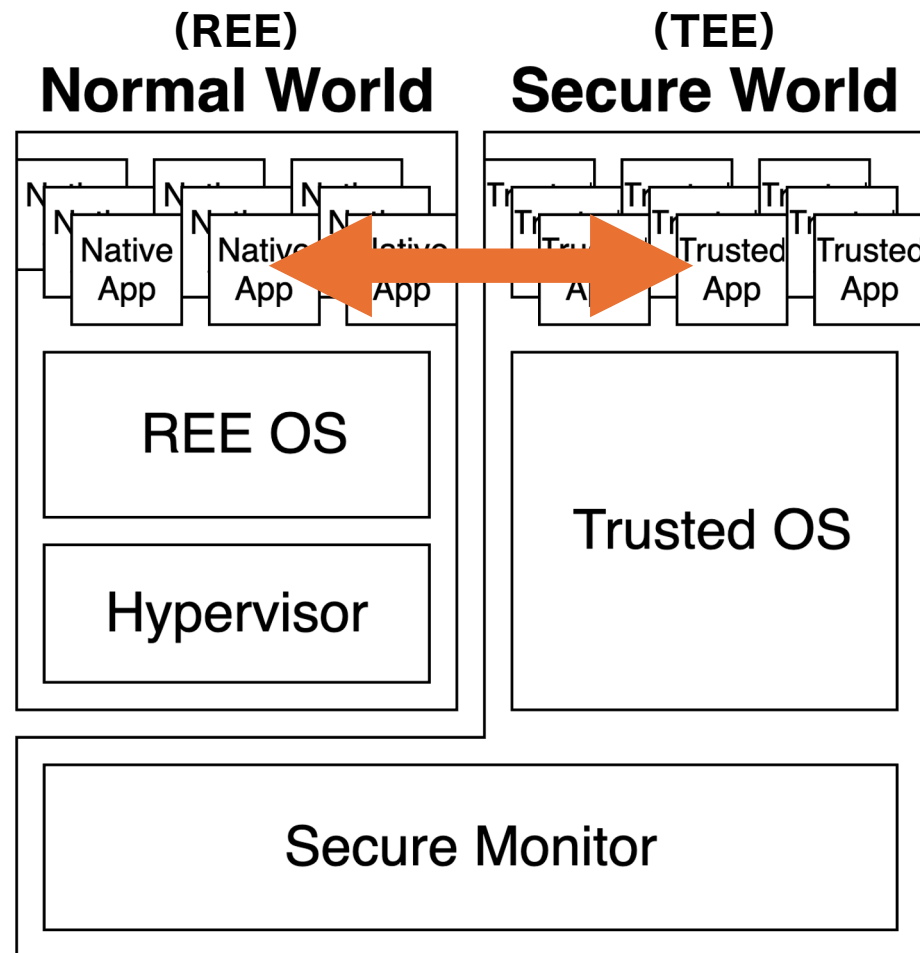
そもそもTEEとは

**暗号化などの“鍵”や処理を守る仕組み
=TEE**

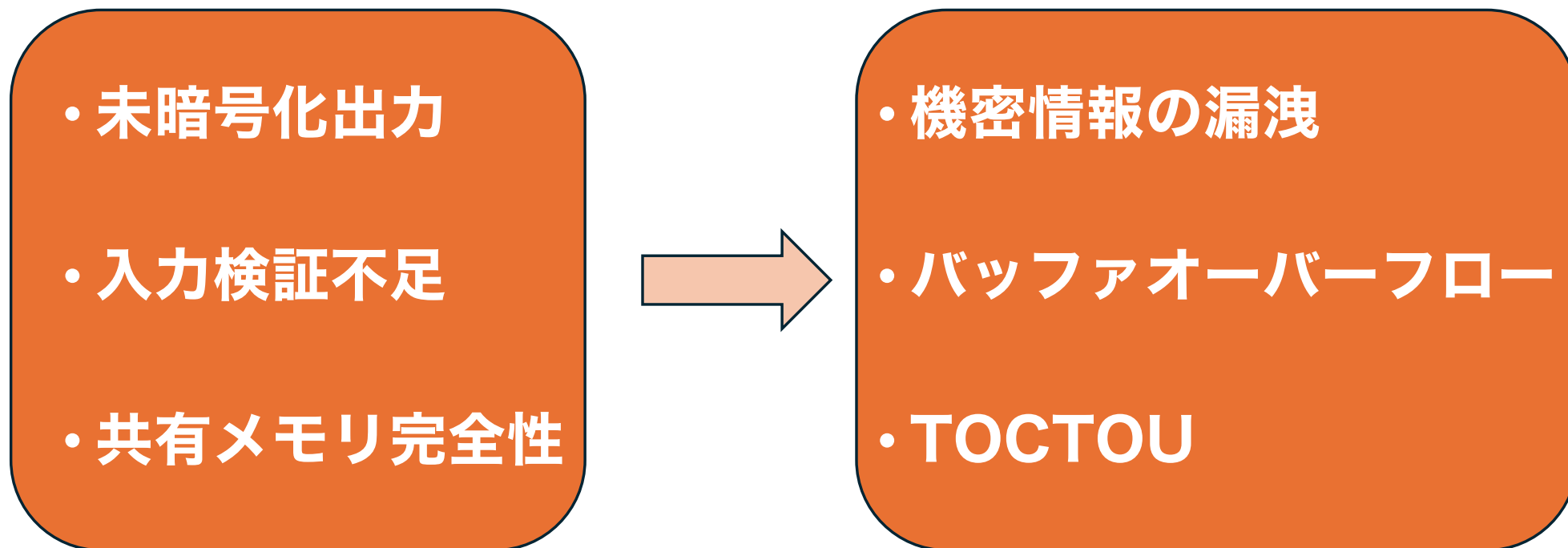
**CPU内にセキュア
な実行環境を構築し
機密処理や
情報を保存する**



Native AppがTrusted Appに処理を依頼する



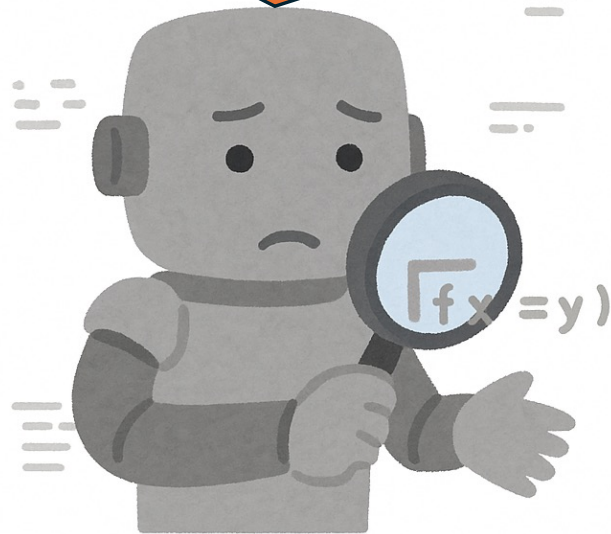
本発表で扱う3つのリスク [1]



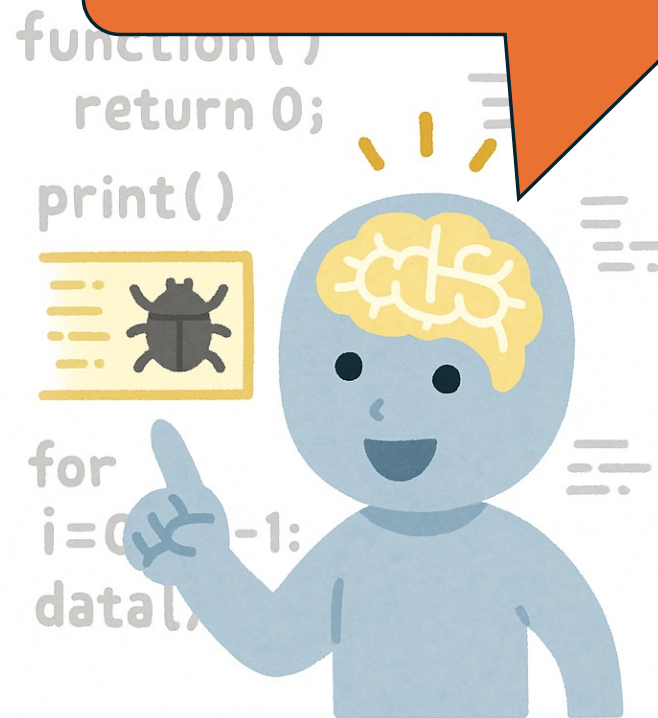
[1] Chengyan Ma et al : "DITING: A Static Analyzer for Identifying Bad Partitioning Issues in TEE Applications", arXiv, eprint={2502.15281}, 2025

問題を検出するために

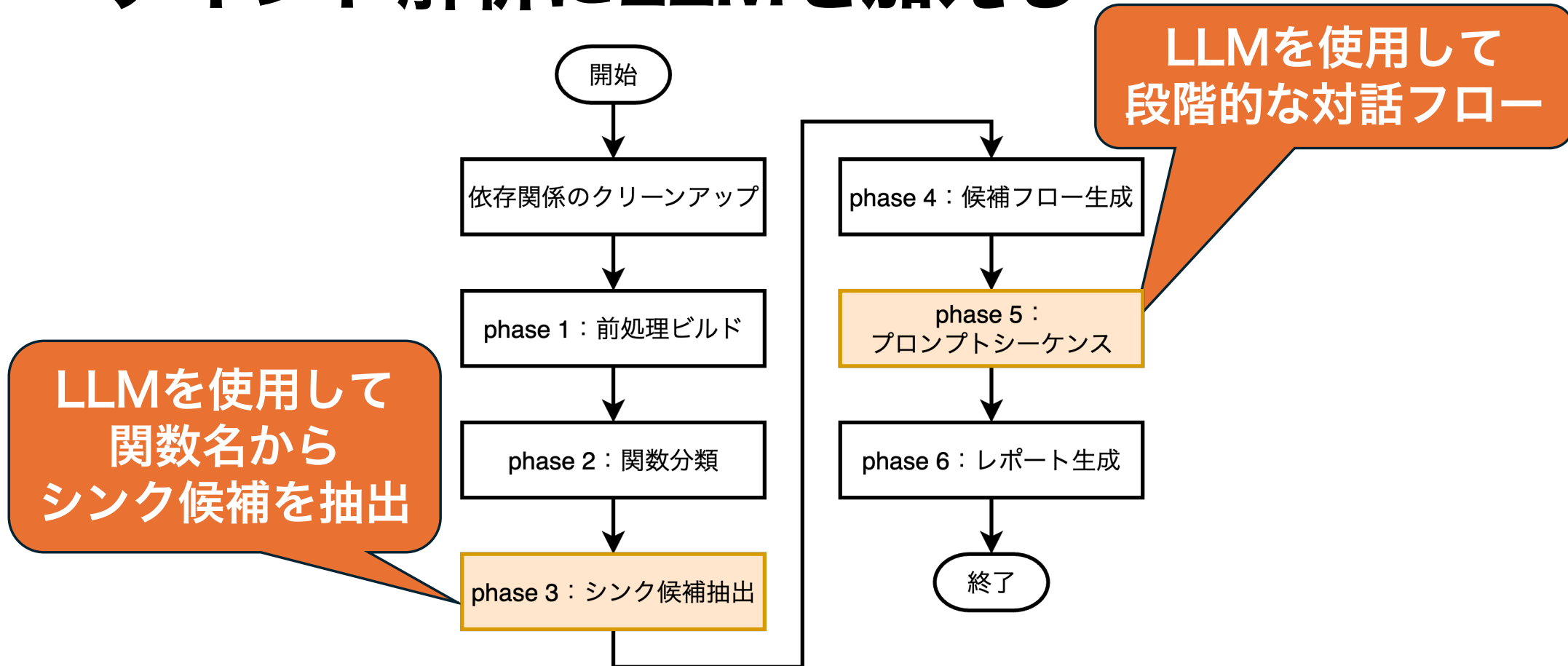
既存のルールベースでは
文脈理解や複雑な処理への
検知が難しい



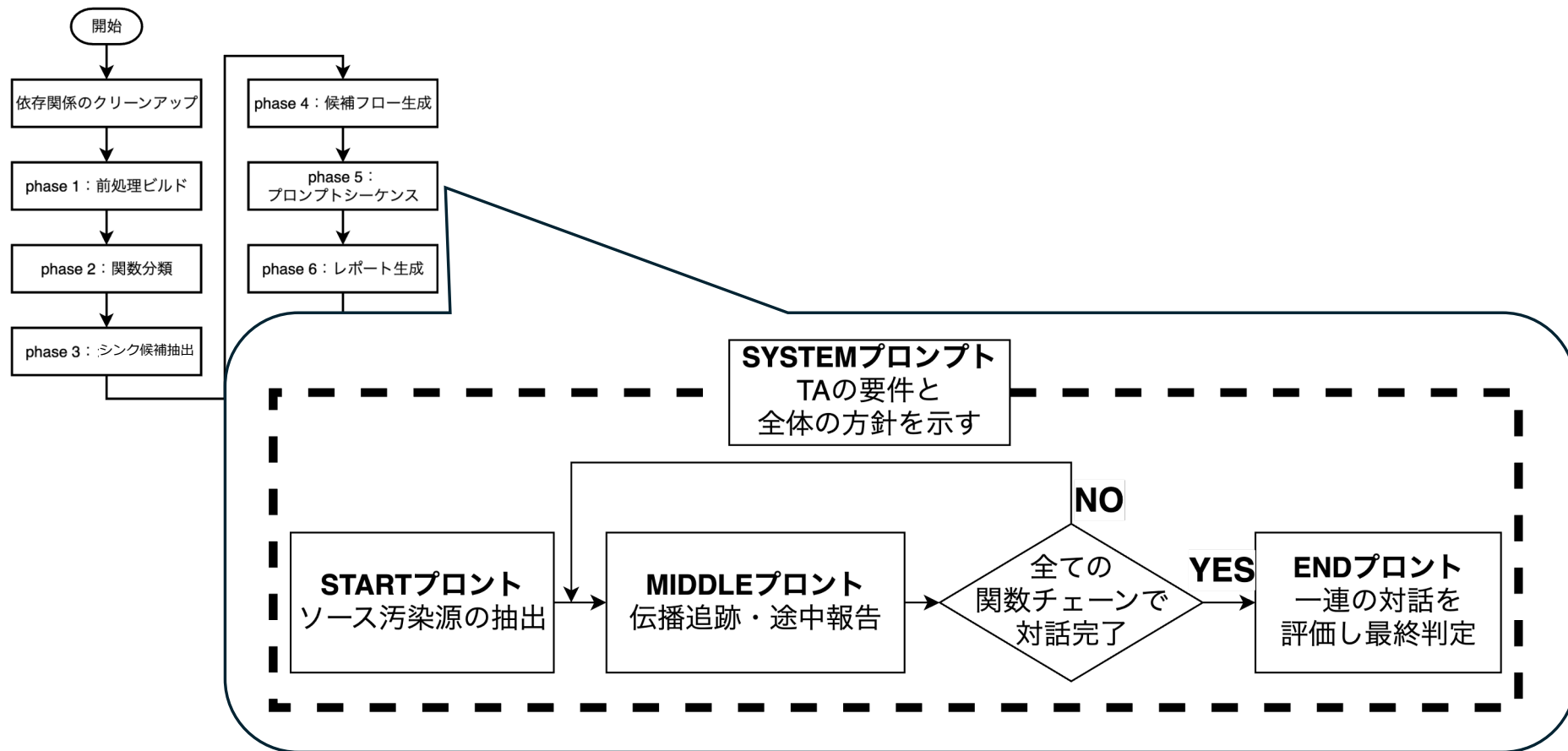
LLMを使用することで
解決できるのでは！？



テイント解析にLLMを加える



関数チェーンの関数ごとにLLMと対話



実験：既存ツール（DITING^[1]）と比較

GPT-5-mini ルールベース

	本システム	DITING
Detected	46	81
TP	34	55
FP	12	26
FN	41	20
Precision	0.739	0.679
Recall	0.453	0.733
F1	0.561	0.705

[1] Chengyan Ma et al : “DITING: A Static Analyzer for Identifying Bad Partitioning Issues in TEE Applications”, arXiv, eprint={2502.15281}, 2025

LLMとルールベース比較

評価軸	LLM検知	ルール検知
① 定型パターン	△～○：過剰/過少に振れる	◎：機械的に強い・安定
② サニタイズ評価	△：見落としが 発生しやすい	△：静的パターン化が 難しい
③ ドメイン文脈	◎：文脈理解で優位	△：細粒度ルールが膨張
④ 再現性	△：プロンプト・LLM依存	◎：安定・監査しやすい

結論：LLMでは文脈理解による検知が容易であった

今後の夢

- 精度向上によるLLMへの置き換え
- TAバイナリへの適応によりAndroidファームウェアでの大規模調査

本システムのGitHub→

