CAL

Automated Code Annotation with LLMs for Establishing TEE Boundaries

Varun Gadey[1], Melanie Götz[2], Christoph Sendner[3], Sampo Sovio[4], Alexandra Dmitrienko[1]

[1]University of Duisburg-Essen, [2]University of Würzburg, [3]University of California, Irvine, [4]Huawei Technologies, Finland

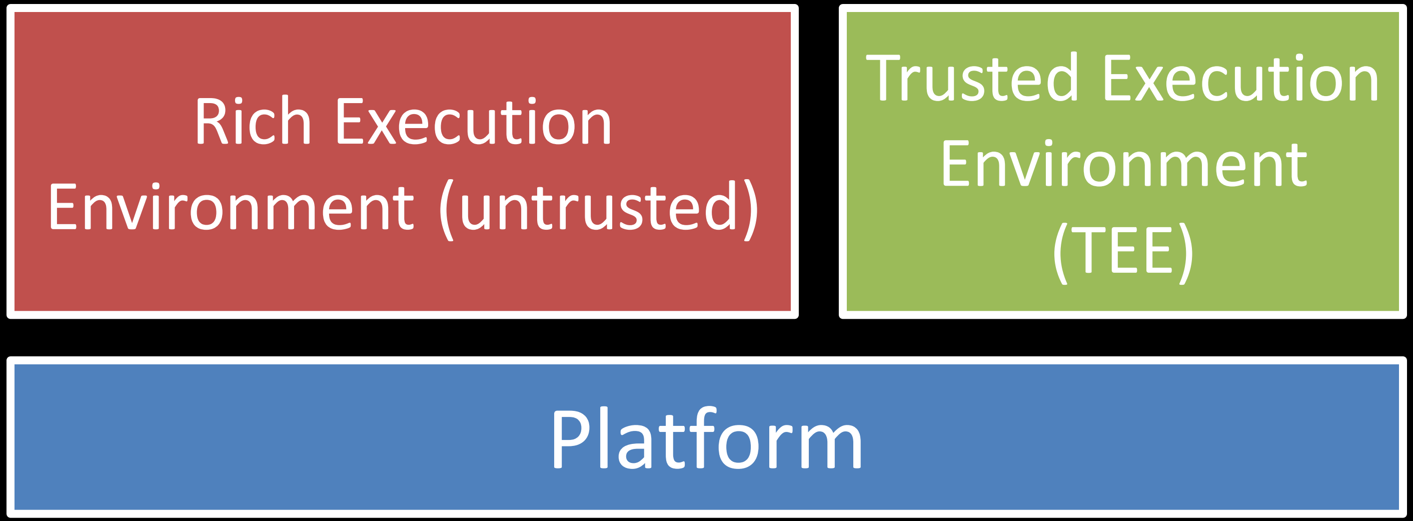
CAL

Automated Code Annotation with LLMs for Establishing TEE Boundaries

Varun Gadey[1], Melanie Götz[1], Christoph Sendner[1],
Sampo Sovio[2], Alexandra Dmitrienko[1]

[1]University of Würzburg, Germany
[2]Huawei Technologies, Finland

Problem



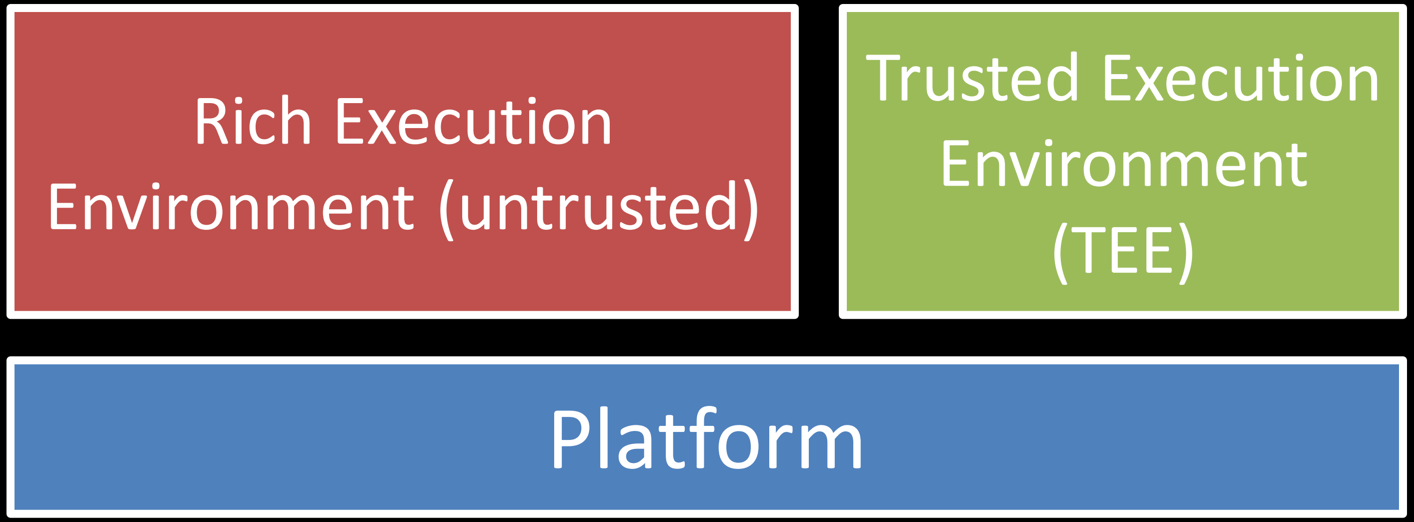
Rich Execution Environment (untrusted)

Trusted Execution Environment (TEE)

Platform

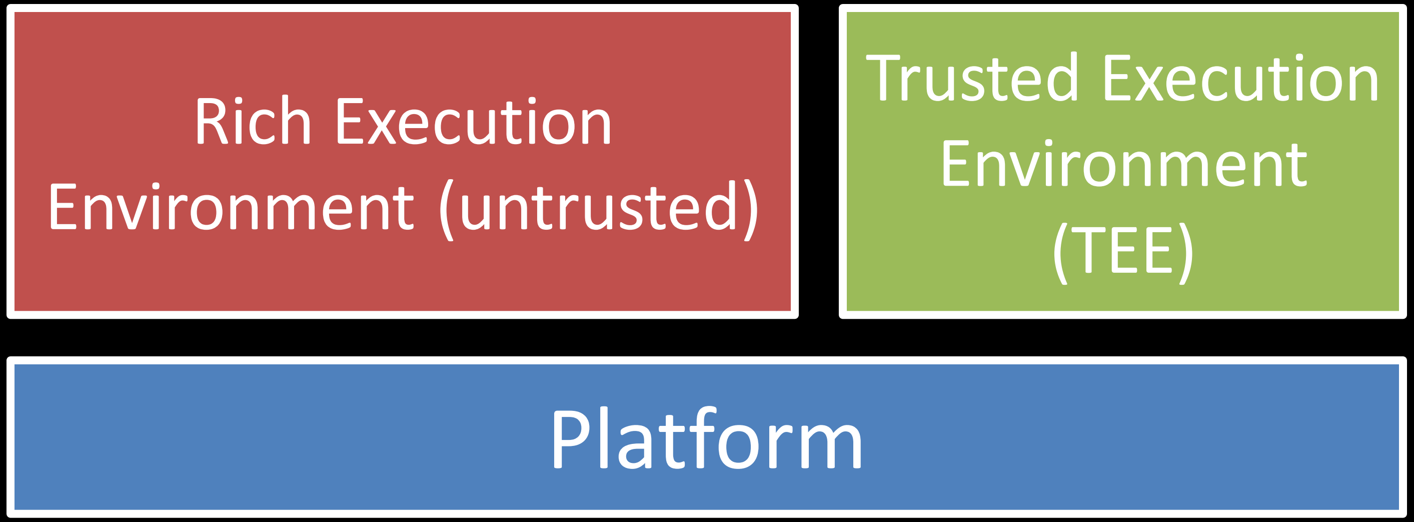
Problem

- Which code is security sensitive and need to run within TEE?


Rich Execution Environment (untrusted)
Trusted Execution Environment (TEE)
Platform

Problem

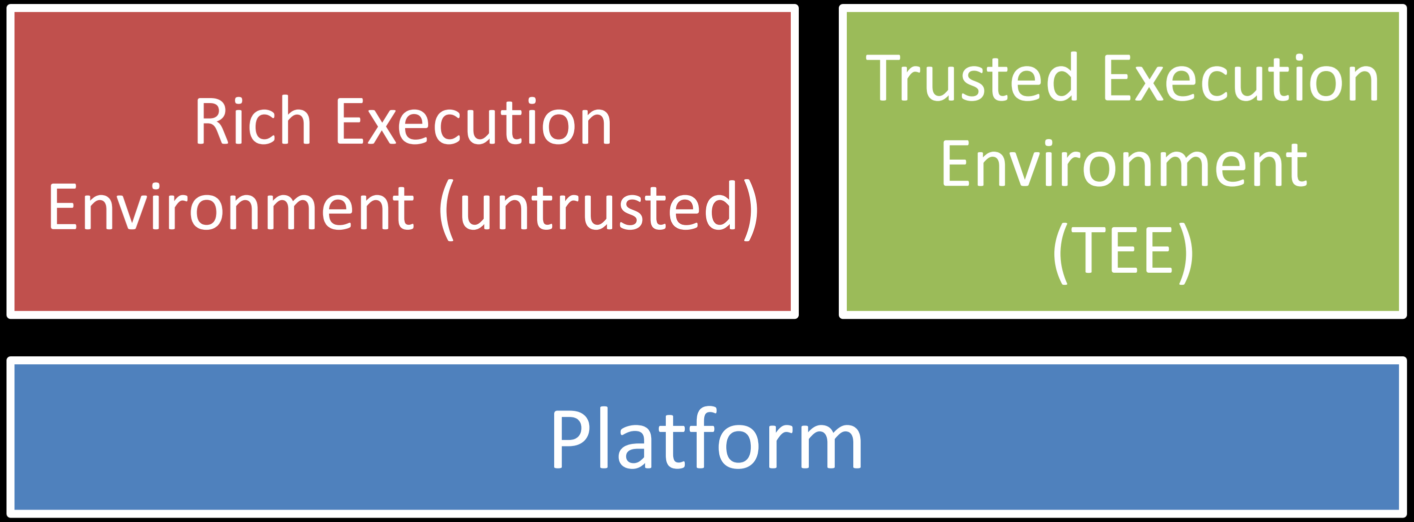
- Which code is security sensitive and need to run within TEE?
- Today, this annotation is done manually (e.g. Soaap[1] and *Datashield[2]*)


Rich Execution Environment (untrusted)
Trusted Execution Environment (TEE)
Platform

1. K. Gudka, R. N. Watson, J. Anderson, D. Chisnall, B. Davis, B. Laurie, I. Marinos, P. G. Neumann, and A. Richardson, “Clean application compartmentalization with soaap,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015,pp. 1016–1031
2. Z. Kong, M. Park, L. Guan, N. Zhang, and C. H. Kim, “TZ-DATASHIELD: Automated Data Protection for Embedded Systems via Data-Flow-Based Compartmentalization,” in Proceedings of the 32nd Network and Distributed System Security Symposium (NDSS2025), San Diego, CA, Feb. 2025.

Problem

- Which code is security sensitive and need to run within TEE?
- Today, this annotation is done manually (e.g. Soaap[1] and *Datashield[2]*)
- Alternatively, entire applications are moved to TEE (e.g. *Graphene[3]* and *Scone[4]*)



Rich Execution Environment (untrusted)

Trusted Execution Environment (TEE)

Platform

1. K. Gudka, R. N. Watson, J. Anderson, D. Chisnall, B. Davis, B. Laurie, I. Marinos, P. G. Neumann, and A. Richardson, "Clean application compartmentalization with soaap," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015,pp. 1016–1031
2. Z. Kong, M. Park, L. Guan, N. Zhang, and C. H. Kim, "TZ-DATASHIELD: Automated Data Protection for Embedded Systems via Data-Flow-Based Compartmentalization," in Proceedings of the 32nd Network and Distributed System Security Symposium (NDSS2025), San Diego, CA, Feb. 2025.
3. C.-C. Tsai, D. E. Porter, and M. Vij, "Graphene-SGX: A practical library OS for unmodified applications on SGX," in 2017 USENIX Annual Technical Conference (USENIX ATC 17), 2017, pp. 645–658.
4. S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'keeffe, M. L. Stillwell et al., "SCONE: Secure Linux containers with Intel SGX," in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp.689–703

Motivation and Goal of CAL

Larger TCB

More potential for vulnerabilities

Motivation and Goal of CAL

Larger TCB

More potential for vulnerabilities

Manual code analysis

Unscalable

Motivation and Goal of CAL

- We aim to fully automate this process!

Larger TCB

More potential for vulnerabilities

Manual code analysis

Unscalable

Motivation and Goal of CAL

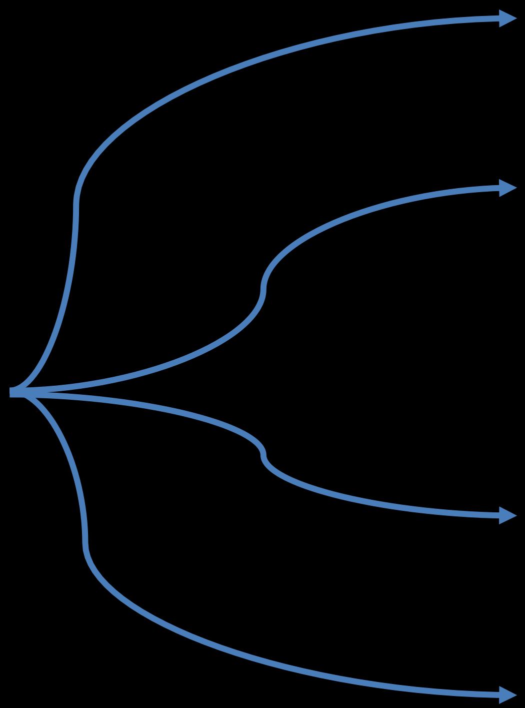
- We aim to fully automate this process!

Larger TCB

More potential for vulnerabilities

Manual code analysis

Unscalable



Using CAL algorithm

Motivation and Goal of CAL

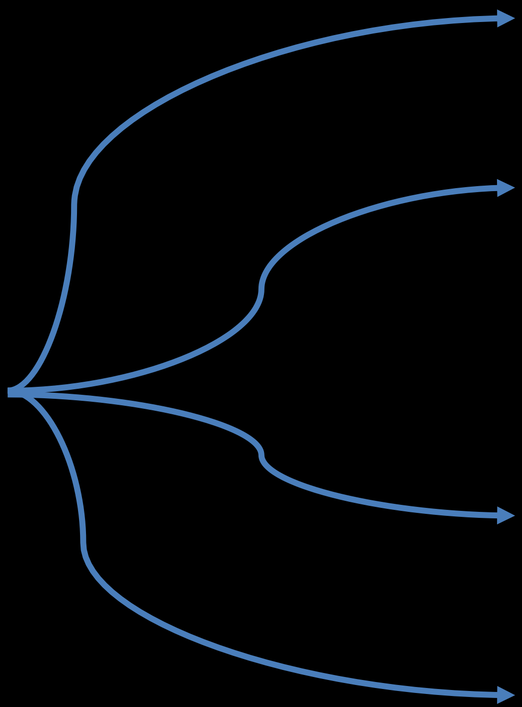
- We aim to fully automate this process!

Larger TCB

More potential for vulnerabilities

Manual code analysis

Unscalable



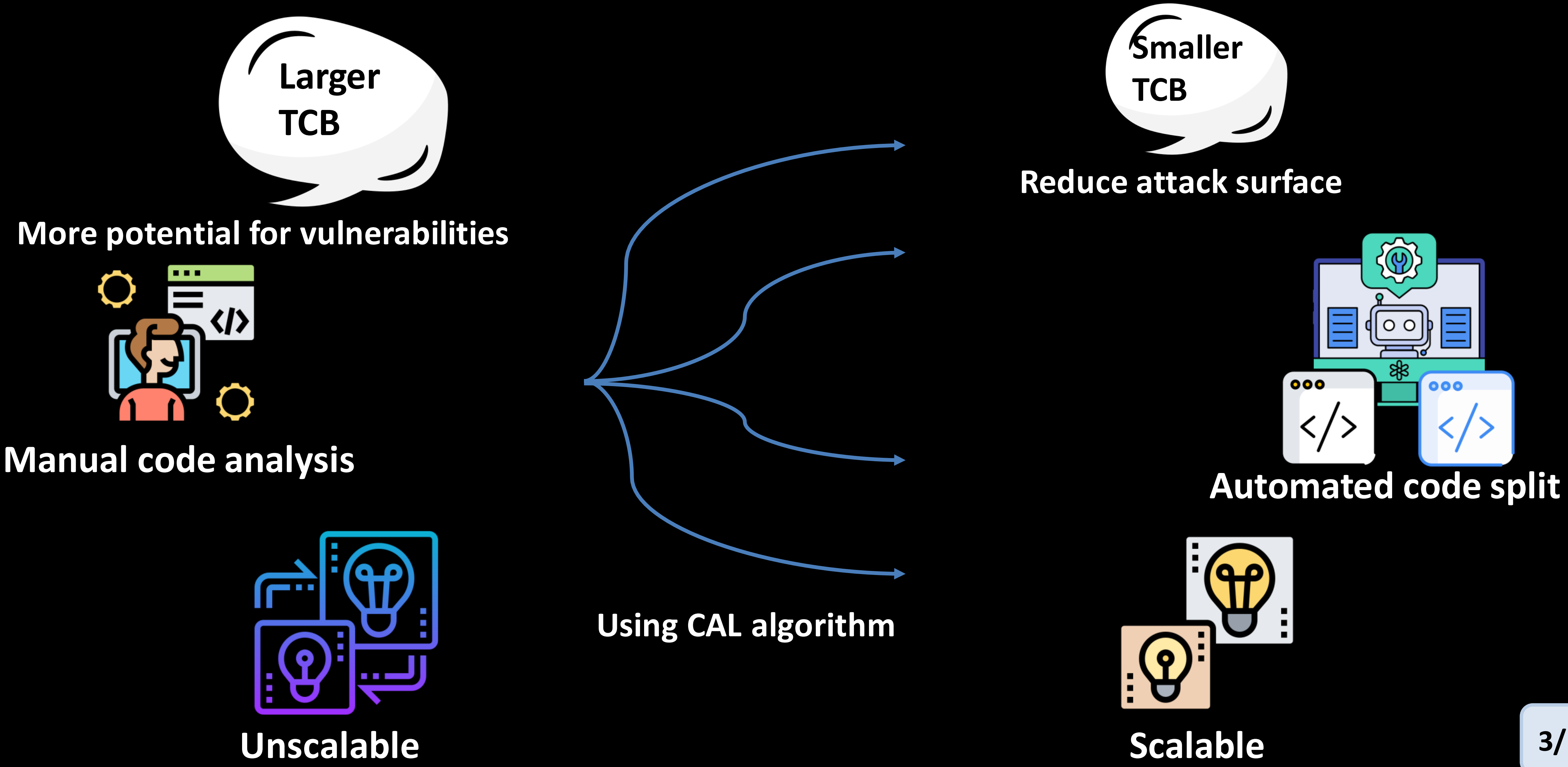
Using CAL algorithm

Smaller TCB

Reduce attack surface

Motivation and Goal of CAL

- We aim to fully automate this process!



Larger TCB

More potential for vulnerabilities

Manual code analysis

Unscalable

Using CAL algorithm

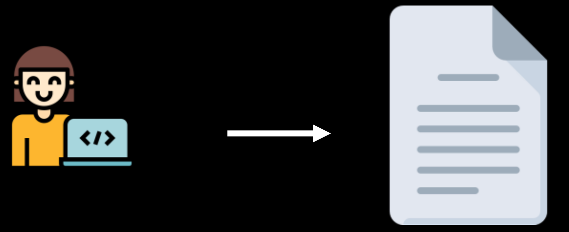
Smaller TCB

Reduce attack surface

Automated code split

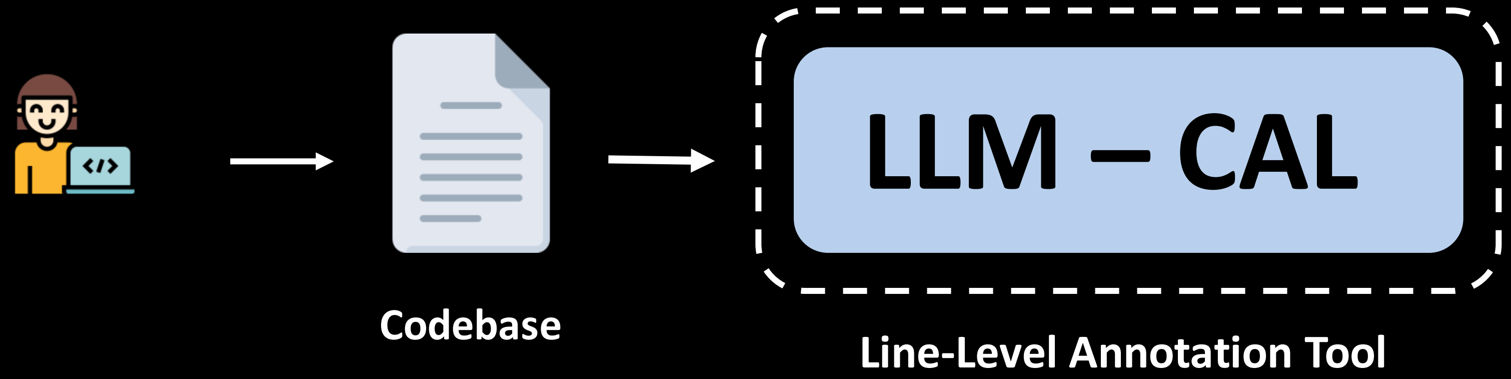
Scalable

LLM-CAL: General Idea

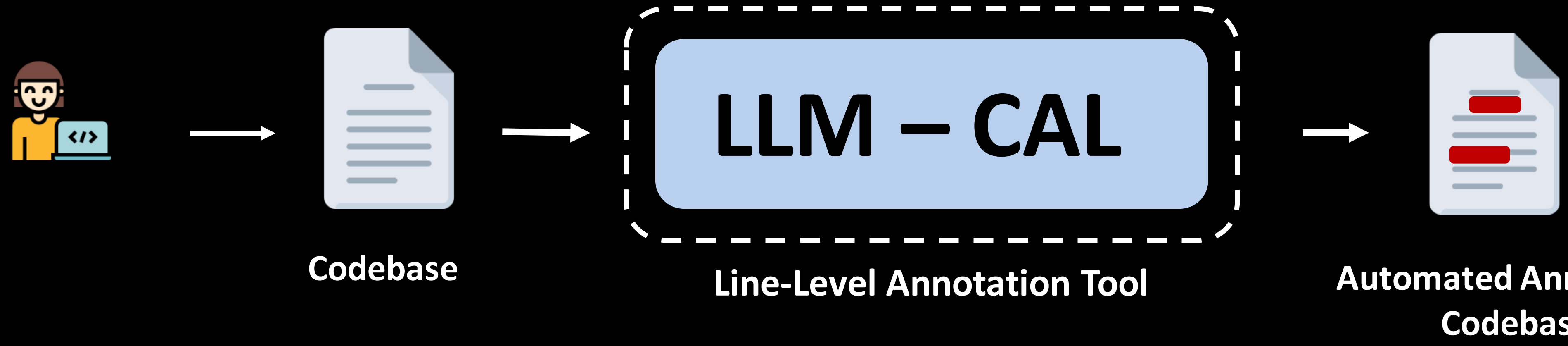


Codebase

LLM-CAL: General Idea



LLM-CAL: General Idea



Contributions

Security Sensitive Code notion and Dataset

Contributions

Security Sensitive Code notion and Dataset

LLM-CAL: A novel, scalable and memory-efficient framework

Contributions

Security Sensitive Code notion and Dataset

LLM-CAL: A novel, scalable and memory-efficient framework

Comprehensive Evaluation & Out-of-distribution case studies

Contributions

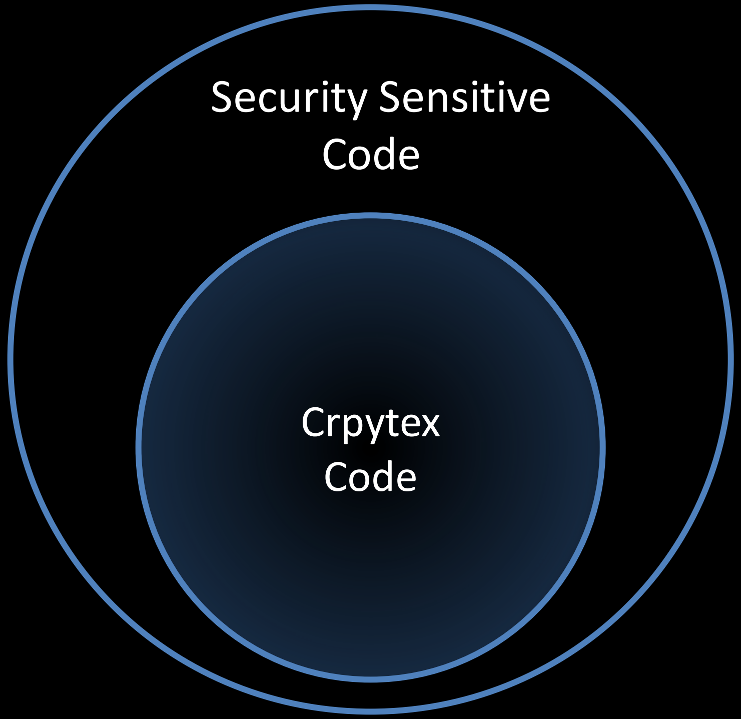
Security Sensitive Code notion and Dataset

LLM-CAL: A novel, scalable and memory-efficient framework

Comprehensive Evaluation & Out-of-distribution case studies

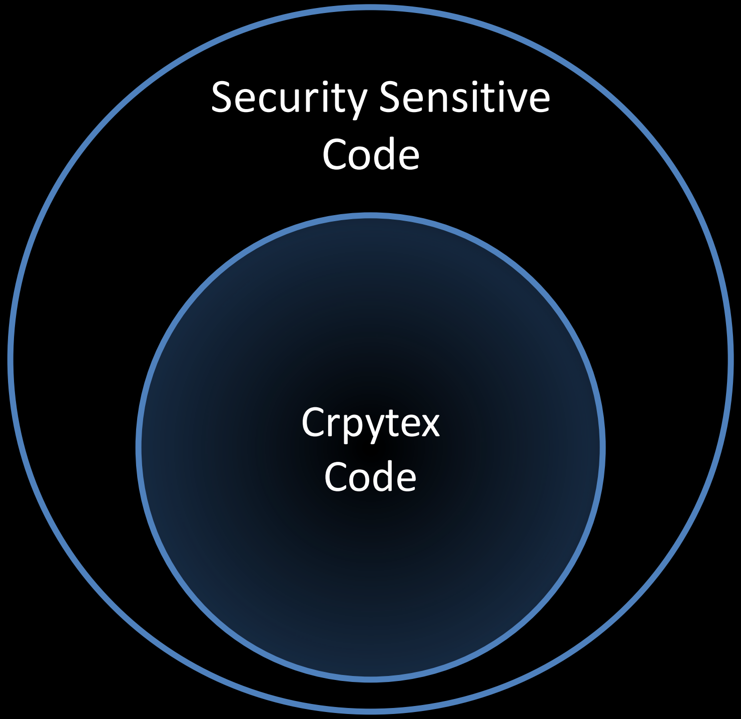
Crpytex Code - Notion

- Centered around cryptographic operations



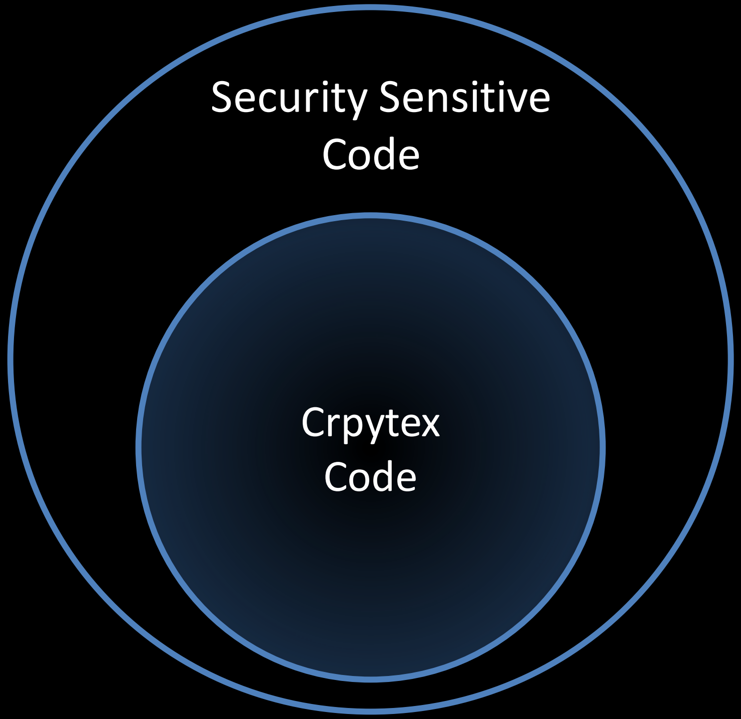
Crpytex Code - Notion

- Centered around cryptographic operations
- Includes inputs and outputs to cryptographic functions



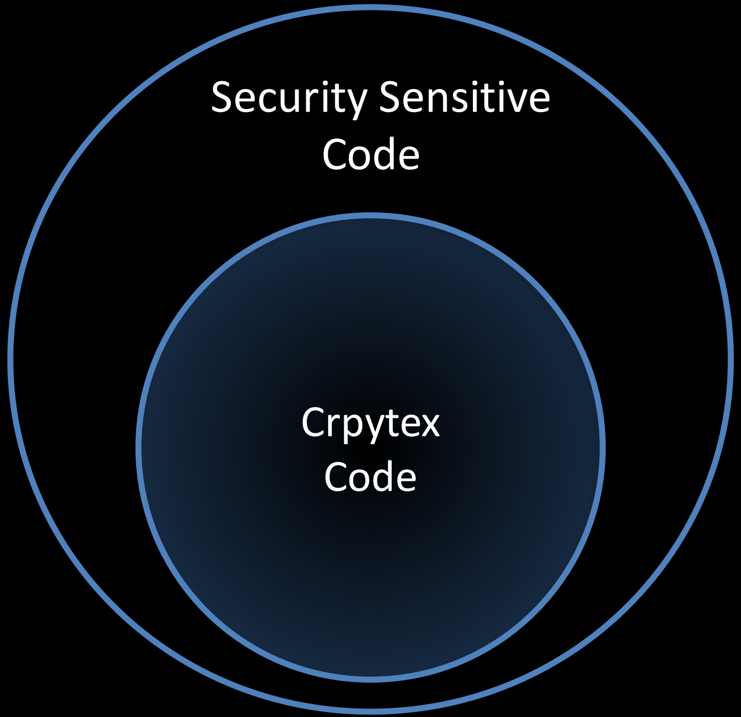
Crpytex Code - Notion

- Centered around cryptographic operations
- Includes inputs and outputs to cryptographic functions
- Follows the dataflow path to crpytographic `sinks`



Crpytex Code - Notion

- Centered around cryptographic operations
- Includes inputs and outputs to cryptographic functions
- Follows the dataflow path to crpytographic `sinks`



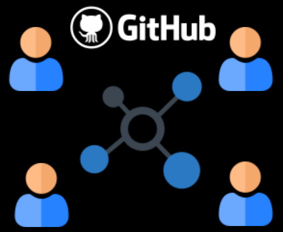
Access Control

Secure Communication

Authentication

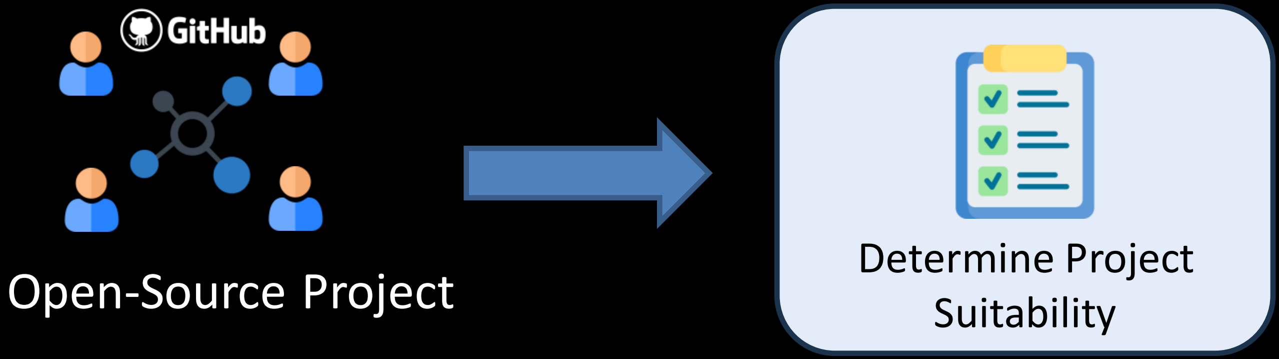
Data Encryption

Manual Dataset Construction

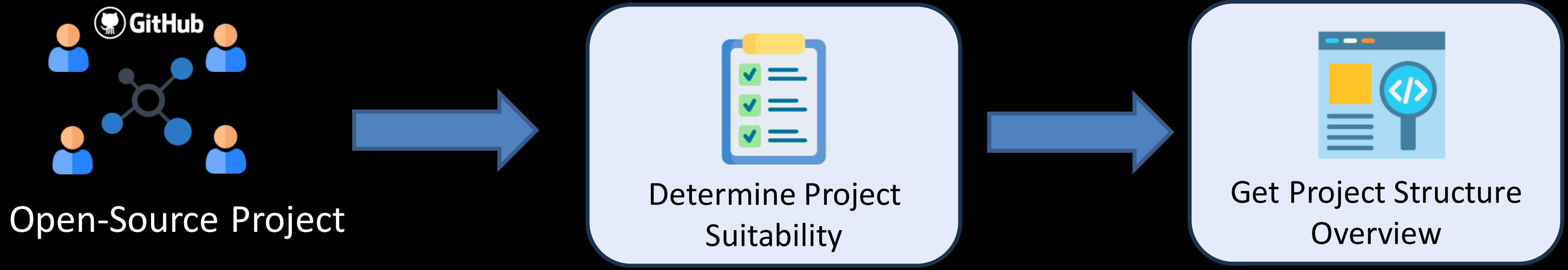


Open-Source Project

Manual Dataset Construction



Manual Dataset Construction

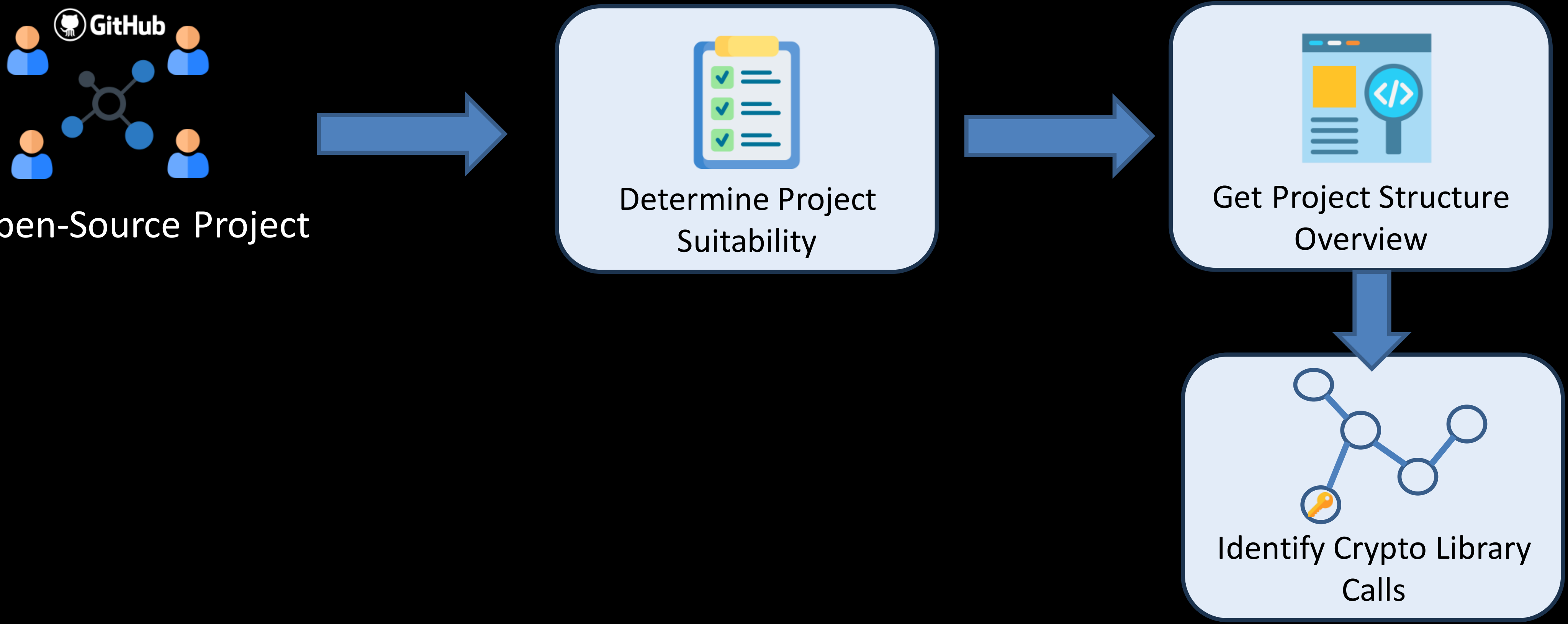


Open-Source Project

Determine Project Suitability

Get Project Structure Overview

Manual Dataset Construction



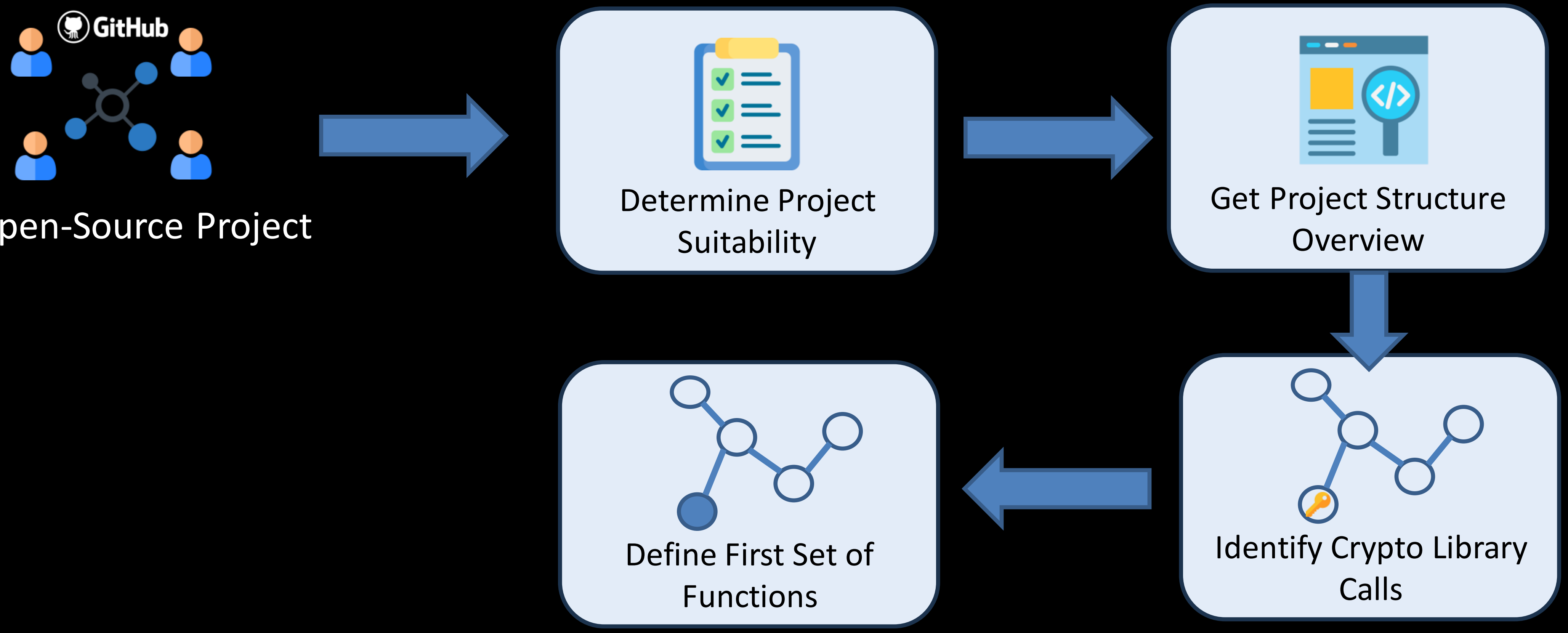
Open-Source Project

Determine Project Suitability

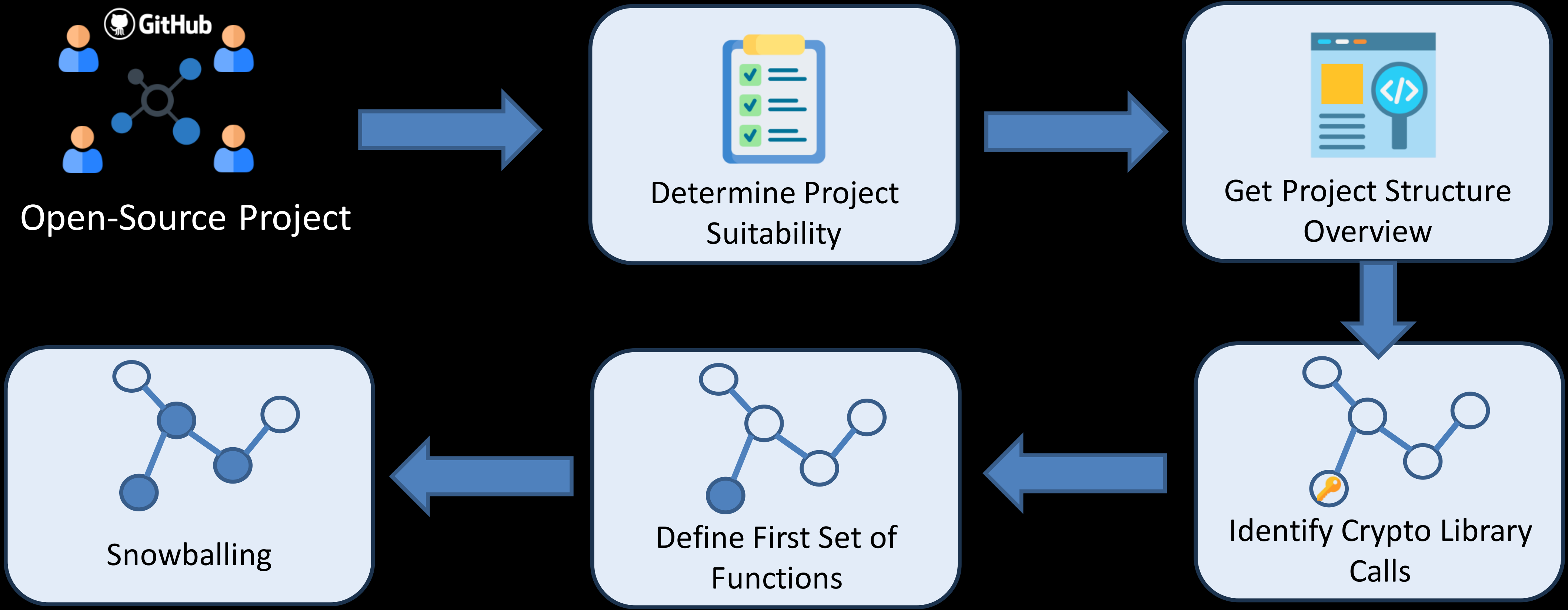
Get Project Structure Overview

Identify Crypto Library Calls

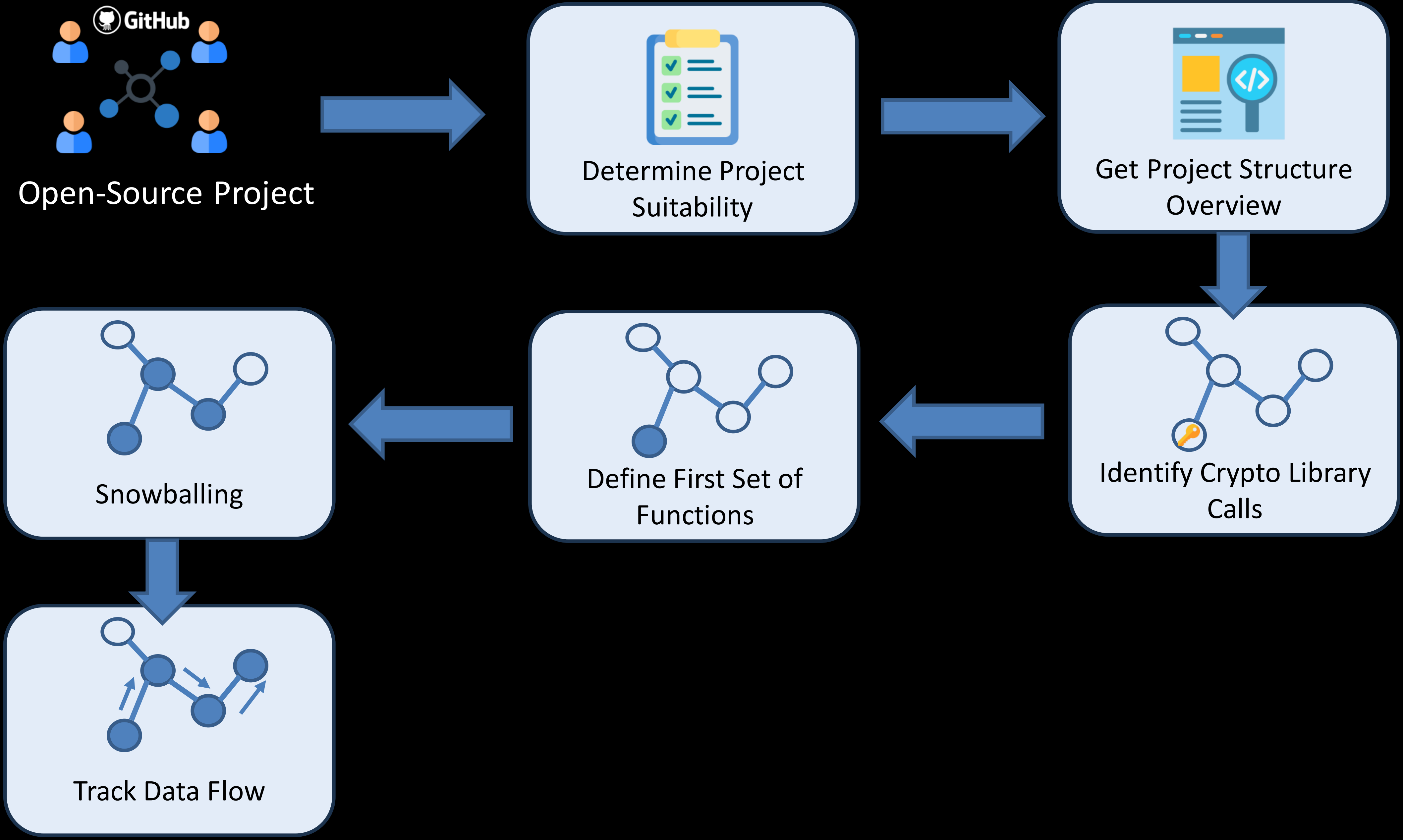
Manual Dataset Construction


GitHub
Open-Source Project
Determine Project Suitability
Get Project Structure Overview
Identify Crypto Library Calls
Define First Set of Functions

Manual Dataset Construction

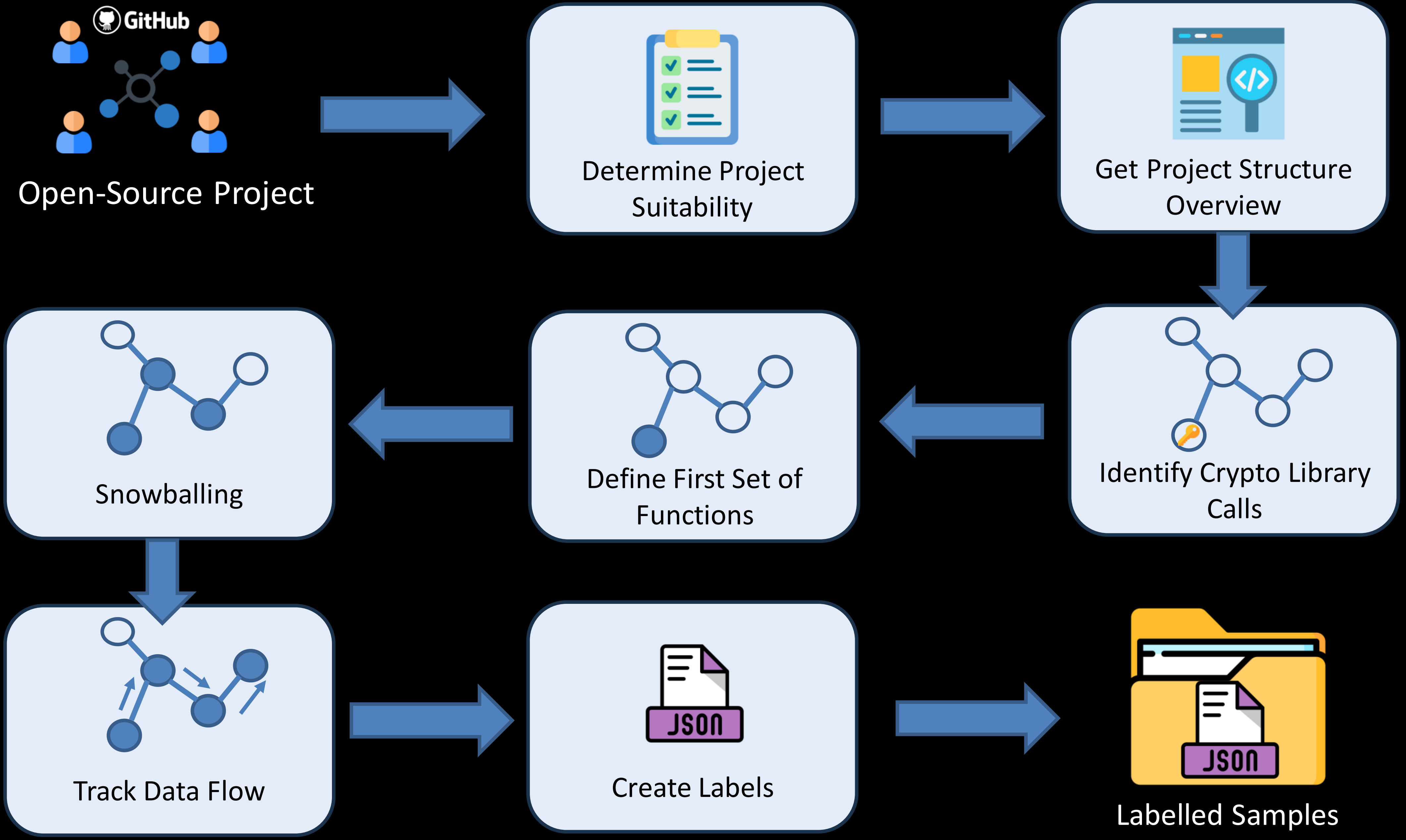

GitHub
Open-Source Project
Determine Project Suitability
Get Project Structure Overview
Identify Crypto Library Calls
Define First Set of Functions
Snowballing

Manual Dataset Construction



GitHub
Open-Source Project
Determine Project Suitability
Get Project Structure Overview
Identify Crypto Library Calls
Define First Set of Functions
Snowballing
Track Data Flow

Manual Dataset Construction



GitHub
Open-Source Project
Determine Project Suitability
Get Project Structure Overview
Identify Crypto Library Calls
Define First Set of Functions
Snowballing
Track Data Flow
JSON
Create Labels
JSON
Labelled Samples

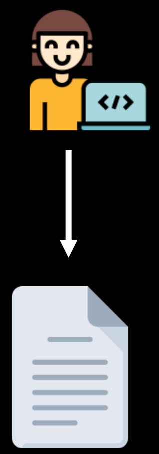
Contributions

Security Sensitive Code notion and Dataset

LLM-CAL: A novel, scalable and memory-efficient framework

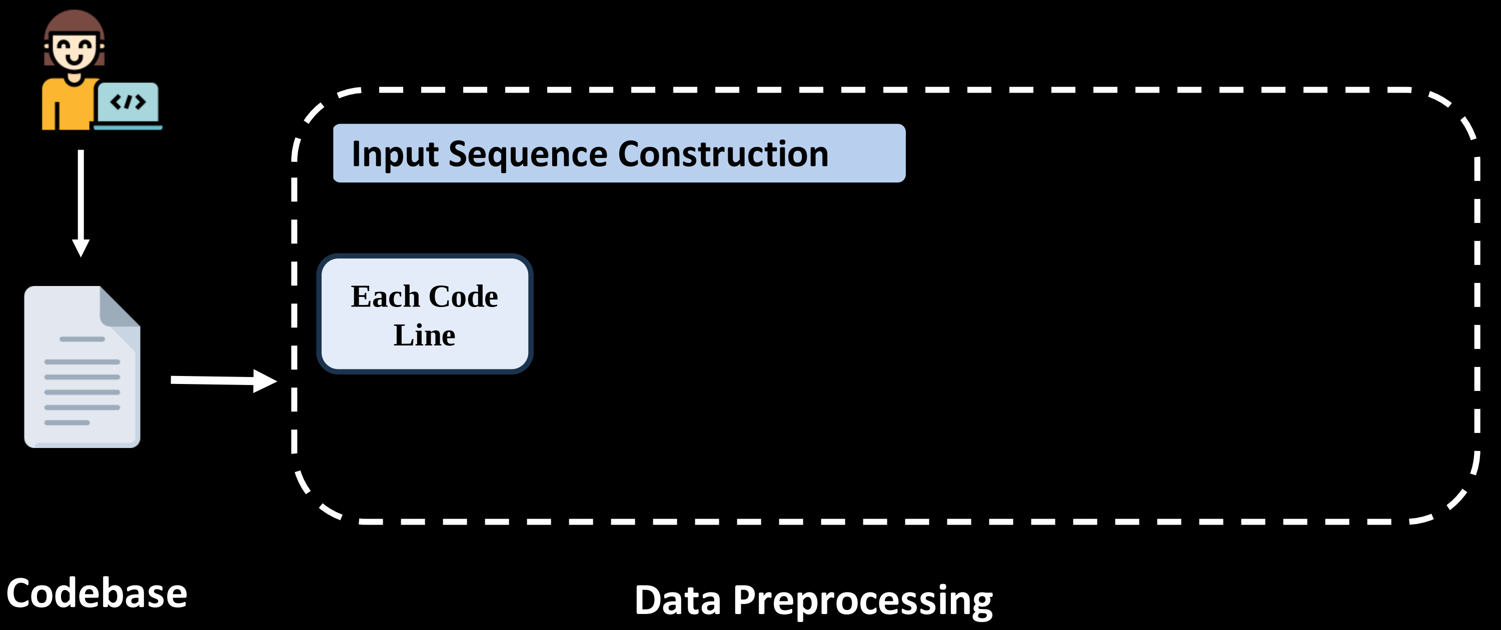
Comprehensive Evaluation & Out-of-distribution case studies

LLM-CAL: Detailed Workflow

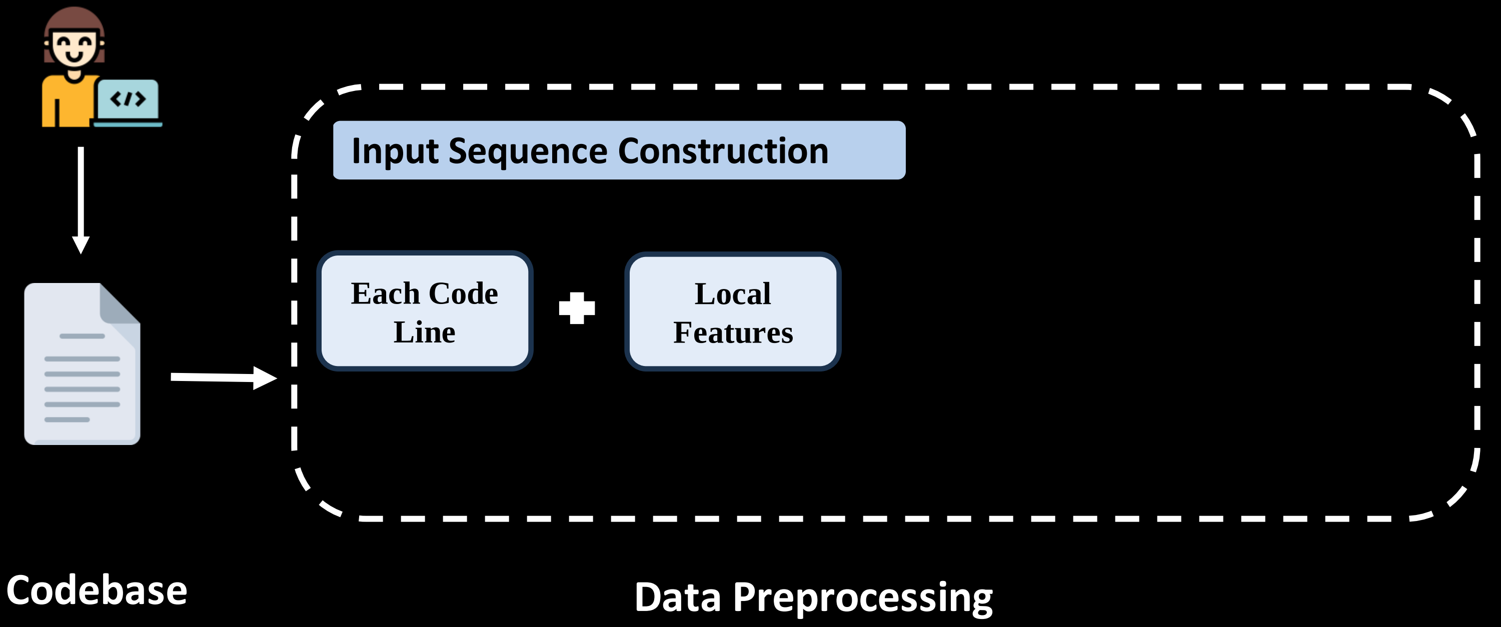


Codebase

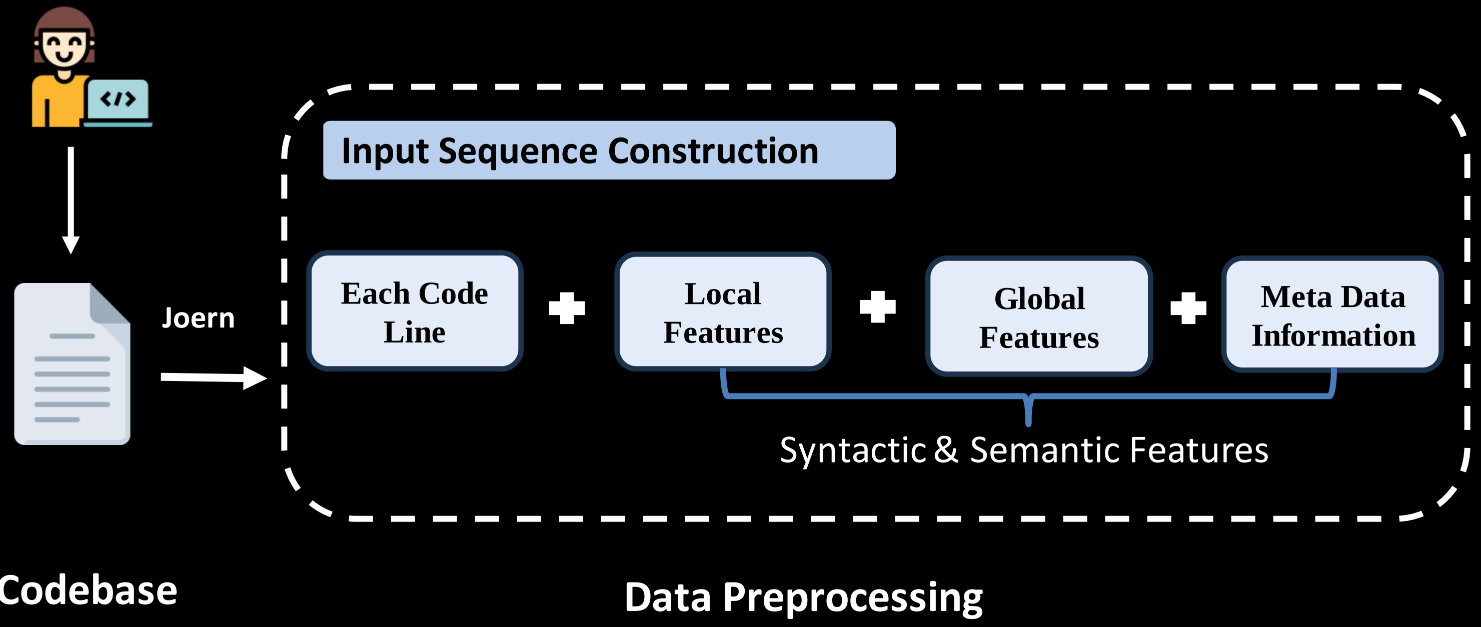
LLM-CAL: Detailed Workflow



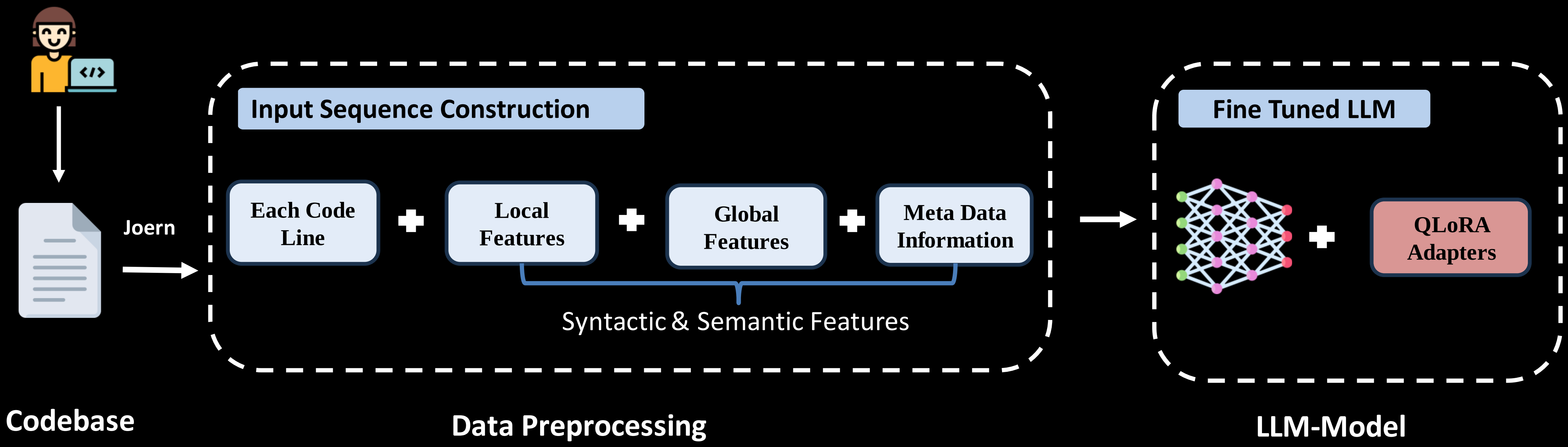
LLM-CAL: Detailed Workflow



LLM-CAL: Detailed Workflow



LLM-CAL: Detailed Workflow



Joern

Input Sequence Construction

Each Code Line + Local Features + Global Features + Meta Data Information

Syntactic & Semantic Features

Fine Tuned LLM

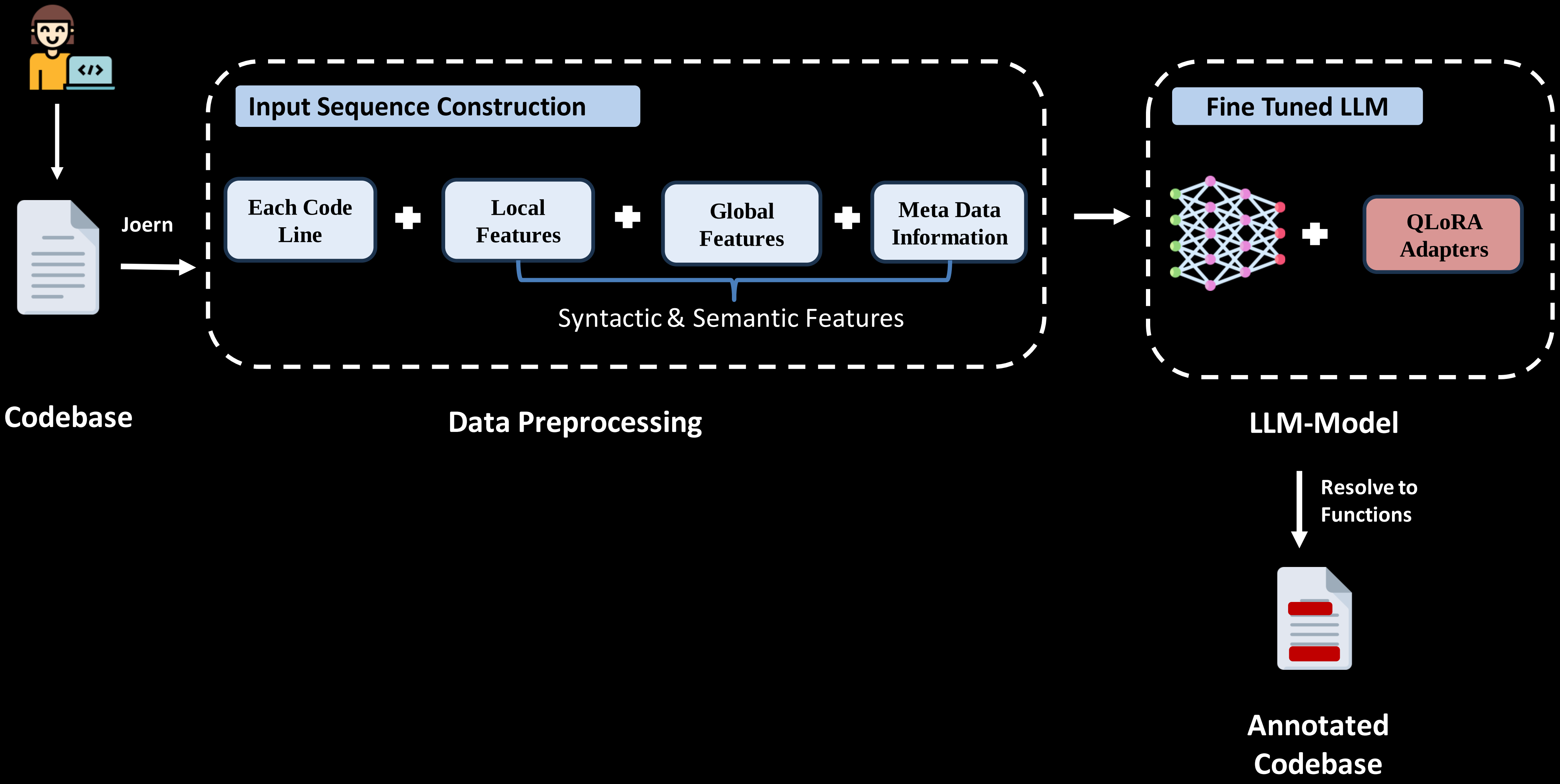
+ QLoRA Adapters

Codebase

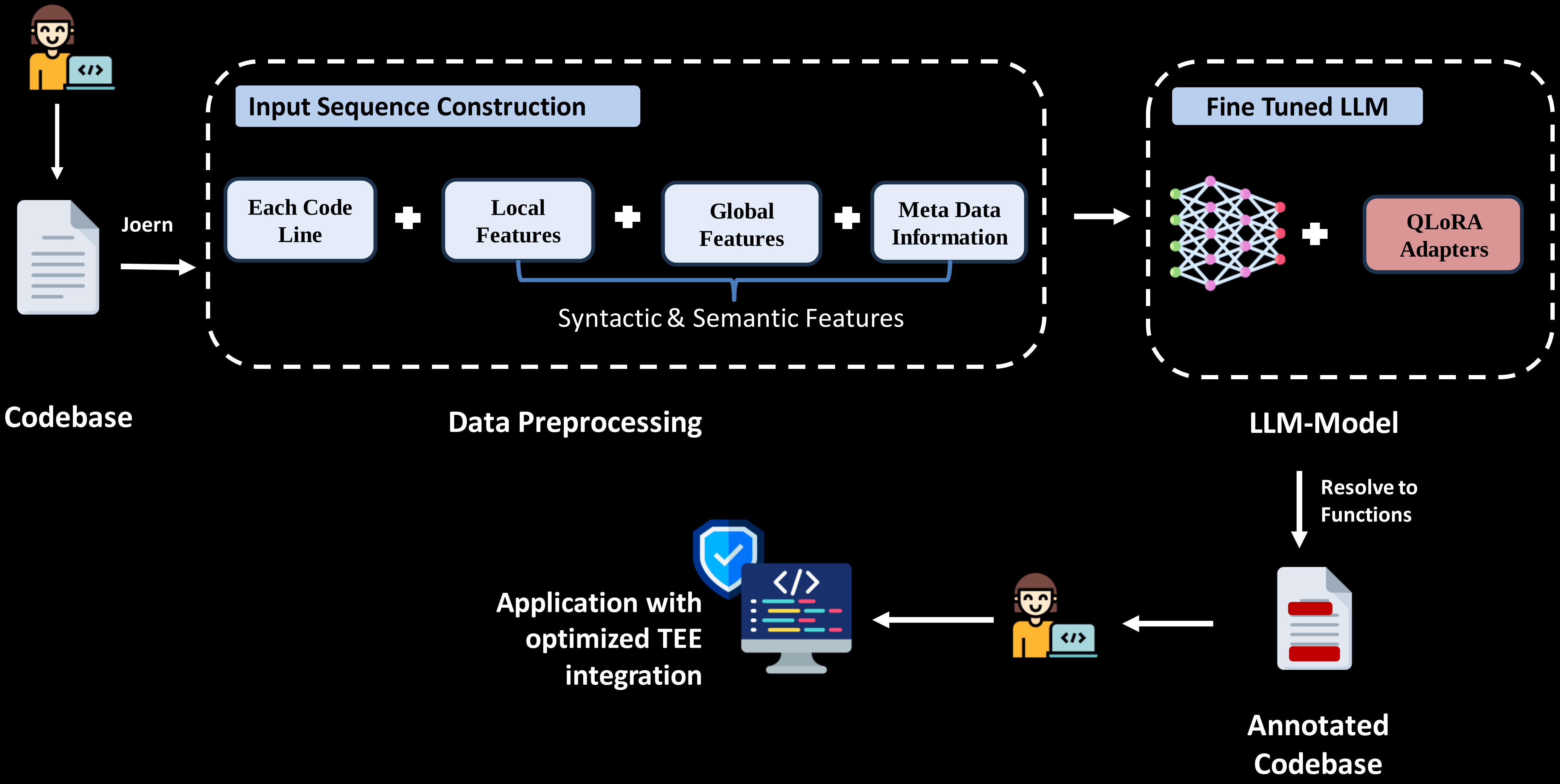
Data Preprocessing

LLM-Model

LLM-CAL: Detailed Workflow



LLM-CAL: Detailed Workflow



Codebase

Joern

Data Preprocessing

Input Sequence Construction

Each Code Line + Local Features + Global Features + Meta Data Information

Syntactic & Semantic Features

LLM-Model

Fine Tuned LLM

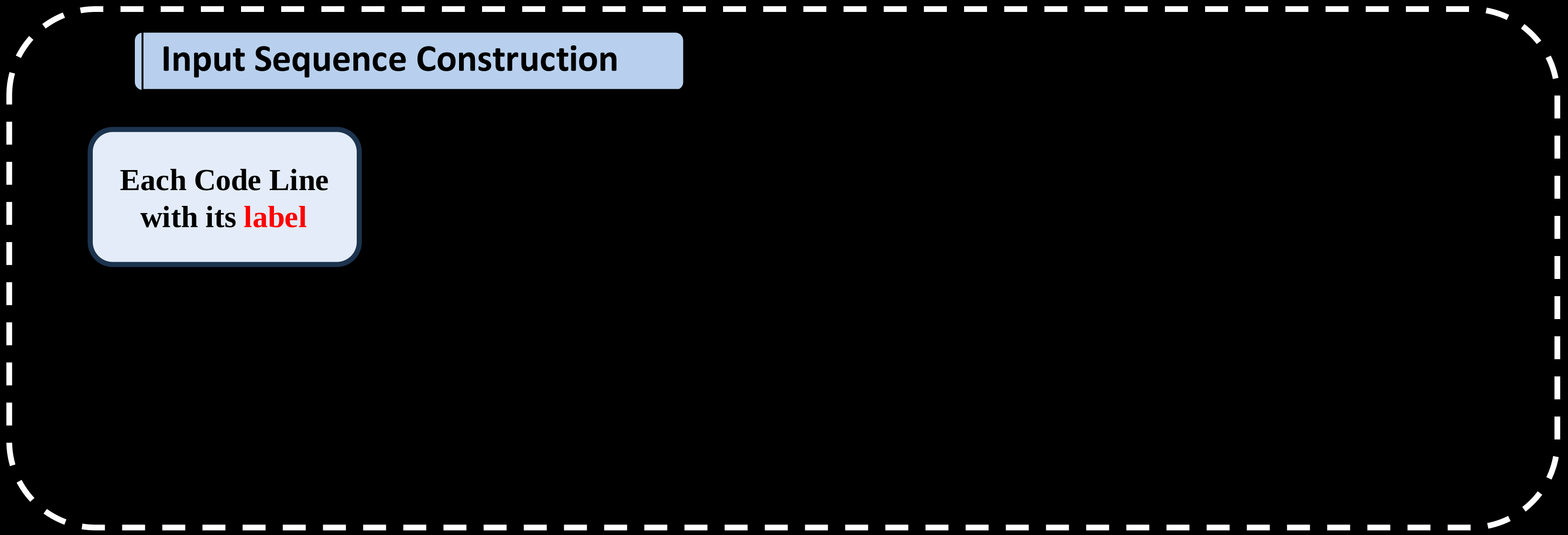
QLoRA Adapters

Resolve to Functions

Annotated Codebase

Application with optimized TEE integration

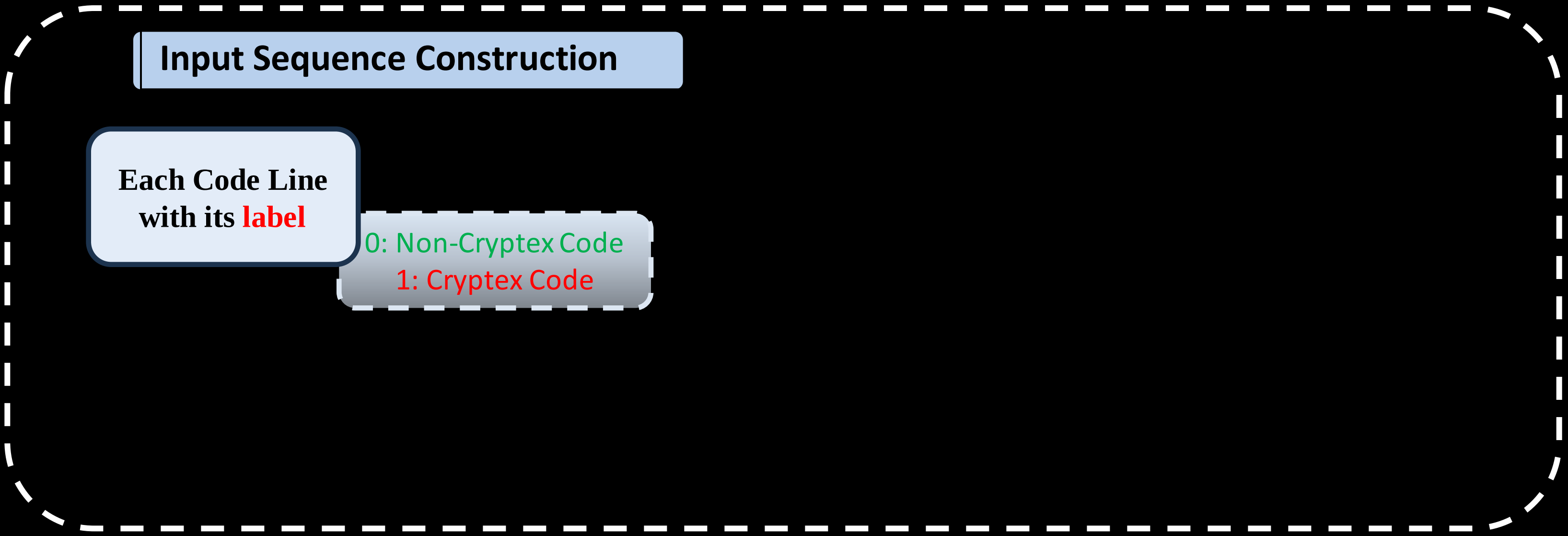
Input Sequence Construction: Local & Metadata Features



Input Sequence Construction

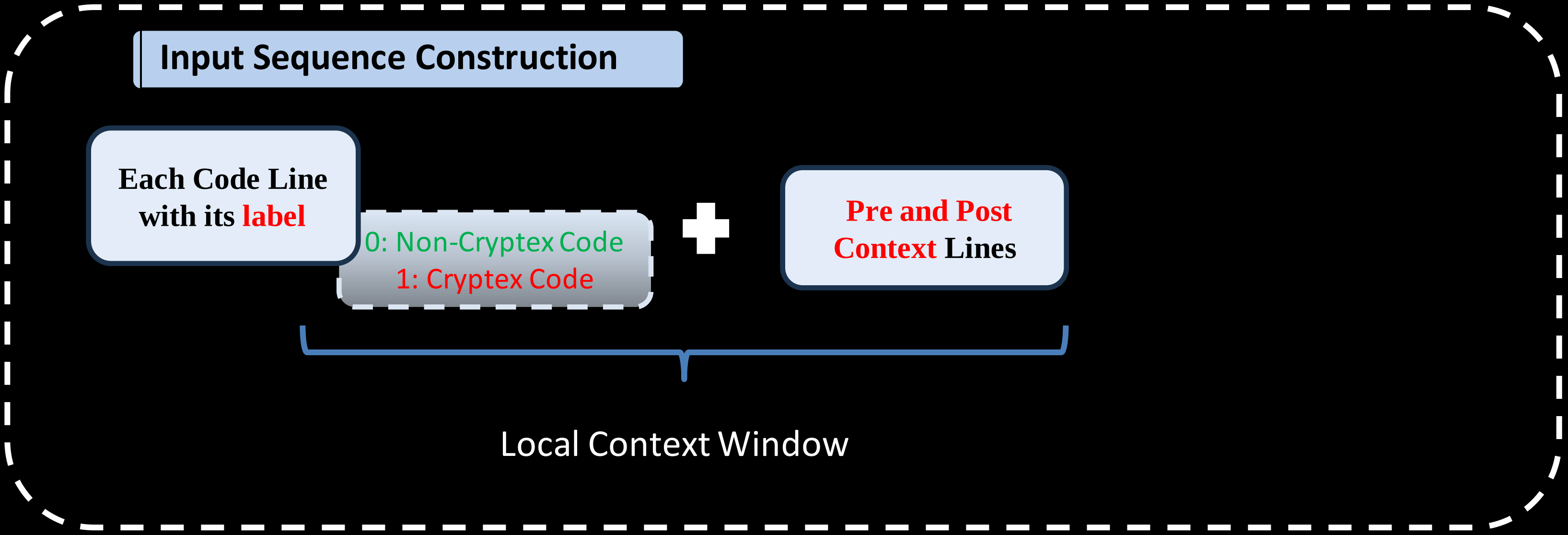
Each Code Line with its label

Input Sequence Construction: Local & Metadata Features



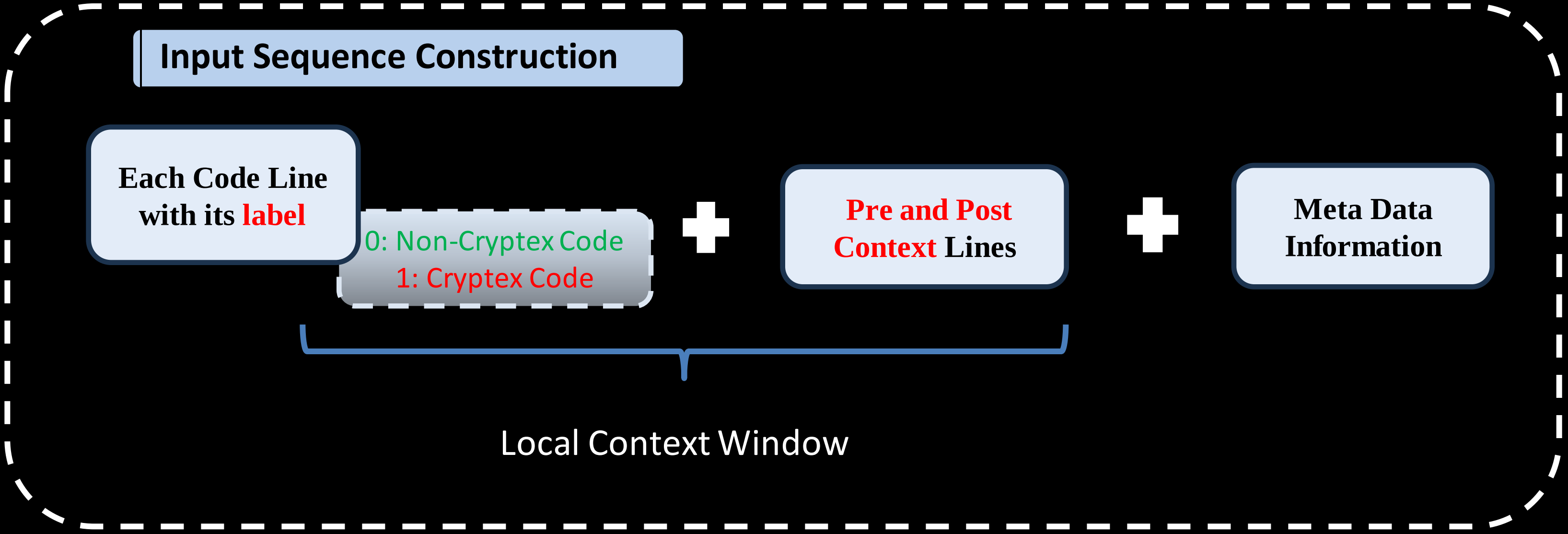
- During training, each code line is guided with its label either 0 or 1

Input Sequence Construction: Local & Metadata Features


Input Sequence Construction
Each Code Line with its label
0: Non-Cryptex Code
1: Cryptex Code
Pre and Post Context Lines
Local Context Window

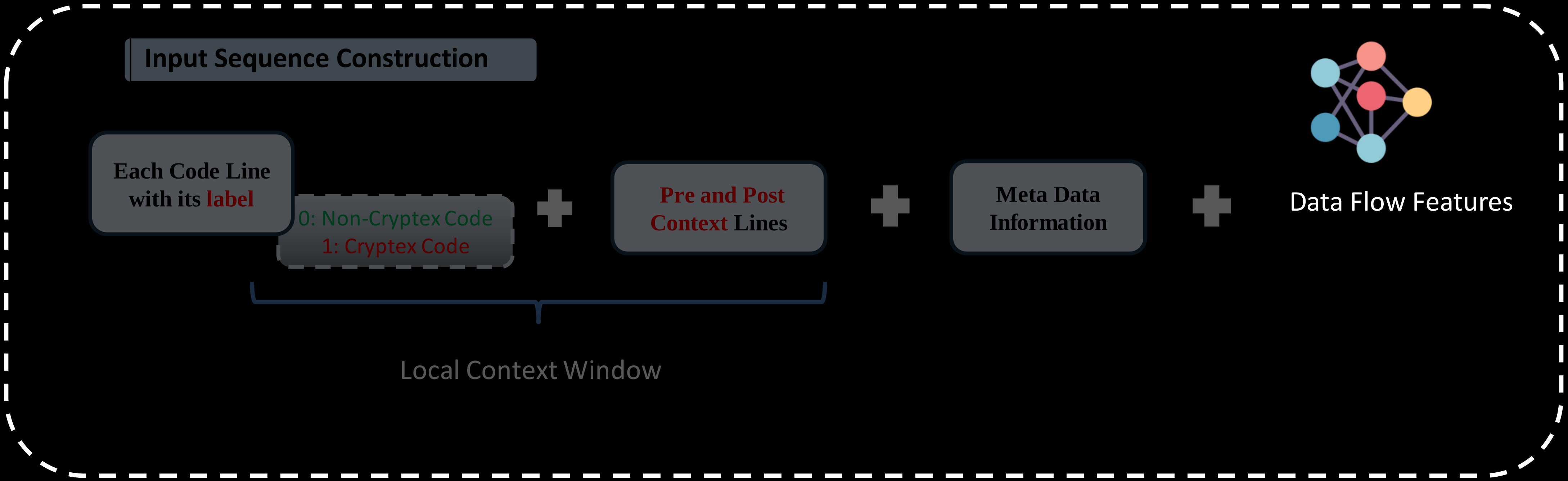
- During training, each code line is guided with its label either 0 or 1
- Each Target Code line is tied with its 4 immediate lines

Input Sequence Construction: Local & Metadata Features


Input Sequence Construction
Each Code Line with its label
0: Non-Cryptex Code
1: Cryptex Code
Pre and Post Context Lines
Meta Data Information
Local Context Window

- During training, each code line is guided with its label either 0 or 1
- Each Target Code line is tied with its 4 immediate lines
- Meta data information helps in resolving to function and file level.

Input Sequence Construction: Global Features



Input Sequence Construction

Each Code Line with its **label**

0: Non-Cryptex Code
1: Cryptex Code

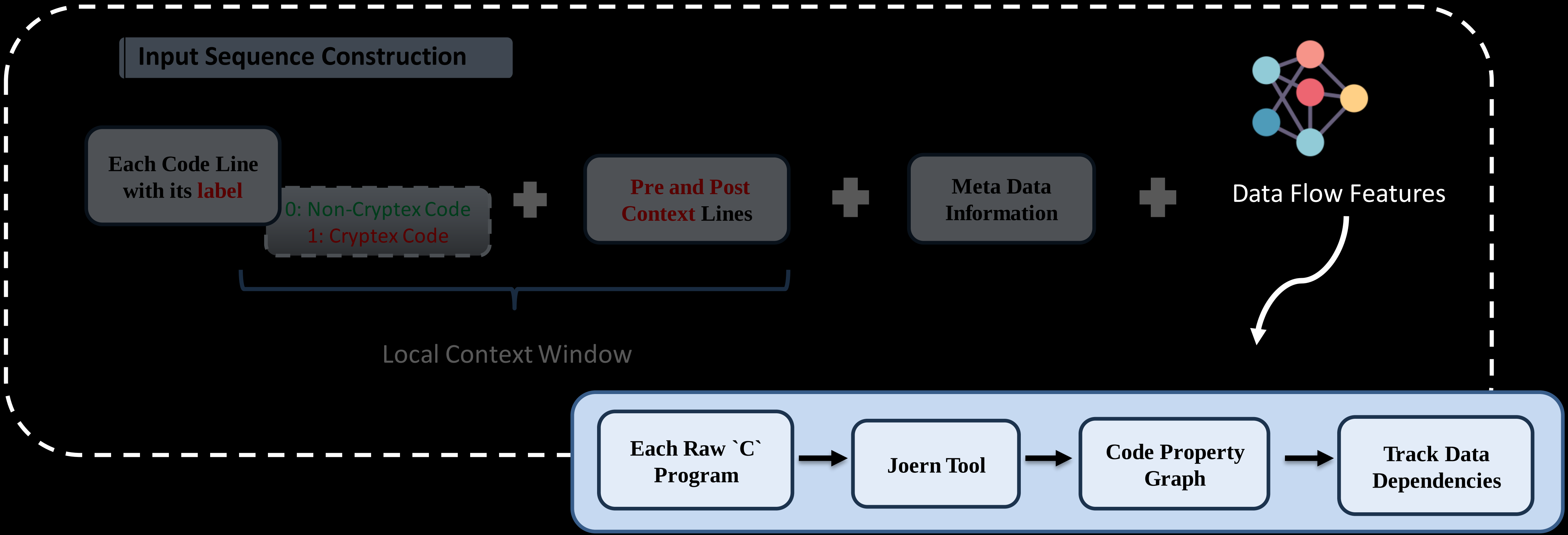
Pre and Post Context Lines

Meta Data Information

Data Flow Features

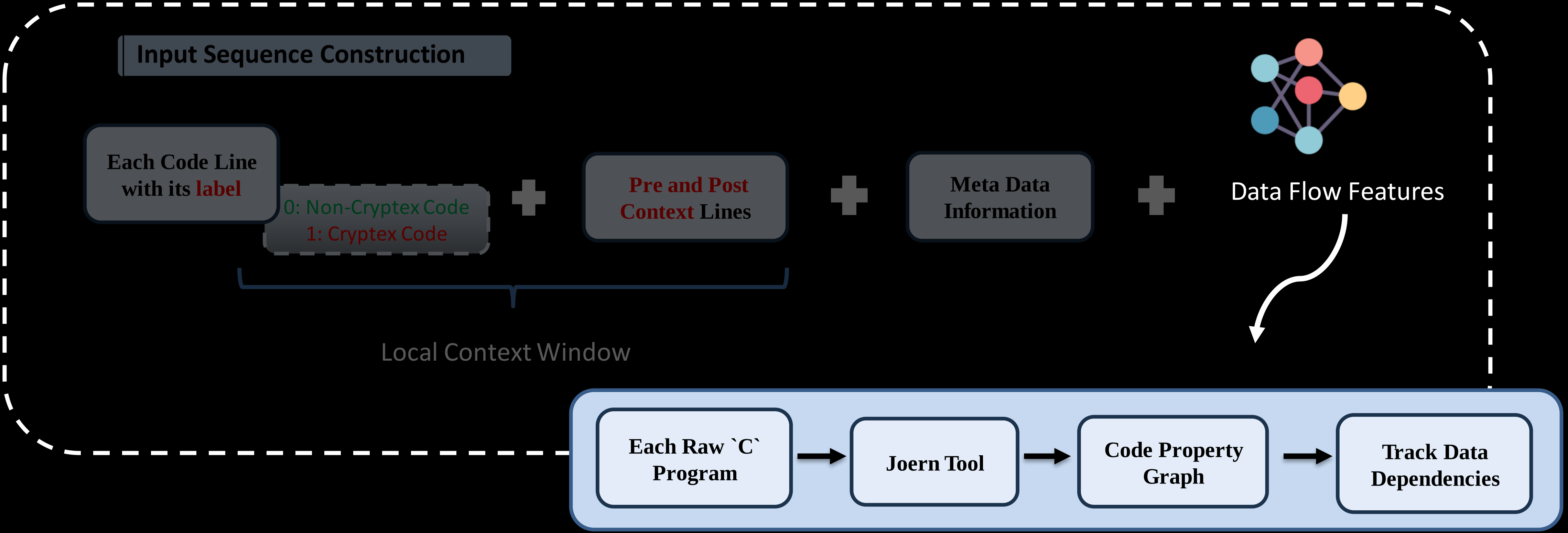
Local Context Window

Input Sequence Construction: Global Features



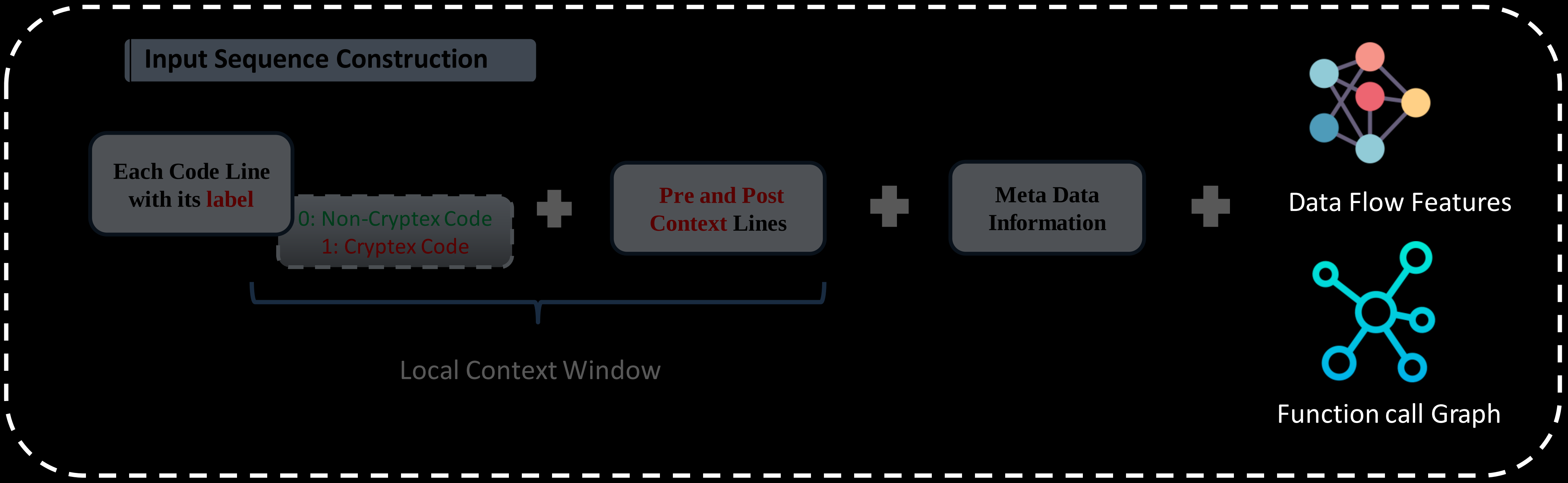
- Pre-Compute and Track all the lines that are semantically reachable to each line

Input Sequence Construction: Global Features



- Pre-Compute and Track all the lines that are semantically reachable to each line
- These Features capture subtle, long-range data flows

Input Sequence Construction: Global Features



Input Sequence Construction

Each Code Line with its label

0: Non-Cryptex Code
1: Cryptex Code

Pre and Post Context Lines

Meta Data Information

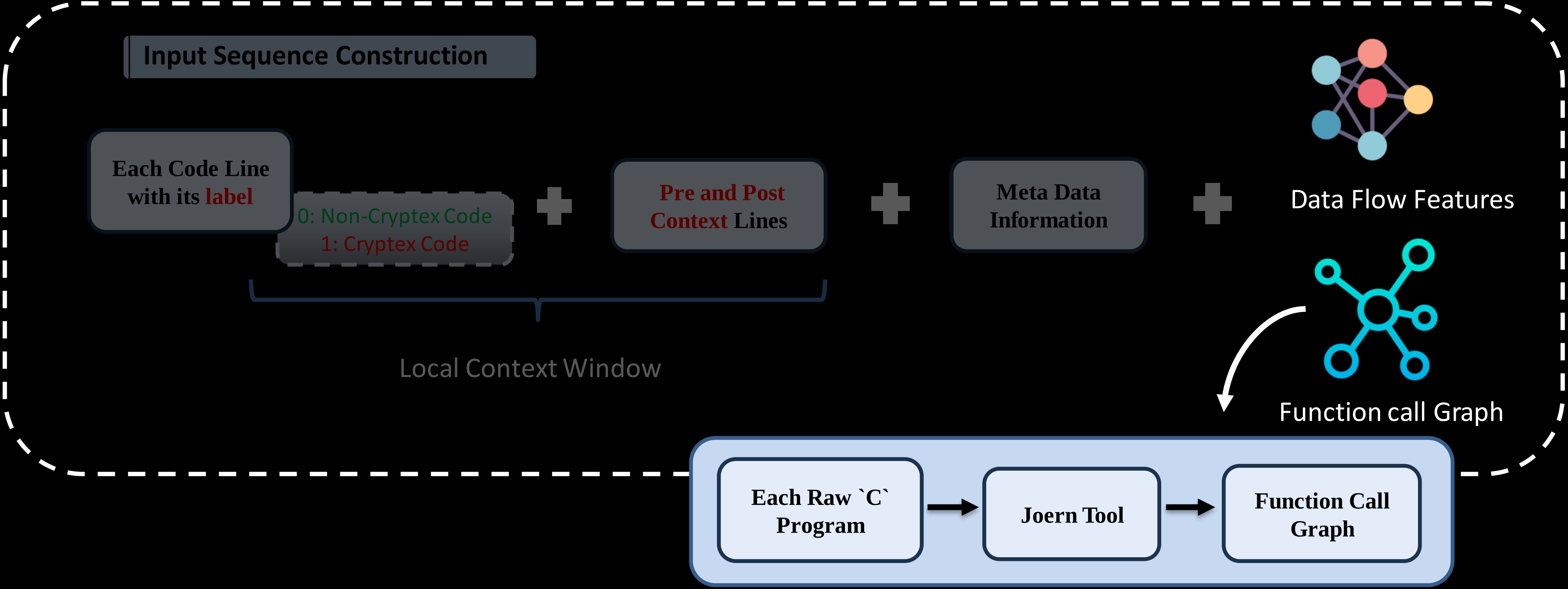
Data Flow Features

Function call Graph

Local Context Window

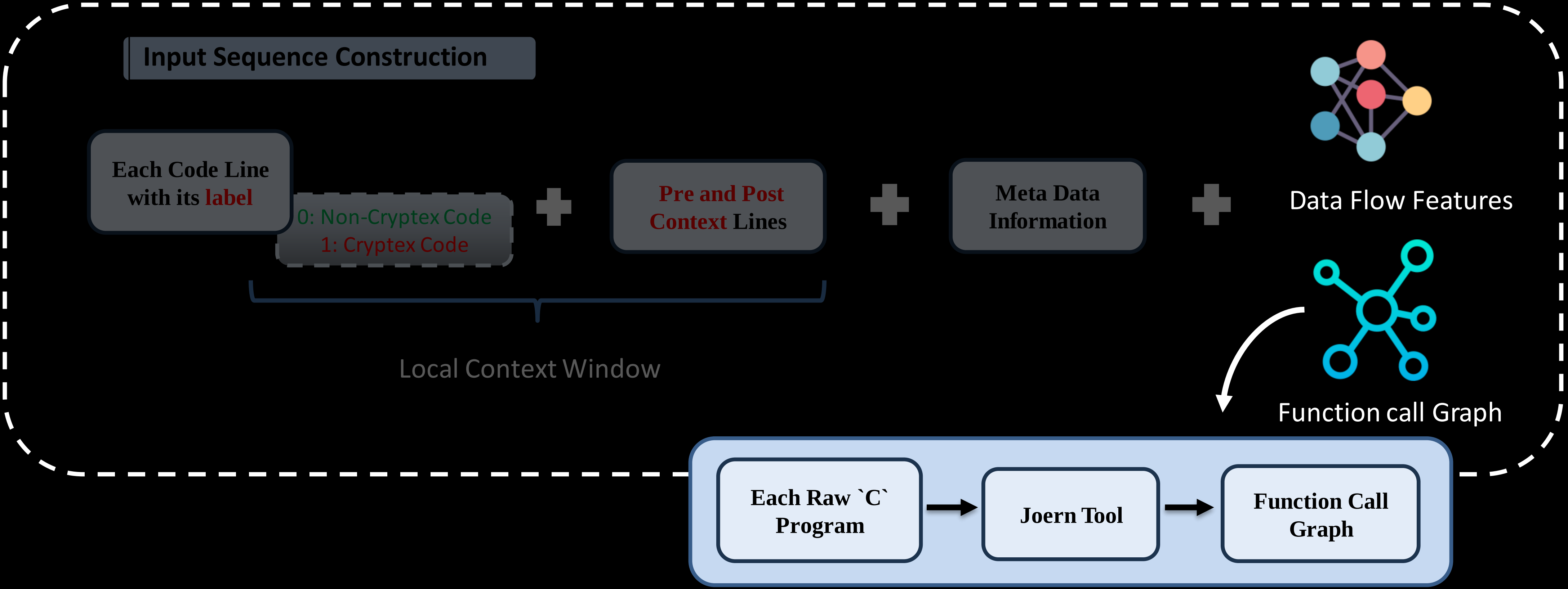
- Pre-Compute and Track all the lines that are semantically reachable to each line
- These Features capture subtle, long-range data flows

Input Sequence Construction: Global Features



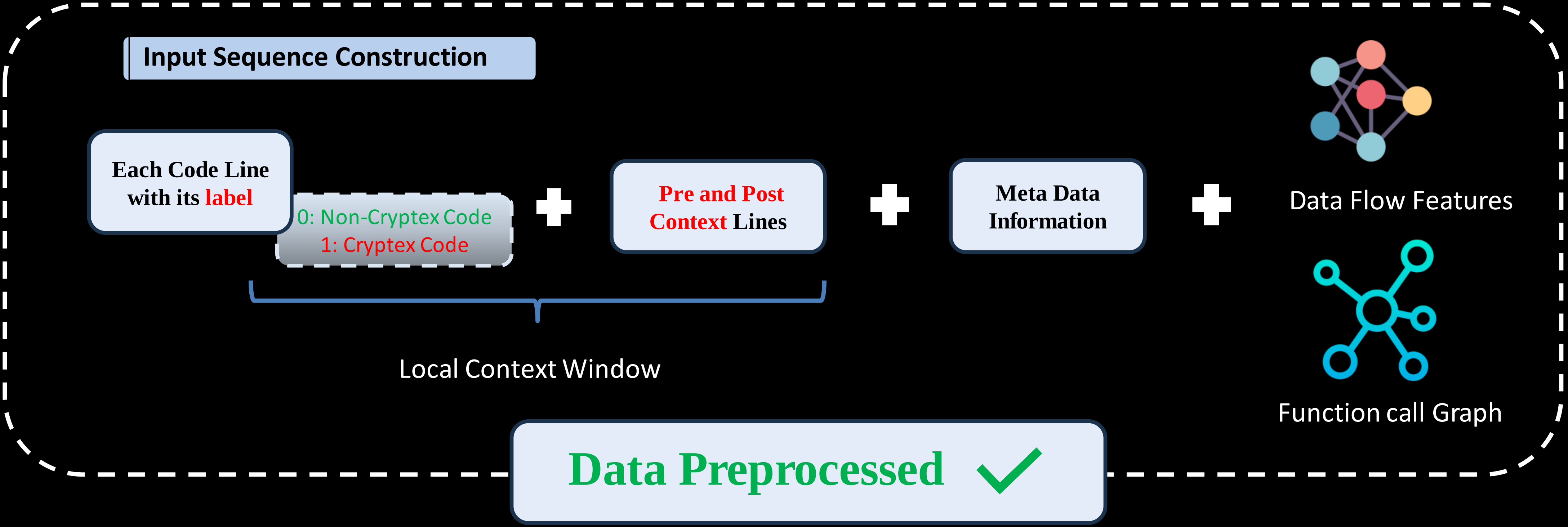
- Pre-Compute and Track all the lines that are semantically reachable to each line
- These Features capture subtle, long-range data flows

Input Sequence Construction: Global Features



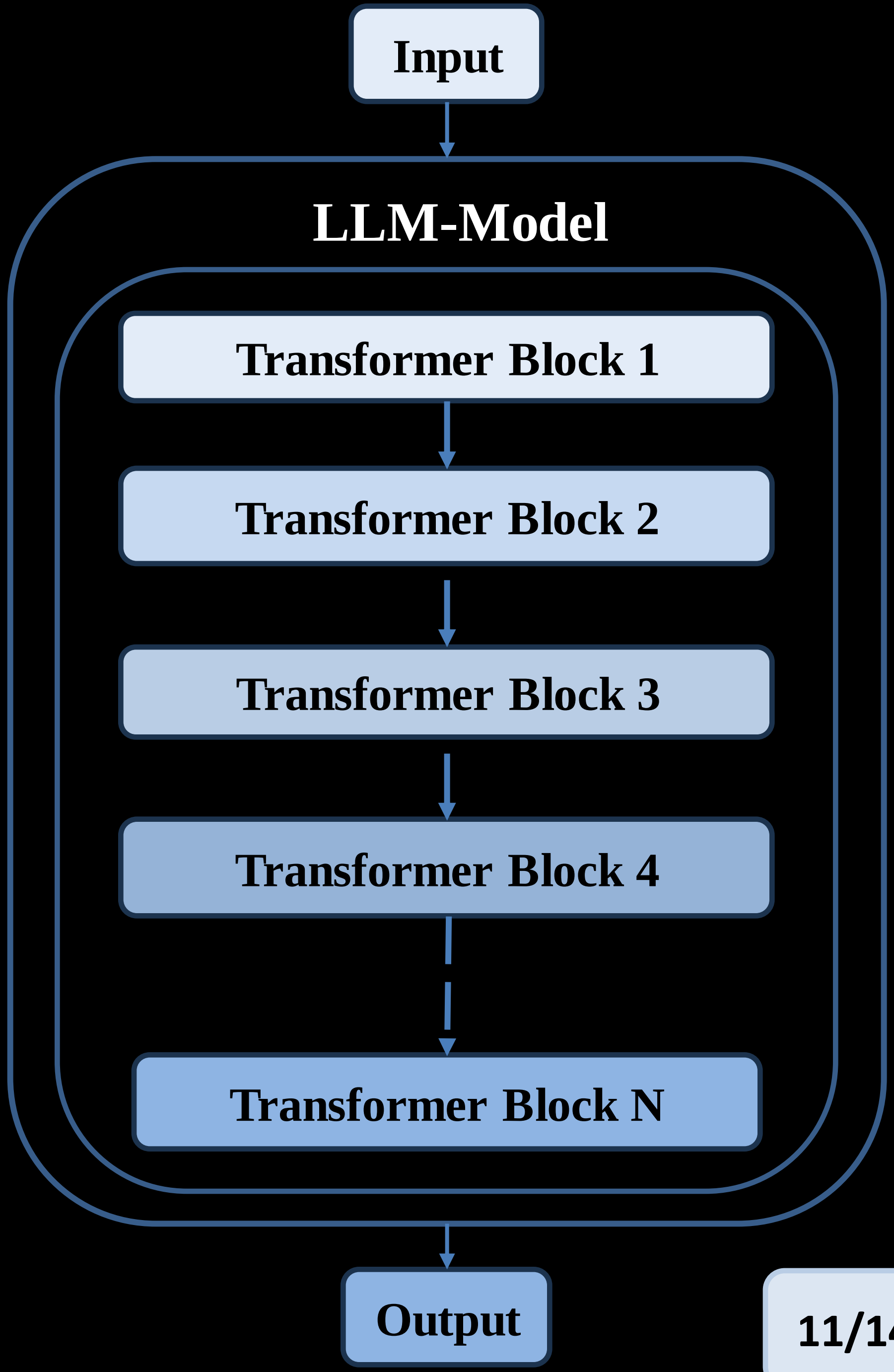
- Pre-Compute and Track all the lines that are semantically reachable to each line
- These Features capture subtle, long-range data flows
- For Each Line, We also identify the API Calls and internal function invocations

Input Sequence Construction: Global Features



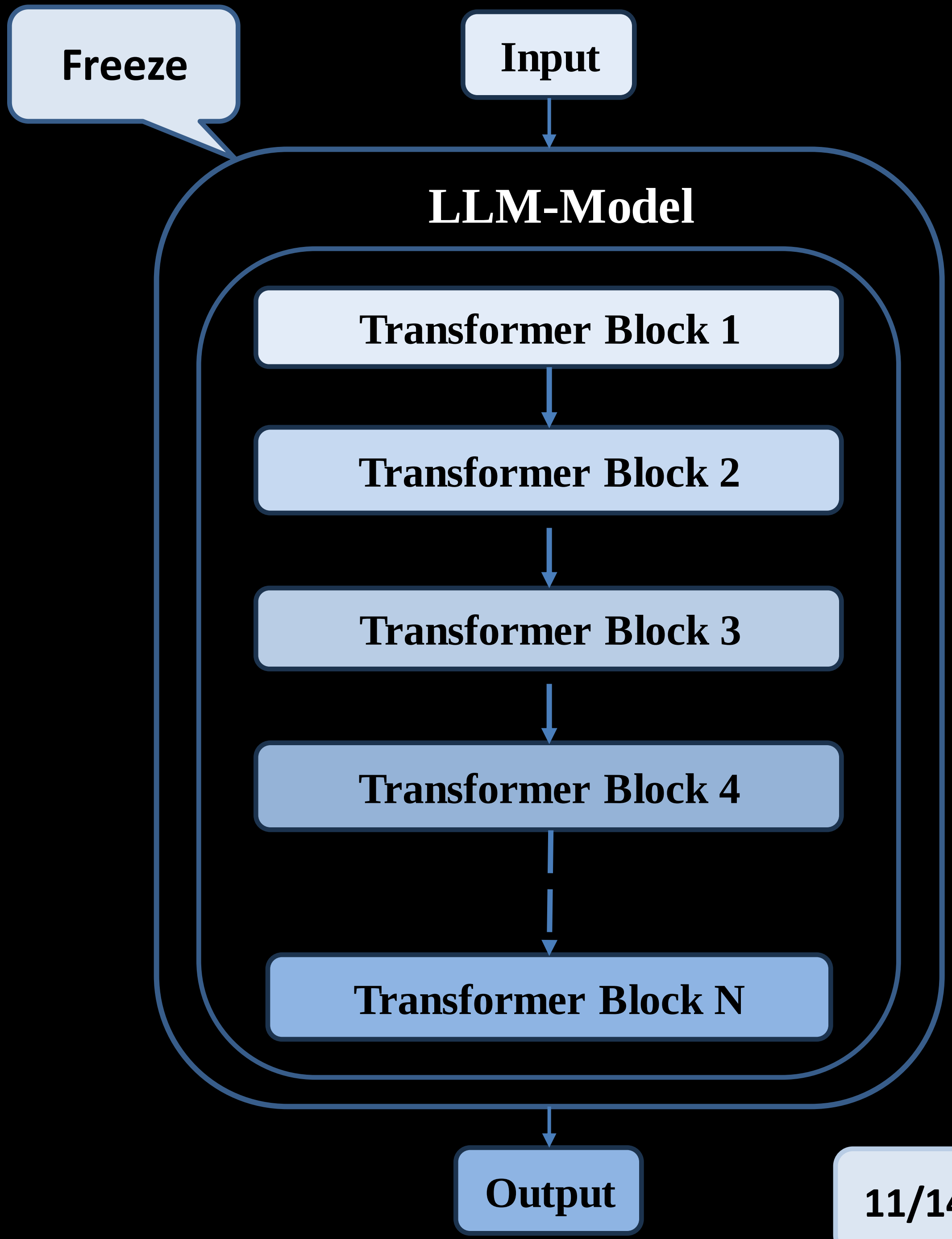
- Pre-Compute and Track all the lines that are semantically reachable to each line
- These Features capture subtle, long-range data flows
- For Each Line, We also identify the API Calls and internal function invocations

LLM-Model & QLoRA Finetuning



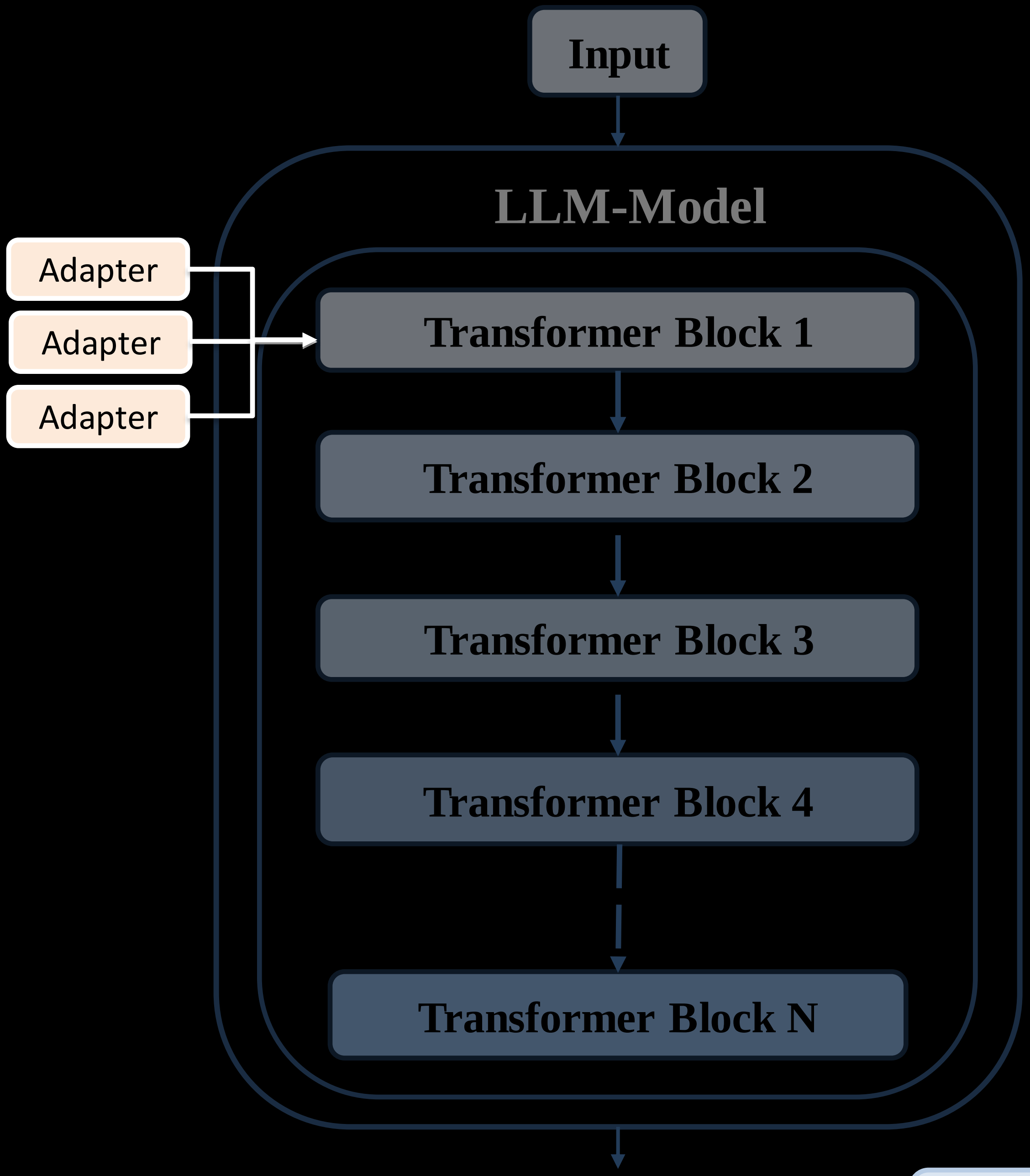
LLM-Model & QLoRA Finetuning

- Freeze the PreTrained LLM



LLM-Model & QLoRA Finetuning

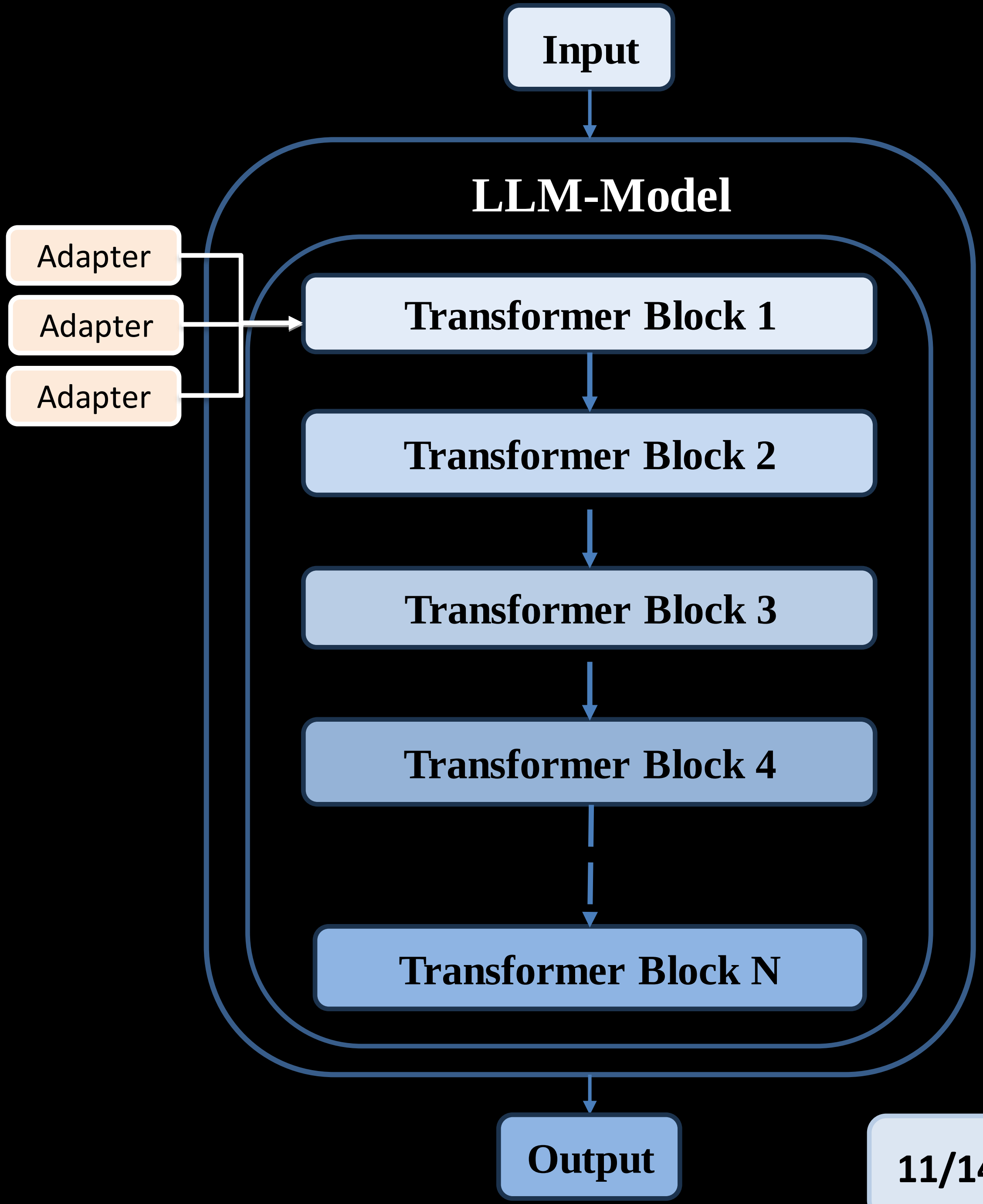
- Freeze the PreTrained LLM
- Add trainable adapters to the LLM architecture



LLM-Model & QLoRA Finetuning

- Freeze the PreTrained LLM
- Add trainable adapters to the LLM architecture

- Adapters are trainable linear layers

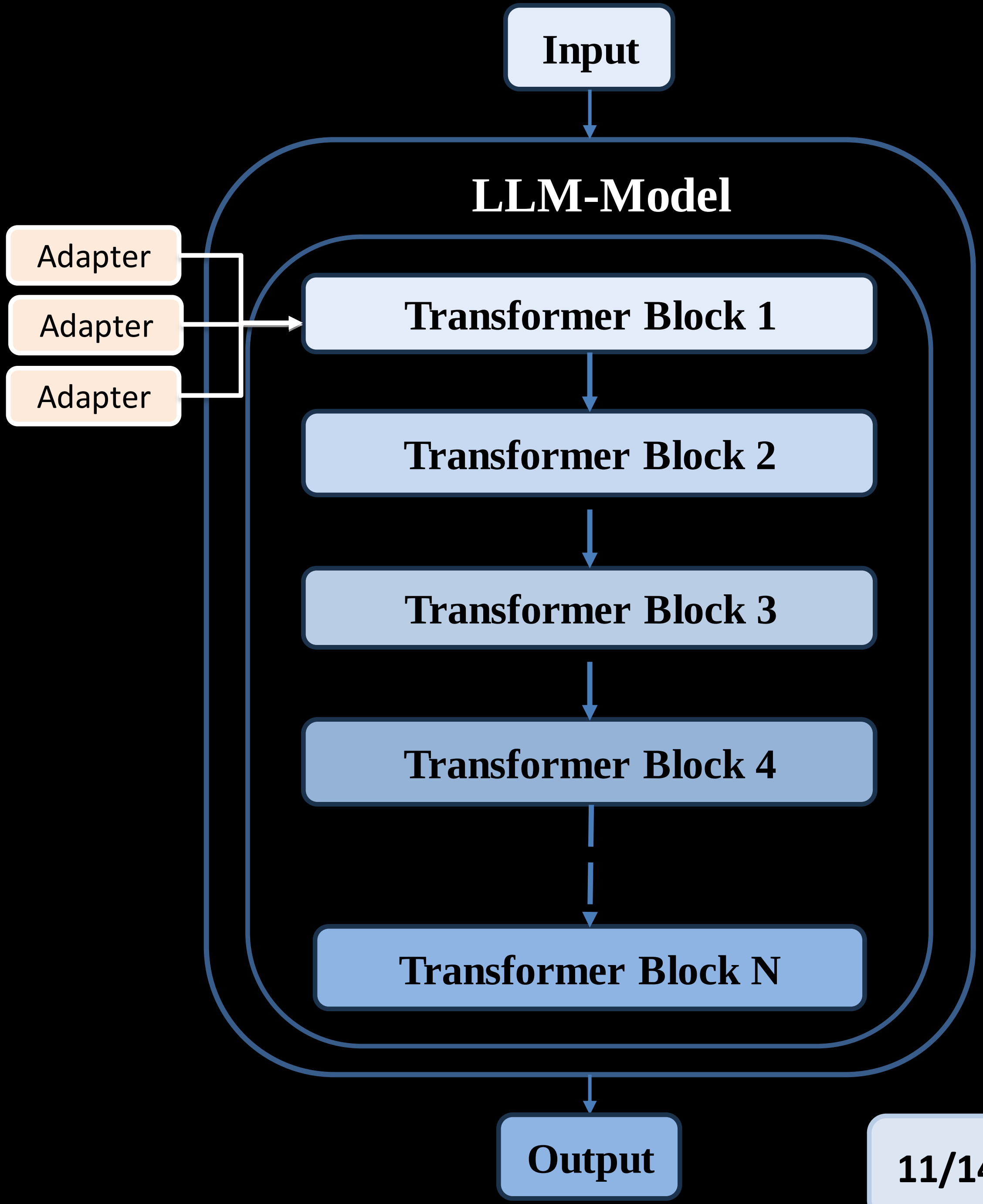


LLM-Model & QLoRA Finetuning

- Freeze the PreTrained LLM
- Add trainable adapters to the LLM architecture

- Adapters are trainable linear layers

- Apply 4-bit Quantization to make the model memory efficient



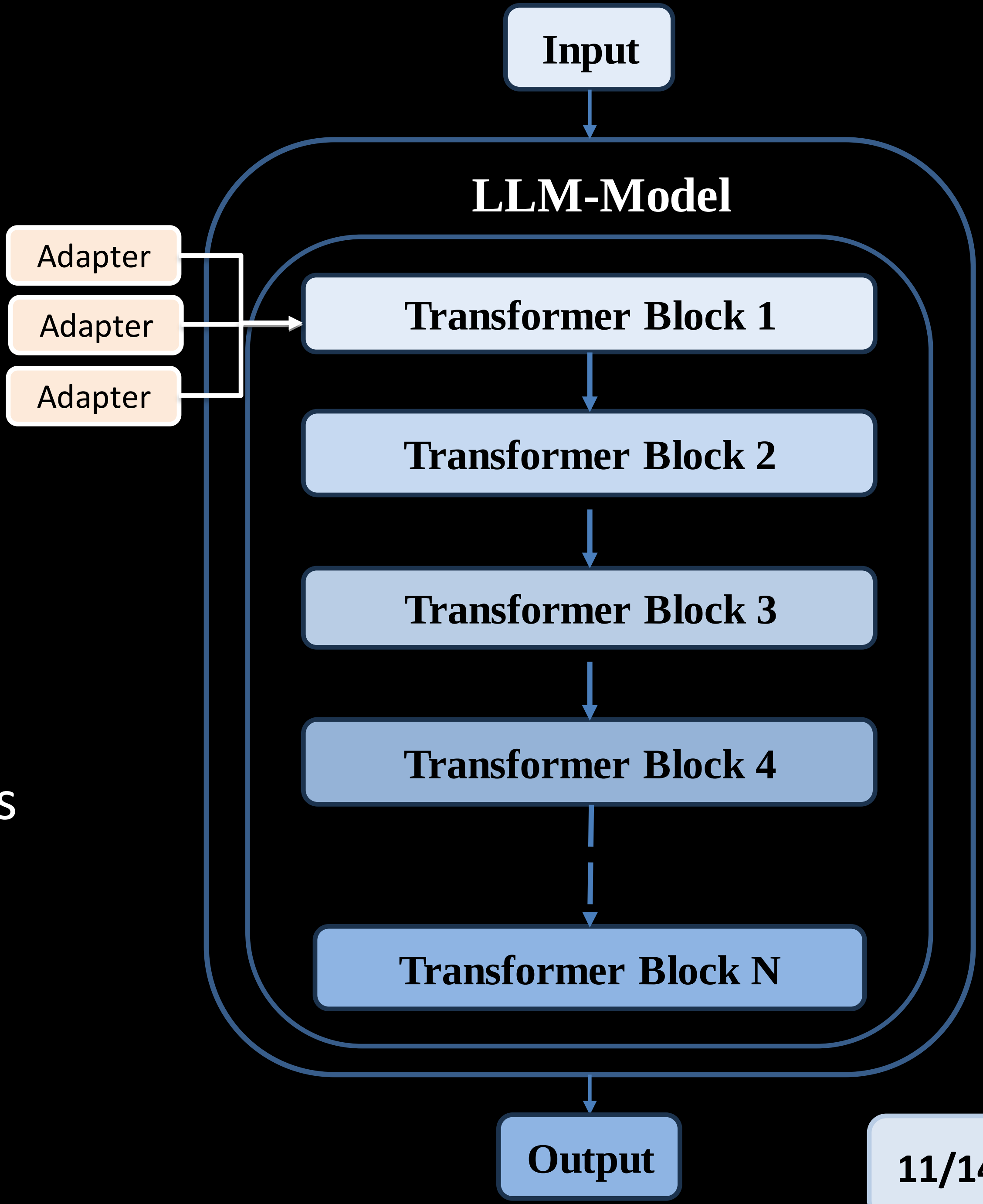
LLM-Model & QLoRA Finetuning

- Freeze the PreTrained LLM
- Add trainable adapters to the LLM Architecture

- Adapters are linear DNN layers

- Apply 4-bit Quantization to make the model memory efficient

- LoRA-Dropout: Drop some weights -> Prevents Over Fitting



LLM-Model & QLoRA Finetuning

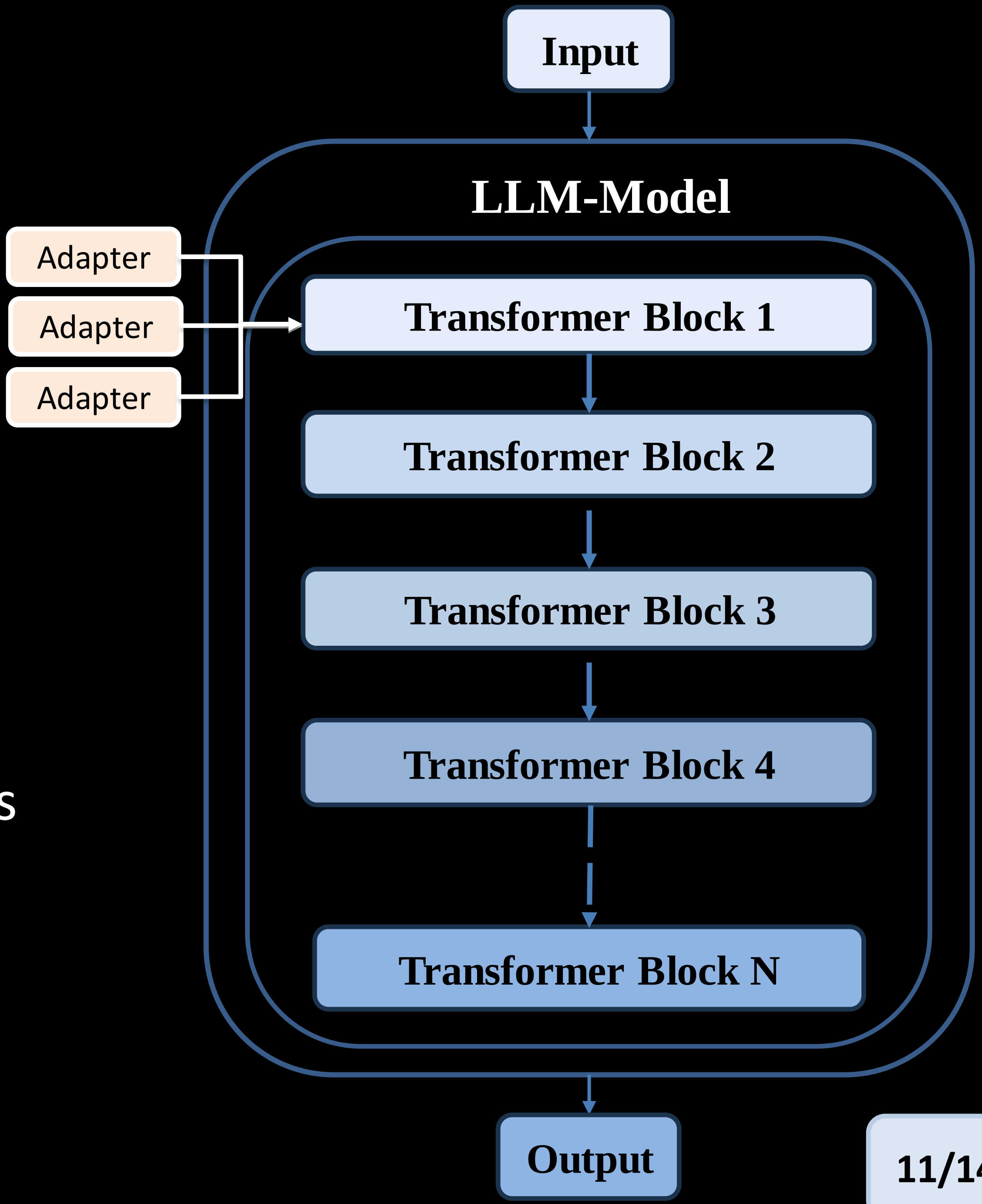
- Freeze the PreTrained LLM
- Add trainable adapters to the LLM Architecture

- Adapters are linear DNN layers

- Apply 4-bit Quantization to make the model memory efficient

- LoRA-Dropout: Drop some weights -> Prevents Over Fitting

- Calculate weights: balances majority and minority classes during finetuning



Contributions

Security Sensitive Code notion and Dataset

LLM-CAL: A novel, scalable and memory-efficient framework

Comprehensive Evaluation & Out-of-distribution case studies

LLM-CAL Evaluation on Test Set

- Models Leveraged: Google Gemma 2B, Code Gemma 2B, Meta Llama 7B

LLM-CAL Evaluation on Test Set

- Models Leveraged: Google Gemma 2B, Code Gemma 2B, Meta Llama 7B
- Fine-Tuning Gemma 2B has produced the best result with better Efficiency

LLM-CAL Evaluation on Test Set

- Models Leveraged: Google Gemma 2B, Code Gemma 2B, Meta Llama 7B
- Fine-Tuning Gemma 2B has produced the best result with better Efficiency

Metric	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
Line-Level	17606	77916	106	452

LLM-CAL Evaluation on Test Set

- Models Leveraged: Google Gemma 2B, Code Gemma 2B, Meta Llama 7B
- Fine-Tuning Gemma 2B has produced the best result with better Efficiency
- Only Few False Positives and False Negatives observed at Line Level

Metric	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
Line-Level	17606	77916	106	452

LLM-CAL Evaluation on Test Set

- Models Leveraged: Google Gemma 2B, Code Gemma 2B, Meta Llama 7B
- Fine-Tuning Gemma 2B has produced the best result with better Efficiency
- Only Few False Positives and False Negatives observed at Line Level

Metric	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
Line-Level	17606	77916	106	452

Accuracy	F1 Score	Recall	Precision
99.04%	98.41%	97.50%	99.40%

LLM-CAL Evaluation on Test Set

- Models Leveraged: Google Gemma 2B, Code Gemma 2B, Meta Llama 7B
- Fine-Tuning Gemma 2B has produced the best result with better Efficiency
- Only Few False Positives and False Negatives observed at Line Level

Metric	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
Line-Level	17606	77916	106	452
Function-Level	684	3755	0	0

LLM-CAL Evaluation on Test Set

- Models Leveraged: Google Gemma 2B, Code Gemma 2B, Meta Llama 7B
- Fine-Tuning Gemma 2B has produced the best result with better Efficiency
- Only Few False Positives and False Negatives observed at Line Level
- LLM-CAL rightly identifies all the cryptex & non-crytex functions

Metric	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
Line-Level	17606	77916	106	452
Function-Level	684	3755	0	0

Case Study : Out- of- Distribution (mbedTLS) Crypto Code

Case Study : Out- of- Distribution (mbedTLS) Crypto Code

- Open-source project[5] that demonstrates AES-GCM Encryption and Decryption

5. T. Leonhardt, "Practical cryptography engineering," https://github.com/tleonhardt/practical cryptography engineering, 2018

Case Study : Out- of- Distribution (mbedTLS) Crypto Code

- Open-source project[5] that demonstrates AES-GCM Encryption and Decryption

Line	Code Block 1 – Decryption	Probability
1	ret = mbedtls_gcm_auth_decrypt (&gcm, plain_len, iv, IV_BYTES, add_data, add_len, tag, TAG_BYTES, output,decrypted) ;	0.6934
2	If (ret != 0) {	0.9334
3	printf ("mbedtls_gcm_auth_decrypt failed to decrypt the ciphertext -tag doesn't match\n");	0.8128
4	goto exit; }	0.5751
5	printf ("decrypted : '%s' (length \%zu) \n", decrypted, strlen ((char *) decrypted)) ;	0.5547

[5] T. Leonhardt, "Practical cryptography engineering," https://github.com/tleonhardt/practical cryptography engineering, 2018

Case Study : Out- of- Distribution (mbedTLS) Crypto Code

- Open-source project[5] that demonstrates AES-GCM Encryption and Decryption

Line	Code Block 1 – Decryption	Probability
1	ret = mbedtls_gcm_auth_decrypt (&gcm, plain_len, iv, IV_BYTES, add_data, add_len, tag, TAG_BYTES, output,decrypted) ;	0.6934
2	If (ret != 0) {	0.9334
3	printf ("mbedtls_gcm_auth_decrypt failed to decrypt the ciphertext -tag doesn't match\n");	0.8128
4	goto exit; }	0.5751
5	printf ("decrypted : '%s' (length \%zu) \n", decrypted, strlen ((char *) decrypted)) ;	0.5547

- LLM-CAL delivers high probability scores to the decryption and error handling lines

[5] T. Leonhardt, "Practical cryptography engineering," https://github.com/tleonhardt/practical cryptography engineering, 2018

Case Study : Out- of- Distribution (mbedTLS) Crypto Code

- Open-source project that demonstrates AES-GCM Encryption and Decryption

Line	Code Block 1 – Decryption	Probability
1	ret = mbedtls_gcm_auth_decrypt (&gcm, plain_len, iv, IV_BYTES, add_data, add_len, tag, TAG_BYTES, output,decrypted) ;	0.6934
2	If (ret != 0) {	0.9334
3	printf ("mbedtls_gcm_auth_decrypt failed to decrypt the ciphertext -tag doesn't match\n");	0.8128
4	goto exit; }	0.5751
5	printf ("decrypted : '%s' (length \%zu) \n", decrypted, strlen ((char *) decrypted)) ;	0.5547

- LLM-CAL delivers high probability scores to the decryption and error handling lines
- LLM-CAL performs accurately on the unseen crypto API calls

5. T. Leonhardt, “Practical cryptography engineering,” https://github.com/tleonhardt/practical cryptography engineering, 2018

Conclusion

Defined Crytex Code Notion and Built a dedicated labeled dataset

Conclusion

- Defined Cryptex Code Notion and Built a dedicated labeled dataset
- LLM-CAL Tool automatically annotates all the Cryptex code with high accuracy

Conclusion

- Defined Cryptex Code Notion and Built a dedicated labeled dataset
- LLM-CAL Tool automatically annotates all the Cryptex code with high accuracy
- LLM-CAL Tool demonstrates lower False Negatives & lowest False Positives

Conclusion

- Defined Cryptex Code Notion and Built a dedicated labeled dataset
- LLM-CAL Tool automatically annotates all the Cryptex code with high accuracy
- LLM-CAL Tool demonstrates lower False Negatives & lowest False Positives
- Helps Developers with highlighting all the sensitive lines as an extension tool

Conclusion

- Defined Crytex Code Notion and Built a dedicated labeled dataset
- LLM-CAL Tool automatically annotates all the Cryptex code with high accuracy
- LLM-CAL Tool demonstrates lower False Negatives & lowest False Positives
- Helps Developers with highlighting all the sensitive lines as a extension tool
- LLM-CAL Tool demonstrates high generalizability and adaptability to unseen code

Q&A