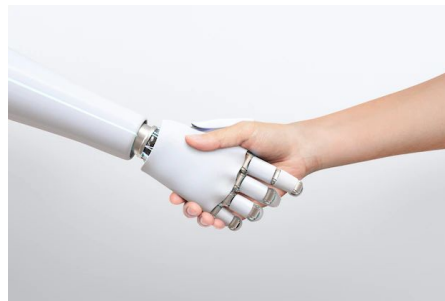
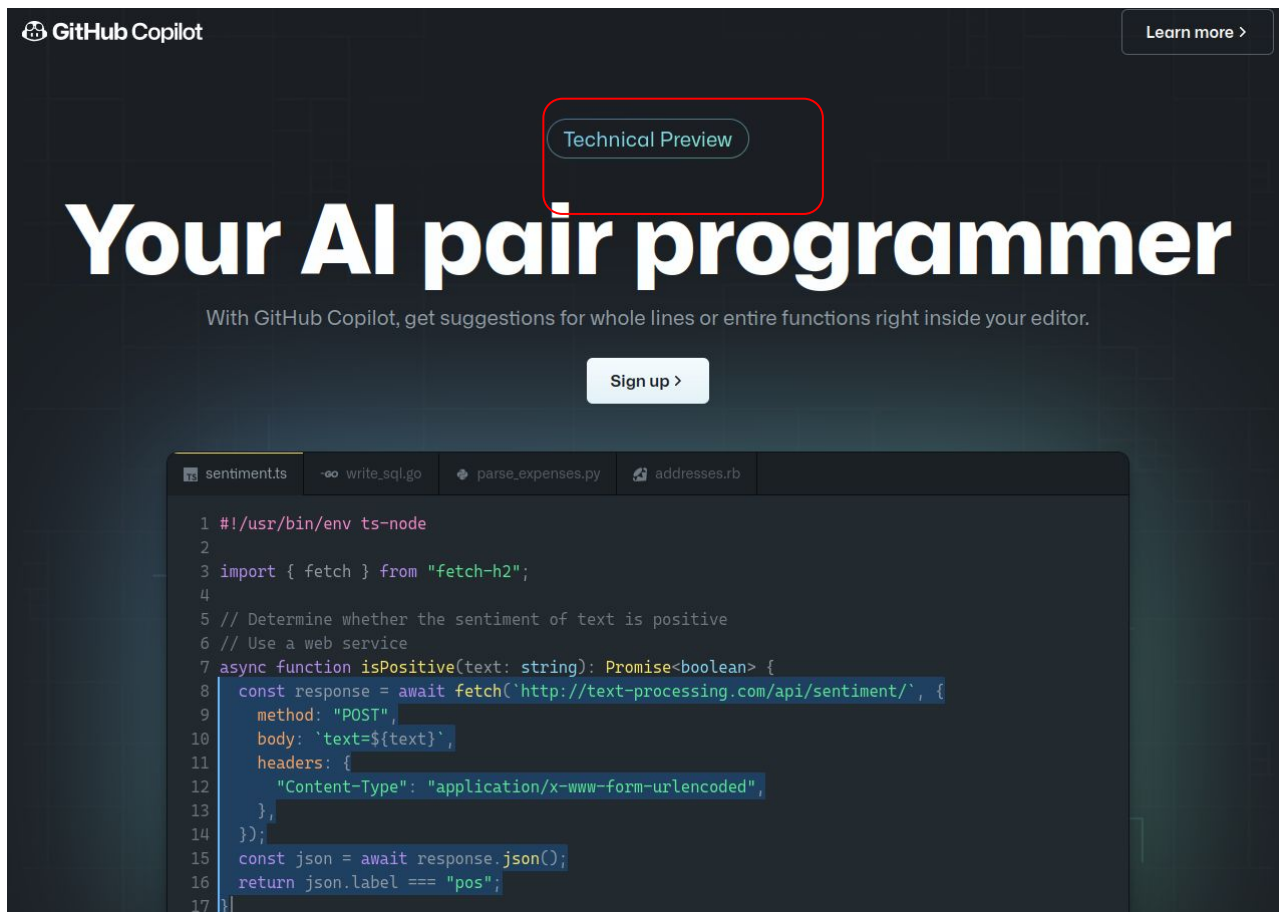


# Lost at C: A User Study on the Security Implications of LLM Code Assistants

**Gustavo Sandoval**, Hammond Pearce, Teo Nys, Ramesh Karri,  
Siddharth Garg and Brendan Dolan-Gavitt  
2023



# June 29, 2021: Future of software development?



The screenshot shows the GitHub Copilot website landing page. At the top left is the GitHub Copilot logo. At the top right is a 'Learn more >' button. In the center, there is a 'Technical Preview' button highlighted with a red rounded rectangle. Below this is the main heading 'Your AI pair programmer' in large white font. Underneath the heading is the text 'With GitHub Copilot, get suggestions for whole lines or entire functions right inside your editor.' and a 'Sign up >' button. At the bottom, there is a code editor window showing a TypeScript file named 'sentiment.ts'. The code in the editor is as follows:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch('http://text-processing.com/api/sentiment/', {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

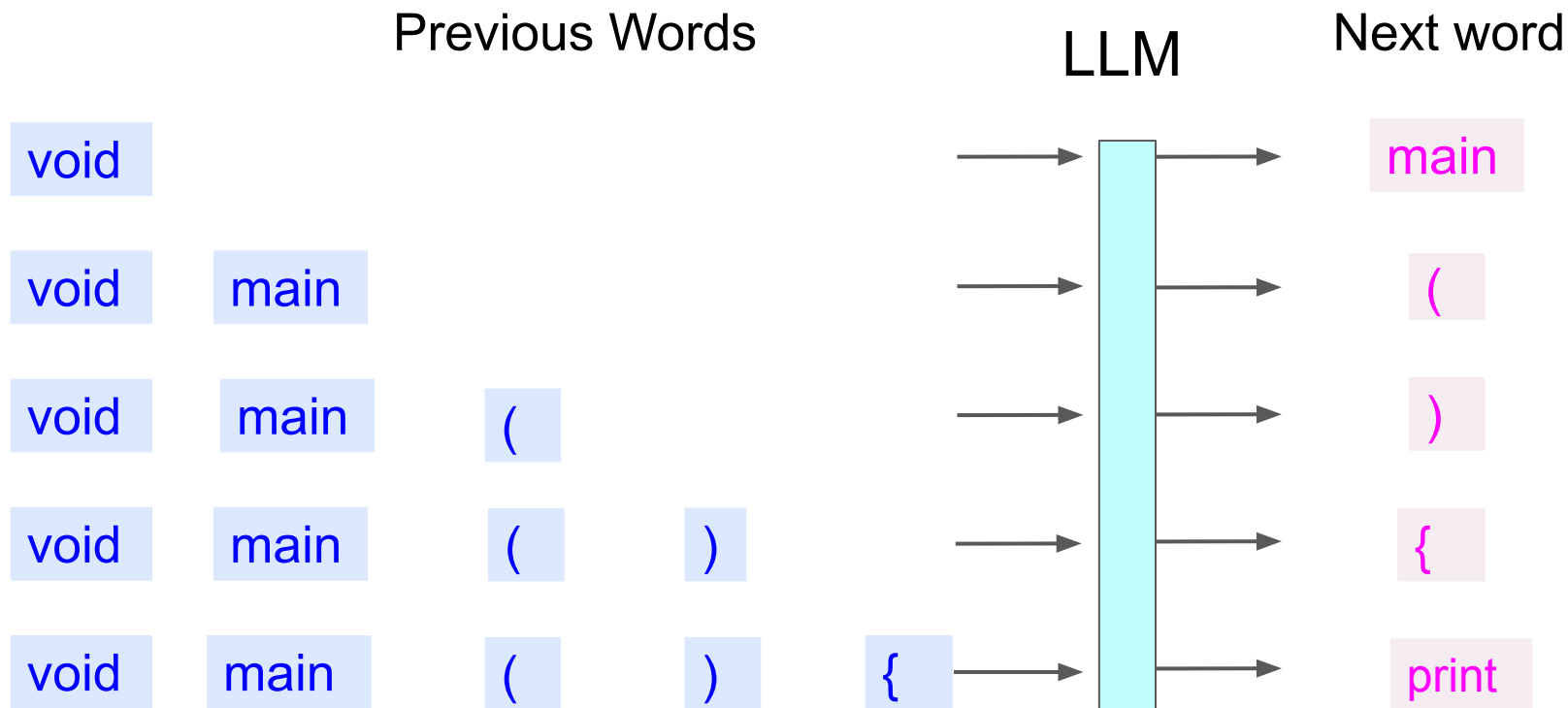


# What are large language models?



LLMs predict the next word given any sequence of words

# What are large language models?



LLMs predict the next word given any sequence of words

## Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions

Hammond Pearce  
Department of ECE  
New York University  
Brooklyn, NY, USA  
hammond.pearce@nyu.edu

Bugs in 40% of security-related completions

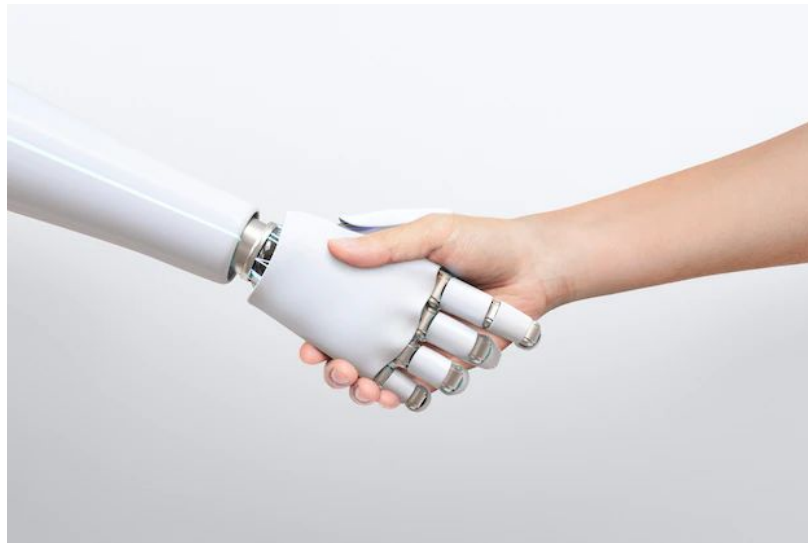
Ramesh Karri  
Department of ECE  
New York University  
Brooklyn, NY, USA  
rkarri@nyu.edu

→ Does this impact code written *by users*?

# First security user study



**Human  
(Control)**

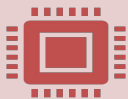


**Human + AI  
(Assisted)**

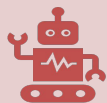
# Research Questions



Does an AI code assistant help novice users write better functional code?



Does code written with AI assistance have an acceptable incidence rate of security bugs?

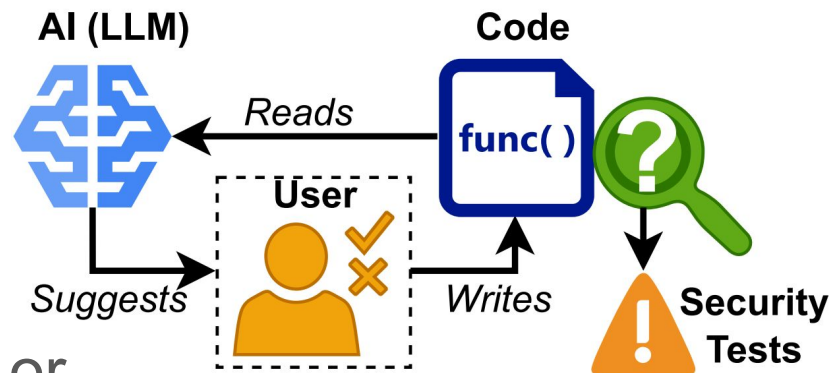


How do AI-assisted users interact with potentially vulnerable code suggestions?



# Methodology for user study

- N = 58, representative sample of CS students
- Participants:
  - Complete a coding task with or without AI assistance
  - 11 functions in **C code** for a shopping list application
- Authors analyze completed code for **functional** and **security** correctness



# Suggestions

```
// print a single list item to an externally allocated string
// This should be in the format of:
// "quantity * item_name @ $price ea", where item_name is a string and
// price is a float formatted with 2 decimal places.
```

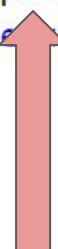
```
int list_item_to_string(node *head, char *str) {
    str = head->quantity;
    return EXIT_SUCCESS;
}
```

*d->price);*

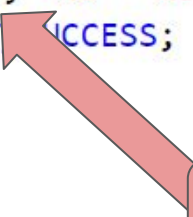
# Completed Code

```
// print a single list item to an externally allocated string
// This should be in the format of:
// "quantity * item_name @ $price ea", where item_name is a string and
// price is a float formatted with 2 decimal places.
int list_item_to_string(node *head, char *str) {
    if (head == NULL) {
        return EXIT_FAILURE;
    }
    sprintf(str, "%d * %s @ $%.2f ea", head->quantity, head->item_name, head->price);
    return EXIT_SUCCESS;
}
```


CWE-787  
Out of bounds write



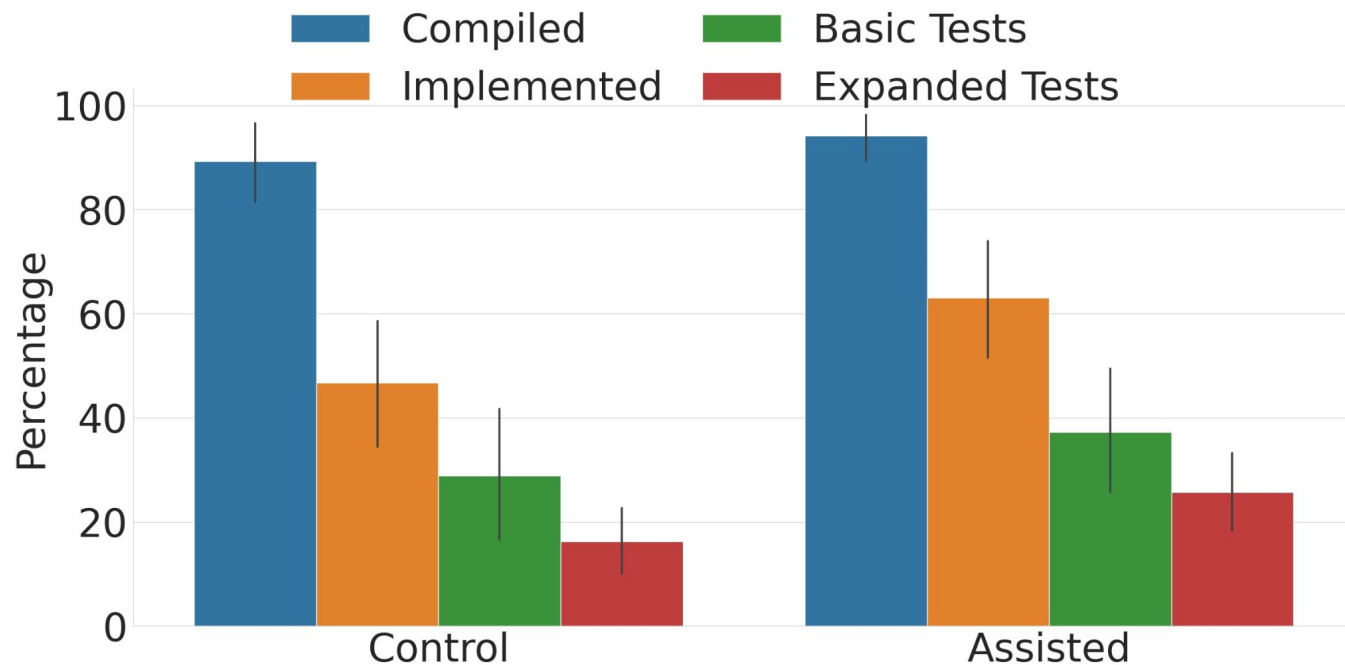
CWE-476  
NULL Ptr Deref



CWE-476  
NULL Ptr Deref

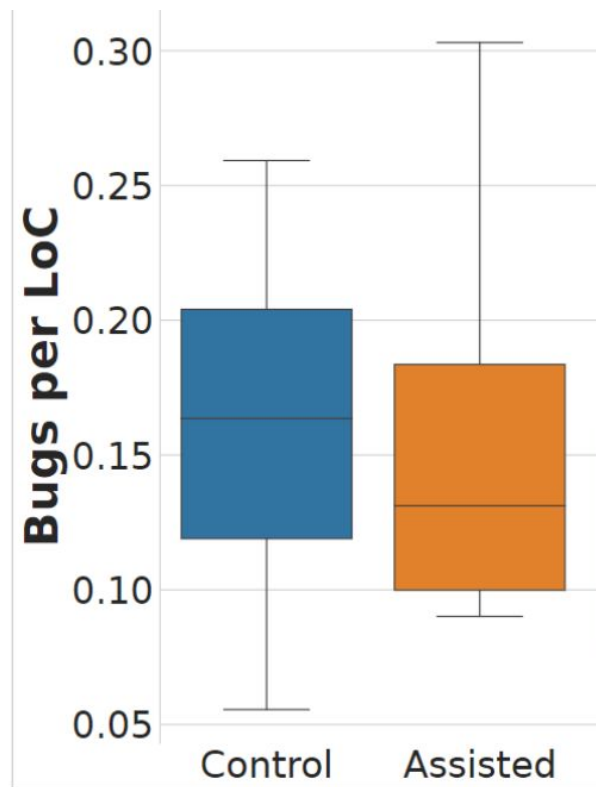
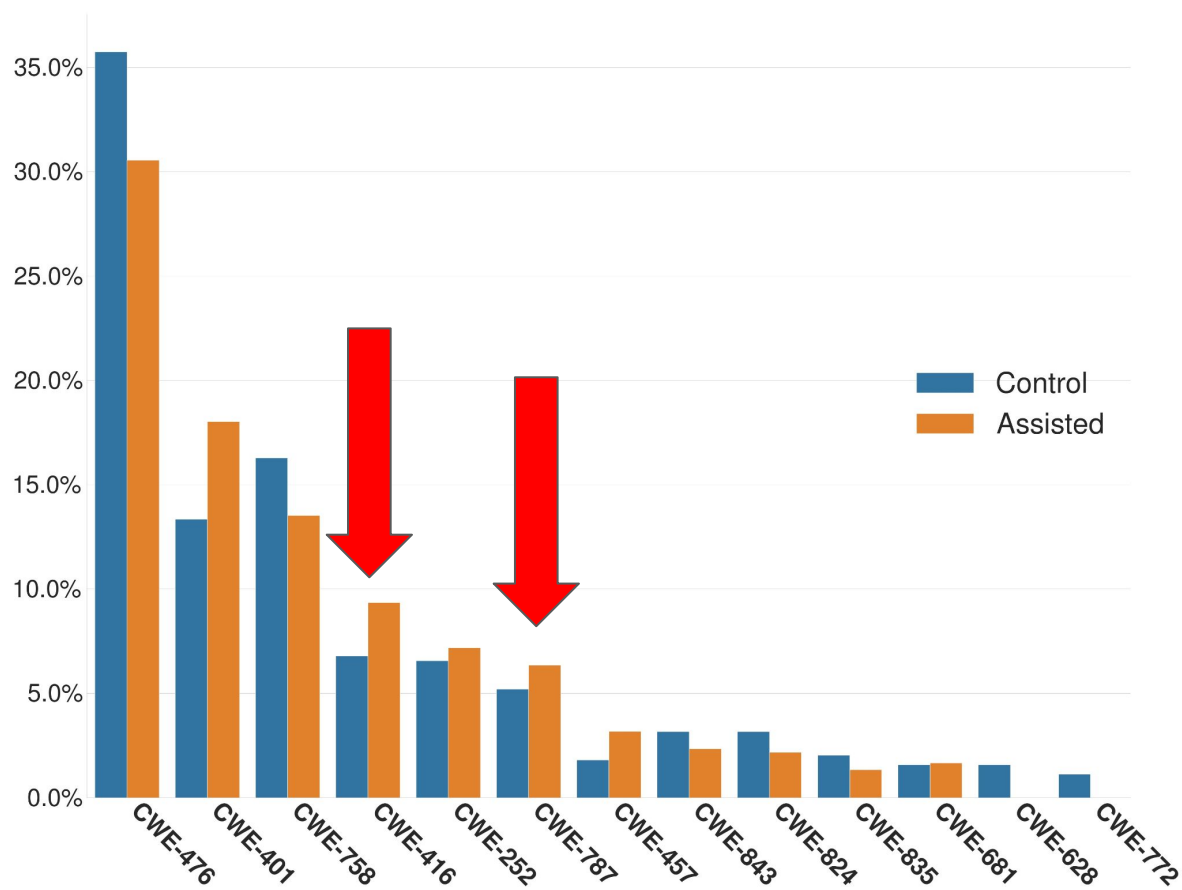


# Results: Functional



→ Assisted were 6-10% more productive

# Results: Security



# Origin of Bugs

```
// This should be in the format of:
// "quantity * item_name @ $price ea", where item_name is a string and
// price is a float formatted with 2 decimal places.
int list_item_to_string(node *head, char *str) {
    // TODO: Implement this function,
    // return EXIT_SUCCESS or EXIT_FAILURE when appropriate
    if (head == NULL) {
        return EXIT_FAILURE;
    }
    sprintf(str, "%d * %s @ $%.2f ea", head->quantity, head->item_name, head->price);
    return EXIT_SUCCESS;
}
```

```
if (head == NULL) {
    return EXIT_FAILURE;
}
sprintf(str, "%d * %s @ $%.2f ea", head->quantity, head->item_name, head->price);
return EXIT_SUCCESS
```

[Suggestion 2022-05-02 17:20:50] (exact match)

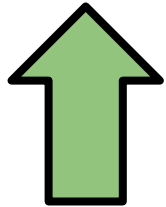
```
// print a single list item to an externally allocated string
// This should be in the format of:
// "quantity * item_name @ $price ea", where item_name is a string and
// price is a float formatted with 2 decimal places.
int list_item_to_string(node *head, char *str) {
    // if head is not NULL, set str to contents of the node
    if (head) {
        sprintf(str, "%d * %s @ $%.2f ea", head->quantity, head->item_name, head->price);
        return EXIT_SUCCESS;
    } else {
        return EXIT_FAILURE;
    }
}
```

```
while(head) {
    // print the item
    printf("%d: %d * %s @ $%.2f ea\n", head->position, head->quantity,
head->item_name, head->price);
    // advance to the next
```

[Suggestion 2022-04-07 20:13:42] (distance: 0.49)

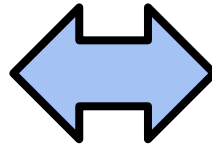
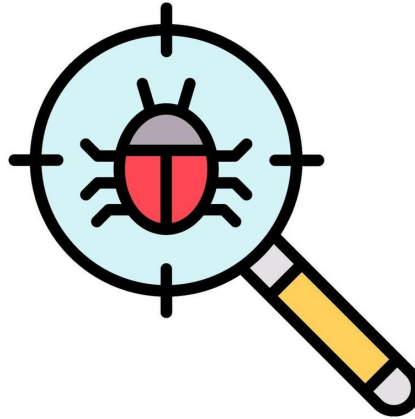
# Results Overview

## Functionality

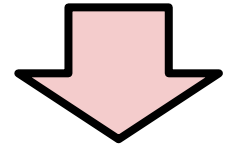


**6%-10%**

## Security Bugs



**Users Blindly accept  
vulnerable suggestions**



**Conclusion:** Large language model code assistants **improve functional correctness** and **do not increase the incidence of severe security bugs for low level C code**

Gustavo  
Sandoval  
@gussand

## Future work:

- In addition to the results here want to continue work.
- We created tooling that assists future user studies. Excited to **collaborate**. **Interested?**

<https://zenodo.org/record/7187359>





## **Anubis:** Large scale cloud workspace environments

- One click in the browser for setup IDE
- The original extension was difficult to install



## **Anubis:** Other considerations

- How do we ensure the latest extension is installed?
- How do we setup base files for participants?





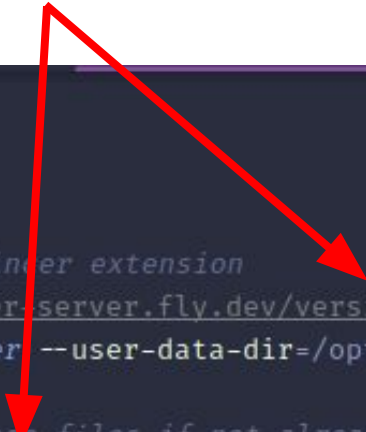
## Run before IDE start

```
#!/bin/bash

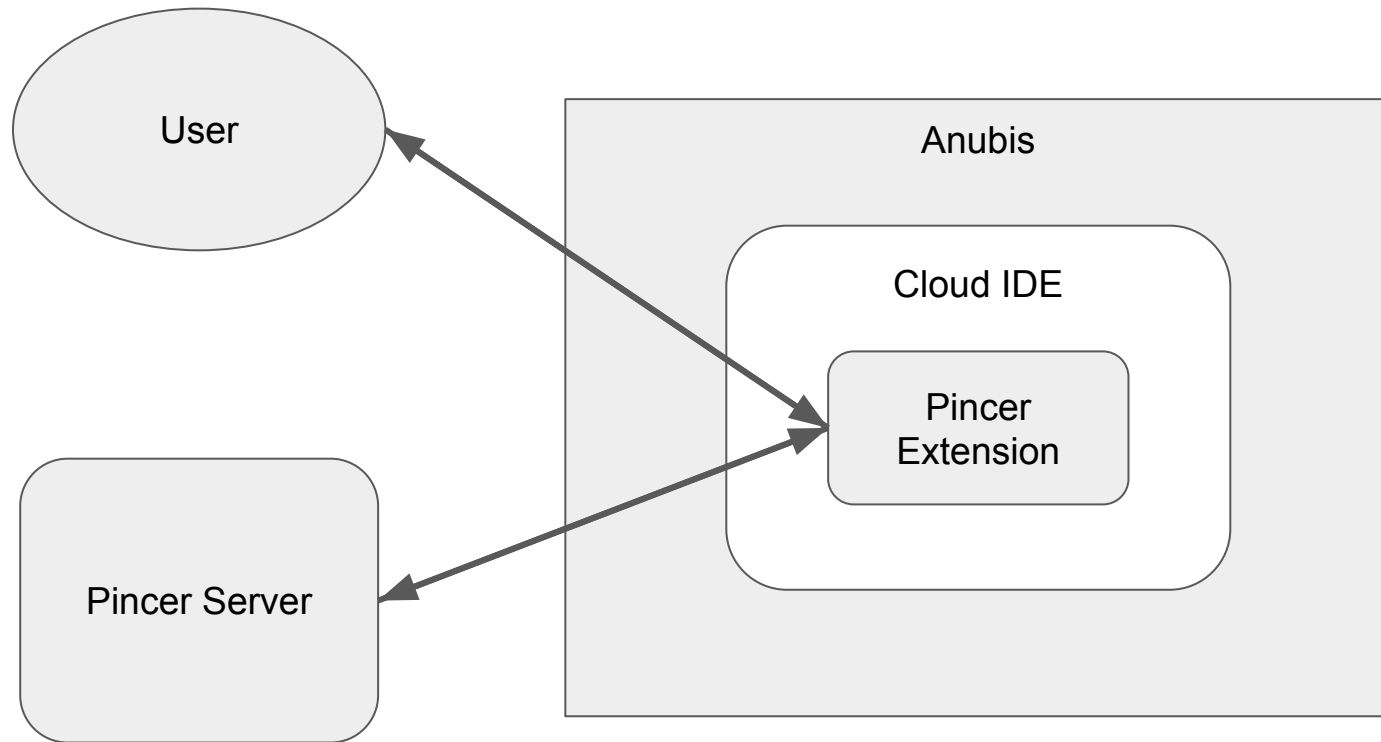
set -xe

# Install latest pincer extension
curl https://pincer-server.fly.dev/version/latest/download -o /opt/code-server/pincer.vsix
/usr/bin/code-server --user-data-dir=/opt/code-server --install-extension /opt/code-server/pincer.vsix

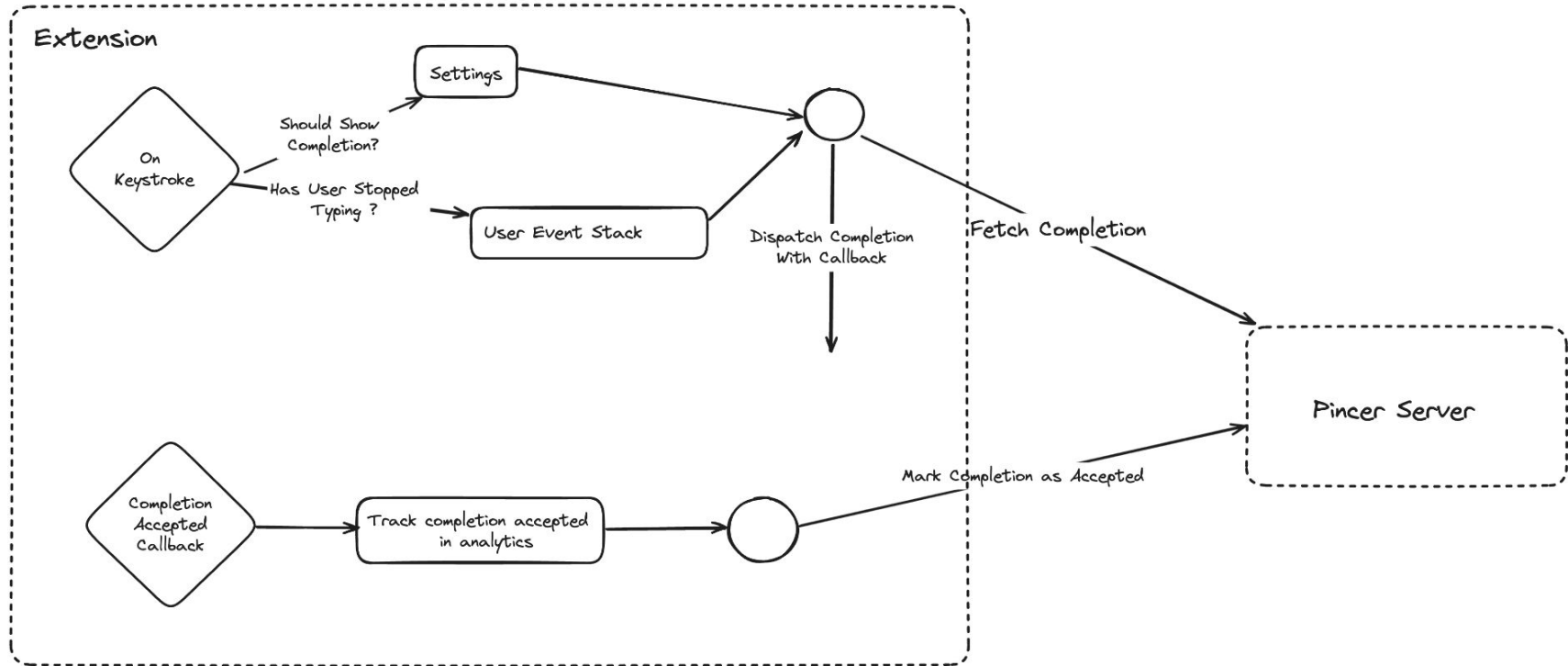
# Download study base files if not already there
if [ ! -d /home/anubis/2023_study ]; then
  git clone https://github.com/GusSand/2023_study.git
  💡
```



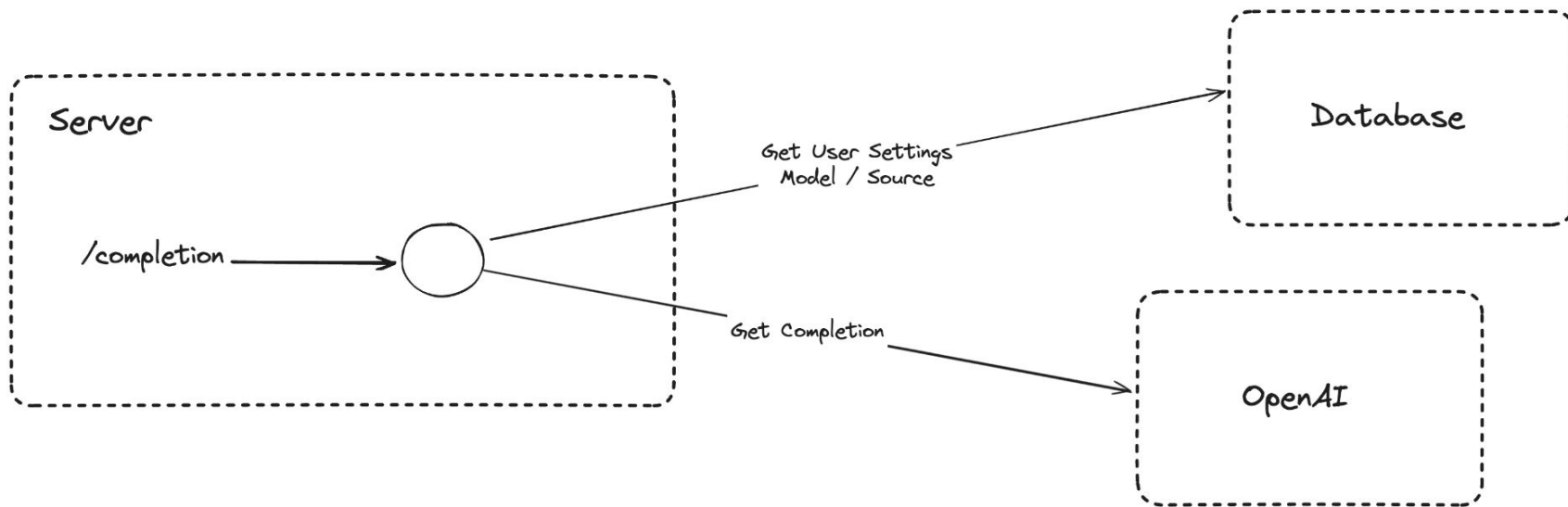
# Infrastructure



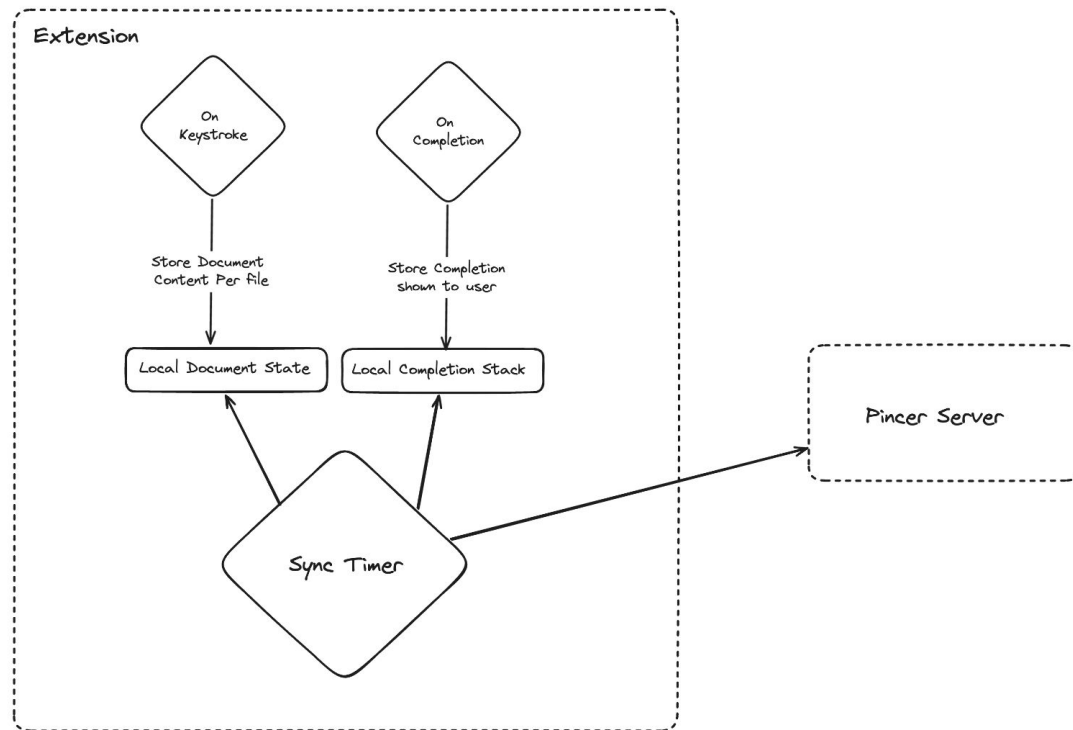
# Completions



# Completion pt2



# Telemetry



**I DON'T ALWAYS TEST MY  
CODE**

**BUT WHEN I DO, I  
DO IT IN  
PRODUCTION**

quickmeme.com