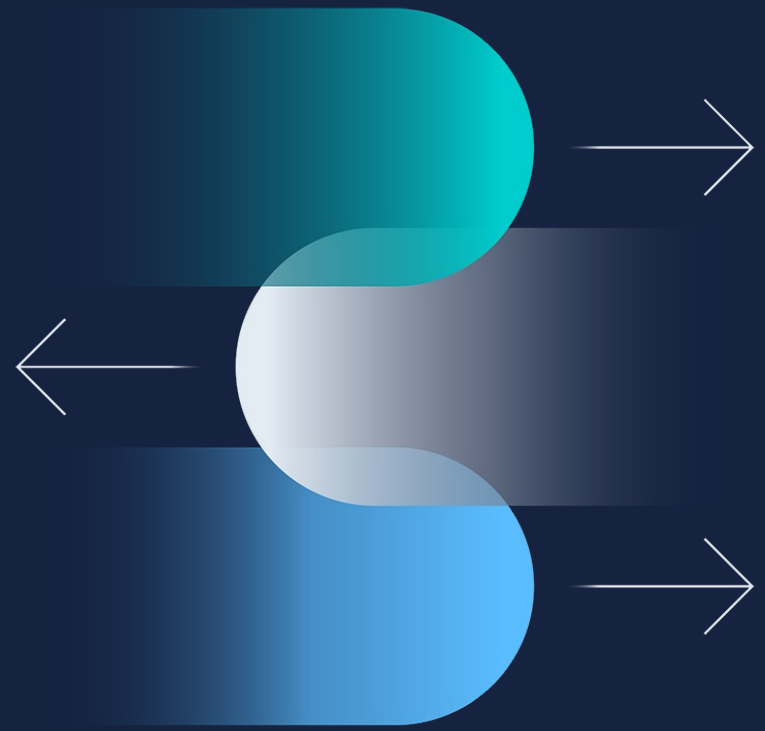
FORM3

Threat Modelling using LLMs

Marcin Niemiec



About Me

Marcin Niemiec

- Cloud Security Engineer at Form3
- Blog: xvnpw.github.io
- X: @xvnpw

About Form3

Real-time payments processing platform

- One single integration to multiple payment schemes
- Multi-cloud platform (AWS, GCP, Azure)
- Go, IaC (Terraform)
- Fully remote
- Pair programming

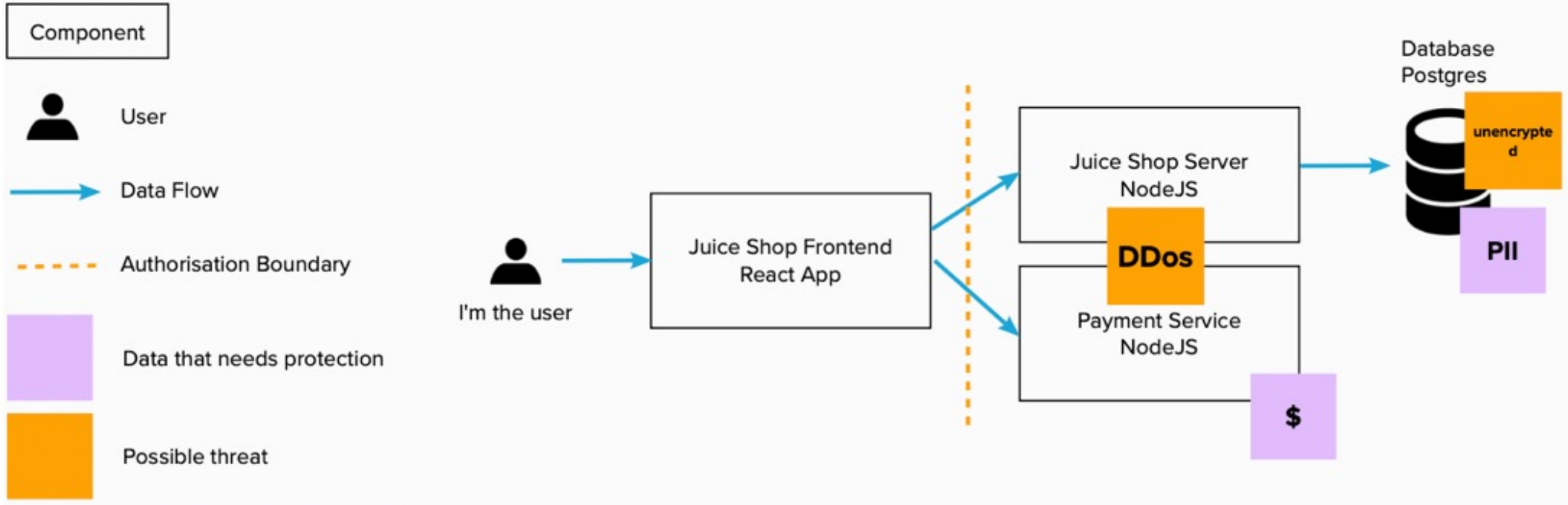
Do you struggle with Threat Modelling?

AI makes Threat Modelling easy!

Threat Modelling 101

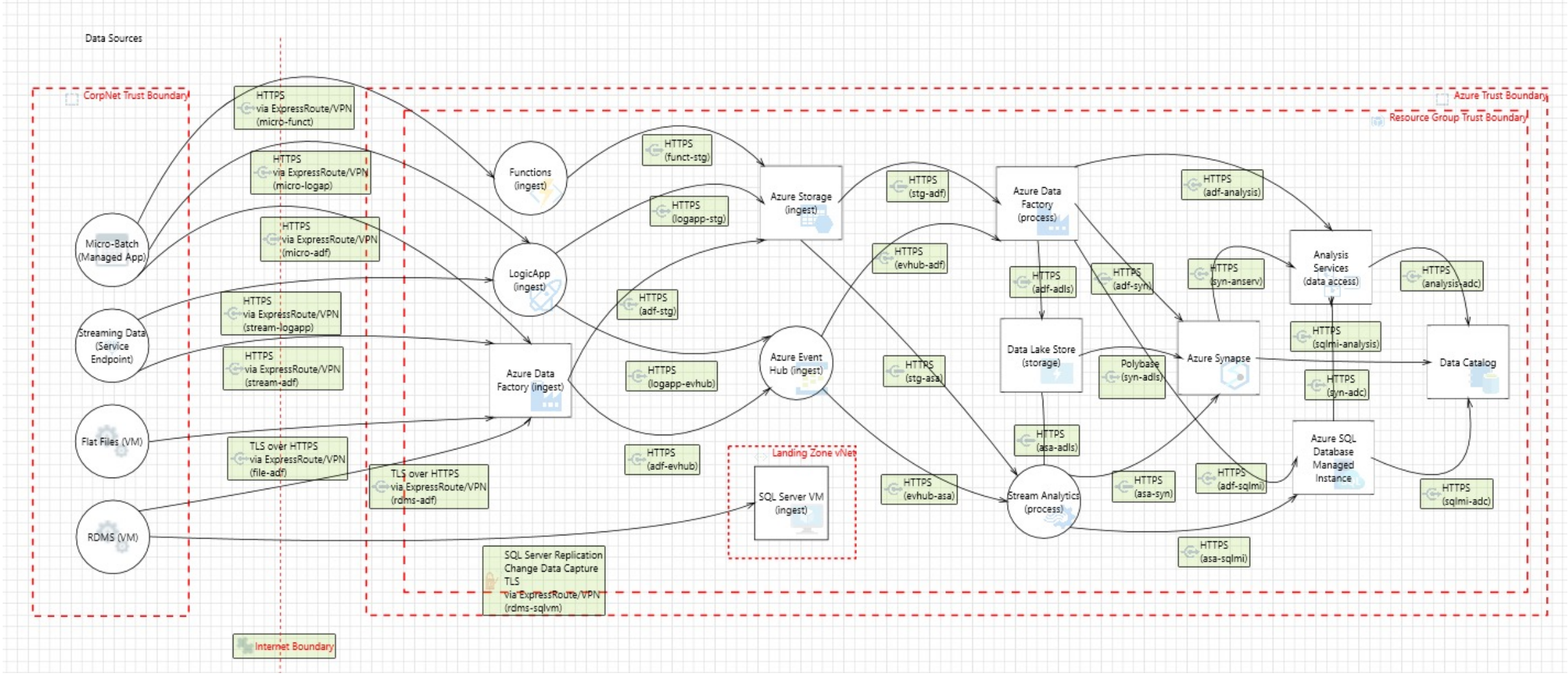
- **What is Threat Modelling?** It's a technique to find security problems **before** implementation happened.
- "Evil brainstorming"
- "Penetration testing on paper"

Threat Modelling – simple?



Source: https://martinfowler.com/

Threat Modelling – hard?



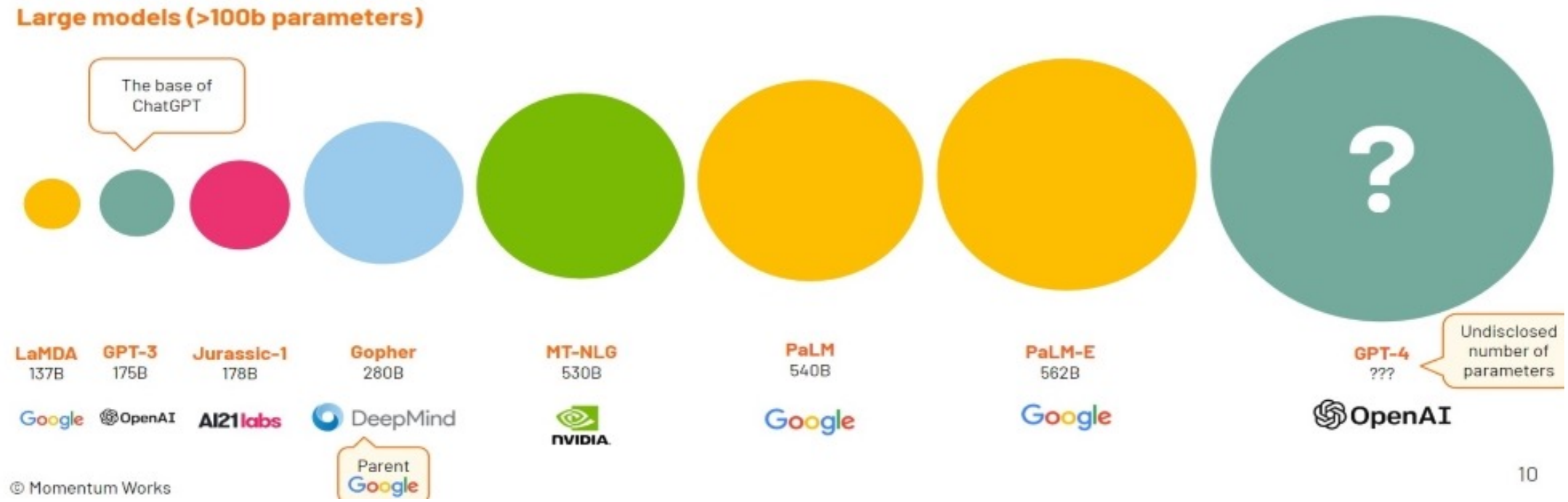
Source: https://github.com/AzureArchitecture/threat-model-templates

FORM3


I LOVE THREAT MODELING
WATCHING PAINT DRY IS FUN TOO
memegenerator.net

Large Language Models 101

What are LLMs? Large language models (LLMs) are deep learning algorithms that can recognize, summarize, translate, predict, and generate content using very large datasets.




5000 YEARS
OF ACCUMULATED
HUMAN
KNOWLEDGE AND KNOWHOW
CHATGPT
imgflip.com

Threat Modelling – get started

To perform threat modelling **manually** we need:

- Description of the “thing” (e.g. architecture of project)
- Developers
- Security Expert

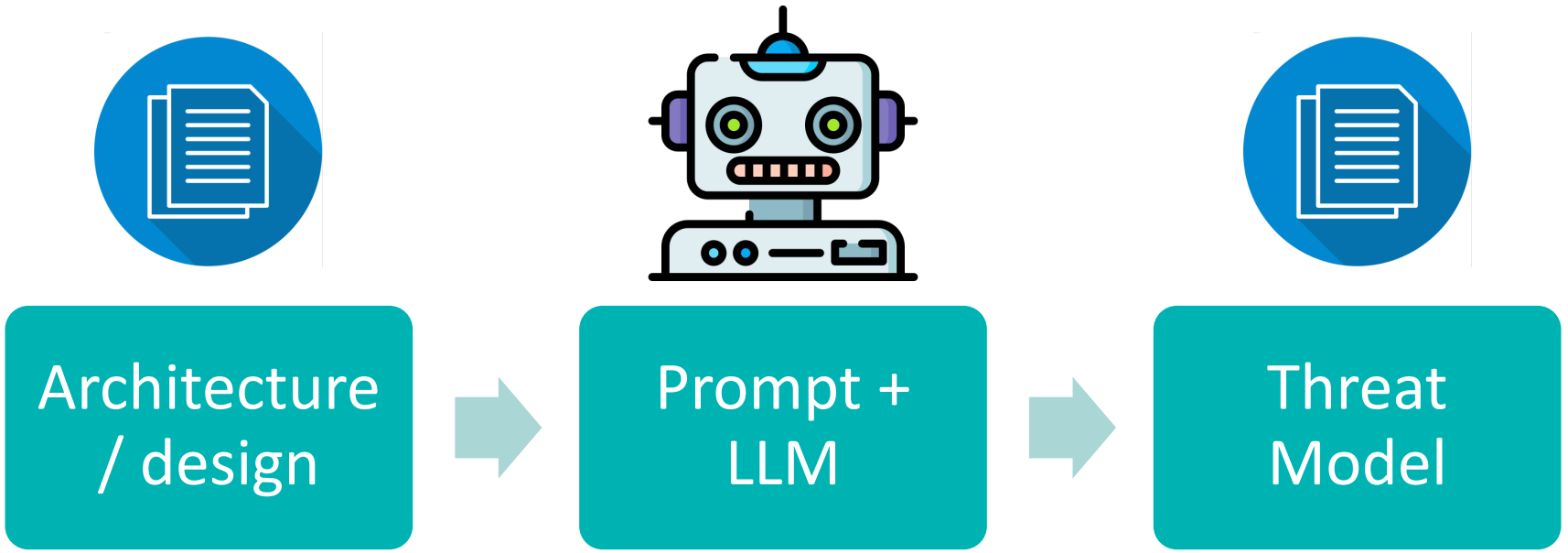
Threat Modelling – get started

To perform threat modelling **with LLM** ✨ we need:

- Description of the “thing” (e.g. architecture of project)
- Developers
- ~~Security Expert~~
- Prompt + LLM

General Idea

Threat Modelling works best in **design stage**. We don't have any code yet. Only architecture / design documents.



General Idea

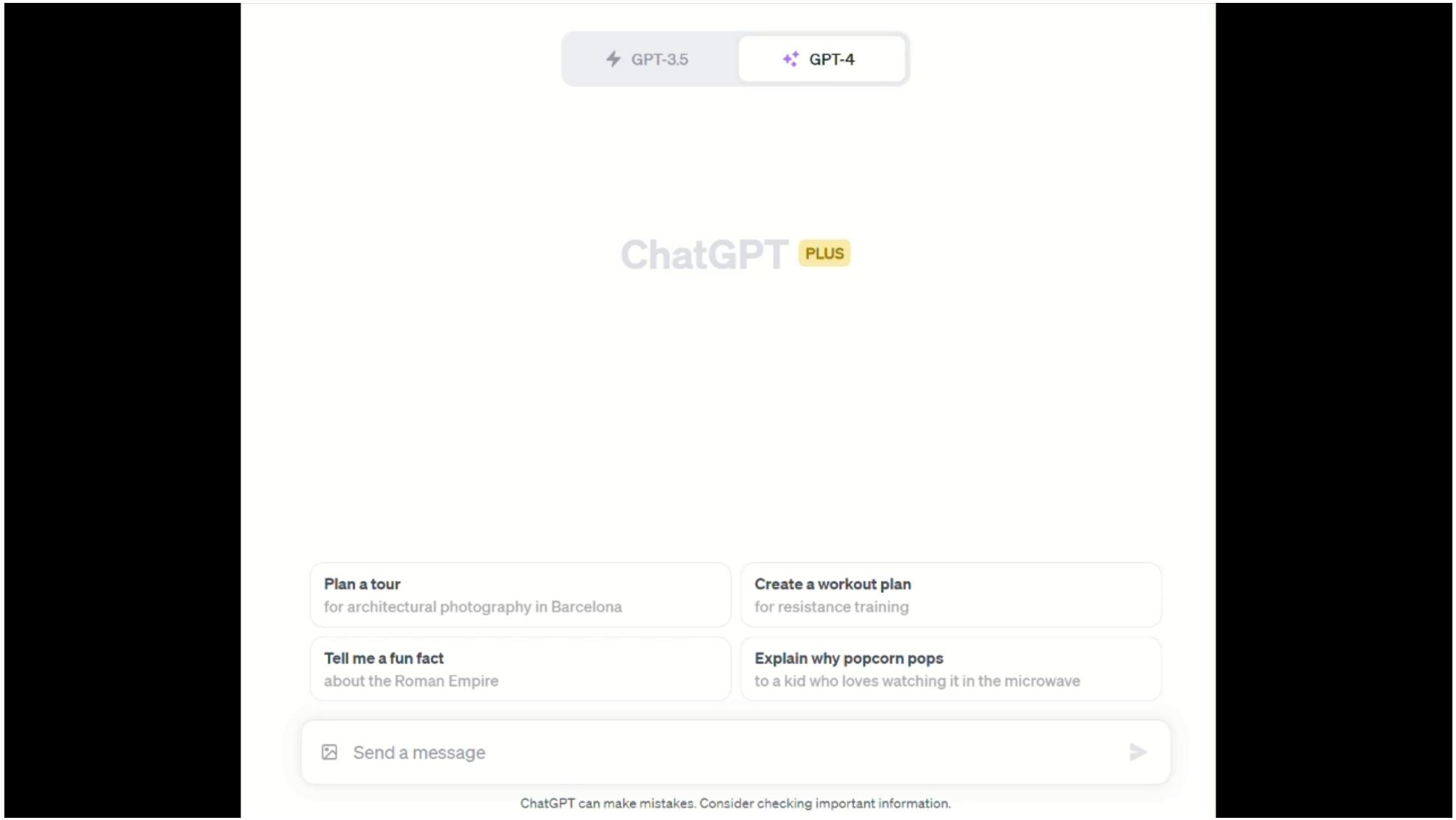
DANIEL MIESSLER ☕ @DanielMiessler · 12 wrz

AI isn't competing with Einstein and Tolstoy and House. AI is competing with:

1. The task not being done at all
2. It being done poorly
3. It being done inconsistently
4. It being done slowly
5. It costing too much

In other words, most things.

Demo


GPT-3.5
GPT-4
ChatGPT PLUS
Plan a tour
for architectural photography in Barcelona
Create a workout plan
for resistance training
Tell me a fun fact
about the Roman Empire
Explain why popcorn pops
to a kid who loves watching it in the microwave
Send a message
ChatGPT can make mistakes. Consider checking important information.

Demo - results

Threat Id	Component name	Threat Name	STRIDE category	Explanation
T001	API Gateway	DDoS Attack	Denial of Service	An attacker may overload the service with high traffic, causing a denial of service for legitimate users.
T002	API Gateway	Man-in-the-Middle (MitM)	Information Disclosure	Data in transit could be intercepted between the API Gateway and clients if TLS is not properly implemented.
T003	API Application	Code Injection	Tampering	An attacker might inject malicious code or scripts into the backend through unvalidated inputs.
T004	API Application	Insecure Deserialization	Elevation of Privilege	Objects could be tampered with before they are deserialized by the API Application.
T005	Web Control Plane	Unauthorized Configuration Changes	Elevation of Privilege	An unauthorized user could gain access to the control plane and alter configurations.

Results overview

- ChatGPT know threat modelling
- ChatGPT can correctly find components
- ChatGPT can write meaningful threats
- **Could we get even better threats?**

What is STRIDE?

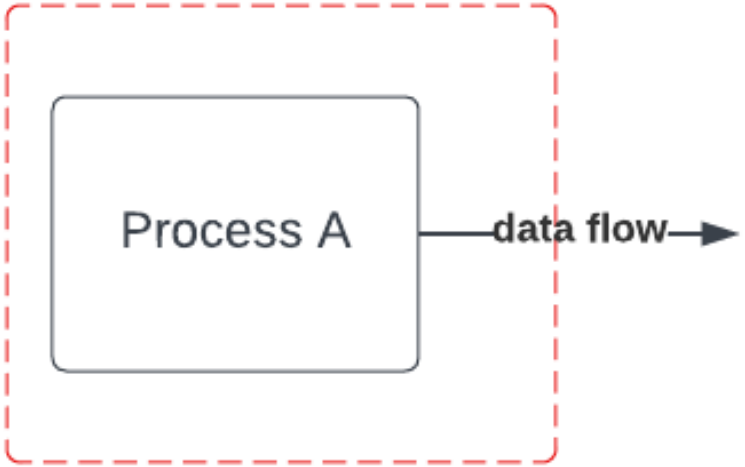
STRIDE	Property Violated	Security Controls
Spoofing	Authentication	MFA, oAuth/OpenID, JWT
Tampering	Integrity	Encryption, hashing, validation, sanitization, encoding
Repudiation	Non-repudiation	Logging, strong authentication
Information disclosure	Confidentiality	Encryption, data protection
Deny of service	Availability	Site/product scalability, rate limiting, backup
Elevation of privilege	Authorization	Access control, least privileges principle

What is STRIDE?

STRIDE	Property Violated	Security Controls
Spoofing	Authentication	MFA, oAuth/OpenID, JWT
Tampering	Integrity	Encryption, hashing, validation,

STRIDE per component

- We apply STRIDE for every component
- As result we get **robust** threat model, that tackle all major security problems
- It’s used by automated tools like: Microsoft Threat Modeler

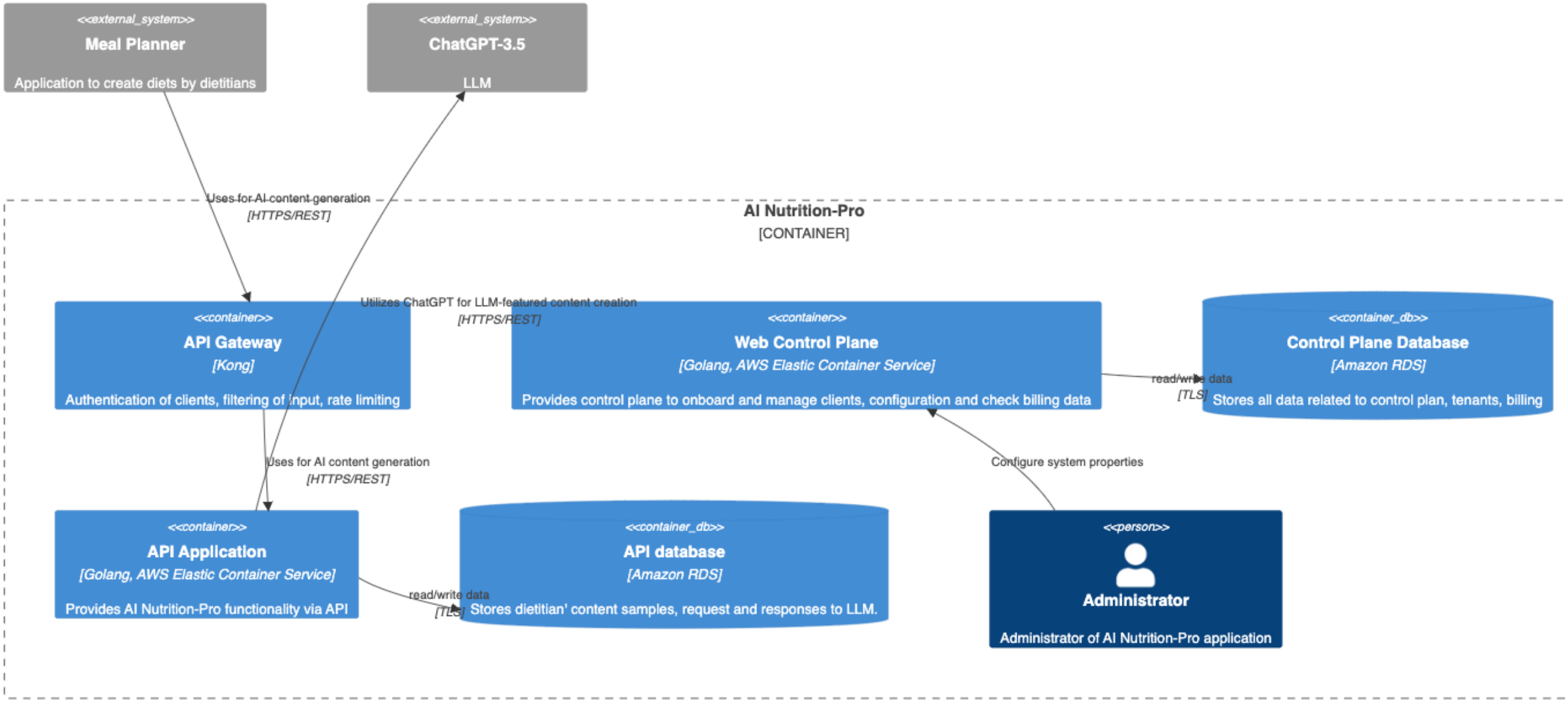


What we need?

To perform threat modelling **with LLM** ✨ we need:

- Description of the “thing” (e.g. architecture of project)
- Developers
- ~~Security Expert~~
- Prompt + LLM

Architecture Description



Architecture Description

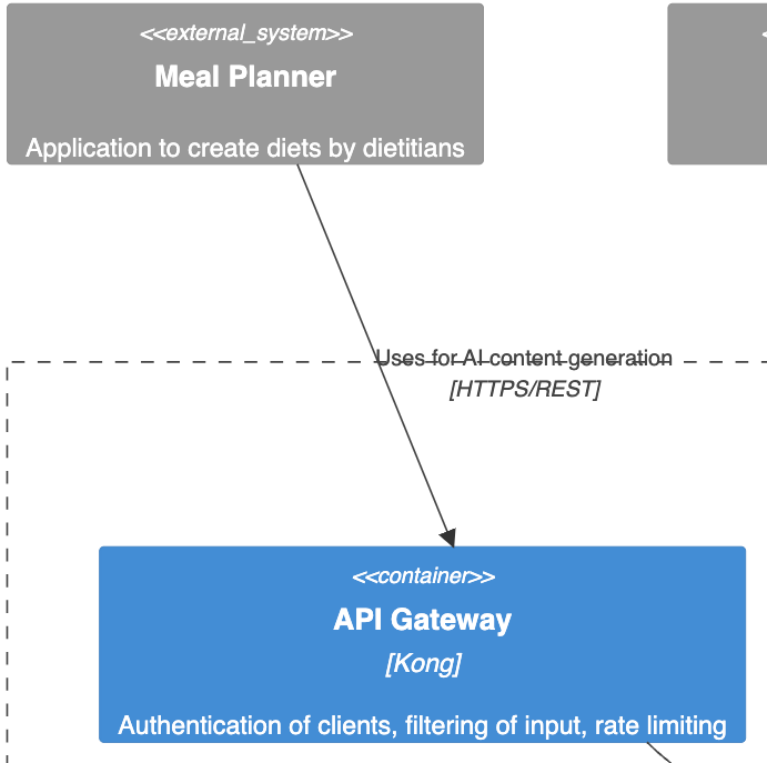
```
7     ```mermaid
8     C4Container
9         title Container diagram for AI Nutrition-Pro
10
11        Container_Boundary(c0, "AI Nutrition-Pro") {
12            Container(api_gateway, "API Gateway", "Kong", "Authentication of clients, filtering of
13            Container(app_control_plane, "Web Control Plane", "Golang, AWS Elastic Container Servic
14            ContainerDb(control_plan_db, "Control Plane Database", "Amazon RDS", "Stores all data r
15            Container(backend_api, "API Application", "Golang, AWS Elastic Container Service", "Pro
16            ContainerDb(api_db, "API database", "Amazon RDS", "Stores dietitian' content samples, r
17            Person(admin, "Administrator", "Administrator of AI Nutrition-Pro application")
18        }
19
```

Diagram is coded in **mermaid** and LLM has no problem to understand it!

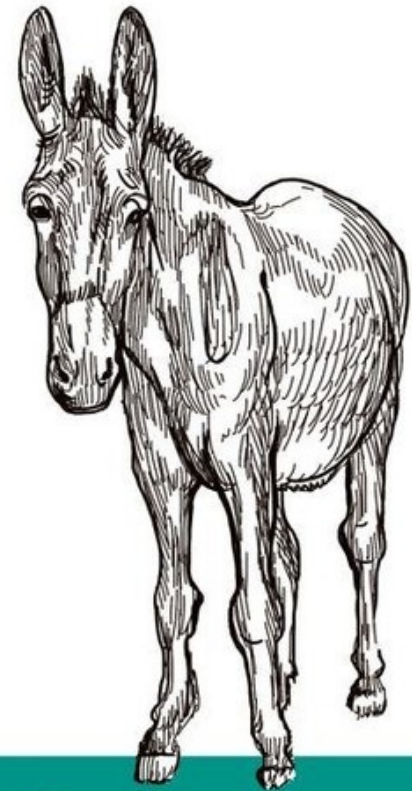
Architecture Description - Security

Security

1. Authentication with Meal Planner applications - each has individual API key.
2. Authorization of Meal Planner applications - API Gateway has ACL rules that allow or deny certain actions.
3. Encrypted network traffic - network traffic between Meal Planner applications and API Gateway is encrypted using TLS.



Where's the fun in just knowing what the code is supposed to do?



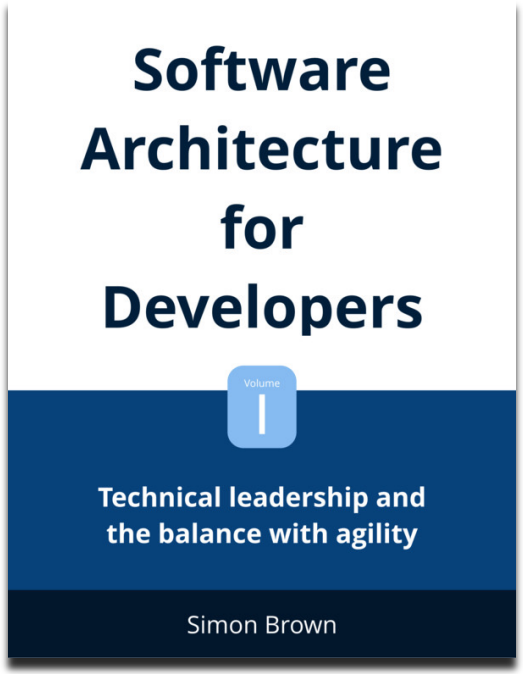
Essential

Excuses for Not Writing Documentation

O RLY?

@ThePracticalDev

Architecture Description

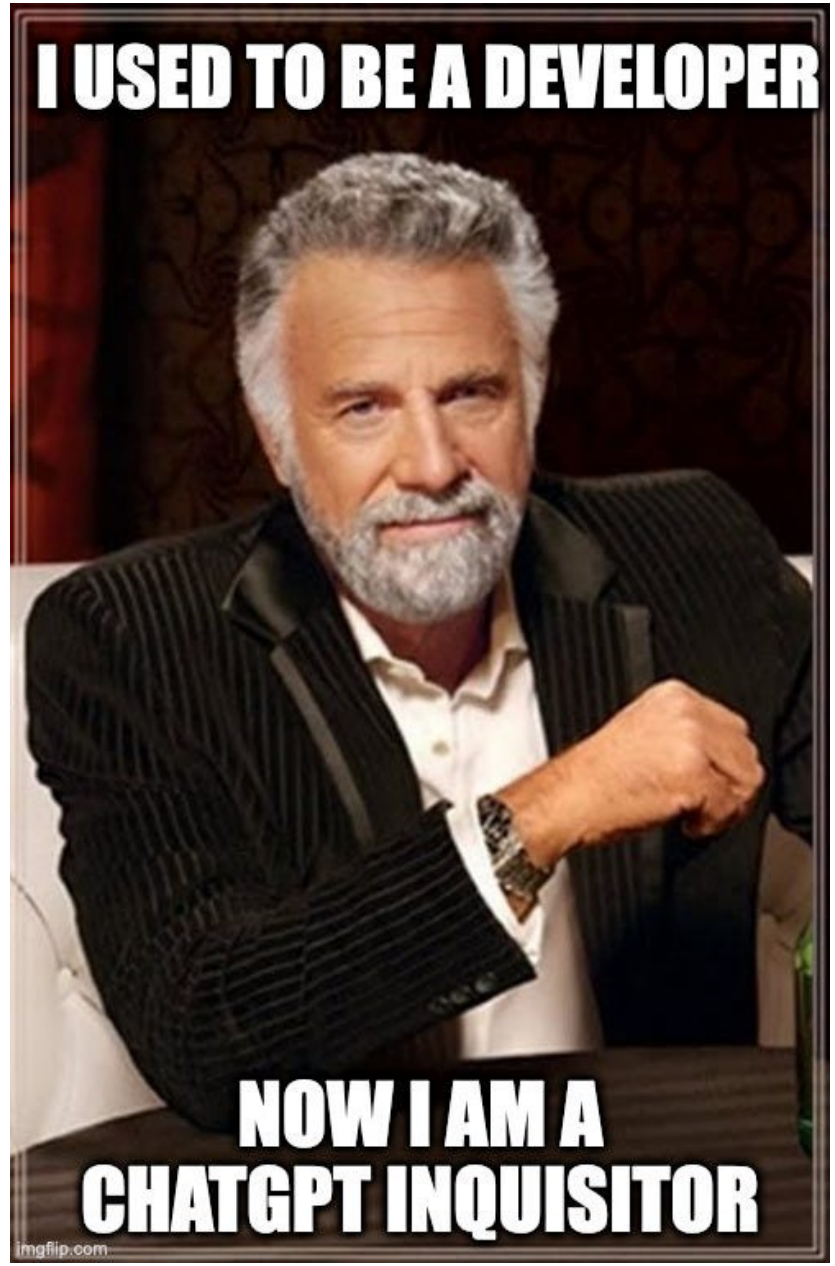


Software Architecture for Developers, Simon Brown

What we need?

To perform threat modelling **with LLM** ✨ we need:

- ☑ Description of the “thing” (e.g. architecture of project)
- ☑ Developer
- ~~Security Expert~~
- Prompt + LLM


I USED TO BE A DEVELOPER
NOW I AM A
CHATGPT INQUISITOR
imgflip.com

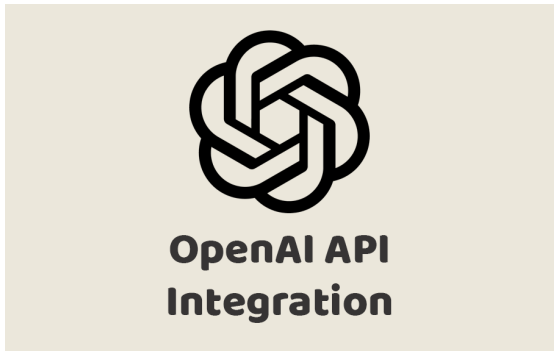
Prompt - key components

You are Security Architect	Setting up persona for AI
Step by step thinking	Best practices if task require logic reasoning
Threat modelling using STRIDE per component	It’s well-defined technique. Output threat model is not perfect, but tackle all major problems

Prompt - key components

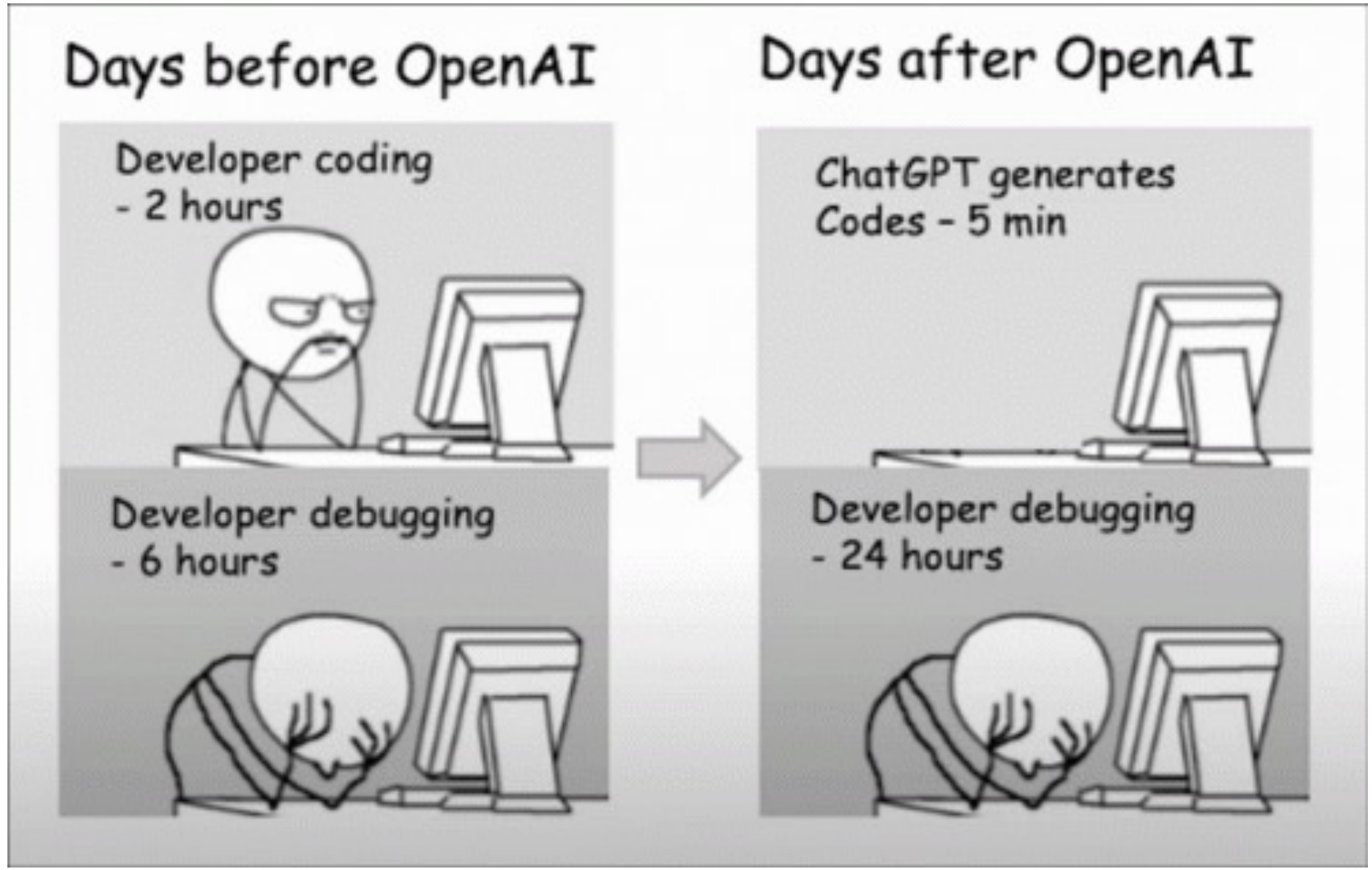
Threat Explanation	Why this threat is important for component in context of Architecture Description
How threat is already mitigated in architecture	Explain if this threat is already mitigated in Architecture Description or not
0 shot prompt	I don’t provide any examples of good work to AI

Which LLMs can we use?



- Ad-hoc
- Easy to experiment and tune prompts
- Not consistent*

- Can be integrated with your existing systems
- Allows to create threat models in place where you keep your documentation


Days before OpenAI
Developer coding
- 2 hours
Developer debugging
- 6 hours
Days after OpenAI
ChatGPT generates
Codes - 5 min
Developer debugging
- 24 hours

Threat Modelling using API

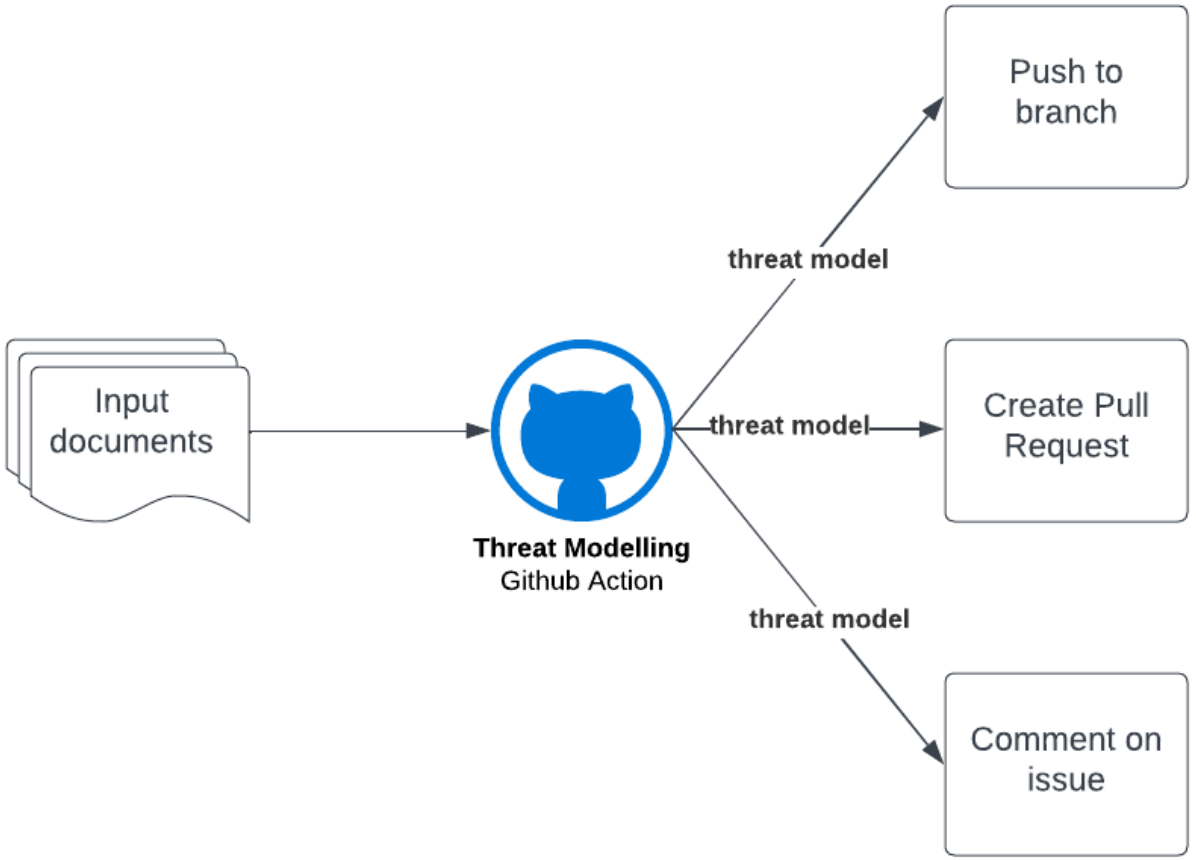
Introducing AI Threat Modeling action

⚠️ This is experimental project

You can:

- Use it in your own project in GitHub
- Fork it, tune prompts for your needs
- Use in any place you want with python scripts: xvnpw/ai-threat-modeling

Threat Modelling using API



Threat Modelling using API

Features:

- High Level Security and Privacy Requirements
- **Threat Model of Architecture**
- Security Acceptance Criteria for User Story
- Review of Architecture Description

Usage Example

```
18        name: Run ai threat modeling action for architecture analysis
19        steps:
20          - name: Checkout repo
21            uses: actions/checkout@v3
22          - name: Generate architecture threat model
```

```
37              add: 'ARCHITECTURE_SECURITY.md'
38              pull: '--rebase --autostash'
```

Usage Example

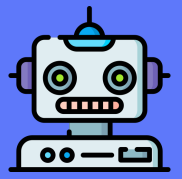
```
18        name: Run ai threat modeling action for architecture analysis
19        steps:
20          - name: Checkout repo
21            uses: actions/checkout@v3
22          - name: Generate architecture threat model
23            uses: xvnpw/ai-threat-modeling-action@v1.2.8
24            with:
25              type: 'architecture'
26              input_files: '["ARCHITECTURE.md"]'
27              output_file: 'ARCHITECTURE_SECURITY.md'
28              temperature: 0.2
29              verbose: true
30              model: 'gpt-4'
31            env:
32              OPENAI_API_KEY: ${{ secrets.OPENAI_API_KEY }}
33          - name: Commit changes
34            uses: EndBug/add-and-commit@v9
35
36
37
38              pull: '--rebase --autostash'
```

Usage Example

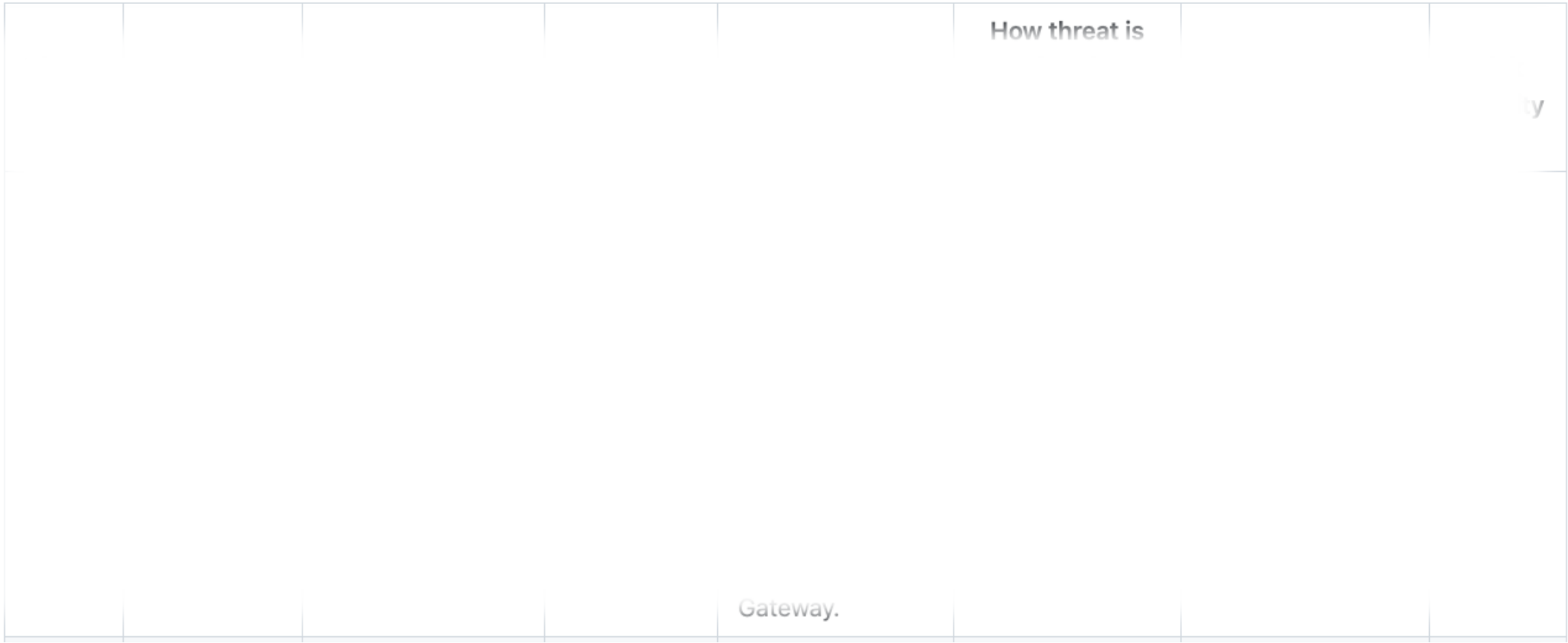
```yaml
18        name: Run ai threat modeling action for architecture analysis
19        steps:
20          - name: Checkout repo
21            uses: actions/checkout@v3
22          - name: Generate architecture threat model
23            uses: xvnpw/ai-threat-modeling-action@v1.2.8
24            with:
25              type: 'architecture'
26              input_files: '["ARCHITECTURE.md"]'
27              output_file: 'ARCHITECTURE_SECURITY.md'
28              temperature: 0.2
29              verbose: true
30              model: 'gpt-4'
31            env:
32              OPENAI_API_KEY: ${{ secrets.OPENAI_API_KEY }}
33          - name: Commit changes
34            uses: EndBug/add-and-commit@v9
35            with:
36              message: 'Project architecture threat model'
37              add: 'ARCHITECTURE_SECURITY.md'
38              pull: '--rebase --autostash'
```

GitHub Action output

AI output



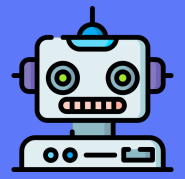
Data flow 1: Meal Planner application -> API Gateway



FORM3

GitHub Action output

AI output



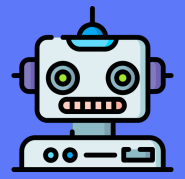
Data flow 1: Meal Planner application -> API Gateway

Threat Id	Co			How threat is		isk erity
1	API Gat					h
				Gateway.		

FORM3

GitHub Action output

AI output



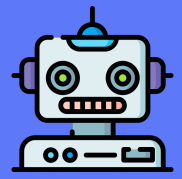
Data flow 1: Meal Planner application -> API Gateway

Threat Id	Component name				How threat is		Risk Severity
1	API Gateway	Sp by gu		Gateway.			gh

FORM3

GitHub Action output

AI output



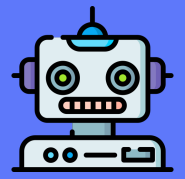
Data flow 1: Meal Planner application -> API Gateway

Threat Id	Component name	Threat Name	S ca		How threat is		isk erity
1	API Gateway	Spoofing attack by using stolen or guessed API keys	Sp	Gateway.			h

FORM3

GitHub Action output

AI output



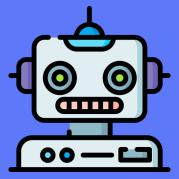
Data flow 1: Meal Planner application -> API Gateway

Threat Id	Component name	Threat Name	STRIDE category	E	How threat is		isk erity
1	API Gateway	Spoofing attack by using stolen or guessed API keys	Spoofing	An cou or key imp leg Pla app gai una acc Gateway.			h

FORM3

GitHub Action output

AI output



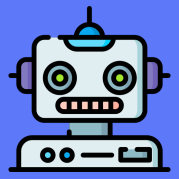
Data flow 1: Meal Planner application -> API Gateway

Threat Id	Component name	Threat Name	STRIDE category	Explanation	How threat is		isk erity
1	API Gateway	Spoofing attack by using stolen or guessed API keys	Spoofing	An attacker could use stolen or guessed API keys to impersonate a legitimate Meal Planner application, gaining unauthorized access to the API Gateway.	Pa mit ind ke aut as arc de		h

FORM3

GitHub Action output

AI output



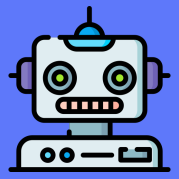
Data flow 1: Meal Planner application -> API Gateway

Threat Id	Component name	Threat Name	STRIDE category	Explanation	How threat is already mitigated in architecture		isk erity
1	API Gateway	Spoofing attack by using stolen or guessed API keys	Spoofing	An attacker could use stolen or guessed API keys to impersonate a legitimate Meal Planner application, gaining unauthorized access to the API Gateway.	Partially mitigated by individual API keys for authentication as stated in the architecture description.	Im fac aut reg API mo un	h

FORM3

GitHub Action output

AI output



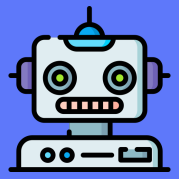
Data flow 1: Meal Planner application -> API Gateway

Threat Id	Component name	Threat Name	STRIDE category	Explanation	How threat is already mitigated in architecture	Mitigations	Risk severity
1	API Gateway	Spoofing attack by using stolen or guessed API keys	Spoofing	An attacker could use stolen or guessed API keys to impersonate a legitimate Meal Planner application, gaining unauthorized access to the API Gateway.	Partially mitigated by individual API keys for authentication as stated in the architecture description.	Implement multi-factor authentication, regularly rotate API keys, and monitor for unusual activity.	High

FORM3

GitHub Action output

AI output

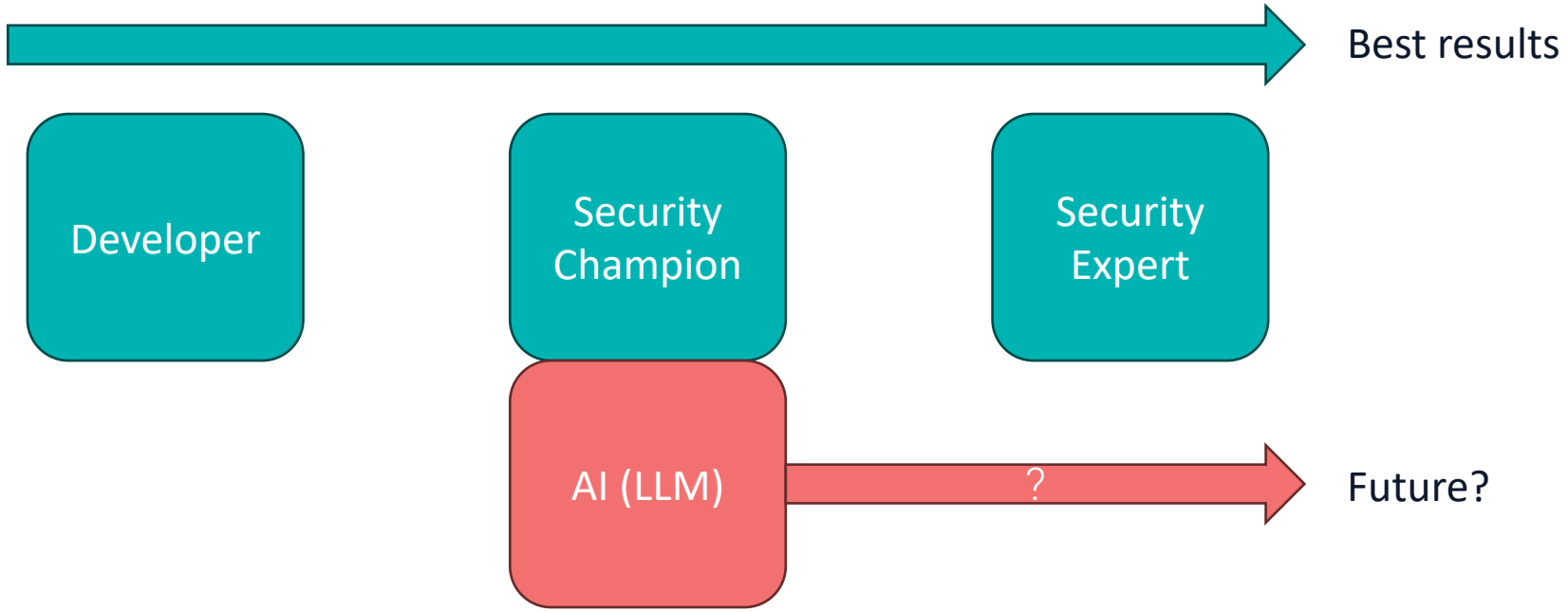


Data flow 1: Meal Planner application -> API Gateway

Threat Id	Component name	Threat Name	STRIDE category	Explanation	How threat is already mitigated in architecture	Mitigations	Risk severity
1	API Gateway	Spoofing attack by using stolen or guessed API keys	Spoofing	An attacker could use stolen or guessed API keys to impersonate a legitimate Meal Planner application, gaining unauthorized access to the API Gateway.	Partially mitigated by individual API keys for authentication as stated in the architecture description.	Implement multi-factor authentication, regularly rotate API keys, and monitor for unusual activity.	High

FORM3

Grading results



Summary

- LLMs can perform threat modelling and logical reasoning
- Good description of design is key for useful results
- GPT-4 and Claude-3 perform well in threat modelling
- GitHub Actions give easy way to integrated existing documents with AI featured flows

Summary

- Experiment inputs and results for GPT-4
- Threat Modeling GitHub Action

More to read

- Matthew Adam Linkedin – analysing JWT flows with GPT4-V
- Clint Gibler on AI and cybersecurity: the current state of the art and where we're headed
- Leveraging LLMs for Threat Modeling - GPT-3.5 vs Claude 2 vs GPT-4
- Leveraging LLMs for Threat Modeling - GPT-3.5
- Reviewing Your Architecture Using LLMs

Thank you!

Questions?