**Folie 1:**

Wichtigsten Begriffe klären – denn wir kommen aus unterschiedlichen Fachrichtungen:

* **Threat Modeling** ist ein strukturierter Prozess, mit dem Sicherheitsschwachstellen schon in der frühen Planungs- oder Designphase systematisch erkannt werden sollen – bevor überhaupt Code geschrieben wird.
* Weil spätere Änderungen – also wenn eine Schwachstelle erst beim Penetrationstest auffällt, **viel teurer und aufwändiger** sind. Threat Modeling hilft, **präventiv zu handeln**, statt nur reaktiv.
  + Typischerweise wird das **STRIDE-Modell** verwendet, das Bedrohungen wie Spoofing, Tampering oder Information Disclosure identifiziert.
* Ein zentrales Werkzeug: **Data Flow Diagram (DFD)**.  
  Wie Daten durch ein System fließen: Welche Prozesse es gibt, welche externen Entitäten beteiligt sind, wo Daten gespeichert werden und wie alles miteinander verbunden ist.
* Das DFD ist Grundlage für Bedrohungsanalyse nach dem STRIDE-Modell
  + das bestimmte Bedrohungskategorien systematisch abfragt, sodass man kritische Schnittstellen erkennt.

Und genau hier kommt unsere Fragestellung ins Spiel:  
**Können LLMs uns dabei unterstützen, diese Analyse einfacher, effizienter oder sogar besser zu machen?“**

**Folie 2:** **Bestehende Studien**

Die bisherigen Studien zeigen ein gemischtes Bild.

In der Studie von Mbaka & Tuma schneiden Gruppen mit LLMs besser ab , sie erkennen mehr reale Bedrohungen. Aber: Sie markieren auch viele falsche. Besonders unerfahrene Nutzer übernehmen zu schnell die LLM-Antworten.

Yang et al. zeigen, dass DFDs automatisiert erstellt werden können – ein vielversprechender Schritt, allerdings noch ungenau und fehleranfällig.

Bei PILLAR geht es um Datenschutzanalyse – auch hier erkennt das LLM viele Bedrohungen, scheitert aber oft am Kontext.

Klassische Ansätze wie bei Tuma & Scandariato sind dafür präziser, aber extrem aufwendig.

**Fazit:** Es gibt Potenzial – aber auch klare Risiken und Grenzen. Entscheidend ist, wie gut das Mensch-Maschine-Verhältnis gestaltet wird.

**Folie 4:**

„Unser Ansatz setzt genau hier an:

Beginnend mit zwei Workshops:  
Der erste mit Expert*innen für Feedback und Ideen zum Interface, der zweite mit Entwickler*innen, die das Tool in der Praxis getestet haben.

Die grobe Idee: ein Chat-Interface entwickelen, einLLM analysiert ein hochgeladenes DFD und generiert konkrete STRIDE-Vorschläge [als Assistenz?]. Was ist dabei wichtig? Chain-of-Tought technik? LLM antworteten sind transparent? Soll der Nutzer alles sehen, bearbeiten und löschen können?

Unser Ziel war **nicht** ein Vergleich von Precision/Recall und auch nicht ob das LLM „besser“ ist als Menschen – sondern qualitative Einsichten:  
Wie arbeiten Menschen mit dem Tool?

**Folie 5: Unser Ansatz**

Unser Ziel war **nicht** ein Benchmark oder ein Vergleich von Precision/Recall und auch nicht ob das LLM „besser“ ist als Menschen – sondern qualitative Einsichten:  
Wie arbeiten Menschen mit dem Tool?

**Funktioniert der Dialog?**  
Verstehen sie die Erklärungen?  
Greifen sie ein – oder übernehmen sie blind?

Diese Fragen helfen uns, das System so zu gestalten, dass es in echten Projekten funktioniert könnte

**Folie 7:**

Diese Fragen leiten die Diskussion. Es geht darum, gemeinsam blinde Flecken zu identifizieren und realistische Anforderungen zu formulieren. Wir freuen uns auf euer Feedback.

**Folie 8:**

1. **Einstieg über das Video: Schrittweise abspielen und pausieren**
   * Teile das Video in 2–3 Abschnitte auf und spiele diese nacheinander ab, jeweils mit einer kurzen Pause dazwischen:
   * Prompt zeigen → wie wurde das LLM „gefüttert“
   * LLM-Antwort zeigen → z. B. erkannte Bedrohungen inkl. STRIDE-Zuordnung und Begründung  
     Nach jedem Abschnitt gezielte Fragen stellen:  
       
     "Was halten Sie von diesem Schritt?"  
     "Wurde etwas Wichtiges vergessen?"  
     "Ist der Output sinnvoll nutzbar?"
2. **Gruppenarbeit oder Kleingruppen nach Rolle**

    Im Anschluss an das Video: Teilnehmende in Kleingruppen aufteilen (LLM, Sicherheit, Entwicklung).  
  
    Jede Gruppe bearbeitet die für sie relevanten Fragen (aus der Erwartungsfolie), z. B.:

* LLM-Expert:innen:
  + "Wie bewerten Sie die Prompt-Struktur?"
  + Wie könnte man die Begründungen verbessern?
* Sicherheitsexpert:innen:
  + "Sind die STRIDE-Kategorisierungen nachvollziehbar?"
  + „Was müsste gegeben sein, damit Sie diesem Tool vertrauen würden?"
* Entwickler:innen:
  + Würden Sie mit solchem Output weiterarbeiten können?"

1. **Live-Voting oder Priorisierung**  
     
   Nach der Demo oder den Gruppenrückmeldungen gezielt Meinungen abfragen:  
     
   "Würden Sie diesem System (so wie gezeigt) vertrauen?"  
   "Welche Form der Darstellung ist für Sie am nützlichsten?"  
   "Wer sollte das letzte Wort bei der Threat-Auswahl haben?"
2. **Live-Dokumentation oder Clustering**  
     
   Sammle zentrale Punkte während der Diskussion in Echtzeit, geordnet nach Themenfeldern:
   * Vertrauen
   * Erklärbarkeit
   * Design

„Was wäre aus Ihrer Sicht das wichtigste Feature eines solchen Systems?“