



KI-Softwareentwicklung

mit Azure OpenAI und MS Semantic Kernel



Misha Hüschen
Managing Director
prodot GmbH



Michael Damatov
Principal Software Engineer
prodot GmbH

Agenda

Generative KIs

... bei der Softwareentwicklung

... in “Mainstream” Software

... in eigener Software

“Let’s build a Copilot”





Generative KIs

Generative KIs

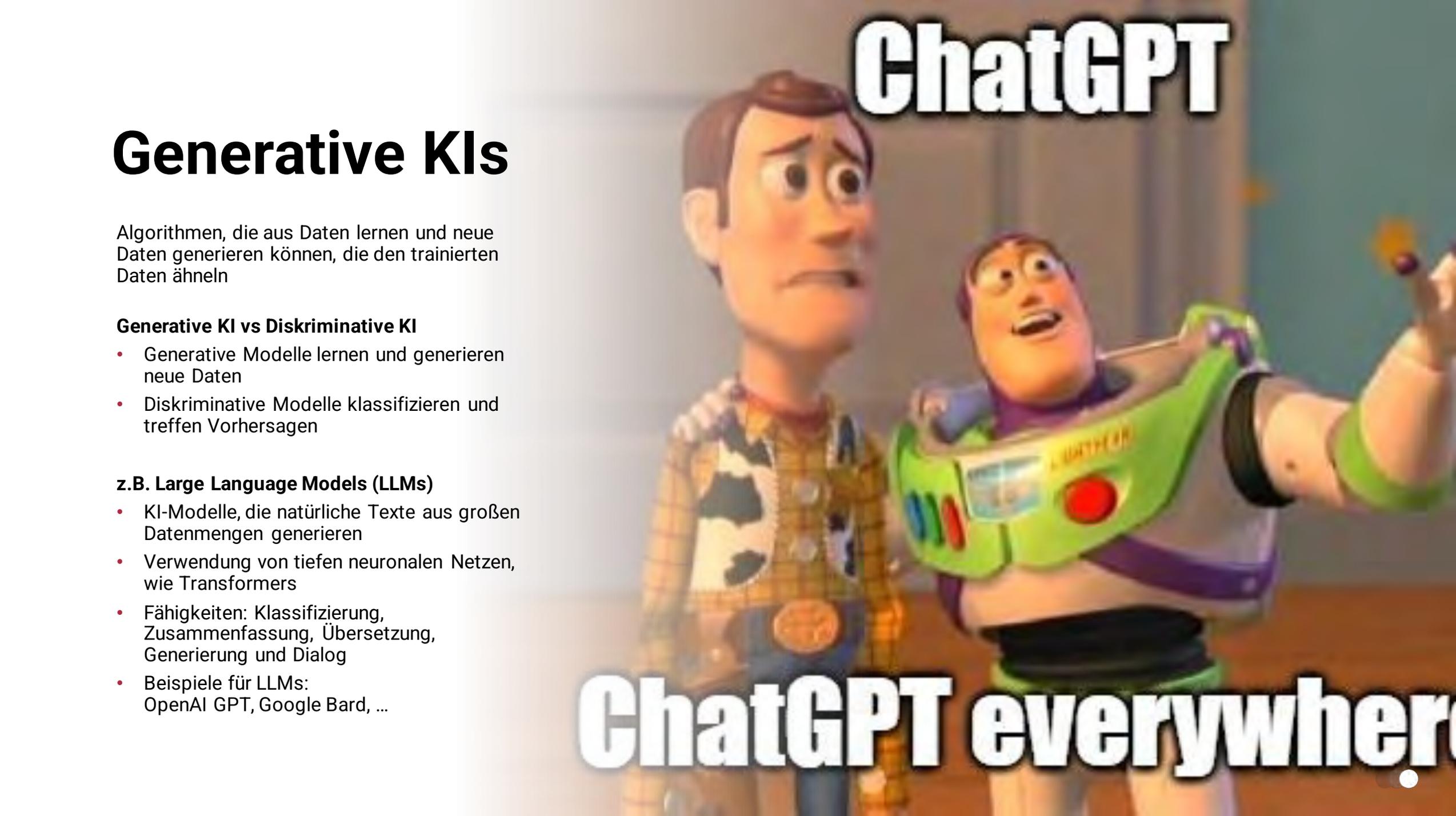
Algorithmen, die aus Daten lernen und neue Daten generieren können, die den trainierten Daten ähneln

Generative KI vs Diskriminative KI

- Generative Modelle lernen und generieren neue Daten
- Diskriminative Modelle klassifizieren und treffen Vorhersagen

z.B. Large Language Models (LLMs)

- KI-Modelle, die natürliche Texte aus großen Datenmengen generieren
- Verwendung von tiefen neuronalen Netzen, wie Transformers
- Fähigkeiten: Klassifizierung, Zusammenfassung, Übersetzung, Generierung und Dialog
- Beispiele für LLMs: OpenAI GPT, Google Bard, ...



ChatGPT

ChatGPT everywhere

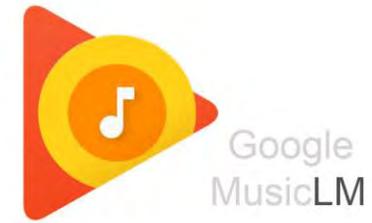
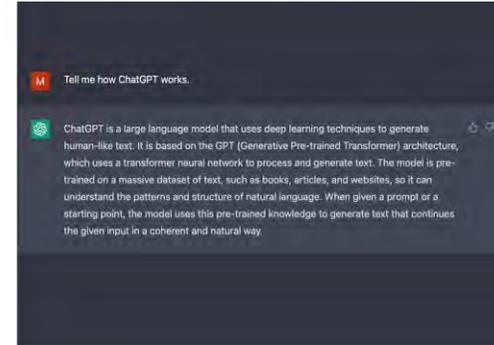
Wozu generative KI?

Generative KI spielt eine Schlüsselrolle bei der Schaffung neuer Inhalte und der Automatisierung von Aufgaben

Beispiele für generative KI

- Textgenerierung **ChatGPT**
- Musikgenerierung **Google MusicLM**
- Bildgenerierung **Midjourney**
- 3D-Generierung **Luma AI**

- und mehr ...



MIDJOURNEY

"marble statue of a greek philosopher"



Zip-NeRF
Anti-Aliased Grid-Based Neural Radiance Fields

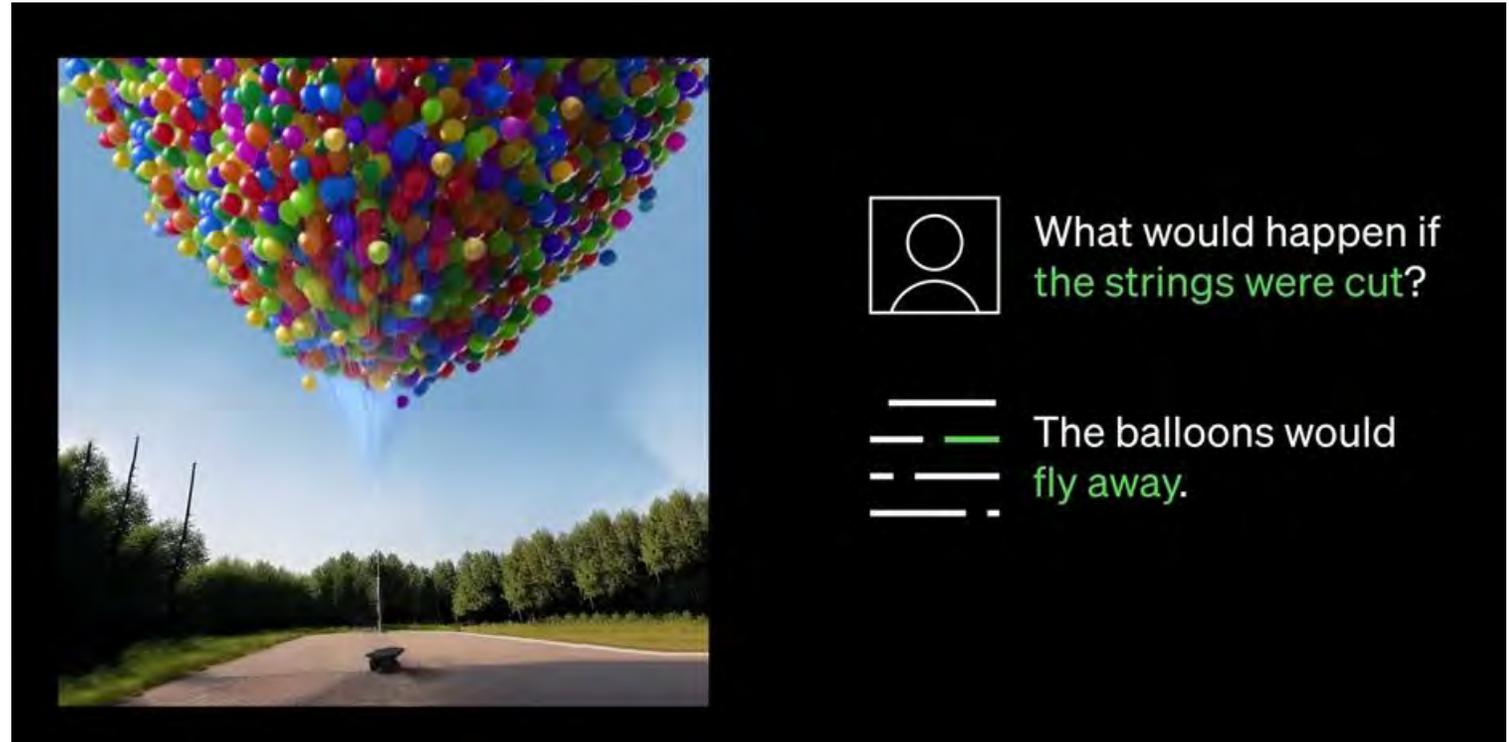


Multimodale KI-Modelle



... bezieht sich auf die Integration mehrerer Modalitäten wie **Text, Bild, Sprache, Video etc.** in ein einziges KI-Modell.

- unterschiedliche Datenformate
- Bildbeschreibung / -analyse
- Potenzial für verbesserte User-Interaktion



Besseres Verständnis von Kontext!

Things are moving fast ...

Generative KI entwickelt sich schnell

- Wöchentlich, teilweise täglich neue Features, Produkte und Integrationen oder mindestens Ankündigungen
- Niemand kommt um dieses Thema herum

Was heißt das für Softwareentwickler?





... bei der Softwareentwicklung

Doc Evil is taking your Job

Generative KI kann Source Code

- ... verstehen und erklären
- ... verbessern
- ... debuggen
- ... auf Sicherheit überprüfen
- ... erzeugen



Was bedeutet das für Softwareentwickler?

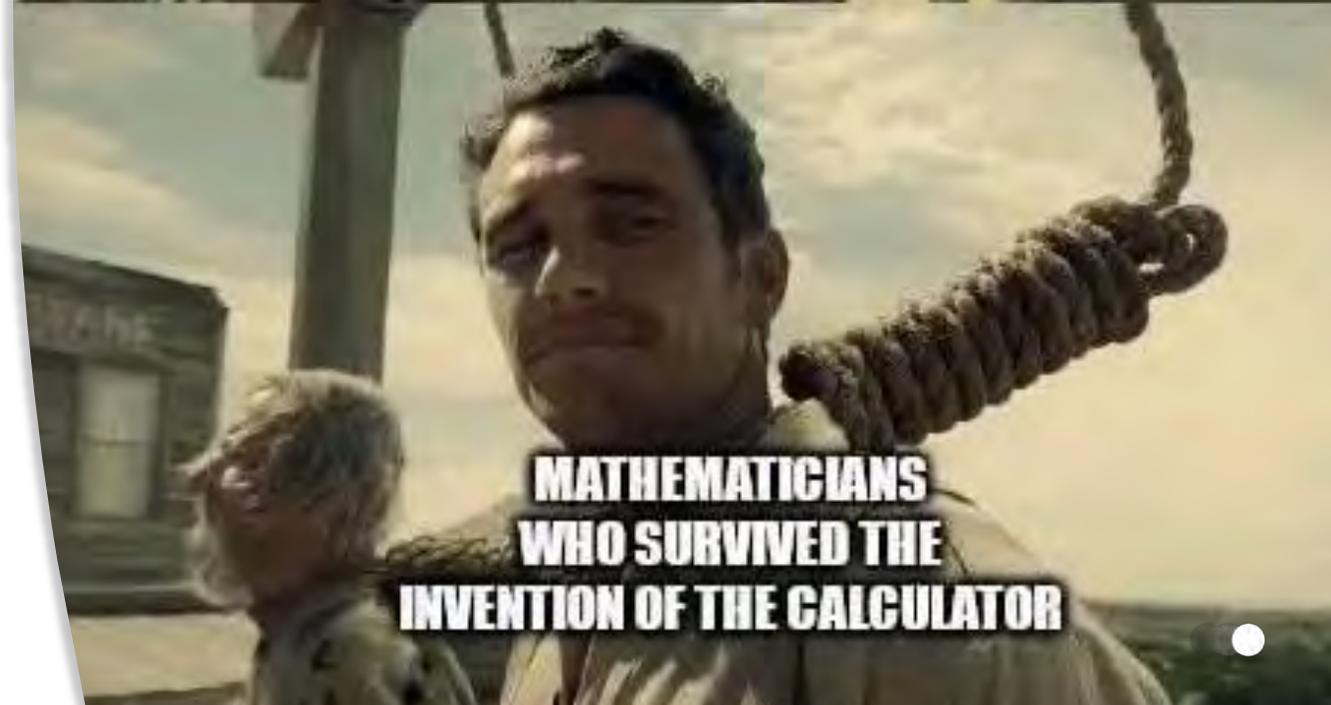
Production-Software ist komplex und kombiniert Dinge wie

- Design, Funktionalität
- Implementierung, Testing
- Maintenance, DevOps
- ... und mehr

Softwareentwickler erstellen Produkte, die

- die Bedürfnisse der User
- und die spezifischen Rahmenbedingungen der gewünschten Lösung
- mit vielen weiteren Faktoren in Einklang bringen
- und betreiben, warten und supporten diese.

Aber die Art und Weise, wie wir das tun, ändert sich!





GitHub Copilot



Redefining developer productivity.

75% more fulfilled

1M+ developers

46%
code written

55%
faster coding

```
runtime.go  JS (days,000000,000000)
1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
8
9 // Get average runtime in seconds
10 func averageRuntimeInSeconds
```

✓ 5,000+
businesses have used it



Github Copilot Features

Lösung größerer Probleme:

- Weniger Boilerplate-Code
- effizientes Arbeiten
- Code-Vorschläge

KI-basierte Vorschläge

- Projekt-spezifische Empfehlungen
- Durchlaufe schnell Codezeilen
- Vervollständigung von Funktionen

Im gewohnten Editor

- Neovim, JetBrains IDEs
- Visual Studio und Visual Studio Code

Sicherer Code mit neuen Sprachen

- Unterstützung bei neuen Sprachen/Frameworks
- Bug-Bewältigung
- Dokumentationsnavigation

Die nächste Version wird noch deutlich mehr können.



Github Copilot X



Github Copilot X - Ankündigungen

Copilot Chat

- ChatGPT-ähnlich im Editor
- Code-Erkennung
- Fehleranalyse
- IDE-Integration

Copilot für Pull-Anfragen

- KI-generierte Beschreibungen
- Anpassbare Vorschläge

Copilot für Dokumentation

- KI-generierte Antworten
- Dokumentationshilfe

Copilot für CLI

- Befehlserstellung
- Syntaxerinnerung

.. ***und viele weitere***

Nutzung von generativer KI bei der Softwareerstellung

- **Effizienzsteigerung**
KI-gestützte Tools reduzieren repetitives Codieren und erhöhen die Produktivität.
- **Qualitätsverbesserung**
Bessere Codevorschläge und Fehlererkennung erhöhen die Softwarequalität.
- **Lernunterstützung**
Unterstützt das Erlernen neuer Sprachen oder Frameworks.
- **Individualisierte Vorschläge**
KI lernt aus deinem Code-Stil und passt Vorschläge an.
- **Zeitersparnis**
Weniger Zeit mit der Suche nach Lösungen im Web oder in der Dokumentation verbringen.
- **Integration**
Nahtlose Integration in gängige IDEs und Editoren.

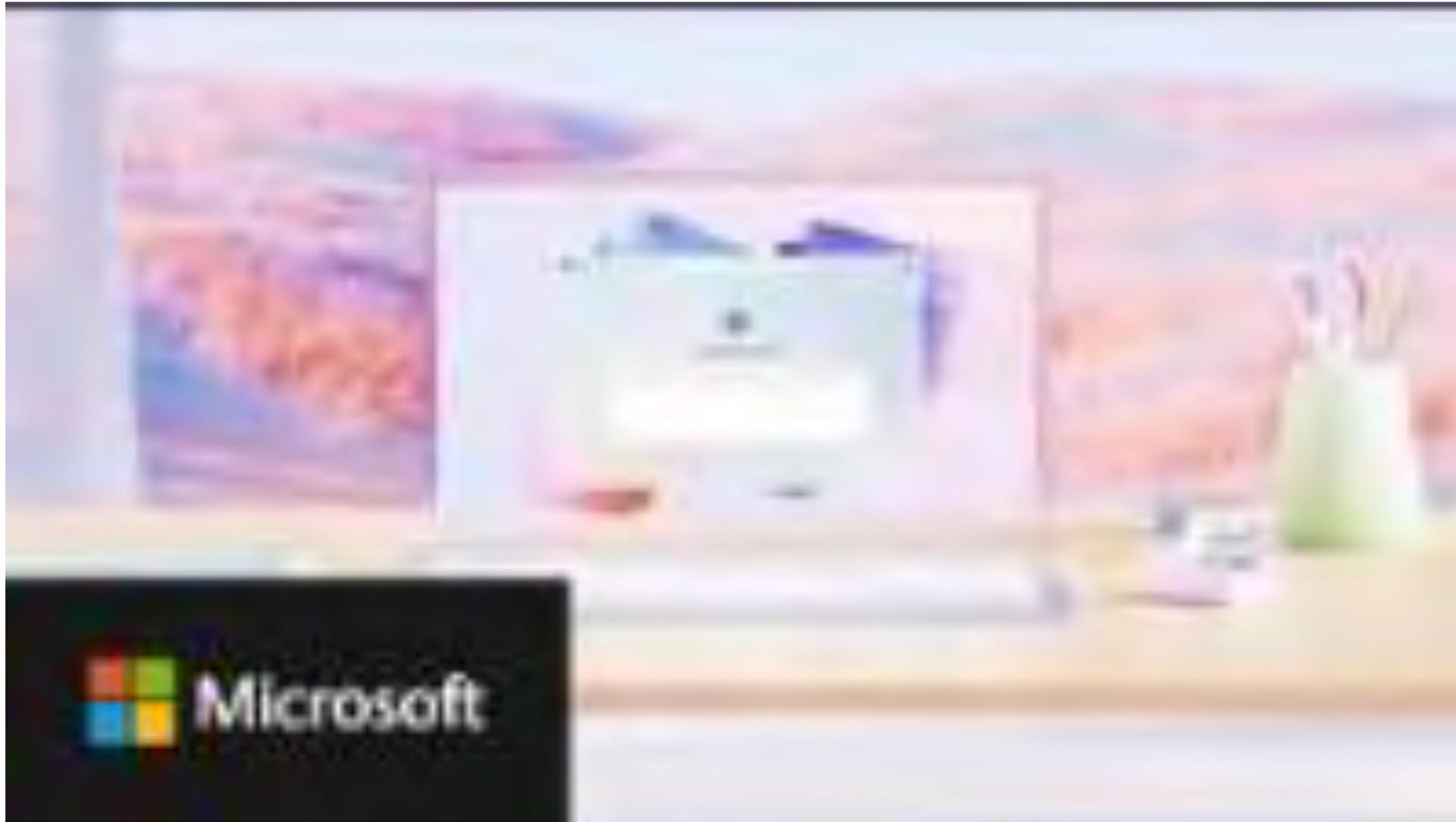


... und was macht generative KI mit der Software selbst?



... in „Mainstream“ Software

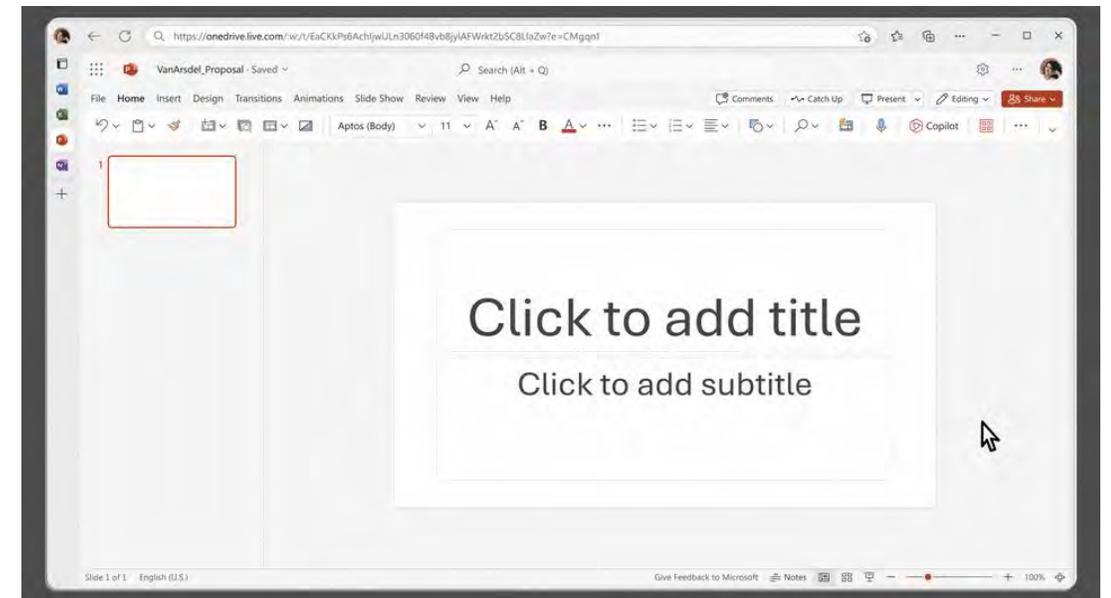
Introducing Microsoft 365 Copilot



Microsoft Copilot

... wird in die gesamte Microsoft Produktpalette integriert

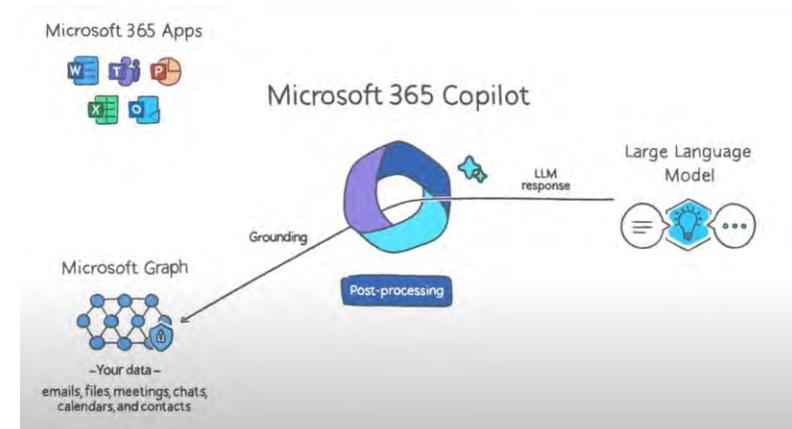
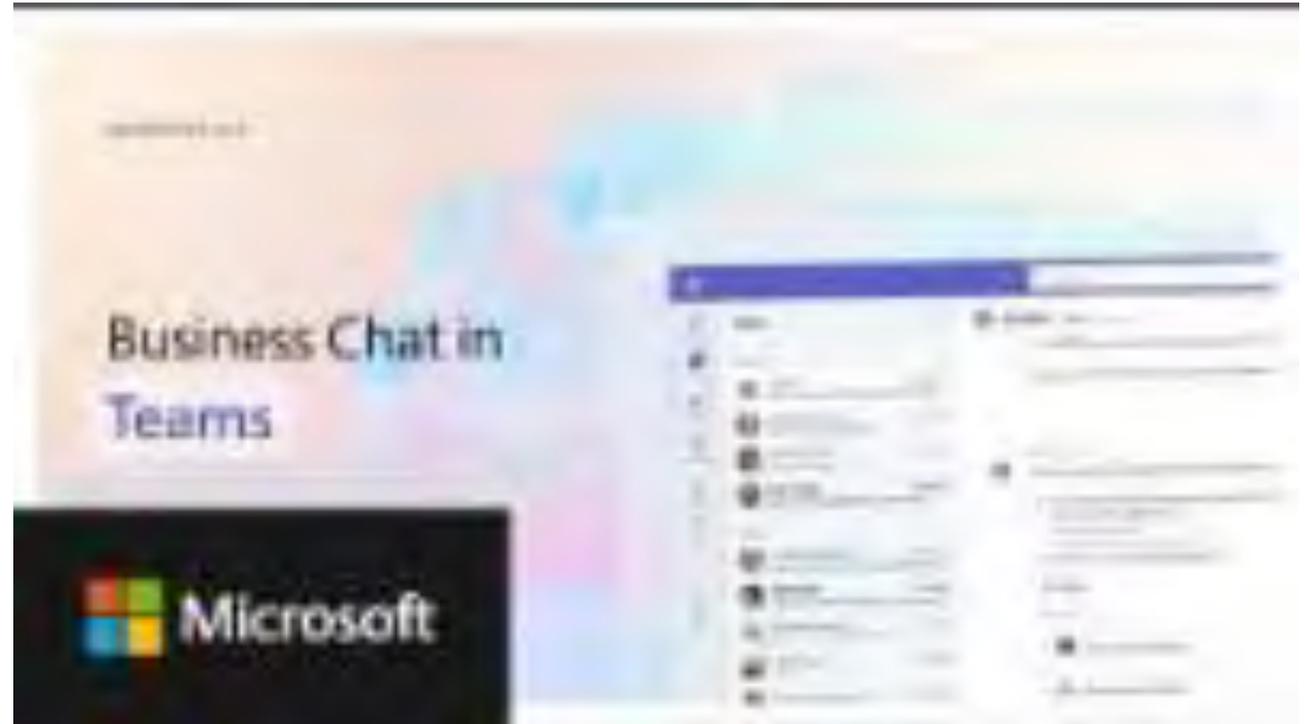
- Outlook
- Word
- PowerPoint
- Excel
- Viva Suite
- PowerApps und PowerApps Designer
- PowerBI
- Security Copilot
- uvm.



M365 Zugriff durch den Copilot

Zugriff auf die Daten des M365 Mandanten

- Teams Chats und Meetings
"Business Chat"
- SharePoint
- Dokumente, Excel Sheets, PowerPoints etc.
- Loop
- E-Mails, Kalender, Kontakte
- Viva Engage
- Microsoft Dataverse
- uvm.



Generative KIs | in Mainstream Software

Country	Customer	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts	Sales	COGS
France	VanArndel	Biodegradable Cleaning Products	Low	4345	\$10	\$352	\$1,529,640	\$15,294	\$1,514,146	\$
USA	VanArndel	Biodegradable Cleaning Products	Low	4245	\$10	\$352	\$1,494,240	\$14,942	\$1,479,298	\$
Australia	VanArndel	Biodegradable Cleaning Products	None	4124	\$10	\$352	\$1,451,648	\$0	\$1,451,648	\$
Mexico	VanArndel	Biodegradable Cleaning Products	Medium	3954	\$10	\$352	\$1,391,808	\$83,508	\$1,308,300	\$
France	Contoso Pharmaceuticals	Biodegradable Cleaning Products	Low	4902	\$10	\$127	\$628,904	\$6,289	\$622,615	\$
France	Contoso Pharmaceuticals	Recycled Can	Low	4903	\$5	\$127	\$622,681	\$6,227	\$616,454	\$
Canada	Proseware	Natural Personal Care Products	Low	1954	\$270	\$302	\$590,108	\$5,901	\$584,207	\$
Canada	Proseware	Biodegradable Cleaning Products	High	1990	\$220	\$302	\$600,980	\$84,137	\$516,843	\$
USA	VanArndel	Compostable Food Packaging	Low	1254	\$3	\$380	\$476,520	\$4,765	\$471,755	\$
USA	Proseware	Natural Personal Care Products	Medium	1465	\$224	\$302	\$442,430	\$26,546	\$415,884	\$
USA	Proseware	Sustainable Snack Packaging	Medium	1410	\$120	\$302	\$425,820	\$25,549	\$400,271	\$
Australia	Proseware	Natural Personal Care Products	Medium	1369	\$224	\$302	\$413,438	\$24,806	\$388,632	\$
USA	VanArndel	Compostable Food Packaging	Medium	1052	\$3	\$380	\$399,760	\$23,986	\$375,774	\$
Canada	VanArndel	Compostable Snack Packaging	Low	890	\$12	\$380	\$337,440	\$3,724	\$333,716	\$
Australia	Proseware	Sustainable Snack Packaging	Medium	1298	\$120	\$302	\$391,996	\$23,520	\$368,476	\$
Australia	VanArndel	Compostable Food Packaging	None	954	\$3	\$380	\$362,520	\$0	\$362,520	\$
Canada	Contoso Pharmaceuticals	Biodegradable Cleaning Products	Low	2785	\$110	\$127	\$353,695	\$3,537	\$350,158	\$

Copilot Chat

Good morning Kati

Here are some ideas that might help you get started:

- Get a summary of everything you need to catch up on, including meetings
- Find and use info that's buried in documents or lost in conversations
- Create content with all of it—drawing from your files, conversations, events, topics, and more

As your Copilot, I'm here to assist you but I do make mistakes, so sources are provided for your review when possible.

Click to add title

Click to add subtitle

Create content with Copilot

A job description for... 10 ideas for... A press release for... A project proposal about...

Describe what you'd like to create, and Copilot will generate a draft



Der “iPhone Moment”

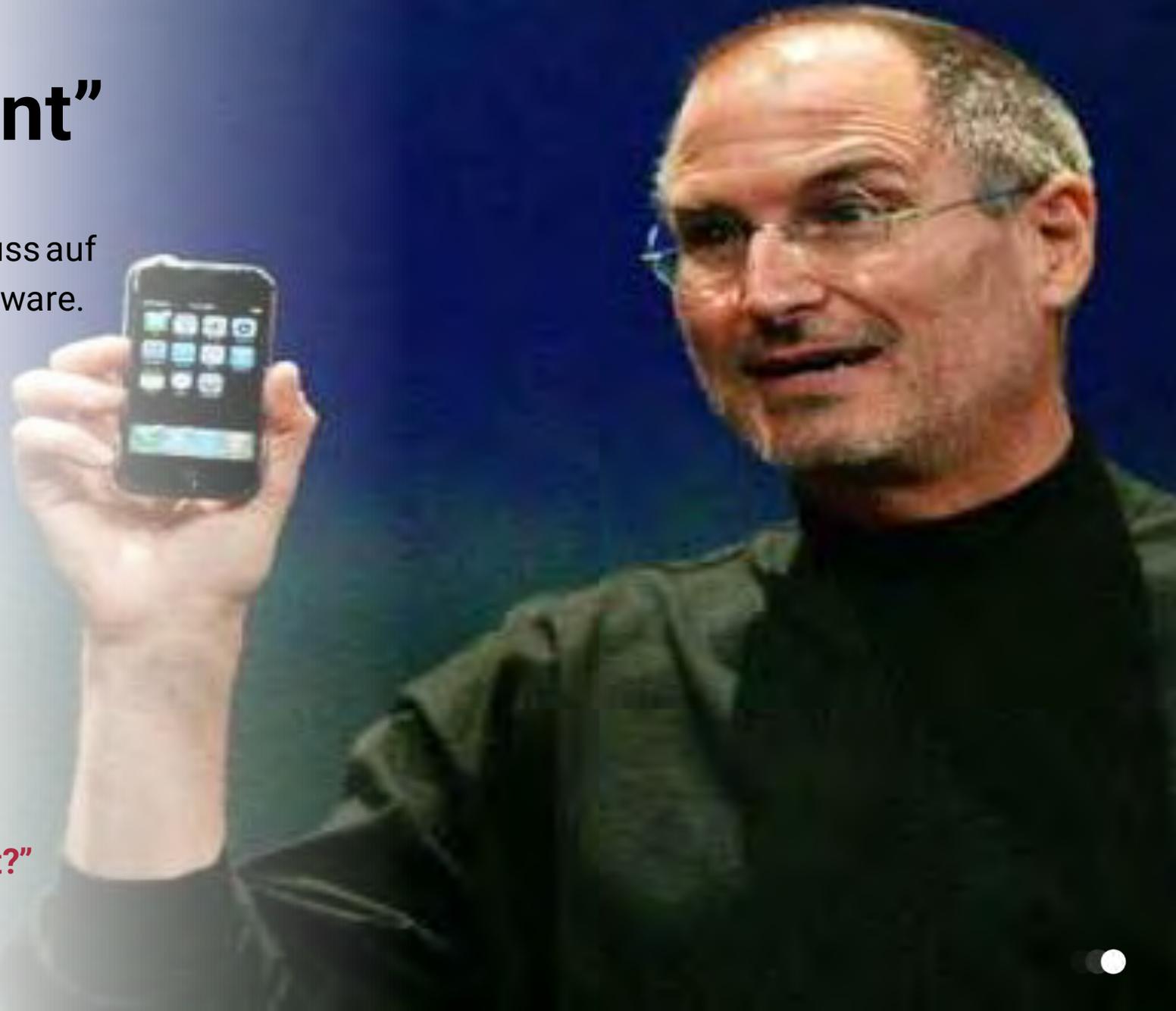
... hatte 2007 einen tiefgreifenden Einfluss auf die Entwicklung von Software und Hardware.

- Es wurden neue “**Standards**” gesetzt

User werden KI-Unterstützung in Zukunft erwarten!

- **Spoiler:**
Es dauert diesmal nicht so lange wie beim iPhone

“Warum kann Deine Software das nicht?”



Wann fange ich mit KI in meiner Software an?

- User werden es in Zukunft erwarten!
- Aktuelle Large Language Models haben noch Probleme
 - Phantasieren
 - teilweise nur online verfügbar
 - Datenschutz
 - Kosten
- Aber sie werden stetig besser und auch offline verfügbar werden
- Google empfindet "Open Source" als die größte "KI-Bedrohung"



Es macht Sinn, sich damit jetzt auseinanderzusetzen



... in eigener Software

Semantische Programmierung

	Syntax	Semantik
Natürliche Sprache	Regeln und Muster, wie Wörter und Sätze gebildet und strukturiert werden.	Bedeutung und Interpretation von Wörtern und Sätzen und wie sie sich auf die reale Welt beziehen.
Programmiersprachen	Regeln und Symbole, wie Code und Befehle geschrieben und strukturiert werden.	Bedeutung und Ausführung von Code und Befehlen und wie sie den Systemzustand beeinflussen.
Neues Mindset	<i>... weg von präzise und zerbrechlich</i>	<i>... hin zu umfassend und flexibel</i>

Wie können wir den Mehrwert von Sprachmodellen bezogen auf diese Flexibilität mit unseren eigenen Daten und Ressourcen nutzbar machen?

Semantische Programmierung

- Wie benutze ich ein Sprachmodell in meiner eigenen Software?
- Wer interpretiert den Wunsch des Users?
- Einen “**Copilot**” baut man nicht nur durch Aufrufen der API eines Sprachmodells
- Prinzip von Tools wie AgentGPT
(<https://agentgpt.reworkd.ai/de>)
Der User gibt ein Ziel vor, die KI ermittelt eine Strategie zur Erreichung!

Wir “plaudern” im Hintergrund mit dem Large Language Model über das Problem des Users!

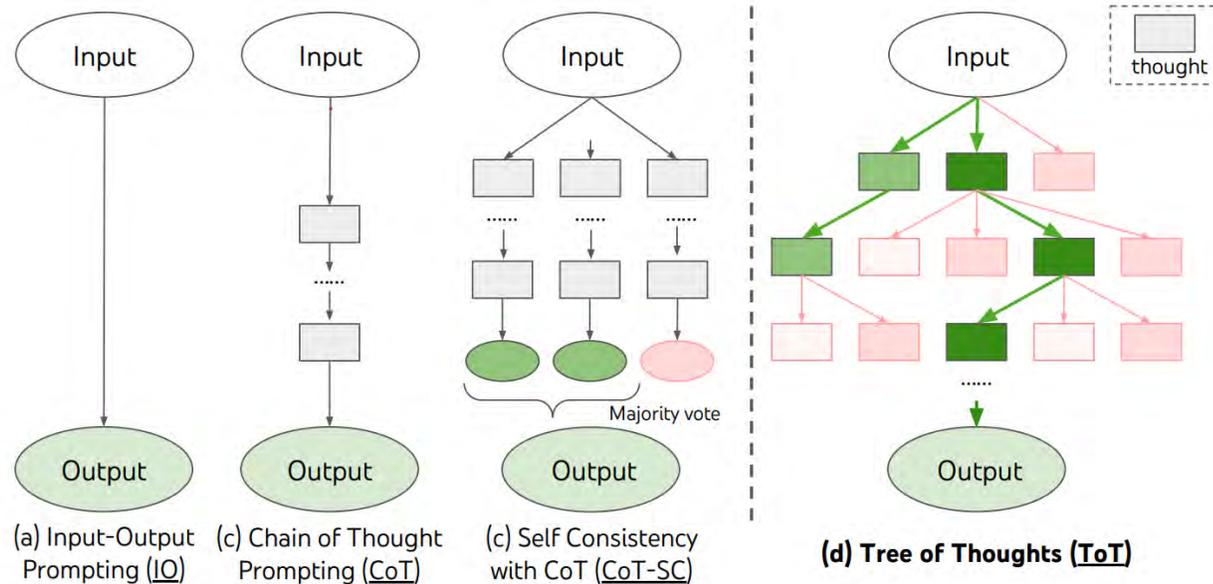
„Tree of Thoughts“

Deliberate Problem Solving with Large Language Models

Shunyu Yao, Princeton University
Dian Yu, Google DeepMind
Jeffrey Zhao, Google DeepMind
Izhak Shafran, Google DeepMind
Thomas L. Griffiths, Princeton University
Yuan Cao, Google DeepMind
Karthik Narasimhan, Princeton University

Erlaubt LLMs Denk- und Lösungswege zu erkunden.
Problem wird als Baumstruktur modelliert.

- 1. Zerlegung von Gedanken**
Zwischenprozesse werden in einzelne Denkschritte zerlegt
- 2. Gedanken Generator**
Strategien werden vorgeschlagen, um Kandidaten für den nächsten Denkschritt zu generieren
- 3. Bewertung von Zuständen**
Zustandsbewerter bewerten den Fortschritt in Bezug auf die Problemlösung
- 4. Suchalgorithmus**
(nach den erfolgversprechendsten Gedanken)
Unterschiedliche Suchalgorithmen,
je nach Struktur des Baumes



ToT ist ein Rahmenwerk, das Sprachmodelle dazu befähigt, autonomer und intelligenter Entscheidungen zu treffen und Probleme zu lösen.

Semantic Kernel (SK)

<https://github.com/microsoft/semantic-kernel>

- **"Best-Guess Practices"** (!) zur Integration von LLM AIs
- **Early Alpha** / Ongoing Development
- Lightweight SDK

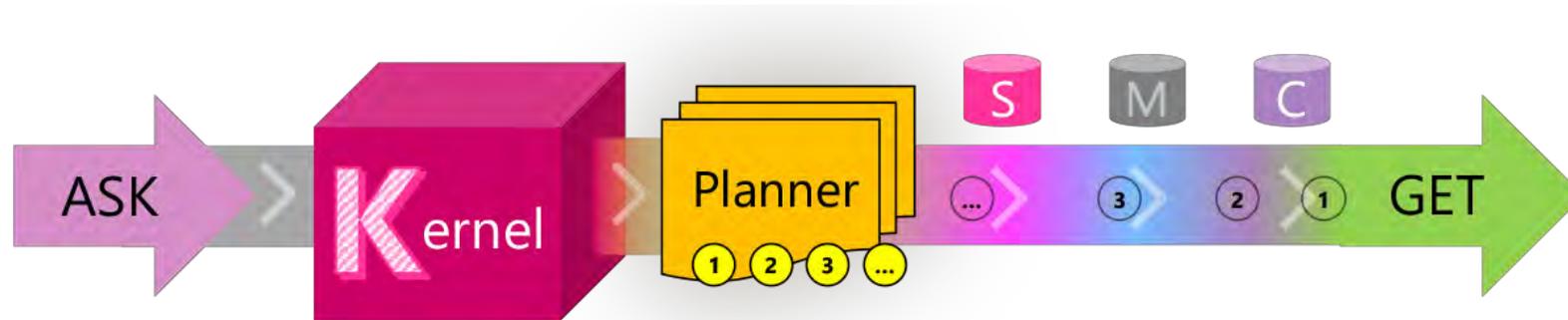
- Vereint konventionelle Programmiersprachen und Large Language Model (LLM) AI Prompts
- Unterstützt aktuell Text-Completion und Text-Embedding-Modelle
- Out-of-the-Box Unterstützung für:
 - OpenAI API
 - Azure OpenAI API

Semantic Kernel (SK)

Grundkomponenten von LLMs im Semantic Kernel

- **Modelle:**
Grundlage für Textgenerierung und Aufgabenbewältigung
- **Prompts und Prompt Templates:**
Eingabeaufforderungen, die die AI-Modelle leiten und vordefinierte Prompts mit Platzhaltern
- **Skills / Semantic Functions:**
Einzelne Funktion oder eine Gruppe von Funktionen um eine bestimmte Funktionalität dem Kernel zur Verfügung zu stellen
- **Connectors:**
Erlauben den Zugriff auf externe APIs, Echtzeitdaten und weitere Ressourcen
- **Embeddings:**
Vektorrepräsentationen von Tokens
- **Vector Databases (Memories):**
Speicherung und Verwaltung von Embeddings
Effiziente Suche und Abruf von relevanten Informationen

Semantic Kernel - Planner

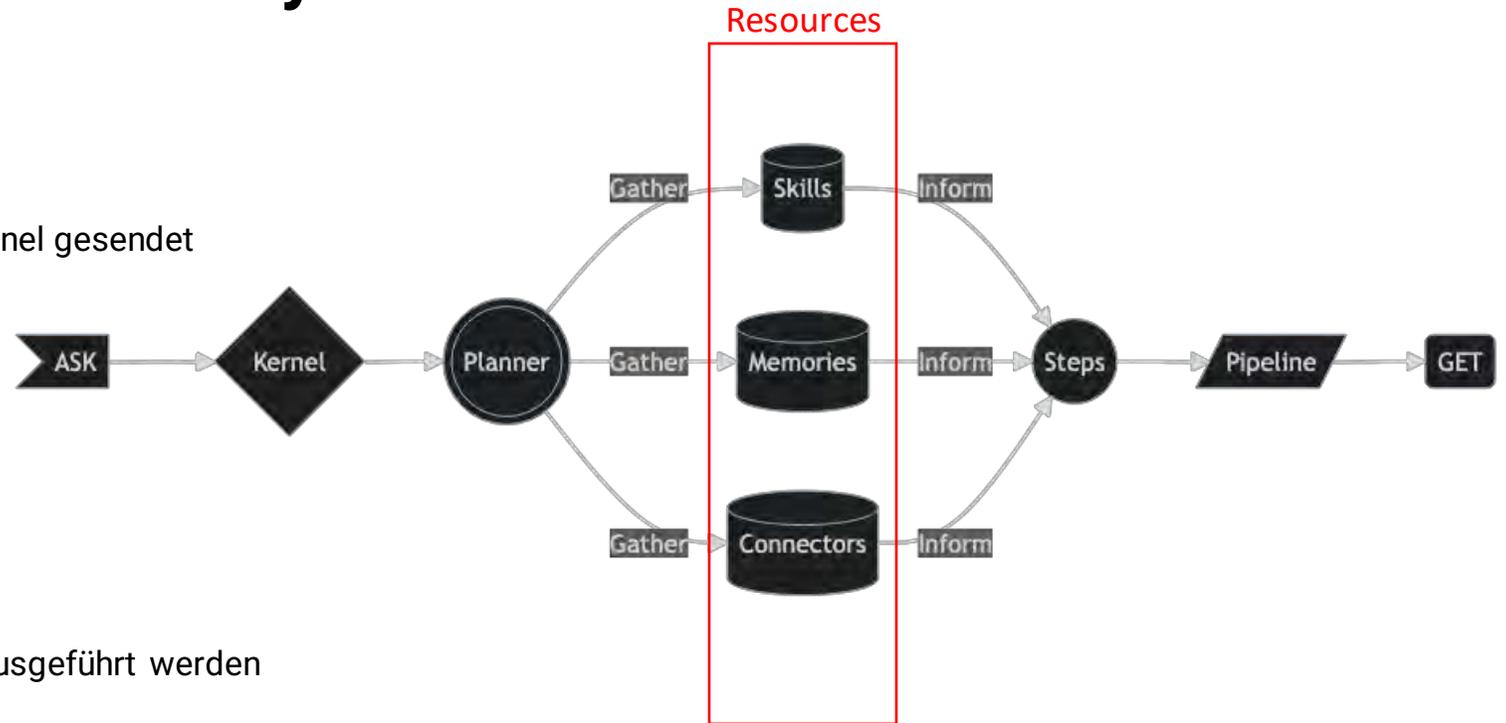


- Planner teilt eine große Aufgabe in viele kleine
- Planner verfolgt das Ziel des Users durch eine optimale Abfolge von Schritten zur Zielerreichung
- Planner arbeitet zielorientiert und arbeitet vom zu erwartenden Ziel rückwärts um Schritte dazu zu ermitteln
- Eine “Black Box” für den Entwickler
- **“Einmal-Werkzeug”**, um ein bestimmtes Ziel zu erreichen

- Der Source Code deutet darauf hin, dass der Planner auf mehreren Ebenen arbeiten und auch mit Teilzielen und Aktionen umgehen können wird.

Semantic Kernel - User Journey

- **ASK**
Wunsch des Users wird als "ASK" an den Kernel gesendet
- **Kernel**
... orchestriert den "ASK" des Users
- **Planner**
... erzeugt einzelne Steps auf Basis der verfügbaren Resources
- **Resources**
Verfügbare Skills, Memories und Connectors
- **Steps**
Ein Plan besteht aus einer Serie von Steps, ausgeführt werden
- **Pipeline**
Das Ausführen der Steps erfüllt den Wunsch des Users
- **GET**
Der User bekommt seinen Wunsch erfüllt.
- **Wir entwickeln in der Regel dabei nur die Ressourcen!**



Semantic Kernel - Resources

- Wissen über die eigene Software liegt beim Entwickler
- Welche Daten gibt es?
- Wo kommen die Daten her?

- Welche Skills und Connectors brauche ich?
 - z.B. Microsoft Graph API, Datenbankabfragen, APIs
- Wie speichere ich „Memories“ und in welcher Form?

Kann man den MS Semantic Kernel schon sinnvoll nutzen?



„Let's build a Copilot“

Copilot in einer Demo Applikation mit Semantic Kernel

- **Konzepte und Demo-Code**
MS Semantic Kernel
am Beispiel von C# Workbooks
- **Ausblick**
Demo App mit Copilot Prototypen
(auf Basis text-davinci-003 Modell von Azure OpenAI)

Wrap-up

- Generative KI wird in Software zukünftig erwartet werden
- SDKs sind aber noch Early Alpha
- Andere SDKs:
z.B. **LangChain** (Python SDK)
- „Mindset Shift“
- Experimentieren lohnt sich
- Kosten:
Ein ganzer Tag Debugging für Prototyp Copilot gegen Azure Open API **“text-davinci-003”** und **“gpt-35-turbo”**
ca. € 11
- Eigene Erfahrungen machen
- Auf Erfahrungen anderer zurückgreifen
z.B. Sam Schillace, MS und seine “Laws”



“Schillace Laws”



Sam Schillace, CVP, Deputy CTO, Microsoft

- **1. Nutze das Modell anstelle von Code**
Schreibe keinen Code, wenn das Modell dies übernehmen kann.
Das Modell wird sich mit der Zeit verbessern, der Code jedoch nicht.
Nutze die Fähigkeiten des Modells zur Planung und Verständnis der Absicht, um hochwertige Programme zu erstellen.
- **2. Tausche eine „Hebelwirkung“ gegen Präzision**
Nutze das LLM so effizient wie möglich, um eine „Hebelwirkung“ zu erzeugen.
Es ist möglich, sehr allgemeine Muster zu erstellen, die mit einfachen Textaufforderungen parametrisiert werden können, um wertvolle und differenzierte Ergebnisse zu erzeugen, wie „Erzeuge einen Report aus der Datenbank“ mit zusätzlicher Parametrisierung.
- **3. Code für Syntax und Prozess / Modelle für Semantik und Absicht**
Code sollte für spezifische Berechnungen und Prozesse verwendet werden, während Modelle besser für das Verständnis von Bedeutung und Zielen geeignet sind.
Die Grenzen zwischen Syntax und Semantik sind die schwierigsten Teile dieser Programme.

“Schillace Laws”



Sam Schillace, CVP, Deputy CTO, Microsoft

- **4. Das System ist so zerbrechlich wie sein zerbrechlichster Teil**
Vermeide unnötiges Hardcoding, um Flexibilität zu gewährleisten.
Nutze so viel Logik und Flexibilität in den Prompts wie möglich und verwende imperativen Code nur minimal.
- **5. Stelle kluge Fragen, um kluge Antworten zu bekommen**
LLM-Modelle sind sehr leistungsfähig, aber sie benötigen präzise Fragen, um detaillierte Antworten zu liefern.
„Garbage in, Garbage out“ gilt auch im Zeitalter der KI.
- **6. Unsicherheit ist eine „Exception“**
Bei Unsicherheit über die Absicht sollte der User eingebunden werden.
Unsicherheit sollte wie eine Exception behandelt werden und an eine höhere Ebene weitergegeben werden, die Klärung bringen oder mit dem User interagieren kann.

“Schillace Laws”



Sam Schillace, CVP, Deputy CTO, Microsoft

- **7. Text ist das universelle Übertragungsprotokoll**
Da LLMs gut in der Analyse von natürlicher Sprache und Absicht sind, ist Text das ideale Format für die Übermittlung von Anweisungen. Natürliche Sprache ist zwar weniger präzise, aber für die meisten Anwendungen weniger zerbrechlich als strukturiertere Sprachen.
- **8. Was für dich schwierig ist, ist auch für das Modell schwierig**
Aufgaben, die für einen Menschen schwierig sind, sind wahrscheinlich auch für das Modell schwierig. Es ist oft hilfreich, komplexe Aufgaben in einfachere Schritte zu unterteilen, um stabilere Ergebnisse zu erzielen. Z.B. kann es Sinn machen zunächst nach der Erklärung eines Ergebnisses zu fragen und im zweiten Schritt nach dem Ergebnis selbst.
- **9. Vorsicht vor „Fehldeutung des Bewusstseins,, / das Modell kann gegen sich selbst verwendet werden**
Es ist einfach, sich ein "Bewusstsein" in einem LLM vorzustellen, aber es gibt bedeutende Unterschiede zur menschlichen Denkweise. Da Modelle keine Erinnerung an vorherige Interaktionen haben, können sie gegen sich selbst eingesetzt werden. Sie können beispielsweise dazu verwendet werden, nach Fehlern oder schädlichem Code zu suchen, selbst wenn sie diesen Code erstellt haben. Sie sind nicht in der Lage, uns "anzulügen", da sie nicht wissen, wo der Code ursprünglich herkommt.



Weitere Informationen

MS Video The Future of Work With AI - Microsoft March 2023 Event	The Future of Work With AI - Microsoft March 2023 Event – YouTube
MS Semantic Kernel GitHub Repo	microsoft/semantic-kernel: Integrate cutting-edge LLM technology quickly and easily into your apps (github.com)
MS Semantic Kernel Discord Channel	https://aka.ms/sk/discord
MS Learn What is Semantic Kernel?	What is Semantic Kernel? Microsoft Learn
MS Semantic Kernel Announcing Copilot Chat	Announcing Copilot Chat Semantic Kernel (microsoft.com)
GitHub Copilot	GitHub Copilot · Your AI pair programmer
MS Azure Request Access to Azure OpenAI Service	Request Access to Azure OpenAI Service (microsoft.com)
Tree of Thoughts Deliberate Problem Solving with Large Language Models	[2305.10601] Tree of Thoughts: Deliberate Problem Solving with Large Language Models (arxiv.org)
LangChain Projekt	Concepts –   LangChain 0.0.177





Diskussion & Fragen



Mischa Hüschen
Managing Director
prodot GmbH

mischa.hueschen@prodot.de



Michael Damatov
Principal Software Engineer
prodot GmbH

michael.damatov@prodot.de