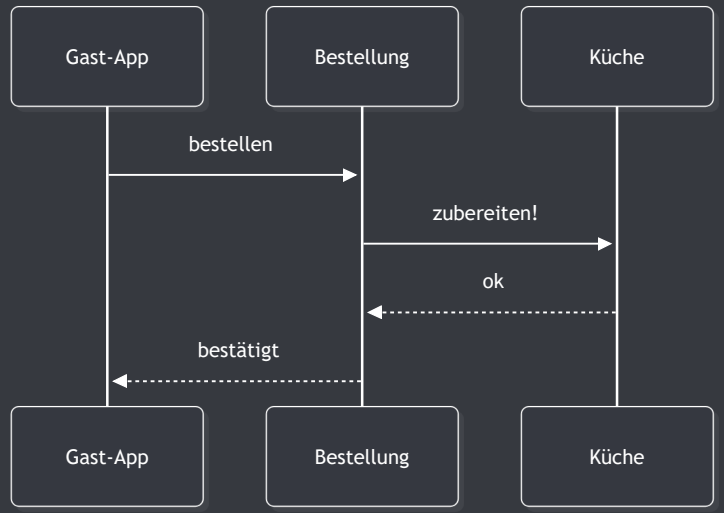
Microservices und Event-getriebene Architektur (MEA)

WASP1 im Wintersemester 2026/27

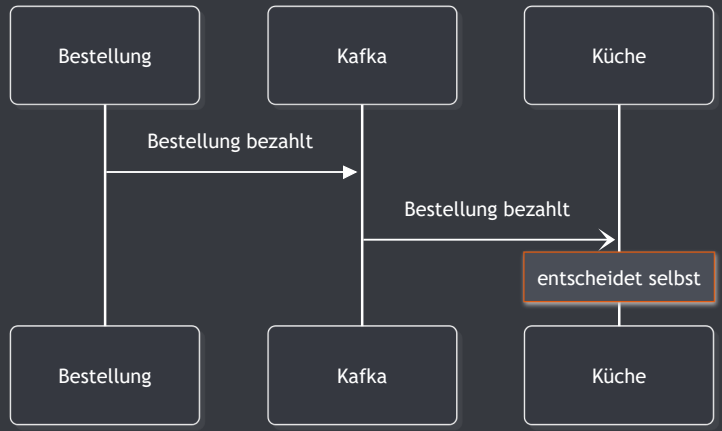
Worum es geht

In ST2



- Ein Modul ruft das andere, wartet auf Antwort
- Module sind (relativ) eng gekoppelt

In MEA



- Jeder Service meldet, was passiert ist
- Niemand wartet
- Services können sich "auf sich selbst" konzentrieren

Eine andere Art, über Abläufe nachzudenken

Ein Aufruf befiehlt

- "Küche, bereite das Essen zu!"
- Der Sender weiß, wer reagiert
- Fällt der Empfänger aus, steht alles

Ein Event meldet eine Tatsache

- "Bestellung bezahlt"
- Der Sender weiß nicht, wer reagiert
- Jeder Service entscheidet selbst, was er damit anfängt

Diese Denkweise erfordert Umdenken - die Gewöhnung daran ist eines der Lernziele dieses Moduls. Außerdem ist der technische Aufwand deutlich höher. Kafka, Container und Pipelines sind Werkzeuge dafür.

Microservices: vom Hype zum Handwerk

Thoughtworks Technology Radar

- 03/2012: *Assess*, beobachtenswert
- 10/2012 bis 01/2014: *Trial*, lohnt sich im Projekt

"We are seeing an uptick in adoption of microservices as a technique for distributed system design, both in Thoughtworks and in the wider community." (01/2014)

Gartner Hype Cycle, Juli 2020

- Microservices im "Trough of Disillusionment": Die Erwartungen sind auf das reale Maß zurückgekommen
- Event-Driven Architecture dagegen kurz vor dem "Peak of Inflated Expectations"

Microservices sind heute kein Versprechen mehr, sondern ein Architekturstil mit bekannten Stärken und bekannten Kosten. Genau die lernen Sie in MEA einzuschätzen.

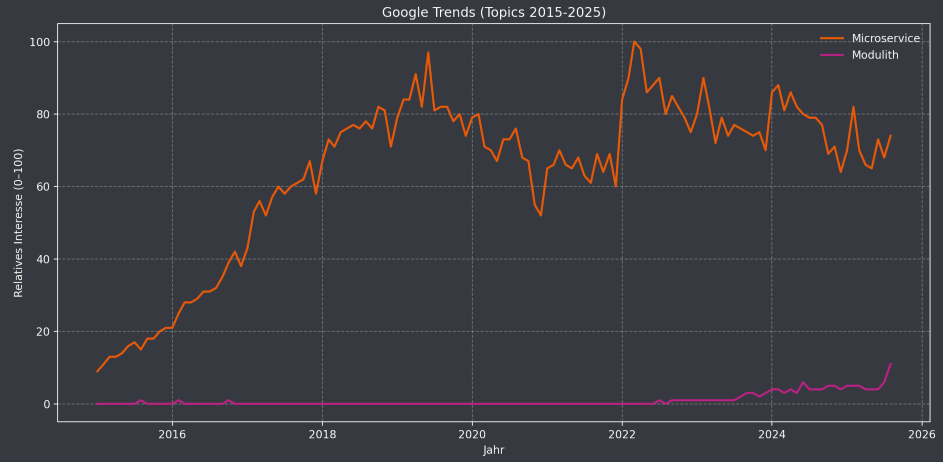
Quellen: thoughtworks.com/radar/techniques/microservices; Gartner, Hype Cycle for Application Architecture and Development, 2020 (G00450406)

Die Gegenposition: erst der Modulith

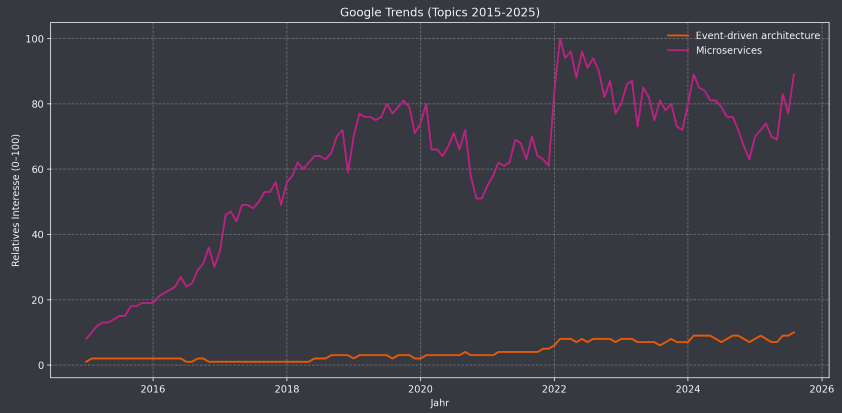
"Rather than start a new system with a collection of separately deployed processes, it's often advisable to start with a well-factored monolith and only break out separately deployable units when the application reaches a scale where the benefits of microservices outweigh the additional complexity inherent in distributed systems."

Thoughtworks Technology Radar, Spring Modulith, 09/2023 (*Assess*)

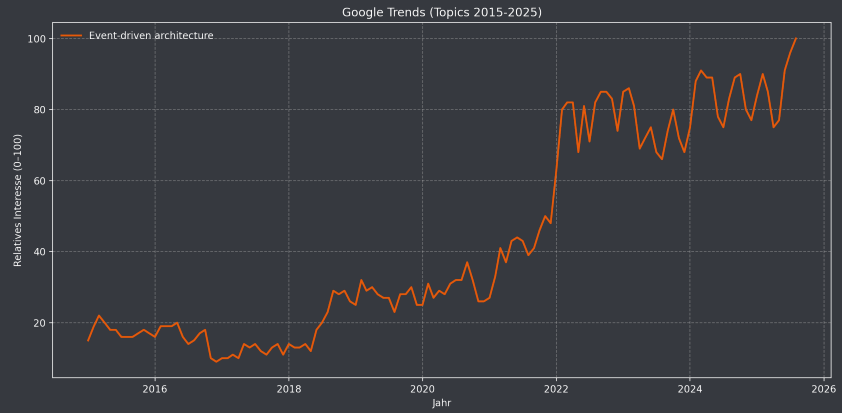
- Auch das ist eine gute Antwort
- Deshalb endet MEA mit der Frage: Wann ist dieser Stil etwas für Sie, und wann nicht?



Event-Driven Architecture: Nische, aber wachsend



Im Vergleich zu Microservices ein Nischenthema …



… aber für sich betrachtet seit 2018 deutlich wachsend

Daten: Google Trends, Topics, 01/2015 bis 08/2025. Der Sprung im Januar 2022 betrifft alle Kurven: Google hat zu diesem Datum die Datenerhebung umgestellt. Der Anstieg davor und danach ist echt.

Die Fallstudie

Das Restaurant-System, das Sie aus ST2 kennen, erweitert um Abläufe, die ohne Events nur mühsam zu bauen wären. Welche es werden, entscheiden wir am EventStorming-Tag.

Kandidaten:

- **Warmhalteboxen:** Die Küche stellt ein fertiges Gericht in eine Box, und der Gast holt es dort ab, ohne die Köchin oder den Koch zu treffen. In welcher Box liegt mein Essen, und wie finde ich sie?
- **Lager:** Zutaten werden verbraucht, gehen zur Neige und müssen nachbestellt werden
- **Mehr Realismus:** Heute müssen zum Beispiel alle Gäste eines Tisches gleichzeitig gehen, was im echten Restaurant niemand tut

Ihren ST2-Code dürfen Sie als Ausgangspunkt verwenden.

Claude Pro für Ihre Projektarbeit

Für die Zeit Ihrer Arbeit an diesem Projekt bekommen Sie kostenlos Claude Pro, und damit auch Claude Code.

- Möglich durch den Anthropic Team Plan for Scientists
  - für 10.000 Wissenschaftler weltweit, ich bin einer davon ;-)

- Wie man im Team mit Agentischer KI arbeitet, ist (Teil-)Thema der Veranstaltung
  - angelegt im Grundlagenteil
  - vertieft im Projektteil

Fünf Phasen



Grundlagenteil, 5 ECTS

Phasen 1 bis 3, Mitte Oktober bis Ende November. Endet am 26.11. mit einem Architekturkonzept, einem laufenden Walking Skeleton und einer Reflexion.

Projektteil, 10 ECTS

Phasen 4 und 5, Ende November bis Mitte Februar. Bis Anfang Januar wird aus dem Konzept ein laufendes Restaurant-System, danach geht es in den Microservice Dungeon.

Nach dem Grundlagenteil können Sie mit eigener Note aufhören. Der Projektteil setzt den Grundlagenteil voraus.

Phase 1: Domänen-Exploration

15.10.2026

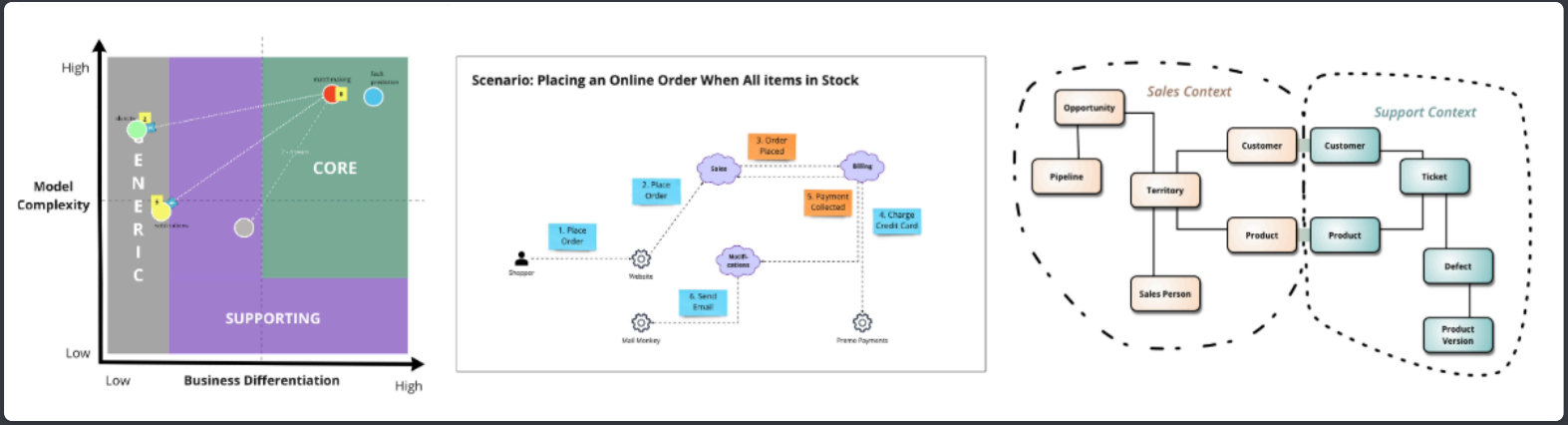
- Ein ganzer Tag EventStorming
- Wir erkunden die Restaurant-Domäne als Zeitleiste von Events
- Ich moderiere und zeige Ihnen dabei, wie die Methode funktioniert
- Ergebnis: die Domäne auf einem Board, mit ersten Kandidaten für Grenzen



Phase 2: Bounded Contexts und Event-Contracts

22.10.2026

- Wo verlaufen die Grenzen zwischen den Bounded Contexts?
- Aus jedem Bounded Context wird der Service eines Teams
- Wir bilden die Teams
- Hausaufgabe: ein erster AsyncAPI-Contract für Ihren Service, mit dem Agenten entworfen



Phase 3: Architekturkonzept und Walking Skeleton

29.10.2026

- Der Umfang des Walking Skeletons: ein reduzierter Ablauf der Fallstudie
- Aus den AsyncAPI-Contracts wird der Event-Katalog des Systems
- Clients: welche gibt es, und wer ist zuständig?
- Entwickeln und Testen ohne Infrastruktur

05.11.2026, dann drei Wochen Teamarbeit bis 26.11.2026

- Backend for Frontend, Kafka-Hilfsbibliothek, Pipeline, Docker Compose und die lokale Umgebung
- Jeder Service kommt über seine Pipeline in die gemeinsame Umgebung und tauscht dort sein erstes Event aus
- Statusrunden per Video am 12.11. und 19.11., jeweils 18:00 bis 19:00
- Am Vormittag des 26.11. führen die Teams ihr Walking Skeleton vor
- Zum Abschluss des Grundlagenteils reflektiert jede:r, wie sich das Denken in Events und asynchroner Kommunikation "angefühlt" hat

Phase 4: Implementierung und Betrieb

26.11.2026 (nachmittags) - 10.12.2026 - 17.12.2026 - 05.01.2027 (vormittags)

- Die Teams bauen ihre Services aus und betreiben sie von Anfang an
- Entwicklung und Betrieb laufen miteinander, nicht nacheinander
- Arbeiten mit dem Agenten im Team: Architekturregeln als Prüfungen in der Pipeline, und was ein Mensch vor dem Merge liest
- Workshops etwa alle zwei Wochen: Input, gemeinsames Coding vor Ort, Integration am Ende des Tages
- Integrationstag am 17.12.2026, Abschlussvorführung des ganzen Restaurant-Systems am 05.01.2027

Phase 5: Microservice Dungeon

05.01.2027 (nachmittags) - 21.01.2027 - 04.02.2027 - 18.02.2027

- Microservice-Entwicklung in einer Landschaft, die es schon gibt
- Sie reagieren auf Events, die andere entworfen haben
- Sie entwickeln einen Player mit einer eigenen Strategie
- Codefight: Wer spielt am besten? Rein sportlich.
- Zum Schluss: Ist dieser Architekturstil etwas für Sie?

Was ist der Microservice Dungeon?

- Ein Spiel, das komplett aus Microservices besteht, die über Events kommunizieren
- Ihr Player ist ein weiterer Service darin und steuert Roboterschwärme
- Dokumentation: docs.microservice-dungeon.de



Bewertung

Grundlagenteil (5 ECTS)

Kriterium	Gewicht
Schnitt des Systems (gemeinsame Note für alle)	1
Event-Contracts (pro Team)	2
Infrastrukturkonzept und Walking Skeleton	4
Reflexion	1
Beteiligung	1
Summe	**9**

Projektteil (10 ECTS)

Kriterium	Gewicht
Service	4
Pipeline und Betrieb	2
Arbeit mit dem Agenten	2
Player: Strategie und Features	4
Player: Architektur und Qualität	2
Player: Deployment und Codefight	2
Abschlussreflexion	1
Beteiligung	1
Summe	**18**

- Jedes Kriterium mit 0 bis 3 Punkten
  - dann gewichtetes Mittel
  - ab 2,75 Punkte = 1,0, 1 Punkt = 4,0, und dazwischen interpoliert
- zwei getrennte Teilnoten
- Entwurf, die genaue Tabelle folgt zum Start
- Planen Sie im Schnitt etwa 2 … 2,5 Tage pro Woche ein (wenn Sie beide Teile machen)

Termine

Donnerstags (eine Ausnahme), ganztägig vor Ort. Die Termine sind noch vorläufig.

Teil	Datum	Phase	Workshop
G	Do 15.10.2026	1	Kickoff und EventStorming
G	Do 22.10.2026	2	Bounded Contexts, Context Map und Repository-Struktur
G	Do 29.10.2026	3	Einstieg in die Implementierung
G	Do 05.11.2026	3	Services verbinden
G	Do 12.11.2026	3	Statusrunde per Video, 18:00 bis 19:00
G	Do 19.11.2026	3	Statusrunde per Video, 18:00 bis 19:00
G/P	Do 26.11.2026	3/4	Abschluss Grundlagenteil, danach Start Projektteil: Agentic Development im Team
P	Do 10.12.2026	4	Eventual Consistency, Robustheit
P	Do 17.12.2026	4	Integrationstag
P	Di 05.01.2027	4/5	Abschluss Restaurant & Kickoff Dungeon
P	Do 21.01.2027	5	Fortgeschrittene DevOps- und Hosting-Konzepte, Infrastructure as Code (geplant)
P	Do 04.02.2027	5	Monitoring (logs, metrics) via Prometheus und Grafana (geplant)
P	Do 18.02.2027	5	Abschluss-Codefight und Reflexion

Teams

- Jedes Team setzt einen Bounded Context als Service um
- Anzahl und Größe der Teams hängen davon ab, wie viele Sie sind
- Jedes Team ist für seinen Service selbst verantwortlich, auch für Pipeline und Betrieb
- Kein DevOps-Team für alle: In jedem Team übernimmt eine Person die Infrastruktur federführend
- Für den Dungeon: dieselben Teams oder neu gemischt? Das entscheiden wir gemeinsam, wenn es so weit ist

Das technische Setting

- Die Kafka-Infrastruktur stellen wir Ihnen bereit
- Für Java gibt es Hilfsbibliotheken, zum Beispiel für Consumer und Producer
- Eine andere Sprache ist erlaubt. Dann bauen Sie die Hilfsfunktionen selbst nach
- Der Event-Katalog ist der Contract, an den sich alle Services halten
- Laufzeitumgebung: Docker Compose auf einem gemeinsamen Server

Fragen?

archi-lab.io