Engineer

Architektur

wie die Teile eines Systems zusammenpassen und warum

Was es ist

Architektur, in der Software auch Software- oder Systemarchitektur genannt, umfasst die grundlegenden Konzepte und Eigenschaften eines Systems in seiner Umgebung. Sie drücken sich in seinen Elementen, deren Beziehungen und den Prinzipien aus, die Entwurf und Weiterentwicklung bestimmen. Diese Formulierung folgt NIST SP 800-160 Volume 1 Revision 1, Engineering Trustworthy Secure Systems, veröffentlicht im November 2022.

Jeder Teil der Definition erfüllt eine Aufgabe. Elemente sind die Bausteine: Dienste, Datenbanken, Warteschlangen, Frontend und die Module darin. Beziehungen beschreiben Abhängigkeiten und das, was zwischen den Teilen fließt. Prinzipien sind Regeln dafür, wohin neue Arbeit gehört. Sie verhindern, dass Elemente und Beziehungen mit dem Wachstum abdriften. „In seiner Umgebung“ macht Nutzer, Nachbarsysteme und Einschränkungen zum Teil der Architektur. „Weiterentwicklung“ bedeutet, dass eine nur am Tag ihrer Zeichnung richtige Beschreibung unvollständig ist.

Nicht jede Entwurfswahl ist eine Architekturentscheidung. Die Gruppe hinter dem Format Architectural Decision Record (ADR) nennt sie „eine begründete Entwurfswahl, die eine architektonisch bedeutsame funktionale oder nicht funktionale Anforderung behandelt“. Bedeutsam ist eine Anforderung, wenn ihr Einfluss auf Architektur und Qualität messbar ist. Die arc42-Vorlage sagt es direkter: Ihr Entscheidungsteil ist für „wichtige, teure, riskante oder umstrittene Entscheidungen“ gedacht.

Weil keine Zeichnung alles zeigt, wird Architektur in Sichten beschrieben. Simon Browns C4-Modell bietet Diagramme wie Karten mit verschiedenen Zoomstufen: einen Systemkontext für Nutzer und Nachbarsysteme, ein Containerdiagramm für Anwendungen und Datenspeicher, ein Komponentendiagramm für einen

Container und, wo hilfreich, ein Codediagramm. arc42 gliedert die schriftliche Beschreibung in zwölf Abschnitte, von Zielen und Randbedingungen bis zu Entscheidungen, Risiken und technischen Schulden.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Architektur?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Die Struktur eines Systems wird zuerst auf dem Papier durchdacht, damit die Teile zusammenpassen und die Kosten feststehen, bevor die erste Zeile Code entsteht.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Ein Infrastrukturplan für 14 Abteilungen — 14 Forschungsabteilungen mit einem gemeinsamen Infrastrukturplan

Ein Infrastrukturplan für 14 Abteilungen Kontakt aufnehmen →

Warum es wichtig ist

Die Architekturfrage eines Teams lautet weniger, welche Struktur es wählt, als welche Entscheidungen es jetzt bewusst und dokumentiert trifft und welche es offenlässt.

Eine früh geklärte Strukturfrage kostet, bevor viel bekannt ist. Die Entscheidung fällt mit den wenigsten Informationen, die das Projekt je haben wird, und verlangsamt die ersten Wochen. Bleibt sie offen, entsteht die Struktur trotzdem: durch jeweils bequeme Änderungen der Person, die zuletzt den Code berührt. Der Ausweg trennt zwei Arten von Entscheidungen. Einige lassen sich nur noch teuer ändern, sobald Code davon abhängt: die Zerlegung des Systems, die Zuständigkeit für Geschäftsregeln, der Datenzugriff und die Hürde, die jede Änderung nehmen muss. Diese sollten früh und bewusst fallen. Der größte Teil des Rests lässt sich günstig überdenken und darf warten, bis der Code seinen Bedarf zeigt.

Aufschreiben ist die zweite Hälfte der Entscheidung. Dort überlebt die Begründung. Ein ADR hält eine Entscheidung mit Gründen, Abwägungen und Folgen fest. Die ADR-Sammlung bildet das Entscheidungsprotokoll des Projekts. Die ADR-Website schreibt die Verbreitung der Idee Michael Nygards Beitrag „Documenting Architecture Decisions“ von 2011 zu. Der Aufwand beträgt eine Seite pro Entscheidung. Dafür kann ein späterer Leser eine bewusste Abwägung von einem Zufall unterscheiden und prüfen, ob der Grund noch gilt.

Qualitätsziele entscheiden zwischen mehreren funktionsfähigen Strukturen. arc42 verlangt die drei wichtigsten, höchstens fünf Qualitätsziele, die den maßgeblichen Beteiligten besonders wichtig sind. Sie sind ausdrücklich nicht die Projektziele. Zwei Entwürfe mit denselben Funktionen können sich bei Ausfall, Skalierung und Änderung völlig unterscheiden. Erst die Qualitätsziele bestimmen, welcher Unterschied zählt.

Wie es in der Praxis aussieht

Das Repository der Plattform begann am 25. November 2025 leer. Die Seite über ihre grundlegenden technischen Entscheidungen argumentiert, solche Entscheidungen seien „am dritten Tag billig und im sechsten Monat kaum umkehrbar“. Zwei der damaligen drei prägten die Struktur. Jede Schicht erhielt eine Aufgabe: Next.js im Frontend, eine validierende und weiterleitende Go-API und PostgreSQL-Funktionen mit den Geschäftsregeln. Der Datenzugriff lief von Anfang an über gespeicherte Funktionen, weil ein späterer Einbau „eine Neufassung jedes Handlers erfordert hätte“. Neun Monate und mehrere Tausend Commits später berichtet die Seite von weiterhin klaren Schichten und keinem direkten Tabellenzugriff aus einem Handler. Eine Gründungsentscheidung wird daran gemessen, ob sie das folgende Arbeitsvolumen übersteht, nicht daran, ob sie gut klang.

Bleibt die Struktur dem Zufall überlassen, wird sie durch unbeabsichtigte Entscheidungen festgelegt. Das zeigt die Entwicklung einer GIS-Karte. Die Teile waren einzeln gewachsen: Geodaten in PostgreSQL, Darstellung in Mapbox, ReactJS vorn und NodeJS hinten. Jeder Teil funktionierte für sich; sie waren „nur nicht so gebaut worden, dass sie zusammenpassen“. Die Rolle umfasste Full-Stack-Entwicklung und technische Leitung. Zuerst wurde die Architektur festgelegt. Danach wurden Zuständigkeiten „immer wieder in ihre richtige Schicht zurückgeschoben, sobald Aufgaben aus einer Schicht in die nächste übergriffen“. Viel Arbeit bestand darin, „kleine Unstimmigkeiten aufzufangen, bevor sie zu Architektur verhärteten“.

Die Seite zur Container-Ebene betrachtet die schriftliche Begründung als eigentliches Ergebnis. Die branchenübliche Containerengine veröffentlicht Ports vor den Regeln der Host-Firewall. Deshalb verwendet die Ebene eine daemonlose Engine mit Units, die der eigene Dienstmanager des Betriebssystems erzeugt, und veröffentlicht überhaupt keine Ports. Eine Abwägung wird ausdrücklich festgehalten: Die Units laufen auf Systemebene statt rootless. Das hält die Automatisierung einfach, geht aber zulasten der strengeren Isolation, die der Rootless-Betrieb bieten würde. Die Notiz

beschreibt, wohin gewechselt werden sollte, wenn Container-Ausbrüche wichtiger werden. Genau das enthält ein ADR: Entscheidung, Begründung, Abwägungen und Folgen. Die Seite bewertet das Aufschreiben samt Grund höher als die Wahl selbst, denn „jeder Artikel wird der nächsten Person raten, den Standard zu verwenden“.

Wo es schiefgeht

Ein sorgfältiger Neuzugang macht eine bewusste Entscheidung rückgängig. Der Grund stand nur in einem Kopf. Deshalb wirkt die Entscheidung wie ein Fehler und der von jedem Tutorial empfohlene Standard wie die Reparatur. Schreiben Sie für schwer umkehrbare Entscheidungen ein ADR: Kontext, erwogene Optionen, Wahl, Folgen und Anlass für eine spätere Neubewertung. Halten Sie es neben dem Code, damit beide gemeinsam geprüft werden.

Geschäftsregeln tauchen in drei Schichten zugleich auf. Jede Änderung legte ihre Logik dorthin, wo es an diesem Nachmittag bequem war. Schließlich prüfen Frontend, API und Datenbank dieselbe Regel jeweils etwas anders. Benennen und dokumentieren Sie die Zuständigkeit jeder Schicht. Verlagern Sie eine Zuständigkeit zurück, sobald sie in eine andere Schicht übergreift, solange die dort entstandene Kopie noch klein ist. Erzwingen Sie die Grenze, wo möglich: etwa durch Funktionsaufrufe der API bei gleichzeitig fehlenden Tabellenrechten.

Eine Regel im sechsten Monat wird zum Aufräumprojekt. Die Entscheidung, von der alles abhängt, blieb offen. Jeder folgende Commit nahm den zufälligen Zustand als gegeben an. Entscheiden Sie die wenigen teuren Dinge zuerst und lassen Sie den Rest bewusst offen. Vermerken Sie das im Entscheidungsprotokoll, damit die offene Frage sichtbar bleibt und nicht still mit der nächsten Änderung beantwortet wird.

Das Architekturdiagramm erklärt niemandem etwas. Eine Zeichnung mischt Nutzer, Dienste, Klassen und Tabellen. Linien bleiben unbeschriftet oder heißen „verwendet“. Zeichnen Sie wie C4 ein Diagramm pro Zoomstufe und benennen Sie an jeder Linie Richtung und Absicht der Beziehung. Zwischen Containern gehört auch Technik oder Protokoll dazu. C4s Notationshinweise verlangen genau das und raten von Einwortbeschriftungen wie „Uses“ ab.

Weiterführende Lektüre

- Architektur im NIST-CSRC-Glossar — die verwendete Definition neben den FIPS- und Sicherheitsarchitekturdefinitionen desselben Eintrags.
- NIST SP 800-160 Vol. 1 Rev. 1 — Engineering Trustworthy Secure Systems als Quelle der Definition.
- Architectural Decision Records — die Merkmale einer Architekturentscheidung, Dokumentationsvorlagen und unterstützende Werkzeuge.
- Das C4-Modell — vier Diagrammebenen vom Kontext bis zum Code mit Notationshinweisen und Prüfliste.
- arc42 — praktische Hinweise und Beispiele für jeden der zwölf Abschnitte der arc42-Vorlage.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Architektur?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/architecture/