

Vorname	Nachname	Beruf	Dauer
Marten	Meißner	Fachinformatiker AE	07.22-07.24

## Wochenbericht KW 23. (2023.06.05. - 11.)

Da wir die in letzter Zeit wenig Theorie Themen besprochen, haben wir dieses Mal viele Präsentationen gehalten zu verschieden Themen. Angefangen hatten wir mit einer Auffrischung zum Thema Hexadezimal. Hexadezimal wird benutzt um große Zahlen kompakter zusammenzufassen, im Vergleich zu dezimal wird die Zahl im Faktor 1/1,6 kompakter, so können bis 255 im dezimalen System noch mit 2 Zeichen bezeichnet werden (FF/255) dies wird bei der nächsten Potenz noch markanter, da hier 4096 mit drei Zeichen bezeichnet werden können. Diese Einsparung ist hier noch nicht so markant, aber das Hexadezimal System wird auch selten verwendet im Verhältnis zu dem Dezimalsystem, da hier die Umrechnung doch sehr komplex werden können. Aber wo es häufig verwendet wird, ist im Binärsystem, hier wird die Logik hinter der Verwendung des Hexadezimalsystems erst richtig interessant, da hier die Informationen häufig in 4 Bit sortiert werden (4 Bit 0000–1111  $\Rightarrow$  0-F) und so immer vier Bit zusammengefasst werden können und so sind hier bis 256 abzubilden schon acht Zeichen nötig und in Hex. nur zwei, besonders häufig kennt man dies bei einem Farbcode, dieser setzt sich aus 256 Informationen je Hauptfarbe zusammen also  $3 * 256$  dies sind 16.777.216 Kombinationsmöglichkeiten diese Zahl wäre in binär 24 Zeichen lang und als Hex wert nur noch sechs Zeichen lang (FFFFFF). Des Weiteren unterhielten wir uns auch über Microservices. Als Microservices wird eine Alternative Software-Architektur beschrieben, die die Programme nicht nur in unterschiedlichen Ebenen aufbaut z. B. MVC, sondern auch noch das Prinzip „Do One Thing, but do it well“ aufgebaut wird, Programme, die mit Microservices arbeiten kann bildlich darstellen wie eine Tabelle mit drei Zeilen jede Zeile steht für eine Abspaltung nach dem MVC z. B. steht die erste Zeile für die View, hier wird nur das Frontend des Programms verwaltet. Die zweite Zeile steht für Models und für den einen Teil des Backends, hier wird die Datenlogik und Architektur verwaltet und bestimmt. Und in der dritten Zeile steht für die andere Hälfte des Backends die Controller, hier wird die Nutzer Interaktion verwaltet, z. B. ein Nutzer klickt auf eine Taste und ändert damit die aktuelle Datenstruktur (State) da hier jede Ebene voneinander abstrahiert wird, um nicht unsauberen Code zu schreiben, wird dann mit dem Controller mitgeteilt das im Frontend eine Aktion ausgelöst wird diese wird abgefangen behandelt und ein Neuaufbau View mit dem geänderten State Daten. Da bei einer solchen Programmaufteilung je Ebene viele Funktionen abfangen müssen und dies im Laufe der Zeit ein riesiges Programm anwächst, dass schwer zu refactoren oder upzudaten ist, wird nach dem Microservices Prinzip nochmal jede Ebene geteilt in einzelne eigenen Programme, die dann immer nur eine Funktion abhandeln, dafür aber spezialisiert werden kann. Das hat viele Vorteile, z. B. kleinere schnelle Programme, die perfekt auf eine Aufgabe abgestimmt werden, können die leicht upgedatet werden können, leicht um Funktionen erweitert werden kann und auch nochmal atomarer im Programm laufen können, um das System noch weniger zu beeinflussen. Dies hat auch nochmal den Vorteil, dass jede Anwendung auch eine andere Programmiersprache verwenden kann. Zu den ganzen Vorteilen gibt es aber auch einige Nachteile und das ist der Aufwand ein solches Programm zu schreiben, da entweder jedes Teil eines Programmes unabhängig und so auch bearbeitet werden sollte, also im Idealfall hat jeder Services hat ein eigenes Team, das hierfür zuständig ist, durch die Atomisierung der einzelnen Funktionen entstehen sehr viele Schnittstellen und erzeugen ein gewisses Sicherheitsrisiko und Aufwand.

Kontrolliert am: \_\_\_\_\_ Unterschrift : \_\_\_\_\_