Ich stehe weniger auf proprietären scheiß und mehr auf öfffentlich zugängliche und erweiterbare technologien Mehr PWA (Browserbased) und weniger proprietär / nativ (?) https://github.com/fancy-flashcard/ffc Projektideen DHBW Richie Vorschlag: mehrere Projekte Issue auf https://github.com/michael-spengler/wwi18seac für jede Projektidee Code Reuse früher: paketbasiert – heute (mit deno) viel spezifischer und feingranularer Meilenstein in der Softwareentwicklung deno (Laufzeitumgebung) / RTE library: https://www.npmjs.com/package/node-nlp Was schreibt mir der User möglicherweise? Inputs NLU Was meint der User damit? Intents Natural Language Processing (NLP) – z.B. relevant für Chatbots wie Siri, Alexa, DHBW Richie) backend (serverseitige entwicklung) Was antworte ich als Maschine? Outputs Fun Facts: https://www.youtube.com/watch?v=ZF-NtGsZ6Zo&feature=youtu.be projekte wie rescam.org sind dadurch möglich artificial intelligence (gpt 3) enable deno <-> ethereum interaction (web3.js --> web3.ts) ethereum (vitalik buterin) flutter (framework) performancevorteile? nativescript (framework) vue Progressive Web Apps react angular frontend (entwicklung von code der beim client (e.g. browser) ausgeführt wird technologien Swift (Programmiersprache) ios & os x Xcode (IDE) Java (Progrmmiersprache) klassischerweise / proprietär Kotlin android DART Android Studio (IDE) chat client interaction simplicity efficiency reliability hosting & operations scalability wenn es trade offs gibt ist es wertvoll die ups and downs kontextspezifisch zu erörtern paypal ethereum in app käufe e.g apple pay monetarisierung Werbung Afiliate Links – e.g. Amazon Partner Program mein code (der code des anwendungsentwicklers) wird vom framework aufgerufen (Beispiel nestjs.com) Exkurse mein code (der code des anwendungsentwicklers) ruft den code aus der library auf library Attraktives Aussehen Benutzerfreundlichkeit Funktioniert zuverllässig Separation of Concerns Leichte Lesbarkeit / Wartbarkeit Loose Coupling Clean Code kontextspezifisch zu bewerten ? Objektorientierung Um zu erläutern WARUM etwas gemacht wird ? Kommentare Skalierbarkeit Performance Technologische Innovation **Hosting & Operations** Bewertungskriterien Datensicherheit für User Schutz gegen z.B. DDOS Attacks --> sicherstellen der availability der app Userdokumentation Dokumentation Begründung bewerten Warum haben wir uns für welche technologie entschieden?? Contributordokumention leicht verständlich dargelegt Fragen konnten auf eine wertvolle Art und Weise beantwortet werden \ Produktpräsentation Interessant und angepasst auf die Zielgruppe Gedanke: competition - todo spengler: explore this Gedanke: Open Source / Reusabilty Mobile First (weil es einfacher ist alles auf einem größerem screen unterzubringen= Responsive

Open Source Fortschrittsgeschwindigkeit

https://github.com/michael-spengler/wwi18seac Separation of Concerns Qualities in Software Loose Coupling Anregung an Spengler: Mehr Theorie als sonst ____ Anregung an Euch: Ein echtes Begreifen geht m.E. am besten durch ausprobieren sachen anhand von stichworten einsortieren können Gruppenzusammensetzung optimieren auf langfristigen Gesamterfolg Ideale Gruppengröße m.E. 3 – 5 – nur eine Anregung Selbstorganisation Minimal Viable Service (MVS) Minimal Viable Product (MVP) Vorgehensempfehlung Start Small Value Proposition Canvas https://www.youtube.com/watch?v=ReM1uqmVfP0

WWI18SEAC