Kriterienkatalog für die Architekturauswahl

Mit diesem Kriterienkatalog sollen Software-Architekten anhand der vorliegenden Problemstellung einen webbasierten Architekturvorschlag (MPA, SPA, Fullstack-Architektur) ableiten können. Entstanden ist das Ganze im Rahmen der Bachelorarbeit "Ist die Trennung zwischen Frontend und Backend in webbasierten Systemen noch zeitgemäß?" von Michael Mertl. Die Kriterien sind dabei aus einer umfangreichen und internationalen Literaturrecherche zu den verschiedenen Ansätzen entstanden, indem die Vor- und Nachteile dieser ermittelt wurden und dementsprechend als Grundlage dienten für die vergebenen Punkte. Außerdem wurde eine praxisorientierte Qualitätsprüfung in Form von Experteninterview bei der Firma XITASO durchgeführt, um den Kriterienkatalog zu verbessern. Ebenso wurden Experimente in den drei Architekturen durchgeführt, um einen direkten Vergleich vorzunehmen und den Kriterienkatalog zu verbessern.

Funktionsweise:

Bitte nur die Kriterien ankreuzen und gewichten, die die vorliegende Problemstellung erfordern. Am Ende die Gesamtpunktzahl ausrechnen und die Architektur mit der höchsten Punktzahl ist diejenige, die am besten geeignet ist, für die vorliegende Problemstellung. Jede Architektur hat für jedes Kriterium vorweg eine Punktzahl bekommen (1 = nicht/schlecht erfüllt, 3 = teilweise erfüllt, 6 = vollständig erfüllt), die aus der Literaturrecherche anhand der Vor- und Nachteile entstanden sind und durch Experteninterviews und Experimente evaluiert wurden. Diese verrechnet mit dem Gewicht ergibt die Punktzahl für das jeweilige Kriterium. Die Gewichte, die für die einzelnen Kriterien vergeben werden, müssen am Ende 100% entsprechen, um eine mathematisch korrekte Berechnung zu gewährleisten.

Kriterium	Multi-Page-Architektur	Single-Page-Architektur	Fullstack-Architektur	Gewicht	Gew
			(orientiert an Blazor Server)		ollt?
Autorisierung mit	Die Autorisierung muss	Die Autorisierung muss in	Die Autorisierung muss nur in	Gewicht:	
OAuth 2.0	nur in einer Anwendung	beiden Anwendungen	einer Anwendung umgesetzt	MPA:	Ш
-> ist die Umsetzung der	umgesetzt werden, da	(dem getrennten Frontend	werden, da bei Blazor Server	SPA:	
Autorisierung mit OAuth	bei der MPA nur eine	und Backend) umgesetzt	nur eine gebaut wird.	Fullstack:	
2.0 in der Architektur einfach?	gebaut wird.	werden. So entsteht ein			
ennacii:		Mehraufwand.			
	-> Punktzahl: 6	-> Punktzahl: 3	-> Punktzahl: 6		
Benutzererfahrung	Jede Benutzerinteraktion	Benutzerinteraktionen	Benutzerinteraktionen lösen	Gewicht:	
-> bietet die Architektur	führt zum neuen Laden	lösen keine ganzen	keine ganzen Seitenaufrufe	MPA:	
eine angenehme	der gesamten Seite,	Seitenaufrufe mehr aus	mehr aus und es können	SPA:	
Benutzererfahrung beim Laden neuer Inhalte?	welche die neuen Inhalte	und es können Animatio-	Animationen genutzt werden,	Fullstack:	
Eddon Hodor Hillard .	dann bereitstellt.	nen genutzt werden, beim	beim Warten auf neue Inhalte.		
	Zusätzlich werden die	Warten auf neue Inhalte.			
	Inhalte im Vergleich zur	Dennoch werden diese			
	Fullstack-Architektur	Inhalte im Vergleich zur			
	langsamer geladen,	Fullstack-Architektur			
	jedoch genauso schnell	langsamer geladen.			
	wie bei der SPA.				
	-> Punktzahl: 1	-> Punktzahl: 3	-> Punktzahl: 6		

Benutzererfahrung	Bei der MPA liegen viele	Bei der SPA liegt nur eine	Bei Blazor Server liegt	Gewicht:	
-> bietet die Architektur	HTML-Seiten vor, daher	HTML-Seite vor auf der	dasselbe Problem wie bei der	MPA:	
eine angenehme	stellt der Browser bereits	mit JS navigiert wird. Das	SPA vor. Die .NET-Umgebung	SPA:	
Benutzererfahrung beim Betätigen von	von selbst sicher, dass	Standardverhalten des	bringt eine Klasse (namens	Fullstack:	
Browsertasten (z.B.	die verschiedenen	Browsers kann nicht direkt	"NavigationManager") mit, die		
Vorwärts/Rückwärts	Browsertasten richtig	genutzt werden und muss	es sehr einfach macht, das		
Schaltfläche)?	funktionieren.	durch Programmierung	Standardverhalten des		
		nutzbar gemacht werden.	Browsers zu nutzen und richtig		
			zu programmieren.		
	-> Punktzahl: 6	-> Punktzahl: 1	-> Punktzahl: 3		
Browseranforderung	Nein, da die MPA auch	Ja	Ja	Gewicht:	
-> muss der Browser	komplett ohne JS			MPA:	
JavaScript fähig sein für	funktionieren würde.			SPA:	
die Webanwendung?				Fullstack:	
	-> Punktzahl: 6	-> Punktzahl: 1	-> Punktzahl: 1		
Entwicklererfahrung	Es muss nur eine	Es werden zwei	Es muss nur eine Anwendung	Gewicht:	
-> welches Fachwissen	Anwendung entwickelt	Anwendungen entwickelt.	entwickelt werden. In dieser	MPA:	
wird für die Architektur	werden. In dieser wird	Für die Kommunikation	wird teilweise Wissen im	SPA:	
benötigt?	aber Fachwissen im	dazwischen muss	Frontend und Backend	Fullstack:	
	Frontend (HTML, CSS	zusätzlich eine API	benötigt. Hier wird kein		
	und JS) und Backend	entwickelt werden. Es wird	JavaScript benötigt, lediglich		
	(z.B. C#) benötigt.	also Fachwissen im	Wissen in C#, HTML und CSS.		
L	1			1	

### Denötigt. ### Denotigt. ### Denoticit is der Regel schneller als bei der SPA, da nur eine Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. ### Denoticit is denoticit is der MPA und der SPA, da nur eine Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. ### Denoticit is denoticit i			Frontend (HTML, CSS und			
Entwicklungszeit -> wie lange dauert die Entwicklung einer Webanwendung mit dieser Architektur? In der Regel schneller als bei der SPA, da nur eine Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die In der Regel schneller als bei der SPA, da nur eine Als bei der MPA und der Fullstack-Architektur, da zwei Anwendung entwickelt werden muss und auf die nötige API für die eine API für die eine API für die einer SPA verzichtet werden kann. -> Punktzahl: 3 -> Punktzahl: 6 In der Regel schneller als bei der SPA, da nur eine Anwendung entwickelt werden muss und auf die nötige API für die einer SPA verzichtet werden kann. Fullstack-Architektur, da zwei Anwendungen geschrieben werden und eine API für die einer SPA verzichtet werden kann. Werden muss.			JS) und Backend (z.B. C#)			
Entwicklungszeit -> wie lange dauert die Entwicklung einer Webanwendung mit dieser Architektur? In der Regel schneller als bei der SPA, da nur eine Anwendung entwickelt werden müss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel schneller als bei der SPA, da nur eine Anwendung entwickelt werden Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Werden MPA: SPA: Fullstack: Fullstack: Fullstack: Fullstack: Fullstack: Fullstack:			benötigt.			
Entwicklungszeit -> wie lange dauert die Entwicklung einer Webanwendung mit dieser Architektur? In der Regel schneller als bei der SPA, da nur eine Anwendung entwickelt werden müss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel langsamer als die In der Regel schneller als bei der SPA, da nur eine Anwendung entwickelt werden Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Werden MPA: SPA: Fullstack: Fullstack: Fullstack: Fullstack: Fullstack: Fullstack:						
bei der SPA, da nur eine Anwendung entwickelt Webanwendung mit dieser Architektur? bei der SPA, da nur eine Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die bei der SPA, da nur eine Anwendung entwickelt werden was und der Fullstack-Architektur, da zwei Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei eine API für die kann. MPA: SPA: Fullstack: Ful		-> Punktzahl: 3	-> Punktzahl: 3	-> Punktzahl: 6		
Entwicklung einer Webanwendung mit dieser Architektur? Anwendung entwickelt werden muss und auf die nötige API für die kommunikation wie bei einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Anwendung entwickelt werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. SPA: Fullstack: F	Entwicklungszeit	In der Regel schneller als	In der Regel langsamer	In der Regel schneller als bei	Gewicht:	
Webanwendung mit dieser Architektur? Werden muss und auf die nötige API für die werden und einer SPA verzichtet werden werden und einer SPA verzichtet werden werden muss. Werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die Anwendung entwickeit werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann. Werden muss und auf die nötige API für die Kommunikation wie bei einer SPA verzichtet werden kann.	· ·	bei der SPA, da nur eine	als bei der MPA und der	der SPA, da nur eine	MPA:	
Architektur? Werden muss und auf die nötige API für die geschrieben werden und kommunikation wie bei einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die	-	Anwendung entwickelt	Fullstack-Architektur, da	Anwendung entwickelt werden	SPA:	
nötige API für die Kommunikation wie bei eine API für die einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die geschrieben werden und eine API für die eine API für die einer SPA verzichtet werden kann. kann. für die Kommunikation wie bei einer SPA verzichtet werden kann.	_	werden muss und auf die	zwei Anwendungen	muss und auf die nötige API	Fullstack:	
einer SPA verzichtet werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die Kommunikation entwickelt werden muss. kann.	/ Horntoktar :	nötige API für die	geschrieben werden und	für die Kommunikation wie bei		
werden kann. Durch die Codetrennung von Frontend und Backend, aber in der Regel langsamer als die werden muss.		Kommunikation wie bei	eine API für die	einer SPA verzichtet werden		
Codetrennung von Frontend und Backend, aber in der Regel langsamer als die		einer SPA verzichtet	Kommunikation entwickelt	kann.		
Frontend und Backend, aber in der Regel langsamer als die		werden kann. Durch die	werden muss.			
aber in der Regel langsamer als die		Codetrennung von				
langsamer als die		Frontend und Backend,				
		aber in der Regel				
Fullstack-Architektur.		langsamer als die				
		Fullstack-Architektur.				
-> Punktzahl: 3 -> Punktzahl: 1 -> Punktzahl: 6		-> Punktzahl: 3	-> Punktzahl: 1	-> Punktzahl: 6		
Erweiterbarkeit Leicht erweiterbar, Leicht erweiterbar, Es wird eine Anwendung Gewicht:	Erweiterbarkeit	Leicht erweiterbar,	Leicht erweiterbar,	Es wird eine Anwendung	Gewicht:	
-> ist die Architektur in der Frontend und Backend Frontend und Backend entwickelt, in der keine MPA:	-> ist die Architektur in der	Frontend und Backend	Frontend und Backend	entwickelt, in der keine	MPA:	
Zukunft leicht erweiterbar sind aber etwas stärker können isoliert erweitert saubere Codetrennung SPA:	Zukunft leicht erweiterbar	sind aber etwas stärker	können isoliert erweitert	saubere Codetrennung	SPA:	

(in Bezug auf neue	verknüpft, da diese in	werden, da zwei	zwischen Frontend und	Fullstack:	
Features)?	einer Anwendung sind.	Anwendungen vorliegen.	Backend erzwungen wird, wie		
			bei der MPA oder SPA, dies		
			kann die Erweiterbarkeit		
			erschweren.		
	-> Punktzahl: 3	-> Punktzahl: 6	-> Punktzahl: 3		
Offlinefähigkeit	Nein	Ja (solange keine neue	Nein	Gewicht:	
-> ist die Architektur in der		Daten-Anfrage		MPA:	
Lage, die Webanwendung		angestoßen wird)		SPA:	
auch offline nutzbar zu machen?				Fullstack:	
machen:	-> Punktzahl: 1	-> Punktzahl: 6	-> Punktzahl: 1		
Performanz	Lädt initial schnell, da nur	Erster Ladezyklus dauert	Lädt initial schnell, da nur eine	Gewicht:	
-> wie schnell ist das	eine HTML-Seite (diese	sehr lang, da die gesamte	statische HTML-Seite und	MPA:	Ш
initiale Laden der	kann statisch oder	Anwendung geladen wird.	zusätzlich eine JavaScript-	SPA:	
Webanwendung?	dynamisch sein) geladen		Bibliothek von Microsoft (diese	Fullstack:	
	werden muss. Dennoch		ist verantwortlich für die		
	dauert das initiale Laden		Übertragung der Benutzer-		
	im Vergleich zur		interaktionen an den Server)		
	Fullstack-Architektur		übertragen werden muss.		
	länger.				
	-> Punktzahl: 3	-> Punktzahl: 1	-> Punktzahl: 6		

Daten fordert eine neue Nachladen neuer Inhalte (ohne die Verwendung von zusätzlichen Caching- Mechanismen)? Daten fordert eine neue HTML-Seite (wieder statisch oder dynamisch) vom Server an. Der Server stellt bzw. rendert die neue HTML-Seite und gibt diese an den Client zum Anzeigen zurück. Lädt genauso schnell wie initial und wie die SPA, aber langsamer als die Fullstack- Architektur. Daten stößt eine Anffrage (z.B. HTTP-Anfrage) an den Server an. Dieser sendet dann eine Antwort (z.B. HTTP-Antwort) mit den benötigten Daten zurück. Die neuen Daten werden dann am Client verarbeitet und es wird nur der Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack- Architektur. Daten stößt eine Anffrage (z.B. HTTP-Anfrage) an die SignalR dem Server gemeldet. Dieser beschafft sich die neuen Daten und tauscht den betroffenen Teil der HTML-Seite auf dem virtuellen Abbild des Servers aus. Dieser geänderte Teil der Seite wird über die SignalR- Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. Sicherheit Sicherheit Server in der Lage, die vertraulichen Daten auf dem Vertra	Performanz	Eine Anfrage nach neuen	Eine Anfrage nach neuen	Eine Anfrage nach neuen	Gewicht:	
(chne die Verwendung von zusätzlichen Caching-Mechanismen)? Server stellt bzw. rendert die neue HTML-Seite und gibt diese an den Client zum Anzeigen zurück. Lädt genauso schnell wie initial und wie die SPA, aber langsamer als die Fullstack- Architektur. Sicherheit Sicherheit Die MPA ist wie Blazor Server stellt bzw. rendert die Neuen Daten und den benötigten Daten werden dann am Client verarbeitet und es wird nur der Seite mid über die SignalR- verarbeitet und es wird nur der Seite mid über die SignalR- verarbeitet und es wird nur der Seite mid über die SignalR- verarbeitet und es wird nur der Seite wird über die SignalR- verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. Sicherheit Sicherheit Server in der Lage, die vertraulichen Daten auf wertraulichen Daten auf den Server gemeldet. Dieser beschafft sich die signalR der HTML-Seite auf dem virtuellen Abbild des Servers aus. Dieser geänderte Teil der Seite wird über die SignalR- Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. Sicherheit Sicherheit Die MPA ist wie Blazor Server in der Lage, die vertraulichen Daten auf daher anfällig für	-> wie schnell ist das	Daten fordert eine neue	Daten stößt eine Anfrage	Daten wird von der JavaScript-	MPA:	
statisch oder dynamisch) wom Server an. Der Server sellt bzw. rendert die neue HTML-Seite und gibt diese an den Client zum Anzeigen zurück. Lädt genauso schnell wie initial und wie die SPA, aber langsamer als die Fullstack-Architektur. Sicherheit Sicherheit Sittisch oder dynamisch) wom Server an. Der sendet dann eine Antwort gemeldet. Dieser beschafft sich die neuen Daten und tauscht den betroffenen Teil der HTML-Seite auf dem virtuellen Abbild des Servers aus. Dieser geänderte Teil der Seite wird über die SignalR-Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der Neuen Daten und tauscht den betroffenen Teil der HTML-Seite auf dem virtuellen Abbild des Servers aus. Dieser geänderte Teil der Seite wird über die SignalR-Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. Sicherheit Sicherheit Sittisch oder dynamisch) vom Server an. Dies sendet dann eine Antwort gemeldet. Dieser beschafft sich die neuen Daten und tauscht den betroffene Teil der Seite wird über die SignalR-Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. Sicherheit Sicherheit Die MPA ist wie Blazor Server in der Lage, die vertraulichen Daten auf dem Vertraulichen Daten auf dem Virtuellen Anten auf dem Virtuellen Anten und tauscht den betroffenen Teil der HTML-Seite auf dem Virtuellen Abbild des Servers aus. Dieser geänderte Teil der Seite wird über die SignalR-Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. Sicherheit Sicherheit Die MPA ist wie Blazor Server in der Lage, die vertraulichen Daten auf dem Vertraulichen Daten auf dem Virtuellen Anten Teil der Mehren Teil der M		HTML-Seite (wieder	(z.B. HTTP-Anfrage) an	Bibliothek von Microsoft über	SPA:	
wom Server an. Der Server stellt bzw. rendert die neue HTML-Seite und gibt diese an den Client zum Anzeigen zurück. Lädt genauso schnell wie initial und wie die SPA, aber langsamer als die Fullstack- Architektur. Sicherheit wie anfällig ist die Architektur für die ungewollte Offenlegung Anzeigen zurück an den Client zug om Server an. Der Server in der Lage, die vorm den Server selt den benötigten Daten zurück. Die neuen Daten werden dann am Client zurück. Die neuen Daten werden dann am Client verarbeitet und es wird nur der Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack- Architektur. Sicherheit wie anfällig ist die Architektur für die ungewollte Offenlegung	,	statisch oder dynamisch)	den Server an. Dieser	die SignalR dem Server	Fullstack:	
die neue HTML-Seite und gibt diese an den Client zum Anzeigen zurück. Lädt genauso schnell wie initial und wie die SPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 Sicherheit -> wie anfällig ist die Architektur für die und gibt diese an den Client zum Anzeigen zurück. Lädt genauso werden dann am Client verarbeitet und es wird nur der Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die und gibt diese an den zurück. Die neuen Daten werden dann am Client verarbeitet und es wird nur der Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 -> Punktzahl: 6 Blazor Server speichert im Gewicht: Gegensatz zu der SPA die vertraullichen Daten auf werden dann am Client virtuellen Abbild des Servers aus. Dieser geänderte Teil der Virtuellen Abbild des Servers aus. Dieser geänderte Teil der Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 6 Blazor Server speichert im Gegensatz zu der SPA die vertraullichen Daten auf dem SPA:	_	vom Server an. Der	sendet dann eine Antwort	gemeldet. Dieser beschafft		
und gibt diese an den Client zum Anzeigen zurück. Lädt genauso schnell wie initial und wie die SPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 Sicherheit -> wie anfällig ist die Architektur für die und gibt diese an den Client zum Anzeigen zurück. Die neuen Daten werden dann am Client verarbeitet und es wird nur der Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die undewöllte Offenleaung Die MPA ist wie Blazor Server in der Lage, die vertraullichen Daten auf	,	Server stellt bzw. rendert	(z.B. HTTP-Antwort) mit	sich die neuen Daten und		
Client zum Anzeigen zurück. Lädt genauso schnell wie initial und wie die SPA, aber langsamer als die Fullstack- Architektur. Sicherheit -> Wenktzahl: 3 Sicherheit -> Wenktzahl: 3 Sicherheit -> Werden dann am Client verarbeitet und es wird nur der Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung Client zum Anzeigen verarbeitet und es wird nur der Seite wird über die SignalR- Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 6 Blazor Server speichert im Gegensatz zu der SPA die vertraulichen Daten auf dem SPA:		die neue HTML-Seite	den benötigten Daten	tauscht den betroffenen Teil		
zurück. Lädt genauso schnell wie initial und wie der Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Architektur. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack-Architektur. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack-Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenieuung zurück. Lädt genauso der Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenieuung zurück. Lädt genauso schnell wie der von den neuen Daten betroffen ist. Lädt genauso schnell wie die NPA; seite wird über die Seite wird über die Seite wird über der Olient geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 6 Sicherheit -> Wie anfällig ist die Architektur für die ungewollte Offenieuung daher anfällig für		und gibt diese an den	zurück. Die neuen Daten	der HTML-Seite auf dem		
schnell wie initial und wie die Seitenteil neu gerendert, der von den neuen Daten betroffen ist. Architektur. Seite wird über die SignalR-Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 3 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung -> Punktzahl: 4 Die MPA ist wie Blazor Server in der Lage, die vertraullichen Daten auf daher anfällig für Seite wird über die SignalR-Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung		Client zum Anzeigen	werden dann am Client	virtuellen Abbild des Servers		
die SPA, aber langsamer als die Fullstack- Architektur. Die MPA ist wie Blazor Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung die SPA, aber langsamer als die Fullstack der von den neuen Daten betroffen ist. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Die MPA ist wie Blazor Server in der Lage, die vertraulichen Daten auf daher anfällig für Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 6 Blazor Server speichert im Gegensatz zu der SPA die vertraulichen Daten auf daher anfällig für Verbindung an den Client geschickt, wo er ebenfalls ausgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird.		zurück. Lädt genauso	verarbeitet und es wird nur	aus. Dieser geänderte Teil der		
als die Fullstack- Architektur. -> Punktzahl: 3 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung -> Punktzahl: 3 -> Punktzahl: 6 Blazor Server speichert im Gegensatz zu der SPA die Vertraulichen Daten auf -> Vertraulichen Daten auf -> Punktzahl: 0 -> Punktza		schnell wie initial und wie	der Seitenteil neu	Seite wird über die SignalR-		
Architektur. Lädt genauso schnell wie die MPA, aber langsamer als die Fullstack-Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung Architektur Daten auf Lädt genauso schnell wie die MPA, aber langsamer als usgetauscht wird und somit der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 3 -> Punktzahl: 6 Blazor Server speichert im Gewicht: Daten am Client und ist daher anfällig für Vertraulichen Daten auf dem SPA:		die SPA, aber langsamer	gerendert, der von den	Verbindung an den Client		
die MPA, aber langsamer als die Fullstack- Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung Die MPA ist wie Blazor Server in der Lage, die vertraulichen Daten auf daher anfällig für der neue Inhalt einem Anwender angezeigt wird. -> Punktzahl: 6 Blazor Server speichert im Gegensatz zu der SPA die vertraulichen Daten auf dem SPA:		als die Fullstack-	neuen Daten betroffen ist.	geschickt, wo er ebenfalls		
als die Fullstack- Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung -> Punktzahl: 3 -> Punktzahl: 6 Die MPA ist wie Blazor Die SPA speichert viele Daten am Client und ist Gegensatz zu der SPA die Vertraulichen Daten auf daher anfällig für Anwender angezeigt wird. -> Punktzahl: 6 Blazor Server speichert im Gewicht: MPA: Vertraulichen Daten auf dem SPA:		Architektur.	Lädt genauso schnell wie	ausgetauscht wird und somit		
Architektur. -> Punktzahl: 3 -> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung Architektur für die ungewollte Offenlegung Architektur. -> Punktzahl: 3 -> Punktzahl: 6 -> Punktzahl: 6 Blazor Server speichert im Gewicht: Gegensatz zu der SPA die vertraulichen Daten auf daher anfällig für -> vertraulichen Daten auf dem SPA:			die MPA, aber langsamer	der neue Inhalt einem		
-> Punktzahl: 3 -> Punktzahl: 6 Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung -> Punktzahl: 3 -> Punktzahl: 6 Die MPA ist wie Blazor Die SPA speichert viele Daten am Client und ist daher anfällig für -> Punktzahl: 6 Blazor Server speichert im Gewicht: MPA: Vertraulichen Daten auf daher anfällig für -> Punktzahl: 6 Blazor Server speichert im Gewicht: MPA: SPA:			als die Fullstack-	Anwender angezeigt wird.		
Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung Die MPA ist wie Blazor Die SPA speichert viele Daten am Client und ist daher anfällig für Die SPA speichert viele Daten am Client und ist daher anfällig für Vertraulichen Daten auf dem SPA:			Architektur.			
Sicherheit -> wie anfällig ist die Architektur für die ungewollte Offenlegung Die MPA ist wie Blazor Die SPA speichert viele Daten am Client und ist daher anfällig für Die SPA speichert viele Daten am Client und ist daher anfällig für Vertraulichen Daten auf dem SPA:						
-> wie anfällig ist die Architektur für die ungewollte Offenlegung Server in der Lage, die Daten am Client und ist daher anfällig für Vertraulichen Daten auf dem SPA:		-> Punktzahl: 3	-> Punktzahl: 3	-> Punktzahl: 6		
Architektur für die vertraulichen Daten auf daher anfällig für vertraulichen Daten auf dem SPA:	Sicherheit	Die MPA ist wie Blazor	Die SPA speichert viele	Blazor Server speichert im	Gewicht:	
vertraulichen Daten auf daner anfallig für vertraulichen Daten auf dem SPA.		Server in der Lage, die	Daten am Client und ist	Gegensatz zu der SPA die	MPA:	
ungewonte Oneniegung dem Server zu bösartiges JavaScript Server Durch die Vermischung Fullstack:		vertraulichen Daten auf	daher anfällig für	vertraulichen Daten auf dem	SPA:	
Social good data on printing and the first a	ungewolite Offenlegung	dem Server zu	bösartiges JavaScript,	Server. Durch die Vermischung	Fullstack:	

von vertraulichen Daten	speichern. Ebenfalls liegt	welches durch Eingaben	von Frontend und Backend,		
durch einen Entwickler?	hier beim Entwickeln	eingeschleust werden	kann es jedoch vorkommen,		
	eine strikte Trennung	kann. Der Entwickler muss	dass beim Entwicklungs-		
	zwischen Frontend und	also beim Entwickeln	prozess unbeabsichtigt		
	Backend vor. Daher ist	aufpassen, dass er keine	vertrauliche Daten auf dem		
	es beim Entwickeln	vertraulichen Daten am	Client verarbeitet werden, da		
	leichter, keine	Client preisgibt.	die Anwendung unübersichtlich		
	vertraulichen Daten		werden kann.		
	ungewollt am Client zu				
	verarbeiten oder				
	offenzulegen.				
	-> Punktzahl: 6	-> Punktzahl: 1	-> Punktzahl: 3		
Serverbelastung	Hoch, da jede Anfrage	Niedrig, da jede Anfrage	Mittel, da jede Anfrage nach	Gewicht:	
-> wie stark wird der	nach neuen Daten eine	nach neuen Daten nur die	neuen Daten wie bei der SPA	MPA:	
Server belastet (ohne die	neue HTML-Seite	genau benötigte	nur die genau benötigte	SPA:	
Verwendung von zusätzlichen Caching-	anfordert. Diese muss	Datenmenge vom Sever	Datenmenge vom Server	Fullstack:	
Mechanismen)?	jedes Mal neu vom	anfordert und dieser somit	anfordert. Die Grundbelastung		
	Server bereitgestellt bzw.	nicht so stark belastet wird	des Servers ist jedoch höher		
	generiert werden. Der	als bei der MPA. Der	als bei der MPA oder der SPA,		
	Zustand und die Logik	Zustand und die Logik in	da bei Blazor Server die Logik		
	der MPA können am	der SPA werden am Client	und die Zustände am Server		
	Client und am Server	verarbeitet und belastet			
	1		1		

	(oder auch in	somit den Server nicht.	verwaltet werden müssen und		
	Kombination) verarbeitet	Dennoch stellt over- und	diesen somit belasten.		
	werden. Dadurch kann	under-fetching eine			
	der Server dann	Herausforderung dar und			
	zusätzlich belastet	muss vermieden werden,			
	werden.	sonst wird der Server			
		unnötig belastet.			
	-> Punktzahl: 1	-> Punktzahl: 3	-> Punktzahl: 3		
Skalierbarkeit	Bei der MPA kann der	Bei der SPA übernimmt	Bei Blazor Server übernimmt	Gewicht:	
-> ist die Architektur in der	Client oder auch der	der Client die Zustands-	der Server die Zustands-	MPA:	
Lage horizontal oder	Server die Zustands-	verwaltung, somit kann	verwaltung. Eine vertikale	SPA:	
vertikal zu skalieren, um mit steigender Anzahl von	verwaltung übernehmen.	der Server zustandslos	Skalierung ist problemlos	Fullstack:	
Benutzern und Anfragen	Hier besteht also die	betrieben werden. Hier	möglich. Eine horizontale		
umzugehen?	Möglichkeit eine	besteht also die	Skalierung ist jedoch aufgrund		
	horizontale oder auch	Möglichkeit eine	dessen und der nötigen		
	eine vertikale Skalierung	horizontale oder auch eine	Verwaltung der SignalR-		
	problemlos	vertikale Skalierung	Verbindungen über mehrere		
	durchzuführen.	problemlos durchzuführen.	Serverinstanzen hinweg, eine		
			größere Herausforderung als		
			bei der MPA oder SPA.		
	-> Punktzahl: 6	-> Punktzahl: 6	-> Punktzahl: 3		

Suchmaschinen-	Es liegen viele HTML-	Es liegt nur eine HTML-	Es liegt nur eine HTML-Seite	Gewicht:	
optimierung (SEO)	Seiten vor, deshalb	Seite vor, deshalb	vor, deshalb funktioniert SEO	MPA:	
-> ist die Architektur in der	funktioniert die SEO sehr	funktioniert SEO nicht	nicht ohne weitere	SPA:	
Lage, die Webanwendung für SEO nutzbar zu machen?	gut.	ohne weitere Technologien.	Technologien.	Fullstack:	
	-> Punktzahl: 6	-> Punktzahl: 1	-> Punktzahl: 1		
Gesamtergebnis				100%	