

JETZT ANFRAGEN!

little

Hyperlinks

A Software Architect's Approach Towards

Using Angular (And SPAs) much

Legacy Apps or *very* new UI isolation?

no

Microservices Aka Microfrontends
Separate Deployment/ mix Technologies?

yes

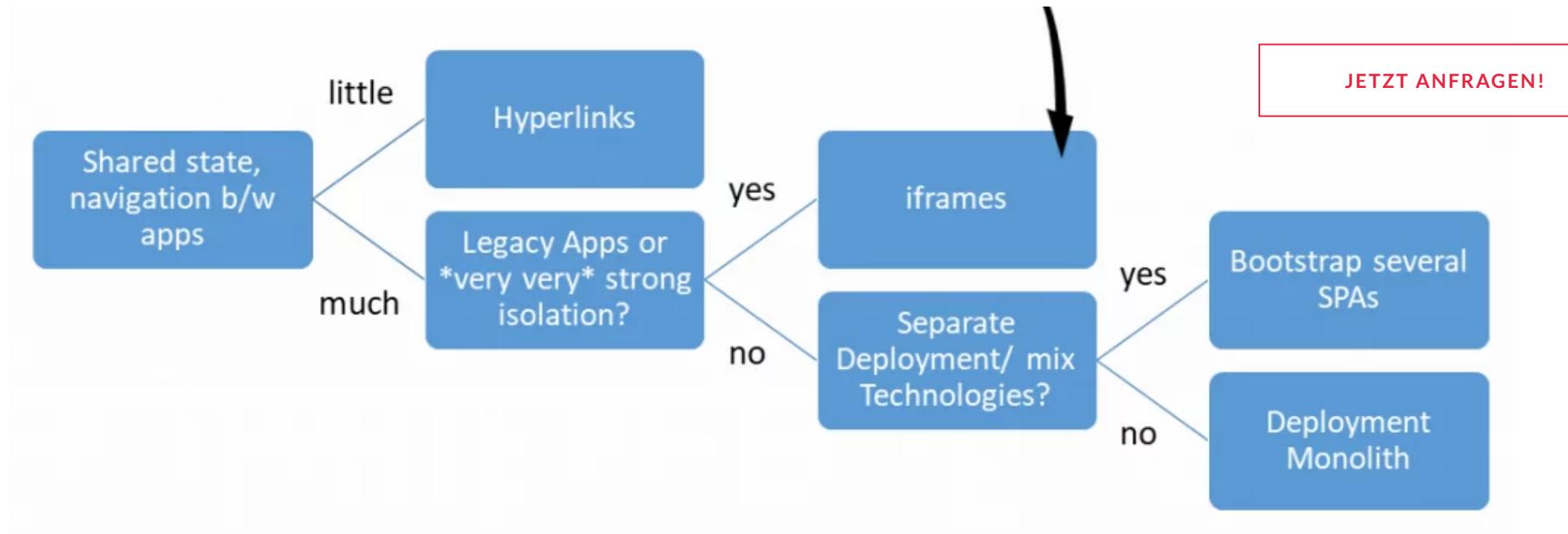
Bootstrap several SPAs

no

Deployment Monolith

03.04.2018 | share  

TLDR: To choose a strategy for implementing micro frontends with SPA, you need to know your architectural goals, prioritize them and evaluate them against the options available. This article does this for some common goals and presents a matrix which reflects the results of the evaluation. To provide some orientation, you also

[JETZT ANFRAGEN!](#)


People ask me on regular basis how to use SPAs and/or Angular in an microservice-based environment. The need for such microfrontends is no surprise, as microservices are quite popular nowadays. The underlying idea of microservices is quite simple: Create several tiny applications – so called microservices – instead of one big monolytic applications. This leads for instance (but not only) to smaller teams (per microservice) that can make decisions faster and chose for the “best” technology that suites their needs.



MANUFACTURED BY THE FEDERAL AGENCY FOR THE USE OF OFFICIAL LANGUAGE IN OFFICIAL ACTIVITIES DURING AUTONOMOUS

complexity into play.

JETZT ANFRAGEN!

Fortunately, there are several approaches for this. Unfortunately, no approach is perfect – each of them has its own pros and cons.

To decide for one, a software architect would evaluate those so called architectural candidates against the architectural goals given for the software system in question. Typical (but not the only) goals for SPAs in microservice-based environments are shown in the next section.

Architectural Goals

Architectural Goal	Description
a) Isolation	Can the clients influence each other in an unplanned way?
b) Separate Deployment	Can the microservices be deployed separately without the need to coordinate with other teams responsible for other microservices?



d) Different SPA-Frameworks

Can we use different SPA frameworks (or libraries) in different versions

JETZT ANFRAGEN!

e) Tree Shaking

Can we make use of tree shaking?

f) Vendor Bundles

Can we reuse already loaded vendor bundles or do we need to load the same framework several times, if it's used by several microfrontends

g) Several microfrontends at the same time

Can we display several microfrontends at the same time, e. g. a product list and a shopping basket

h) Prevents version conflicts

Does the approach prevent version conflicts between used libraries?

i) Separate development

Can separate teams develop their microfrontends independently of other ones

j) One optimized solution (bundle)

Everything is compiled into one optimized solution. You don't have to duplicate libraries or frameworks for different parts of the system.


[JETZT ANFRAGEN!](#)

Evaluation

The following table evaluates some architectural candidates for microfrontends against the discussed goals (a – g).

Architectural Candidate	a	b	c	d	e	f	g	h	i	j
I) Just using Hyperlinks	x	x		x	x			x	x	
II) Using iframes	x	x	x	x	x		x	x	x	
III) Loading different SPAs into the same page	s	x	x	x		x	x		x	
IV) Plugins *	s	x	x				x			x
V) Deployment Monolith using Packages (npm, etc.) *	s			x		x	x			x
VI) Deployment Monolith using Monorepos * **	s		x		x	x	x	x		x

**JETZT ANFRAGEN!**

An X means that the goal in question is supported. The S in the column Isolation means that you can leverage Shadow DOM as well as EcmaScript Modules to archive some amount of isolation.

* This doesn't lead to a typical microservice-based solution. But as the main goal is to reach defined goals, it would be wrong to don't consider these options.

** You can use monorepos with different applications. In this case you would use one of the other integration strategies outlined here.

If you are interested into some of those candidates, the next table provides some additional thoughts on them:

Nr	Remarks
I) Just using Hyperlinks	We could save the state before navigating to another microfrontend. Using something like Redux (@ngrx/store) could come in handy b/c it manages the state centrally. Also consider that the user can open several microservices in several browser windows/ tabs.



Also, iframes are not the most popular feature 😊

JETZT ANFRAGEN!

III) Loading different SPAs into the same page

A popular framework that loads several SPAs into the browser is [Single SPA](#). The main drawback seems to be the lack of isolation, cause all applications share the same global namespace and the same global browser objects. If the latter ones are monkey patched by a framework (like zone.js) this affects all the loaded SPAs

IV) Plugins

Dynamically loading parts of a SPA can be done with Angular but webpack and so the CLI demands on compiling everything together. Switching to SystemJS would allow to load parts that have been compiled separately

V) Packages

This means, providing each frontend as a package via (a private) npm registry or the monorepo approach and consuming it in a shell application. This also means, that there is one compilation step that goes through each frontend.

VI)
Deployment
Monolith

To structure your deployment monolith into parts which can be developed and tested separately, you can e.g. use npm packages or a monorepo.

VII) Web Components

This is similar to III) but Web Components seem to be a good fit here, b/c they can be used with any framework – at least, in theory. They also provide a bit of isolation when it comes to rendering and CSS due to the usage of Shadow DOM. [Angular Elements](#) come very handy for this. The idea is to compile Angular Components down to Web Components.



important the architectural goals are for you. E. g. I've seen many teams writing successful applications

leveraging libraries and at companies like Google and Facebook there is a long tradition of u

Also, I expect that Web Components will be used more and more due to the growing framework and browser support.

[JETZT ANFRAGEN!](#)

Some guidance

One way to work with the presented matrix is to extend it with all the other goals you have and evaluate it against your specific situation. Even though this seems to be straight forward, in practice it can be quite difficult. That's why I've decided to give you some (biased) guidance by pointing out the strength of the approaches presented:

If ...

1) You need just use one microfrontend at one time and have little/ no communication on UI level between them

Try ...

Hyperlinks

2) Otherwise: Legacy and need for very strong isolation

iframes



4) Otherwise

Bootstrap several SPAs in one browser w
DOM for Isolation, consider Web Components

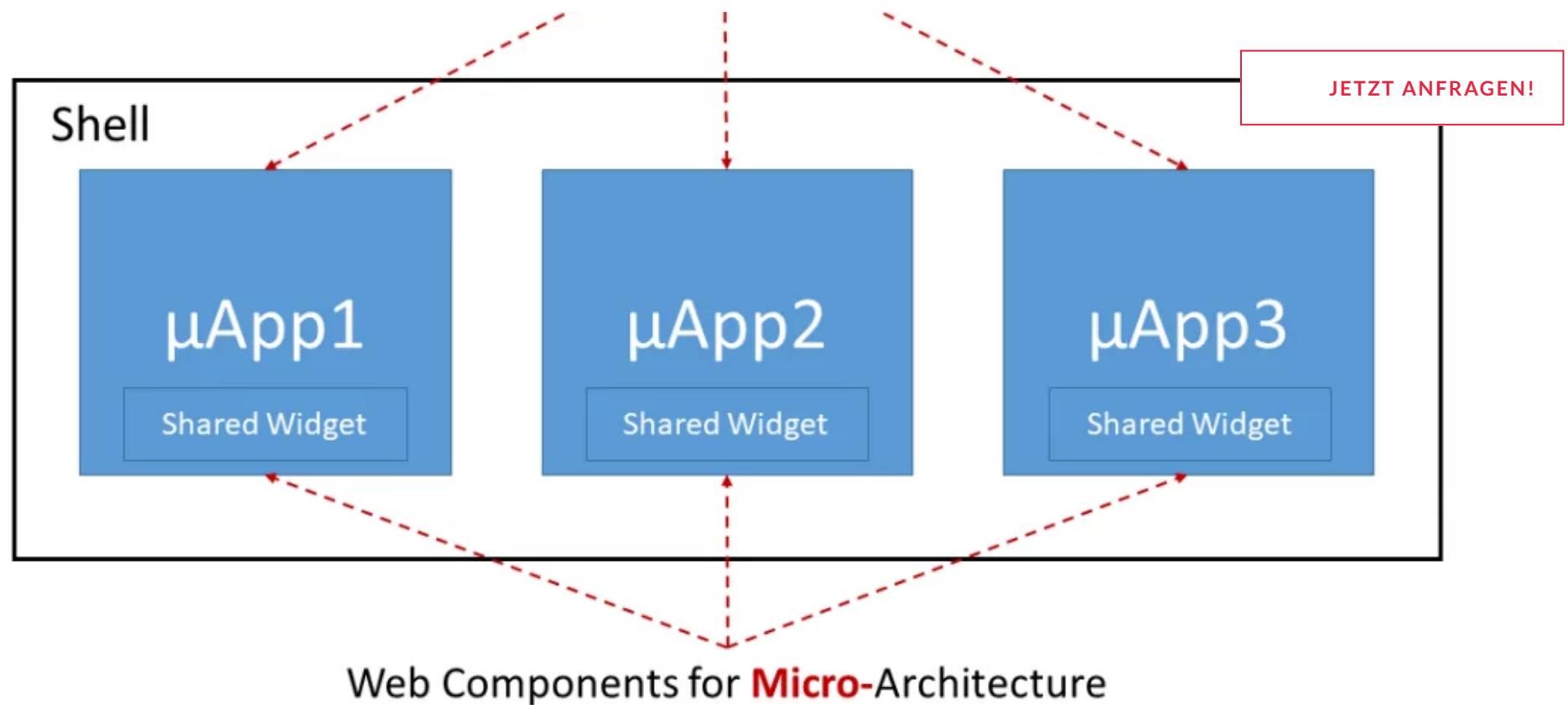
JETZT ANFRAGEN!

If you don't have a legacy system and you've already decided to go with microservices, you should consider 1)
and 4) in this very order.

Distinguish between Macro and Micro Architecture

When choosing a solution, please keep in mind that you need solutions for two different architecture levels.

I'm calling them the macro and the micro architecture:



While the macro architecture is the glue between your micro apps, the micro architecture is the architecture within them. For me, this separation turned out to be handy, because you can choose for different solutions on those two levels. While this picture uses web components for both, one could also decide for using iframes



ANGULAR SCHULUNG | BERATUNG | PRINT | VORTRÄGE | BLOG | TEAM

JETZT ANFRAGEN!

It's not an “either/or thing”!

Mixing those approaches can also be a good idea. For instance, you could go with Hyperlinks for the general routing and when you have to display widgets from one microfrontend within an other one, you could choose for libraries or web components.

Enjoyed this Article?

Feel free to share it on social media
and subscribe to our newsletter



ANGULAR SCHULUNG | BERATUNG | PRINT | VORTRÄGE | BLOG | TEAM

Subscribe to our newsletter to get all the information about Angular.

JETZT ANFRAGEN!

Business EMail Address*:

Country*

Subscribe*

* By subscribing to our newsletter, you agree with our [privacy policy](#).



ANGULAR SCHULUNG | BERATUNG | PRINT | VORTRÄGE | BLOG | TEAM



Angular Schulung: Strukturierte Einführung

In dieser strukturierten Einführung lernen Einsteiger und Autodidakten alle Building-Blocks...

[MEHR INFORMATIONEN](#)

[JETZT ANFRAGEN!](#)



Angular Architektur Workshop

In diesem weiterführenden Intensiv-Kurs lernen Sie, wie sich große und...

[MEHR INFORMATIONEN](#)

[JETZT ANFRAGEN!](#)

[JETZT ANFRAGEN!](#)

Micro Frontends mit

Lernen Sie große Angular-Lösungen und Micro Frontends zu strukturieren...

[MEHR INFORMATIONEN](#)

[JETZT ANFRAGEN!](#)

Professional Angular Testing

Reaktive Angular-Architekturen mit RxJS und NGRX (Redux)

Professional NGF Advanced State Management Best Practices

[MEHR INFORMATIONEN](#)[JETZT ANFRAGEN!](#)

Angular Migration Workshop

Wir zeigen Ihre Optionen für eine Migration nach Angular auf und erstellen mit Ihnen einen Proof-of-Concept in Ihrer Codebasis.

[MEHR INFORMATIONEN](#)[JETZT ANFRAGEN!](#)[MEHR INFORMATIONEN](#)[JETZT ANFRAGEN!](#)

Moderne .NET-Backends für Angular

Microservices mit .NET (Core)

[MEHR INFORMATIONEN](#)[JETZT ANFRAGEN!](#)[JETZT ANFRAGEN!](#)

TIO

[JETZT ANFRAGEN!](#)



ANGULAR SCHULUNG | BERATUNG | PRINT | VORTRÄGE | BLOG | TEAM

JETZT ANFRAGEN!

Aktuelle Blog-Artikel

ALLE ARTIKEL



ANGULAR SCHULUNG | BERATUNG | PRINT | VORTRÄGE | BLOG | TEAM



VON: MANFRED STEYER, GDE

4 WAYS TO PREPARE FOR ANGULAR'S UPCOMING STANDALONE COMPONENTS

With the introduction of Standalone Components, NgModules will become optional. But how to prepare for...

[MEHR ERFAHREN](#)



VON: MANFRED STEYER, GDE

STATE-MANAGEMENT MIT

In diesem Tutorial lernen Sie alles über NGRX in Angular. Lern Sie wie Sie State-Management mit NgRx und NgRx Effects verwenden.

[MEHR ERFAHREN](#)

Nur einen Schritt entfernt!



ANGULAR SCHULUNG | BERATUNG | PRINT | VORTRÄGE | BLOG | TEAM

JETZT ANFRAGEN!



ebenfalls berufsbegleitend Computer Science in Hagen studiert und eine vier-semestrige Ausbildung im Bereich der Erwachsenenbildung abgeschlossen.

JETZT ANFRAGEN

manfred.steyer@softwarearchitekt.at

NEWSLETTER ABONNIEREN



zum Zwecke des Versands des Newsletters verarbeiten kann.

DATENSCHUTZ.

Email:

ZU NEWSLETTER ANMELDEN!



ANGULAR SCHULUNG | BERATUNG | PRINT | VORTRÄGE | BLOG | TEAM

JETZT ANFRAGEN!