

Documentation-as-Code – Dokumentation kontinuierlich und automatisiert erstellen

FALK SIPPACH // EMBARC

OOP 2022, Online

Donnerstag, 03.02.2022



1

Dokumentation kontinuierlich und automatisiert erstellen

Im Gegensatz zu den klassischen Ansätzen (Textverarbeitung, Wiki, Netzlaufwerke, ...) verfolgt Docs-as-Code den Ansatz, die in Softwareprojekten relevante Dokumentation genau wie den Quelltext zu behandeln. Für die entwickelnden Team-Mitglieder entsteht somit kein Medienbruch. Sie können die gleichen Werkzeuge (Texteditor, IDE, Build-Tools, CI/CD-Pipeline) verwenden, um die Inhalte als leichtgewichtige Textformate neben dem Sourcecode in der Versionsverwaltung abzulegen, die Änderungen gemeinsam mit dem Quellen zu reviewen, als Release zu markieren und auszuliefern sowie die zielgruppenorientierte Zusammenstellung der Ergebnisse in den Build-Prozess einzubinden. Um Redundanzen zu vermeiden, können aus den vorhandenen Projektmodellen (Sourcecode, DB-, UML-Modell, ...) automatisiert textuelle Inhalte und Diagramme generiert werden.

Jedwede Art von Dokumentation gewinnt somit an Sichtbarkeit, durch die Eingliederung in die Entwicklungsprozesse und die damit verbundene kontinuierliche Weiterentwicklung steigt die Qualität und damit die Akzeptanz bei den Lesern. Dokumentation kann sogar ausgeführt werden, um zum Beispiel eingebettete Architektur-Regeln regelmäßig automatisiert zu testen. Die Zuhörer erfahren in diesem Vortrag, wie sie mit Documentation as Code starten können, welche typischen Fallstricke sie umschiffen sollten und welche konkreten Tools bekannt sein sollten.



2

Falk Sippach

- Softwarearchitekt, Berater, Trainer bei embarc
- früher bei Orientation in Objects (OIO), Trivadis

Schwerpunkte:

- Architekturberatung und -bewertung
- Cloud- und Java-Technologien



✉ fs@embarc.de

🐦 @sipsack

➔ xing.to/fsi



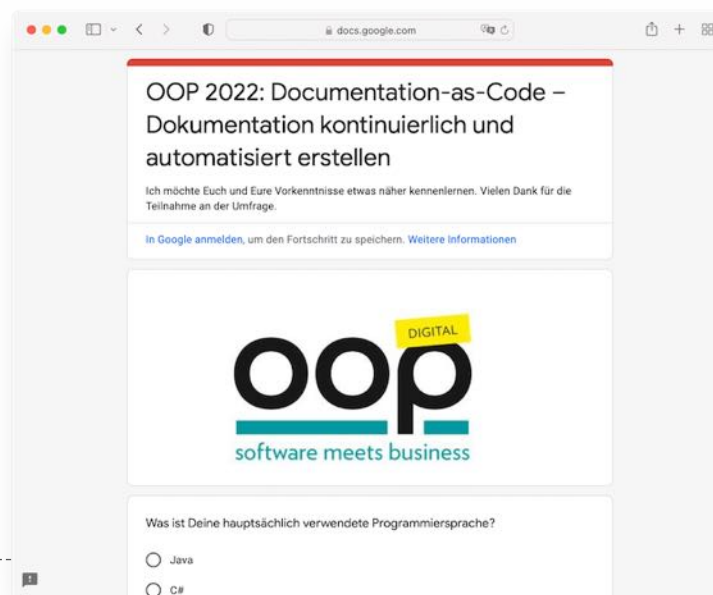
Documentation-as-Code

embarc.de

3

3

Wer seid Ihr?



OOP 2022: Documentation-as-Code –
Dokumentation kontinuierlich und
automatisiert erstellen

Ich möchte Euch und Eure Vorkenntnisse etwas näher kennenlernen. Vielen Dank für die
Teilnahme an der Umfrage.

[In Google anmelden](#), um den Fortschritt zu speichern. [Weitere Informationen](#)

oop DIGITAL
software meets business

Was ist Deine hauptsächlich verwendete Programmiersprache?

☐ Java

☐ C#



Documentation-as-Code

6

6



sw_architecture_handson

@HandsonSw

...

Antwort an @HandsonSw @grafandreas und 3 weitere Personen

I think this "docs-as-code", "architecture-as-code" thing is currently at the beginning. People just start to find out how much sense it makes to embed the architecture documents into your git repo in a branch/mergable way. Sure we will see some great solutions in the future.

[Tweet übersetzen](#)

11:56 vorm. · 2. Okt. 2021 · Twitter Web App

4 Retweets 1 Tweet zitieren 24 „Gefällt mir“-Angaben



Documentation-as-Code

embarc.de

7

7

Agenda



- 1 Einstieg und Motivation
- 2 Architektur dokumentieren
- 3 Docs-as-Code Maturity Level
- 4 Fazit und Ausblick



Documentation-as-Code

embarc.de

8

8

Agenda



1

- 1 Einstieg und Motivation
- 2 Architektur dokumentieren
- 3 Docs-as-Code Maturity Level
- 4 Fazit und Ausblick

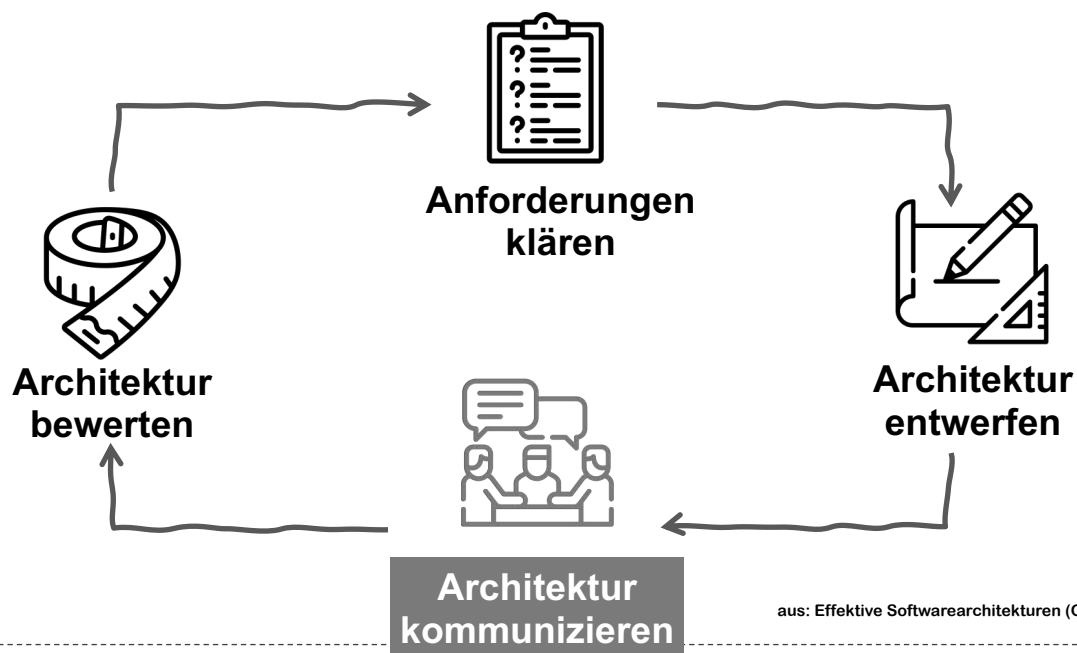


Documentation-as-Code

embarc.de

9

9



aus: Effektive Softwarearchitekturen (G. Starke)



Documentation-as-Code

embarc.de

10

10

10

Where's the fun in just knowing what the code is supposed to do?



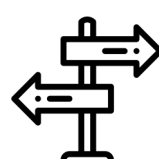
Essential
Excuses for Not Writing Documentation

ORLY? @ThePracticalDev

Grafik von The Practical Dev (CC0 Public Domain Lizenz)



Entwurfs-
unterstützung



Frage nach
dem Warum



Bewertbarkeit



Neue Mitarbeiter



Kommunikation

Documentation-as-Code embarc.de 11 **11**

11



Liest sowieso keiner!

Ungeeignete Werkzeuge

Veraltet

Medienbruch

Handarbeit

Dateiablage

Textwüsten

Redundanzen

Documentation-as-Code embarc.de **12**

12

Agenda



2

- 1 Einstieg und Motivation
- 2 Architektur dokumentieren**
- 3 Docs-as-Code Maturity Level
- 4 Fazit und Ausblick



Documentation-as-Code

embarc.de

13

13

7 Regeln für gute Dokumentation



1. Schreibe aus Sicht des Lesers
2. Vermeide unnötige Wiederholungen
3. Vermeide Mehrdeutigkeiten
 3. a) Erkläre Deine Notation
4. Verwende eine Standardstrukturierung
5. Halte Begründungen für Entscheidungen fest
6. Halte die Dokumentation aktuell, aber auch nicht zu aktuell
7. Überprüfe Dokumentation auf ihre Gebrauchstauglichkeit

„Documenting Software Architectures:
Views and Beyond“
Clements, et.al, 2. Auflage 2010



Documentation-as-Code

embarc.de

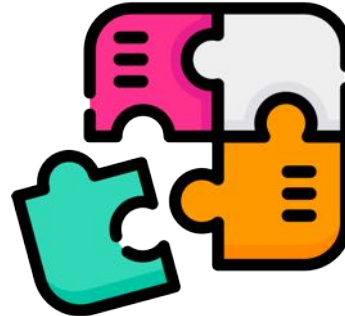
14

14

Architekturentwurf



Anforderungen/
Architekturziele



Lösungsansätze/
Entscheidungen



Documentation-as-Code

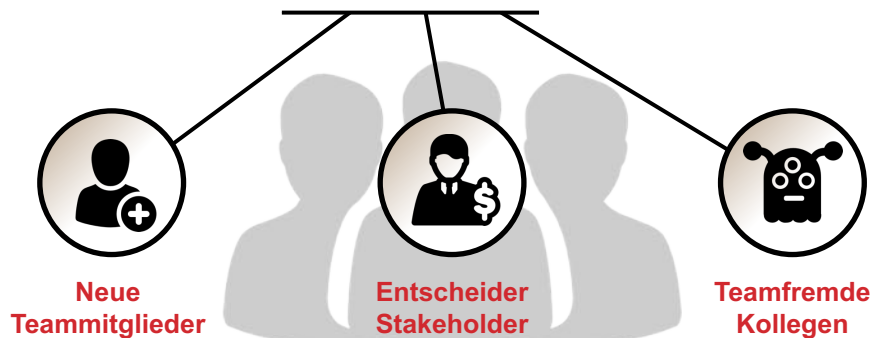
embarc.de

15

15

Was ist ein Architekturüberblick?

Ein Architekturüberblick macht die zentralen Lösungsansätze Eurer Softwarearchitektur für Außenstehende nachvollziehbar.



Documentation-as-Code

embarc.de

16

16

Zutaten für einen Architekturüberblick



Qualitätsziele



Vorgaben



Systemkontext



Metapher
"Produktkarton"



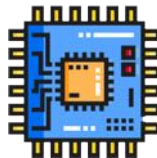
Informeller
Überblick



Architektur-Stil/-
Muster/...



Entscheidungen



Technologien



Documentation-as-Code

embarc.de

17

17

Top-Qualitätsziele Corona-Warn-App

Ziel	Beschreibung
 Höchster Datenschutz	Der Schutz der personenbezogenen Daten hat oberste Priorität. (<i>Sicherheit</i>)
 Effektive Warnfunktionalität	Die App ist ein effektiver Baustein bei der Pandemie-Bekämpfung. (<i>Funktionale Eignung</i>)
 Attraktive Lösung für App-Nutzer	Die App ist leicht zu installieren sowie intuitiv und effizient zu bedienen. (<i>Benutzbarkeit</i>)
 Hohe Zuverlässigkeit	Die Lösung geht mit Lastspitzen wegen hoher Nutzer- oder Infektionszahlen ebenso souverän um, wie mit böswilligen Angriffen. (<i>Zuverlässigkeit</i>)
 Gute Änderbarkeit	Die Software lässt sich leicht anpassen, wenn z. B. Nutzer/-innen oder neue Forschungsergebnisse es erfordern. (<i>Wartbarkeit/Erweiterbarkeit</i>)

Die Reihenfolge gibt Orientierung bezüglich der Wichtigkeit.



Documentation-as-Code

embarc.de

18

18

Zentrale Rahmenbedingungen der CWA



Technisch

- Betrieb in der **Cloud**
- **Native mobile** Clients
- Einsatz des **Exposure Notification Framework**

Organisatorisch

- **Große Medienaufmerksamkeit**, gewisse **Skepsis** am Mehrwert innerhalb der Bevölkerung
- Konsortium aus **zwei Auftragnehmern** (SAP und Deutsche Telekom)
- **Enger Zeitrahmen**
- Hoher **politischer Druck**, viele Parteien involviert (Ministerien, Behörden, RKI)
- **Hohe Datenschutzerfordernisse**



Documentation-as-Code

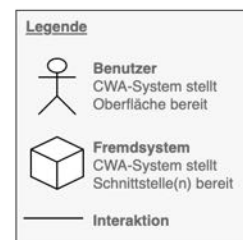
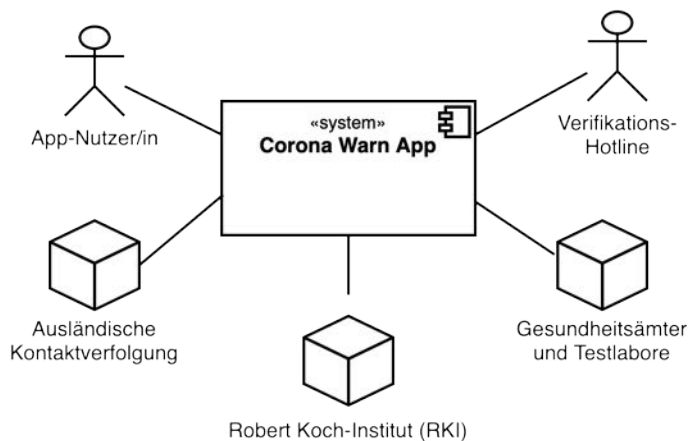
embarc.de

19

19

Kontextabgrenzung

Dieser fachliche Systemkontext zeigt das Corona-Warn-App-System im Zusammenspiel mit den wichtigsten Benutzern und Fremdsystemen.



Documentation-as-Code

embarc.de

20

20

Fachlicher Systemkontext, Akteure

Kurze Erläuterungen zu den Benutzern und Fremdsystemen

Akteur	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Robert Koch-Institut (RKI)	Stellt Inhalte ("Content") für die App zur Verfügung und bestimmt Parameter für die Messung der Kontakte ("Risiko-Ermittlung"). Empfängt Auswertungen, etwa aus der Datenspende.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



Documentation-as-Code

embarc.de

21

21

Mission Statement

Die **Corona-Warn-App** ist eine App, die hilft, **Infektionsketten** des SARS-CoV-2 (COVID-19-Auslöser) in Deutschland **nachzuverfolgen** und zu **unterbrechen**.

Die App basiert auf Technologien mit einem **dezentralisierten Ansatz** und informiert Personen, wenn sie mit einer infizierten Person in Kontakt standen.

Transparenz ist von entscheidender **Bedeutung**, um die Bevölkerung zu schützen und die **Akzeptanz zu erhöhen**.



Quelle: <https://www.coronawarn.app/de/>



Documentation-as-Code

embarc.de

22

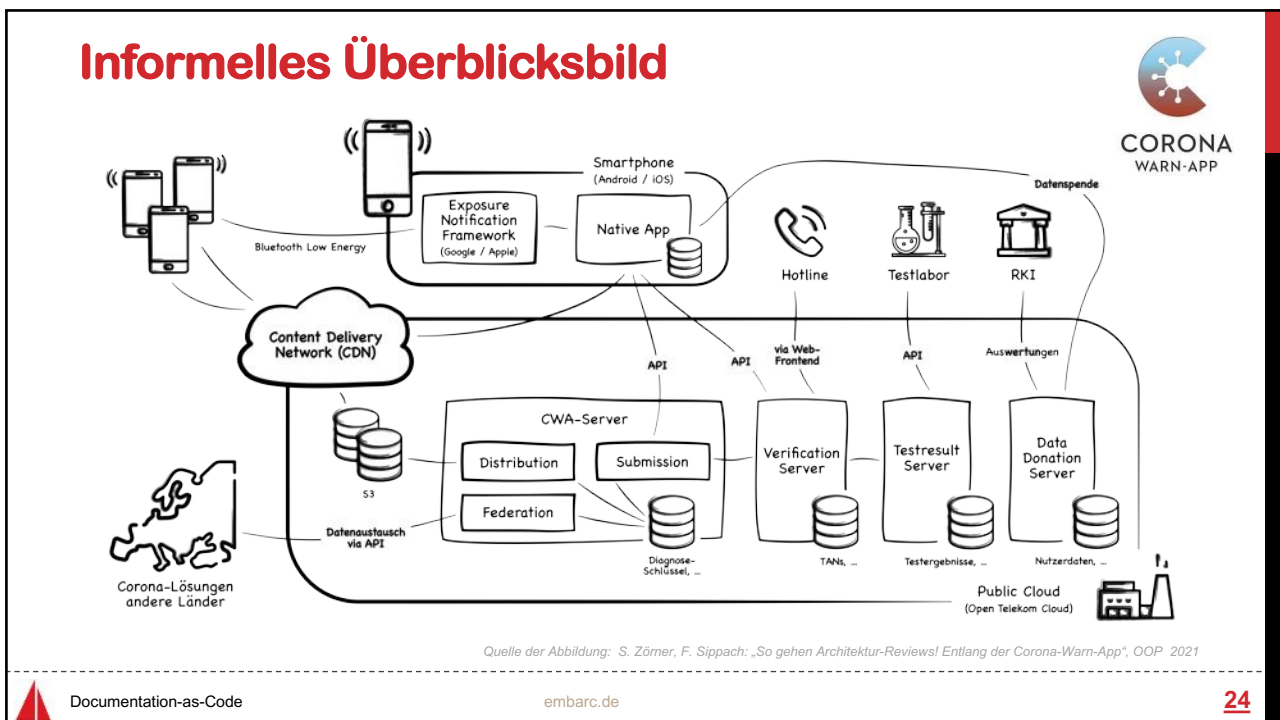
22

Bewerbungs Genie

Ganz einfach überzeugende Bewerbungen erstellen

- Simple Gestaltung mit professionellen Design-Vorlagen
- Perfekt formulierte Textbausteine
- Individuelle Lebensläufe und Anschreiben (auch auf Englisch)
- Ideal für Print- und Onlinebewerbungen
- Für Einsteiger, Young Professionals und Berufserfahrene

23



24

Lösungsstrategie Corona-Warn-App

Ziel	Passende Lösungsansätze (Auswahl)
 Höchster Datenschutz	• Speicherung der Daten lokal • Verschlüsselung aller Bewegungsdaten • Senden der Daten nur nach Aufforderung • Transparente Entwicklung (Open Source)
 Effektive Warnfunktionalität	• Verwendung Exposure Notification Framework • Digitale Abläufe bevorzugt • Optionales, manuelles Kontakt-Tagebuch
 Attraktive Lösung für App-Nutzer	• Native Clients (L&F) • Übersichtliche Gestaltung und simple Bedienung
 Hohe Zuverlässigkeit	• Microservices, Docker, Kubernetes, Public Cloud • hohe Testabdeckung und automatisierte Builds • Bereitstellung von zu lesenden Daten über CDN
 Gute Änderbarkeit	• Hoher Modularisierungsgrad • Open Source Projekt, gute Dokumentation • Verwendung von Standard & Open Source Libraries • Konsortium von mehreren Auftragnehmern • Code-Qualität (SonarQube, SwiftLint, Checkstyle, ...)



Documentation-as-Code

embarc.de

25

25

Technologie-Stack



- **Native Clients** in Swift bzw. Kotlin für iOS und Android
- **SQLite**



- **Java 11**
- **Spring Boot/Cloud/Data**
- Lombok, Guava, ...
- **REST, Protobuf**
- **OpenAPI**, Micrometer
- Liquibase



- Maven, Gradle
- **Docker, Kubernetes**
- **Open Telekom Cloud** (OpenStack)
- PostgreSQL, **S3**, **CDN**
- Keycloak

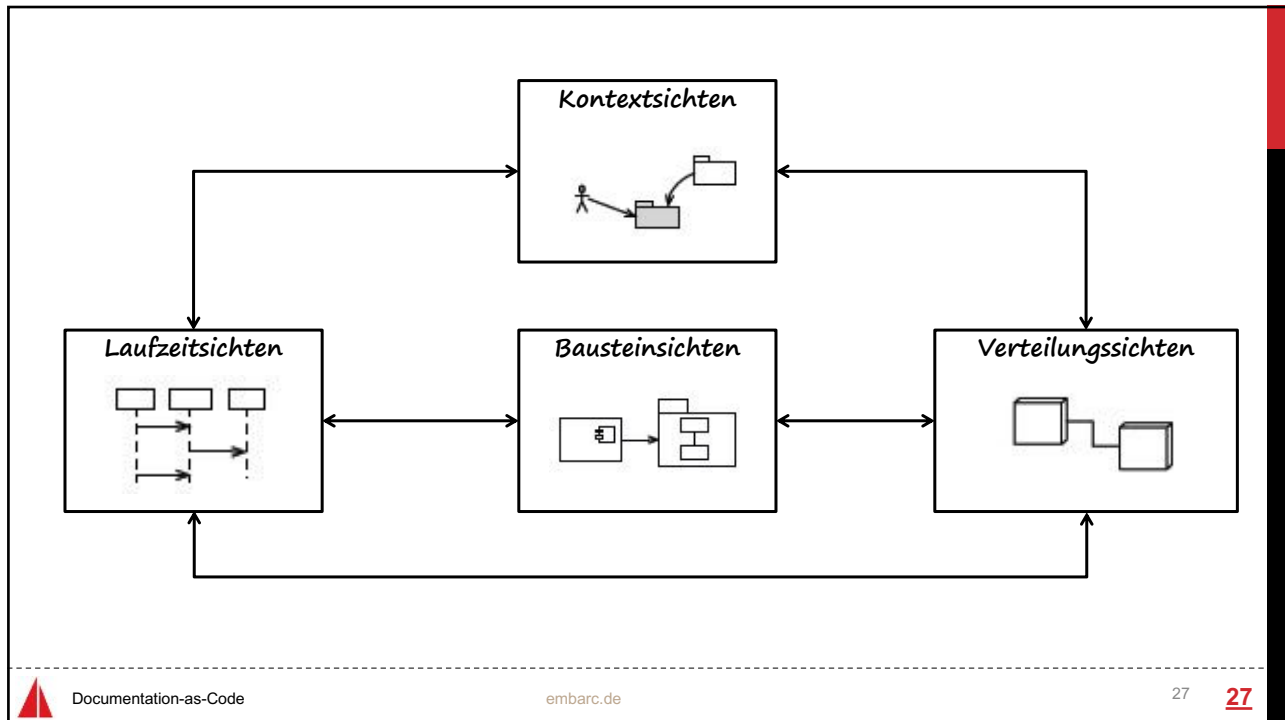


Documentation-as-Code

embarc.de

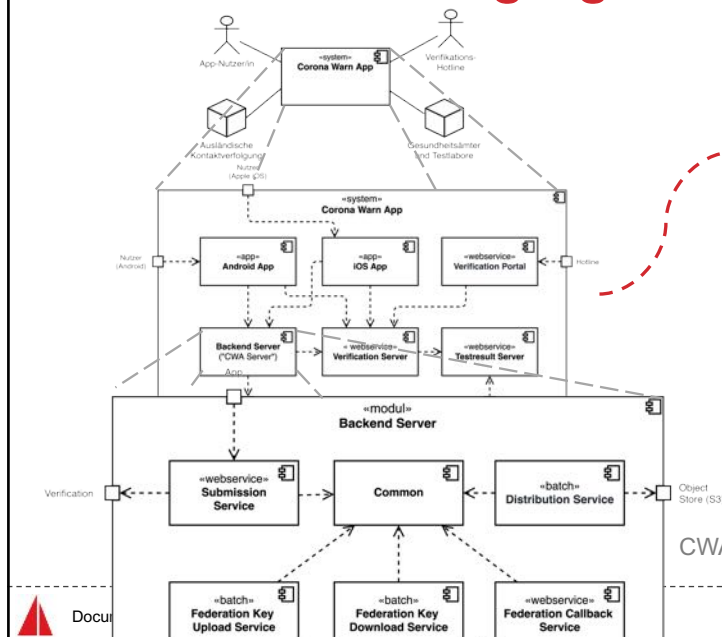
26

26



27

Tatsächliche Zerlegung im Quelltext



„Bausteinsicht, Ebene 1“

GitHub-Repositories

[https://github.com/corona-warn-app/...](https://github.com/corona-warn-app/)

- cwa-app-android
- cwa-app-ios
- cwa-server („Backend“)
- cwa-testresult-server
- cwa-verification-server
- cwa-verification-portal

CWA-Server (Backend), Bausteinsicht, Ebene 2

28

Technische Themen/Cross-cutting concerns

Ziele

- Einheitliche Lösungen, Konsistenz
- weniger Redundanzen
- Bessere Wartbarkeit

Lösungsraum

- (Architektur-)muster
- Bibliotheken, Frameworks
- Aspektorientierte Programmierung
- Generierung



Softwarearchitektur

Eine der Definitionen ...

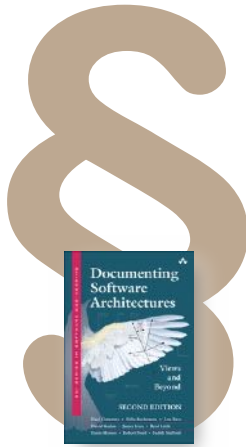


*“Softwarearchitektur ist die **Menge der Entwurfsentscheidungen**, die, **wenn falsch** getroffen, Dein Projekt zum **Scheitern bringen kann**.“**
(Eoin Woods)

* Im Englischen eigentlich: “Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.”



7 Regeln für gute Dokumentation



„Documenting Software Architectures:
Views and Beyond“
Clements, et.al, 2. Auflage 2010

1. Schreibe aus Sicht des Lesers
2. Vermeide unnötige Wiederholungen
3. Vermeide Mehrdeutigkeiten
 3. a) Erkläre Deine Notation
4. Verwende eine Standardstrukturierung
- 5. Halte Begründungen für Entscheidungen fest**
6. Halte die Dokumentation aktuell, aber auch nicht zu aktuell
7. Überprüfe Dokumentation auf ihre Gebrauchstauglichkeit



Documentation-as-Code

embarc.de

31

31

Ein Werkzeug

Zum Entscheidungen treffen und festhalten ...



Documentation-as-Code

embarc.de

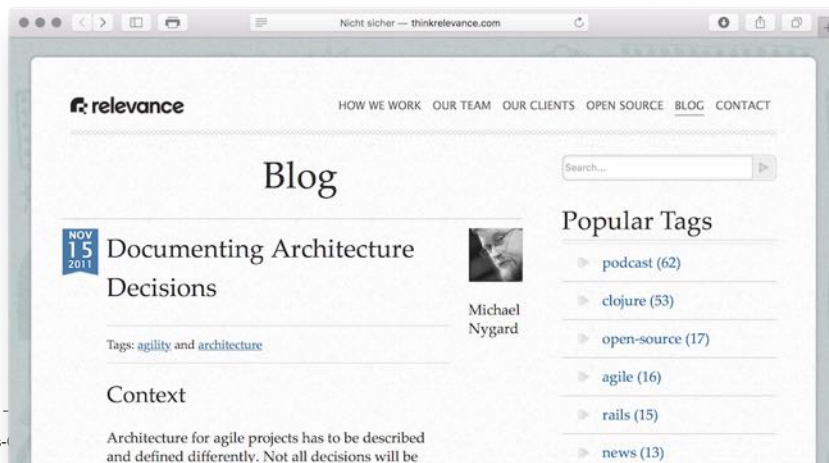
32

32

Alternative: ADRs

Architecture Decision Records (Michael Nygard, 2011)

→ <http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions>



Was ist ein Szenario?



Ein Qualitätsszenario (auch: Bewertungsszenario) ...

- ... ist ein kurzer Text (1-3 Sätze).
- ... beschreibt **beispielhaft** die Verwendung des Systems, und zwar so dass ein **Qualitätsmerkmal** die Hauptrolle spielt.

Wie konkret?



Man muss sinnvoll drüber reden können.
Man muss es (theoretisch) überprüfen können.
(Kein Abnahmekriterium, kein Testfall!)

Agenda



- 1 Einstieg und Motivation
- 2 Architektur dokumentieren
- 3 Docs-as-Code Maturity Level**
- 4 Fazit und Ausblick

3



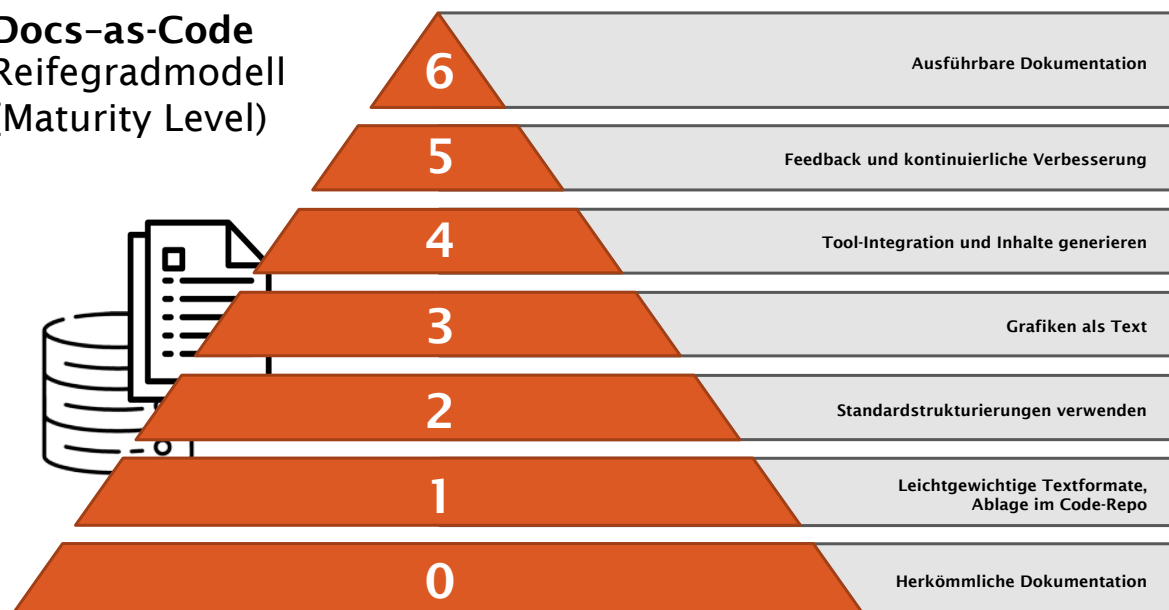
Documentation-as-Code

embarc.de

37

37

Docs-as-Code Reifegradmodell (Maturity Level)



Documentation-as-Code

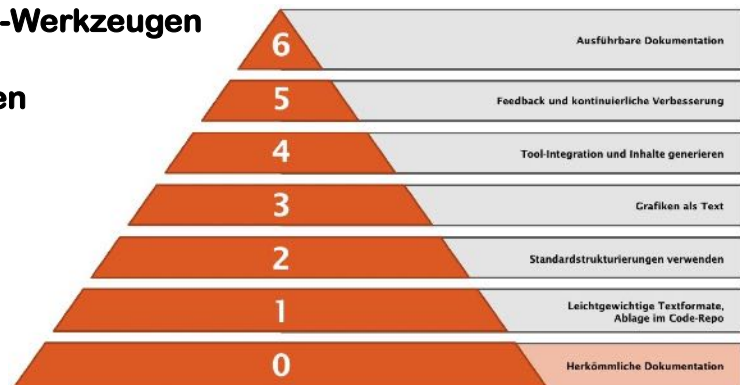
embarc.de

38

38

Herkömmliche Dokumentation

- Dokumentation wird getrennt vom Code verwaltet
- Bearbeitung in WYSIWYG-Werkzeugen
- Ablage auf Netzlaufwerken
- Verwendung von Wikis



Documentation-as-Code

embarc.de

39

39

... der Ort, wo die Dokumente zum Sterben hingehen



Wiki = Where Information Kills Itself

CC BY-SA 3.0, https://de.wikipedia.org/wiki/Ward_Cunningham#/media/File:Ward_Cunningham_-_Commons-1.jpg



Documentation-as-Code

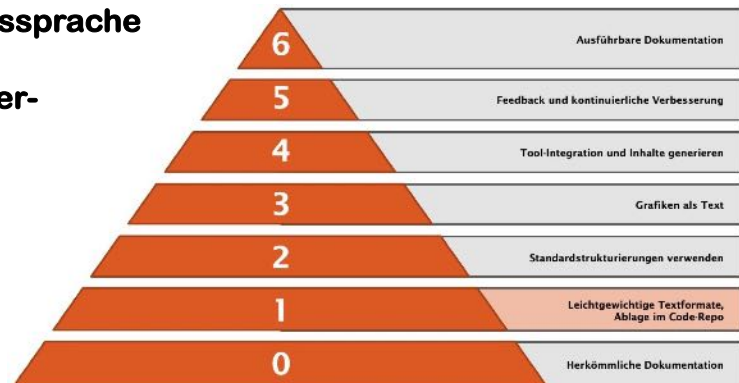
embarc.de

40

40

Leichtgewichtige Textformate, Ablage im Repo

- Dokumentation ist gemeinsam mit dem Quelltext versionierbar
- Wahl einer Auszeichnungssprache
- Bearbeitung mit Entwickler-Werkzeugen



Documentation-as-Code

embarc.de

41

41

Hauptsache,
du machst es
nicht mit
Word!



Documentation-as-Code

embarc.de

42

42

Probleme mit Word und Co.

Binärformat

Mehrbenutzer schwierig

Dokument per Mail verschicken und Änderungen wieder zusammenbringen

Historisierung, Versionen vergleichen

Word ist seitenorientiert (Seitenzahl, Header, Footer)

Subdokumente und Bild-Referenzen (möglich, aber wird kaum verwendet)



Documentation-as-Code

embarc.de

43

43

Unser täglich Entwickler-Brot

- Plain-Text
- Entwicklungsumgebung
- Kommandozeilenwerkzeuge
- **Versionsverwaltung**

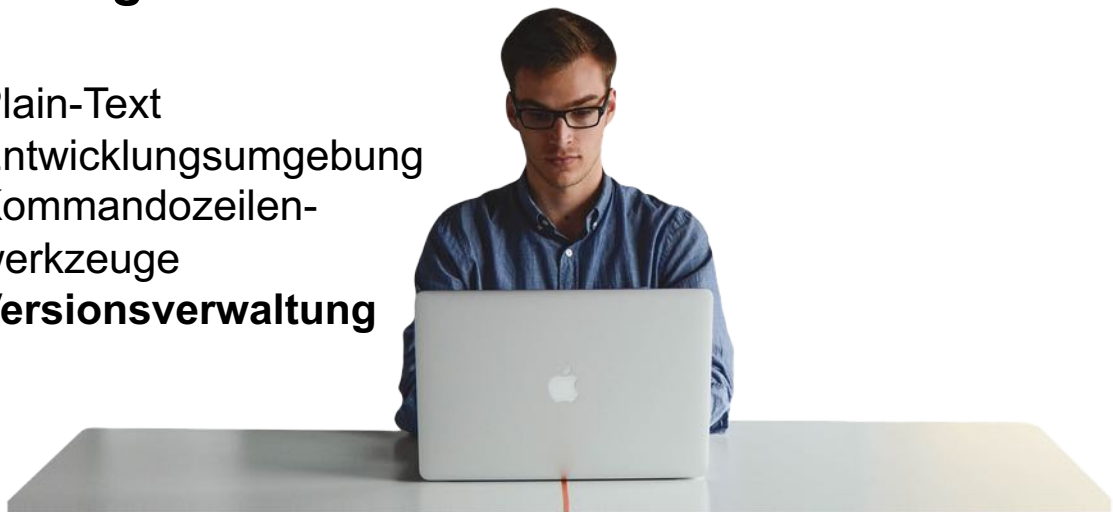


Foto von geralt: <https://pixabay.com/de/unternehmer-start-start-up-karriere-696976/> (CC0 Public Domain Lizenz)



Documentation-as-Code

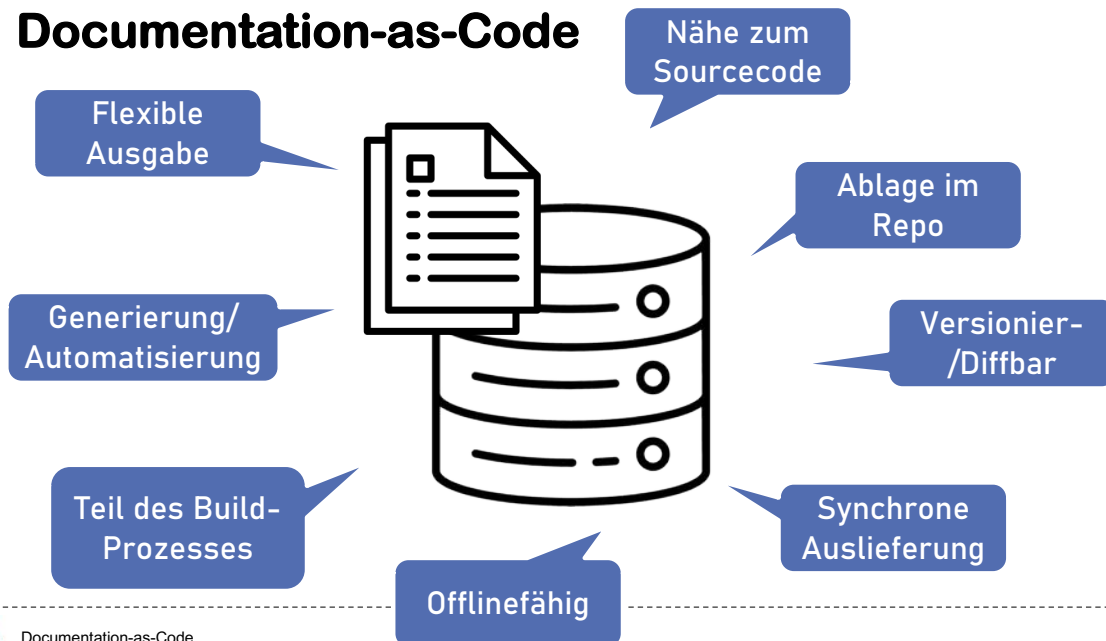
embarc.de

44

44

44

Documentation-as-Code



Documentation-as-Code

45

45



Michael Simons
@rotnrolle65



Folge ich

Think I'm more a Markdown person than AsciiDoc. The results are great, but Markdown needs less concentration to write.



Documentation-as-Code

embarc.de

46

46

Markdown

Markdown

- Toller Standard für einfache Auszeichnungen
- Features

Span Elements
 Links
 Emphasis
 Code
 Images
 Block Elements
 Paragraphs and Line Breaks
 Headers
 Blockquotes
 Lists
 Code Blocks
 Horizontal Rules



TOC
 Tables (Feature Rich)
 Includes (Level-Offset)
 PlantUML
 Admonitions
 Attributes
 Anchors
 Footnotes, Index, Glossary
 Videos
 Syntax Highlighting
 Callouts
 Math Rendering
 Outputformats



Documentation-as-Code

embarc.de

47

47

Markdown

Wir brauchen eine Erweiterung!

Welche wählen wir?

- | | | |
|---|---|--|
| • CommonMark | • Leanpub Flavored Markdown | • Python Markdown |
| • CriticMarkup | • Litedown | • R Markdown |
| • Discount | • Lunamark | • Redcarpet |
| • DocFX | • Madoko | • Remarkable |
| • ExtraMark | • Markdown | • Rhythmus |
| • Ghost's Markdown/Haunted Markdown | • Markdown 2 | • Scholarly Markdown |
| • GitHub Flavored Markdown | • Markdown Extra | • Showdown |
| • GitLab Flavored Markdown (with login) | • Markdown-it | • StackOverflow's Markdown |
| • Haroopad Flavored Markdown | • Markua | • Taiga Markdown |
| • iA Writer's Markdown | • Maruku | • Trello's Markdown |
| • Kramdown | • MultiMarkdown | • vfm |
| | • Pandoc's Markdown | • Xcode/Swift Playgrounds Markup |
| | • PHP Markdown Extra Extended | |

<https://github.com/commonmark/commonmark-spec/wiki/markdown-flavors>



Documentation-as-Code

embarc.de

48

48

Markdown

einen Dialekt

Wir brauchen eine ~~Erweiterung!~~

Welche wählen wir?

Funktioniert dann unsere Toolchain noch?

Gibt es einen Editor-Preview?



AsciiDoc / Asciidoctor

leistungsfähige Syntax für technische Dokumentation

in Ruby geschrieben

mit Opal nach JavaScript transpiliert

mit jRuby auf der JVM gewrapped

=> Keine Dialekte!





Michael Simons
@rotnroll666



Folge ich

Think I'm more a MarkDown person than AsciiDoc. The results are great, but MarkDown needs less concentration to write.



Ralf D. Müller
@RaifDMueller



Folgen

@rotnroll666 but the possibilities of AsciiDoc are better! Include code directly from the repository, render plantUML, subdocuments etc...



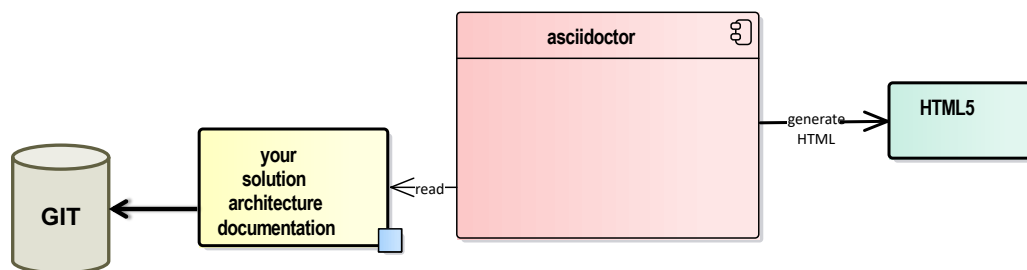
Documentation-as-Code

embarc.de

52

52

Basic Docs-as-Code with AsciiDoc



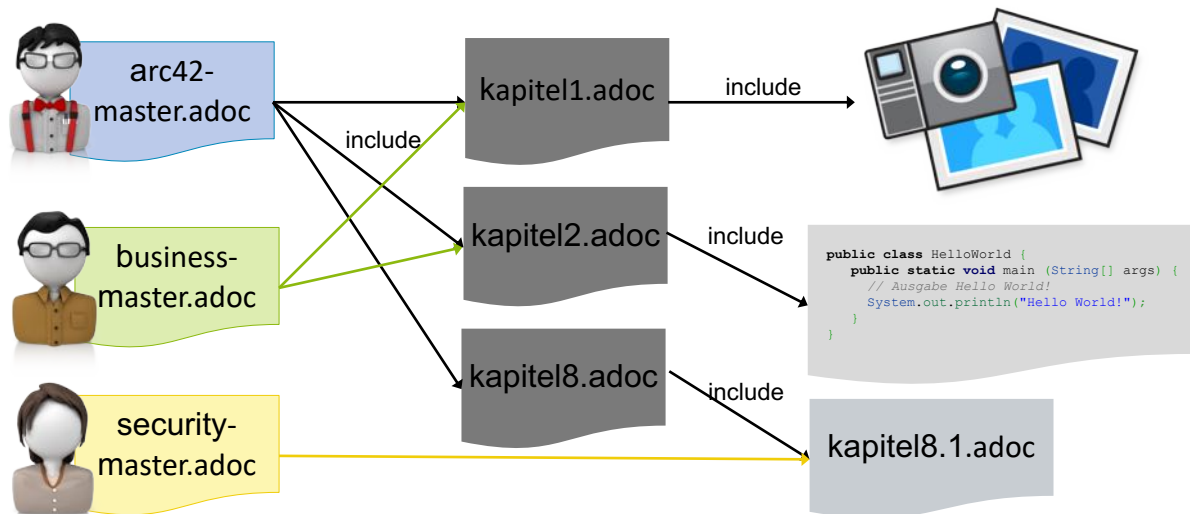
Documentation-as-Code

embarc.de

53

53

Modulare Dokumentation



Out-of-the-Box Features

„ablenkungsfrei“ – Dokumentation wie E-Mails schreiben

Gliederung in Unterdokumente

Neugliederung je nach Stakeholder

Bilder werden referenziert, nicht eingebettet

leichte Versionierung „handle Docs-as-Code“

Formatierung von Source-Code

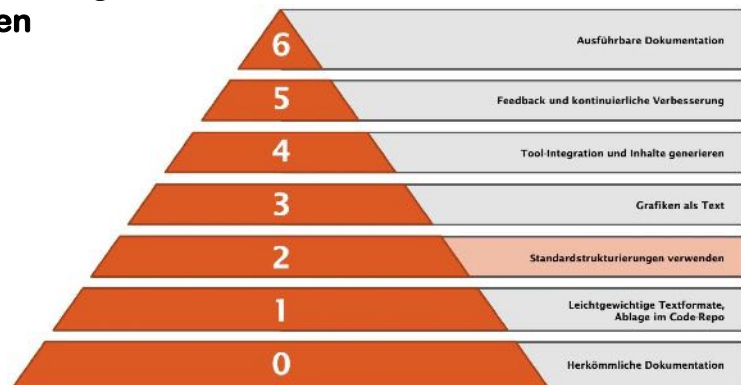
Reviews, Pull-Requests, Versionierung durch Git

Konvertierung nach HTML5 und DocBook



Standardstrukturierungen verwenden

- Leser finden sich leichter zurecht
- Schreiber erhalten Orientierung bei der Erstellung von Inhalten



Documentation-as-Code

embarc.de

56

56

7 Regeln für gute Dokumentation



1. Schreibe aus Sicht des Lesers
2. Vermeide unnötige Wiederholungen
3. Vermeide Mehrdeutigkeiten
 3. a) Erkläre Deine Notation
- 4. Verwende eine Standardstrukturierung**
5. Halte Begründungen für Entscheidungen fest
6. Halte die Dokumentation aktuell, aber auch nicht zu aktuell
7. Überprüfe Dokumentation auf ihre Gebrauchstauglichkeit

„Documenting Software Architectures:
Views and Beyond“
Clements, et.al, 2. Auflage 2010



Documentation-as-Code

embarc.de

57

57

„Abheften“ in arc42

1. Einführung und Ziele

- 1.1 Aufgabenstellung
- 1.2 Qualitätsziele
- 1.3 Stakeholder

2. Randbedingungen

- 2.1 Technische Randbedingungen
- 2.2 Organisatorische Randbedingungen
- 2.3 Konventionen

3. Kontextabgrenzung

- 3.1 Fachlicher Kontext
- 3.2 Technischer- oder Verteilungskontext

4. Lösungsstrategie

5. Bausteinsicht

- 5.1 Ebene 1
- 5.2 Ebene 2
- ...

6. Laufzeitsicht

- 6.1 Laufzeitszenario 1
- 6.2 Laufzeitszenario 2
- ...

7. Verteilungssicht

- 7.1 Infrastruktur Ebene 1
- 7.2 Infrastruktur Ebene 2
- ...

8. Konzepte

- 8.1 Fachliche Strukturen und Modelle
- 8.2 Typische Muster und Strukturen
- 8.3 Persistenz
- 8.4 Benutzungsoberfläche
- ...

9. Entwurfsentscheidungen

- 9.1 Entwurfsentscheidung 1
- 9.2 Entwurfsentscheidung 2
- ...

10. Qualitätsszenarien

- 10.1 Qualitätsbaum
- 10.2 Bewertungsszenarien

11. Risiken

12. Glossar



Dr. Peter Hruschka

<http://www.peterhruschka.eu/>



Dr. Gernot Starke

<http://gernotstarke.de/>



<https://arc42.de/>



Documentation-as-Code

embarc.de

58

58

1. Requirements & Goals

2. Constraints

3. Scope & Context

4. Solution Strategy

5. Building Block View

6. Runtime View



7. Deployment View

8. Crosscutting Concepts

9. Decisions

10. Quality Scenarios

11. Risks & Tech Debt

12. Glossary



Documentation-as-Code

embarc.de

59

59

Ein anderer Blick: Simon Brown



Simon Brown:
Software Architecture for Developers

Technical leadership by coding, coaching, collaboration, architecture sketching and just enough up front design

→ <http://leanpub.com/>



Documentation-as-Code

embarc.de

60

60

Ein anderer Blick: Simon Brown



Simon Brown:
Software Architecture for Developers, Volume 2

Visualize, document and explore your software architecture

→ <http://leanpub.com/>



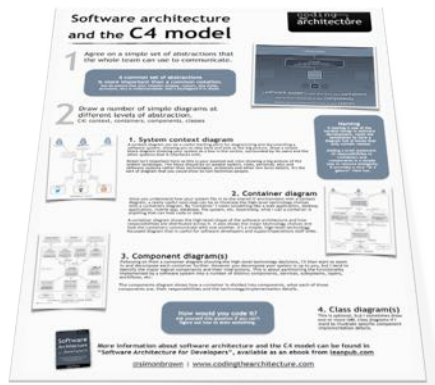
Documentation-as-Code

embarc.de

61

61

Simon Brown: C4 Model



- Agree on a simple set of abstractions that the whole team can use to communicate.
- Draw a number of simple diagrams at different levels of abstraction.
- C4: context, containers, components, classes (code)

→ <http://static.codingthearchitecture.com/c4.pdf>



Documentation-as-Code

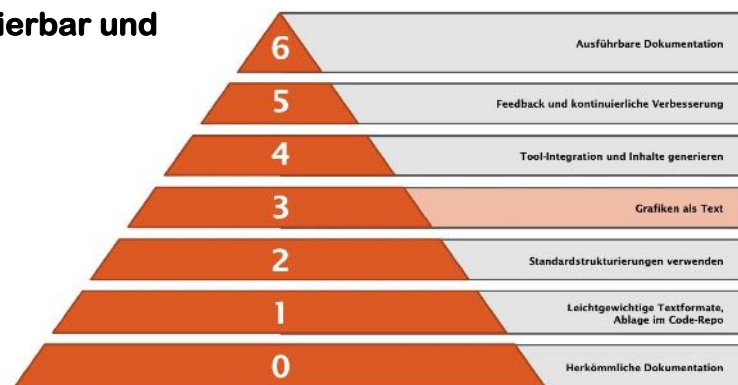
embarc.de

62

62

Grafiken als Text

- **Mindert Risiko des Vendor-Lock-In bei Diagrammen**
- **macht Graphiken versionierbar und diffbar**



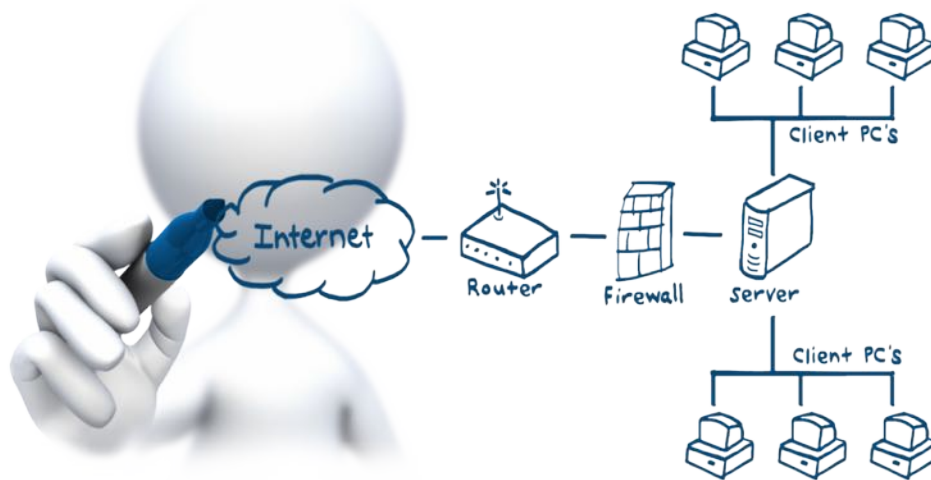
Documentation-as-Code

embarc.de

63

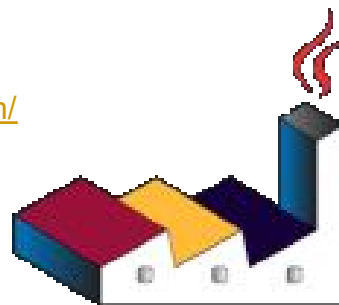
63

Diagramme



Diagramme

- <http://plantuml.com/>
- <http://asciidoctor.org/docs/asciidoctor-diagram/>
- Komplexe Diagramme als einfachen Text verwalten



PlantUML

```

1 @startuml
2
3 User -> Server: via Browser
4   Server -> Database: fetch profile
5   Server <-- Database
6 User <-- Server
7
8 @enduml

```

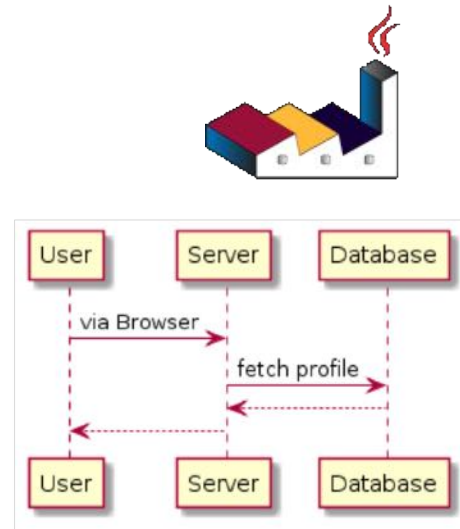


Diagramme als textuelle Beschreibung

```

@startuml
skinparam handwritten true

actor :Conference Participant:

cloud "DukeCon Infra" #lightgray {
    [DukeCon] <<Service>>
}

cloud "DOAG" [
    [Backoffice] <<Service>>
    [Vortragsplaner] <<Service>>
]

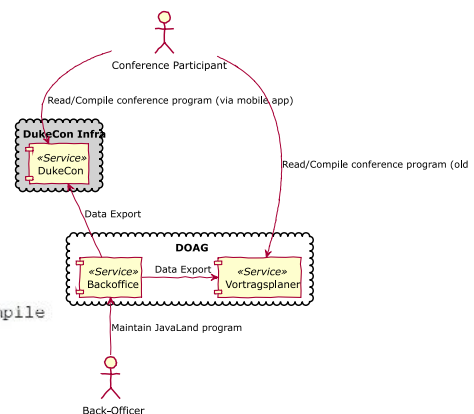
:Conference Participant: -right-> (DukeCon) : Read/Compile conference program (via mobile app)
:Conference Participant: -right-> (Vortragsplaner) : Read/Compile conference program (old style)

(Backoffice) -left-> (DukeCon) : Data Export
(Backoffice) -left-> (Vortragsplaner) : Data Export

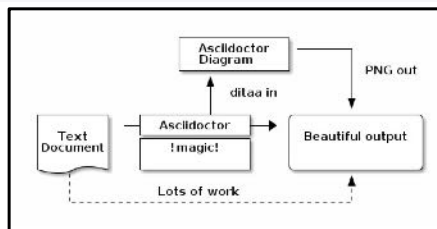
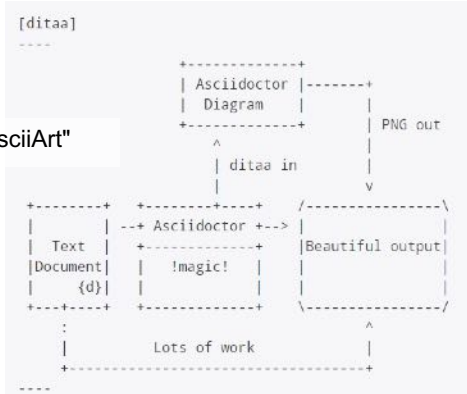
actor :Back-Officer:

["plantuml", "dukecon-systemcontext", "svg"]
include:../systemcontext.puml[]
@enduml

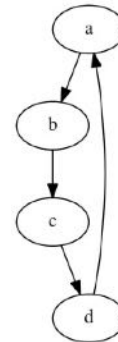
```



"AsciiArt"



```
[graphviz, dot-example, svg]
-----
digraph g {
  a -> b
  b -> c
  c -> d
  d -> a
}
-----
```



Documentation-as-Code

embarc.de

68

68

Kroki

Creates **diagrams** from **textual** descriptions!

Kroki provides a unified API with support for BlockDiag (BlockDiag, SeqDiag, ActDiag, NwDiag, PacketDiag, RackDiag), BPMN, Bytefield, C4 (with PlantUML), Ditaa, Erd, Excalidraw, GraphViz, Mermaid, Nomnoml, PlantUML, SvgBob, UMLet, Vega, Vega-Lite, WaveDrom... and more to come!

#Features

Ready to use

Diagrams libraries are written in a variety of languages: Haskell, Python, JavaScript, Go, PHP, Java... some also have C bindings. Trust us, you have better things to do than install all the requirements to use them. Get started in no time!

Simple

Kroki provides a unified API for all the diagram libraries. Learn once use diagrams anywhere!

Free & Open source

All the code is available on GitHub and our goal is to provide Kroki as a free service.

Fast

Built using a modern architecture, Kroki offers great performance.

<https://kroki.io/>

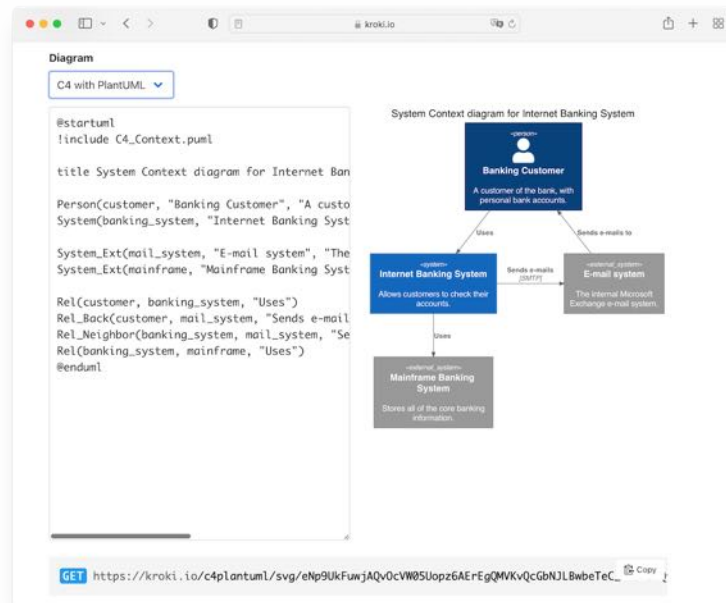


Documentation-as-Code

embarc.de

69

69



Documentation-as-Code

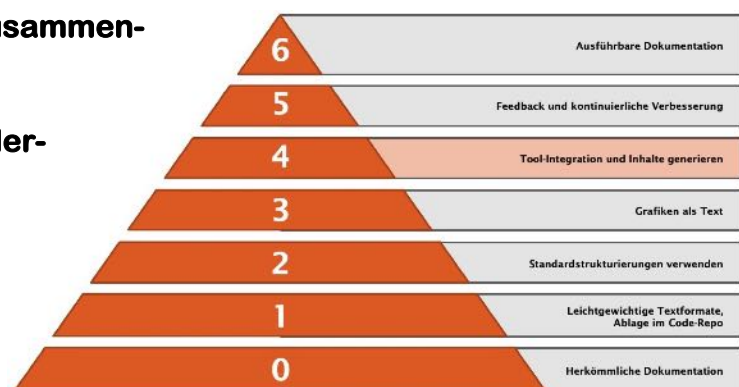
embarc.de

70

70

Tool-Integration und Inhalte generieren

- automatisierte Erstellung im Build-Prozess
- zielgruppenorientierte Zusammenstellung der Ergebnisse
- Verwendung der Entwicklerwerkzeuge ohne Kontextwechsel
- einfache Im- und Exporte



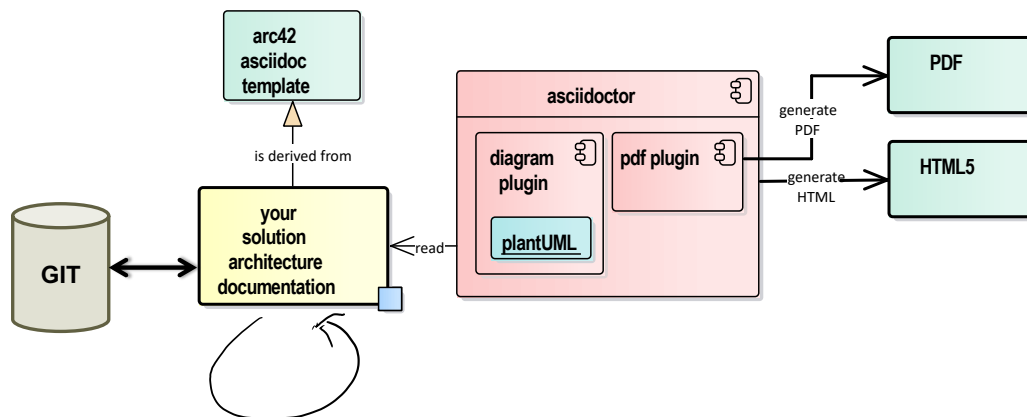
Documentation-as-Code

embarc.de

71

71

Einfacher Workflow mit AsciiDoctor



72

Docs-as-Code in der IDE



73

Docs-as-Code in der IDE

```

22 == Autovervollständigung
23
24 image::|
25
26 {docfile} (C:/Users/ralfd/AppData/Roaming/JetBrains/IntelliJ
27 auto-attribute.png (845x330)
28 autovervollstaendigung.png (590x508)
29 autovervollständigung.png (884x502)
30 grammar_check.png (881x166)
31 live template.png (558x373)
32 pictures.png (400x289)
33 scratch.html
34 scratch.java
35 scratch.kts
36 scratch.md
37

```

Press Eingabe to insert, Tabulator to replace. [Next Tip](#)



Docs-as-Code in der IDE

```

22 == Autovervollständigung
23
24 include::p
25
26 part2.adoc
27 pictures.png (400x289)
28 auto-attribute.png (845x330)
29 autovervollstaendigung.png (590x508)
30 autovervollständigung.png (884x502)
31 grammar_check.png (881x166)
32 chapter_1.adoc
33 live template.png (558x373)
34
35
36
37

```

Press Strg+. to choose the selected (or first) suggestion and insert a dot afterwards [Next Tip](#)



Docs-as-Code in der IDE

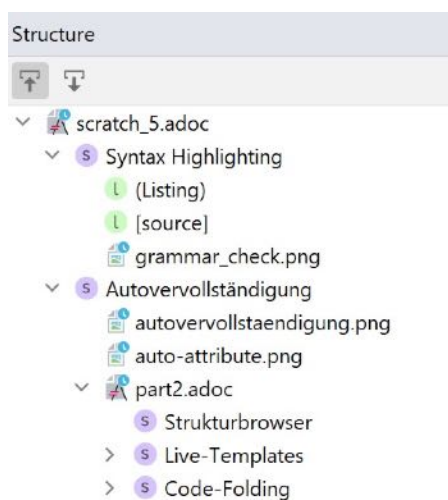
```

22 == Autovervollständigung
23
24 :app-id: 4711
25
26 {app}
27 app-id (4711) scratch_5.adoc
28 app-name (not set)
29 appendix-caption (Appendix) scratch_1.adoc
30 appendix-number (A)
31 appendix-refsig (Appendix)
32 Strg+Unten and Strg+Oben will move caret down and up in the editor. Next Tip
33
34
35
36
37

```



Docs-as-Code in der IDE



Docs-as-Code in der IDE

```

7  == Code-Folding
8
9  === Abschnitt ausgeklappt
10
11  Lorem Ipsum
12
13  === Abschnitt eingeklappt ...

```



Docs-as-Code in der IDE

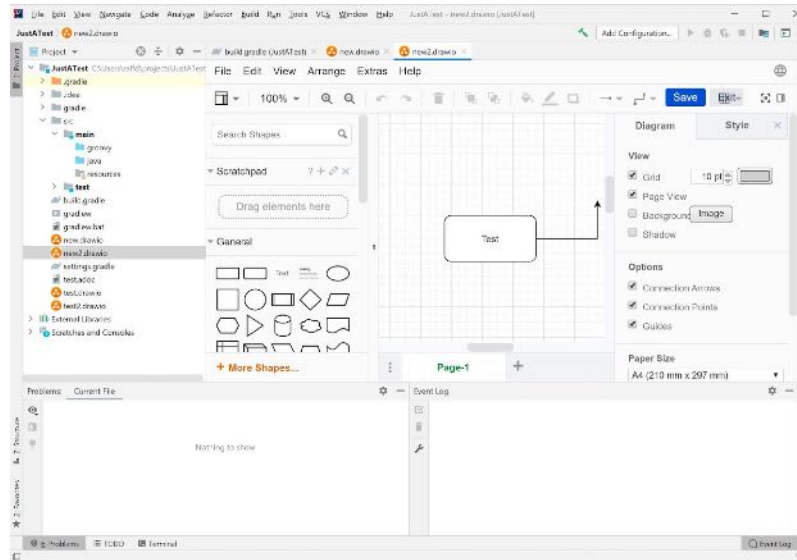
```

3  == Live-Templates
4
5  ad-t
6  ad-table2          Table with two columns
7  ad-tag-include     AsciiDoc Tags to be used with include macro
8  ad-title1          Title 1
9  ad-title2          Title 2
10 ad-title3          Title 3
11 ad-title4          Title 4
12 ad-title5          Title 5
13 ad-title6          Title 6
14 ad-config-toc      config-attributes for table of contents and s...
15 Press Eingabe to insert, Tabulator to replace Next Tip

```



Draw.io/Diagrams.net Integration



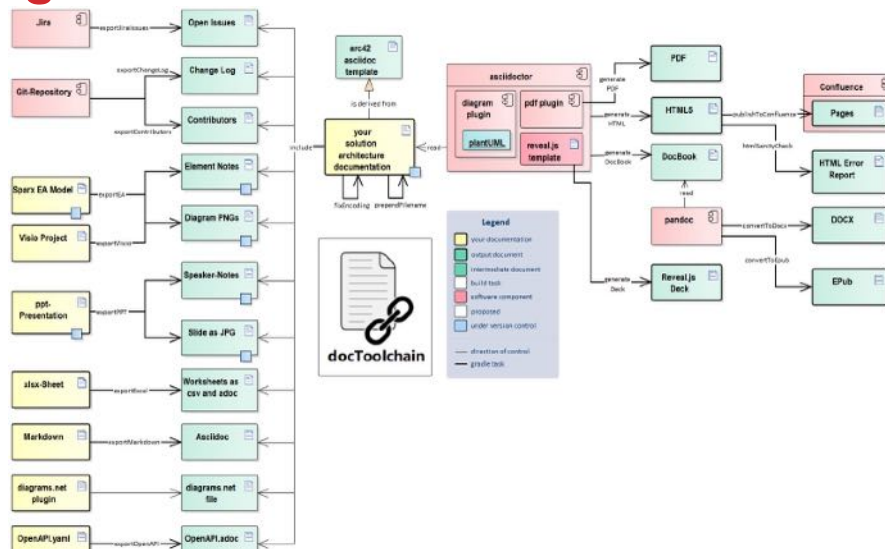
Documentation-as-Code

embarc.de

80

80

Werkzeugkette: docToolchain



Documentation-as-Code

embarc.de

81

81



docToolchain @docToolchain · 26. Sep.

...

A full blown docu toolchain with @arc42Tipps as template installed in just one tweet:

```
wget doctoolchain.github.io/dtcw
chmod +x dtcw
./dtcw downloadTemplate
./dtcw generateSite
```

=> doctoolchain.org



Documentation-as-Code

embarc.de

82

82

```
fish /Users/falk/doc-project
~/doc-project tree
├── docToolchainConfig.groovy
├── dtcw
├── src
│   └── docs
│       ├── arc42
│       │   ├── arc42.adoc
│       │   └── chapters
│       │       ├── 01_introduction_and_goals.adoc
│       │       ├── 02_architecture_constraints.adoc
│       │       ├── 03_system_scope_and_context.adoc
│       │       ├── 04_solution_strategy.adoc
│       │       ├── 05_building_block_view.adoc
│       │       ├── 06_runtime_view.adoc
│       │       ├── 07_deployment_view.adoc
│       │       ├── 08_concepts.adoc
│       │       ├── 09_design_decisions.adoc
│       │       ├── 10_quality_scenarios.adoc
│       │       ├── 11_technical_risks.adoc
│       │       ├── 12_glossary.adoc
│       │       ├── about-arc42.adoc
│       │       └── config.adoc
│       └── images
│           ├── 05_building_blocks-DE.png
│           ├── 08-Crosscutting-Concepts-Structure-DE.png
│           └── arc42-logo.png
5 directories, 20 files
~/doc-project
```



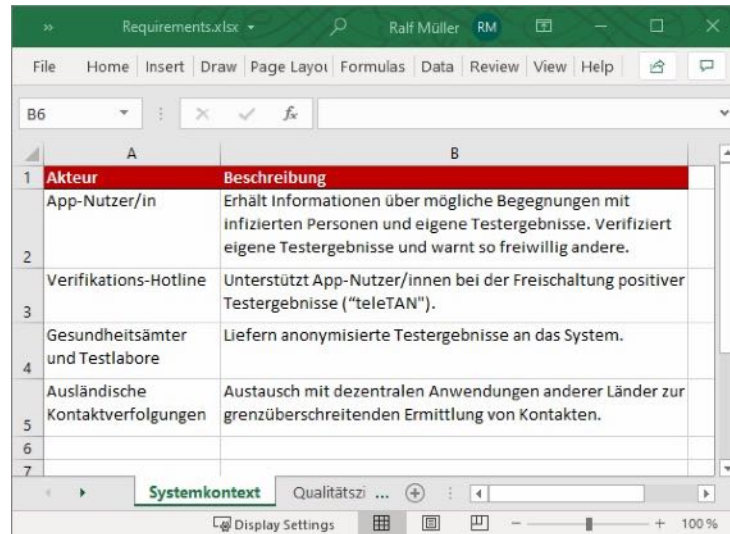
Documentation-as-Code

embarc.de

83

83

Tabellen!?



Akteur	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



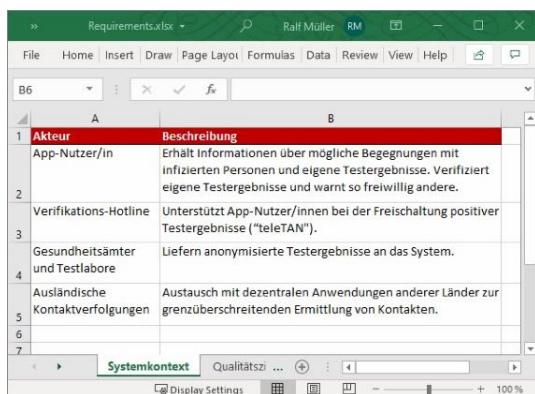
Documentation-as-Code

embarc.de

84

84

Tabellen!?



Akteur	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



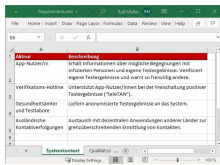
Documentation-as-Code

embarc.de

85

85

Tabellen!?



Name	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Robert Koch-Institut (RKI)	Stellt Inhalte ("Content") für die App zur Verfügung und bestimmt Parameter für die Messung der Kontakte ("Risiko-Ermittlung"). Empfängt Auswertungen, etwa aus der Datenspende.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



Akteur	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Robert Koch-Institut (RKI)	Stellt Inhalte ("Content") für die App zur Verfügung und bestimmt Parameter für die Messung der Kontakte ("Risiko-Ermittlung"). Empfängt Auswertungen, etwa aus der Datenspende.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



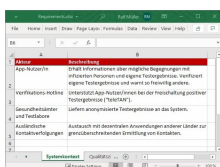
Documentation-as-Code

embarc.de

86

86

Tabellen!?



Name	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Robert Koch-Institut (RKI)	Stellt Inhalte ("Content") für die App zur Verfügung und bestimmt Parameter für die Messung der Kontakte ("Risiko-Ermittlung"). Empfängt Auswertungen, etwa aus der Datenspende.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



Akteur	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Robert Koch-Institut (RKI)	Stellt Inhalte ("Content") für die App zur Verfügung und bestimmt Parameter für die Messung der Kontakte ("Risiko-Ermittlung"). Empfängt Auswertungen, etwa aus der Datenspende.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



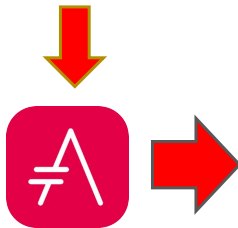
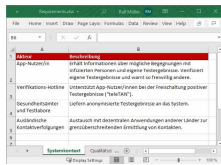
Documentation-as-Code

embarc.de

87

87

Tabellen!?



Akteur	Beschreibung
App-Nutzer/in	Erhält Informationen über mögliche Begegnungen mit infizierten Personen und eigene Testergebnisse. Verifiziert eigene Testergebnisse und warnt so freiwillig andere.
Verifikations-Hotline	Unterstützt App-Nutzer/innen bei der Freischaltung positiver Testergebnisse ("teleTAN").
Gesundheitsämter und Testlabore	Liefern anonymisierte Testergebnisse an das System.
Robert Koch-Institut (RKI)	Stellt Inhalte ("Content") für die App zur Verfügung und bestimmt Parameter für die Messung der Kontakte ("Risiko-Ermittlung"). Empfängt Auswertungen, etwa aus der Datenspende.
Ausländische Kontaktverfolgungen	Austausch mit dezentralen Anwendungen anderer Länder zur grenzüberschreitenden Ermittlung von Kontakten.



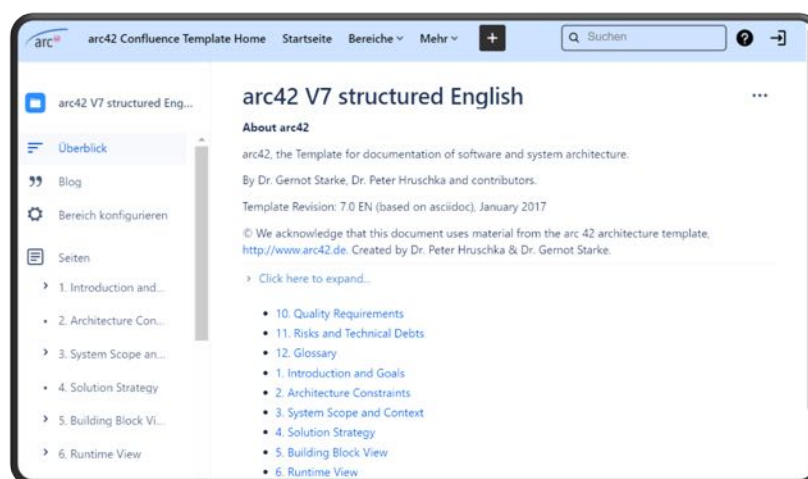
Documentation-as-Code

embarc.de

88

88

publishToConfluence



Documentation-as-Code

embarc.de

89

89

Code einbetten

```

1  :source-highlighter: pygments
2
3  [source,java]
4  ----
5  include::MyTest.java[]
6  ----
7
8
9
10
11
12 [source,java,indent=10]
13 ----
14 include::MyTest.java[linea=2..4]
15 ----
16 <1> method declaration
17 <2> print statement
18
19
20
21
22
23 :sourceDir: ../../src/main/java
24
25 [source,java]
26 ----
27 include::(sourceDir)/de/olo/vavz/functions/CheckedFunction.java[linea=12..14]
28 ----

```

```

public class MyTest {
    public static void main(String[] args) { (1)
        System.out.println("Hello world!"); (2)
    }
}

```

```

    public static void main(String[] args) { (1)
        System.out.println("Hello world!"); (2)
    }

```

1. method declaration
2. print statement

```

CheckedFunction1<Integer, Integer> readFunction = i -> readFromFile(i);
integers.stream().map(readFunction.unchecked());

```



Schnittstellenbeschreibung generieren


Swagger2Markup / swagger2markup
Watch 15
Star 294
Fork 54

[Code](#)
[Issues 7](#)
[Pull requests 0](#)
[Pulse](#)
[Graphs](#)

A Swagger to AsciiDoc or Markdown converter to simplify the generation of an up-to-date RESTful API documentation by combining documentation that's been hand-written with auto-generated API documentation.

Spring REST Docs SUCCESS

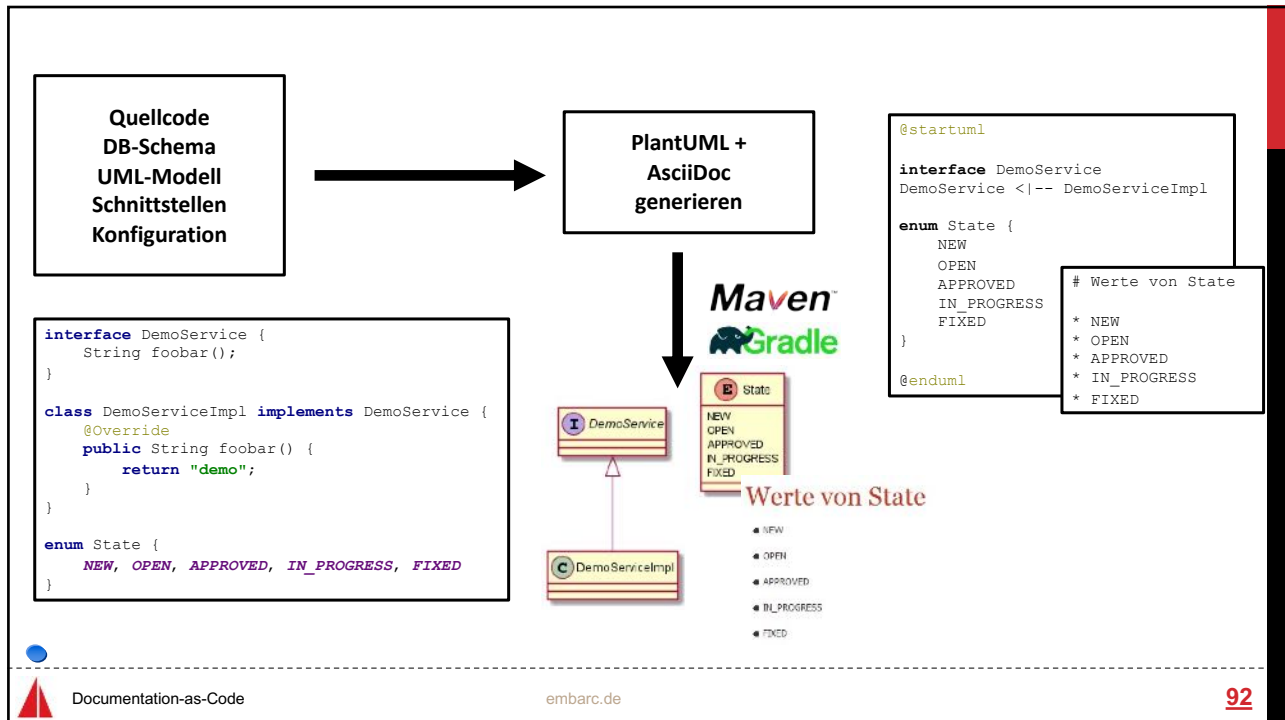
Overview

The primary goal of this project is to make it easy to document RESTful services by combining content that's been hand-written using [AsciiDoctor](#) with auto-generated examples produced with the [Spring MVC Test](#) framework. The result is intended to be an easy-to-read user guide, akin to [GitHub's API documentation](#) for example, rather than the fully automated, dense API documentation produced by tools like [Swagger](#).

JAX-RS Analyzer

Generates an overview of all JAX-RS resources in a project by bytecode analysis.



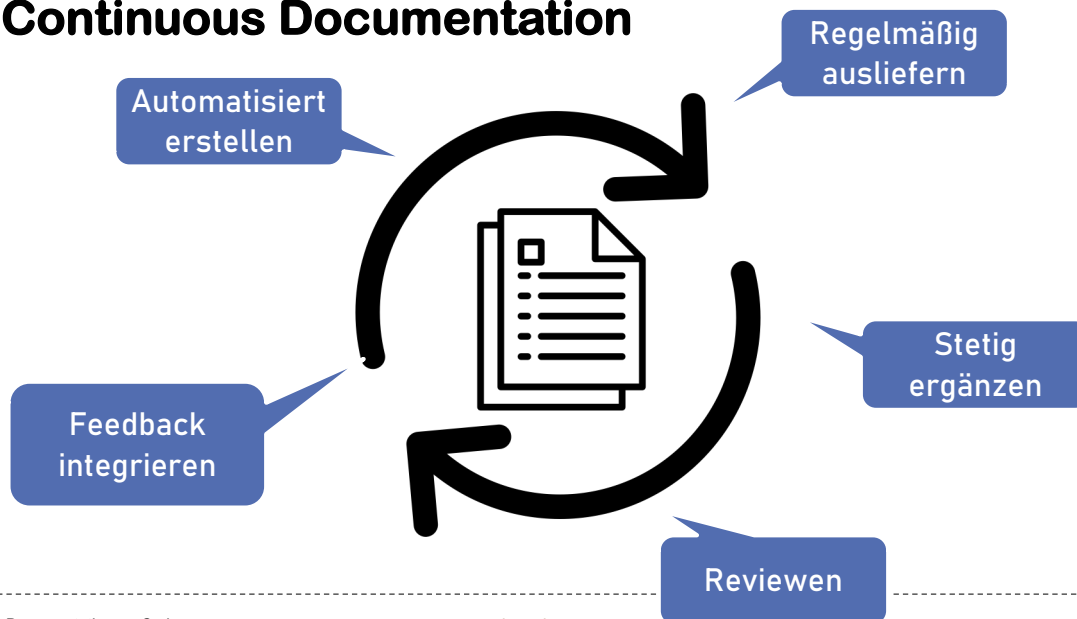


92



93

Continuous Documentation



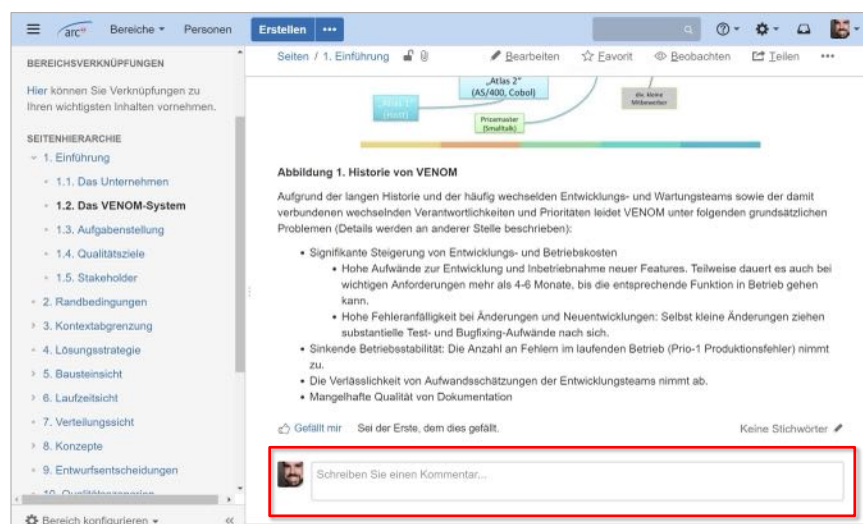
Documentation-as-Code

embarc.de

94

94

Zusammenarbeit



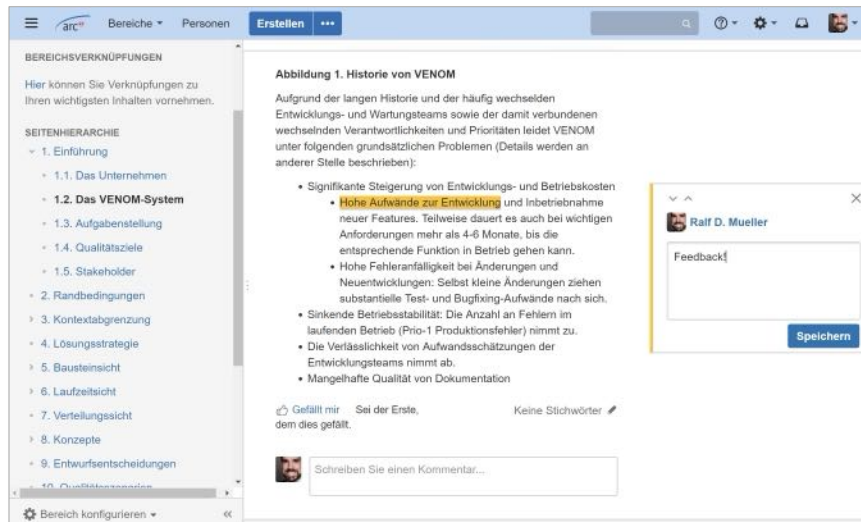
Documentation-as-Code

embarc.de

95

95

Zusammenarbeit



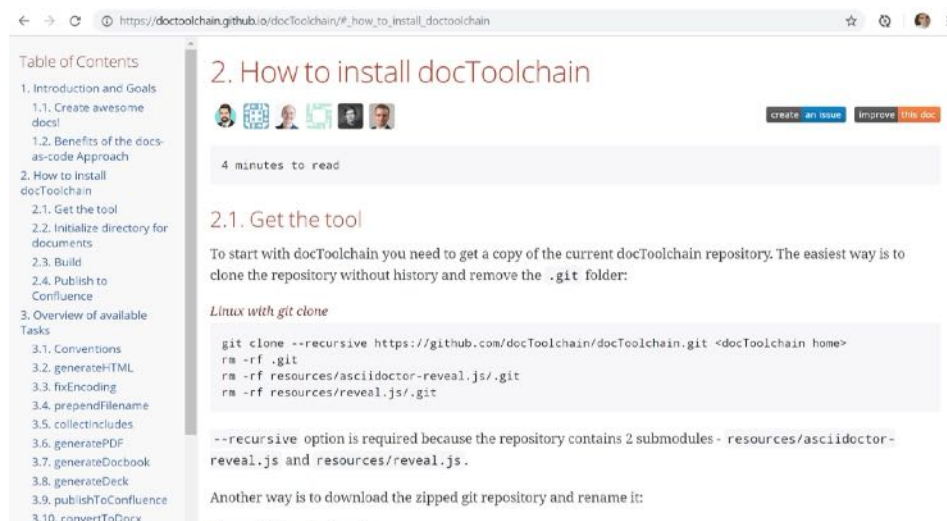
Documentation-as-Code

embarc.de

96

96

Besser: statische Seiten



Documentation-as-Code

embarc.de

97

97

Besser: statische Seiten

STATION

create an issue improve this doc

of the current docToolchain repository. The easiest way is to
the .git folder:



Documentation-as-Code

embarc.de

98

98

Besser: statische Seiten

improve this doc

docToolchain / docToolchain

Unwatch 25 Unstar 159 Fork 52

Code Issues 78 Pull requests 2 Wiki Insights Settings

docToolchain / src / docs / manual / 02_install.adoc or cancel

Edit file Preview changes Spaces 4 Soft wrap

```

1 :filename: manual/02_install.adoc
2
3 # How to install docToolchain
4
5 include::feedback.adoc[]
6
7 # Get the tool
8
9 To start with docToolchain you need to get a copy of the current docToolchain repository.
10 The easiest way is to clone the repository without history and remove the '.git' folder:
11
12 .Linux with git clone
13 [source,bash]
```



Documentation-as-Code

embarc.de

99

99

Besser: statische Seiten

create an issue

Search or jump to... Pull requests Issues Marketplace Explore

docToolchain / documentation

Unwatch 1 Star 0 Fork 0

Code Issues 2 Pull requests 0 Projects 0 Wiki Insights Settings

Title

Write Preview

[Enter feedback here]

#page:manual/02_install.adoc

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

Submit new issue

Assignees
No one—assign yourself

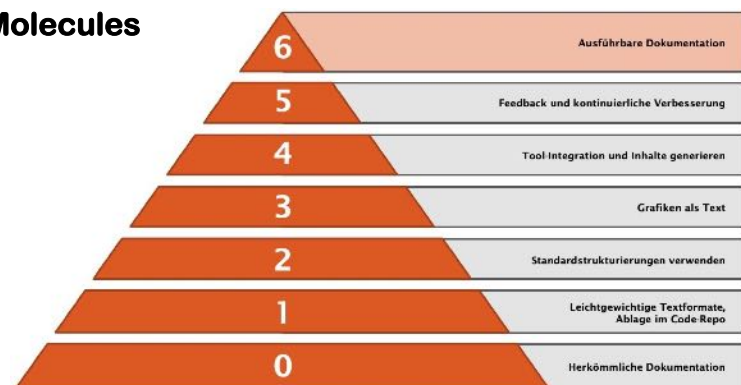
Labels
None yet

Projects
None yet

Milestone
No milestone

Ausführbare Dokumentation

- Strukturvorgaben dokumentieren und automatisiert testen
- jQAssistant, ArchUnit, xMolecules



Agenda



- 1 Einstieg und Motivation
- 2 Architektur dokumentieren
- 3 Docs-as-Code Maturity Level
- 4 Fazit und Ausblick**

4



Documentation-as-Code

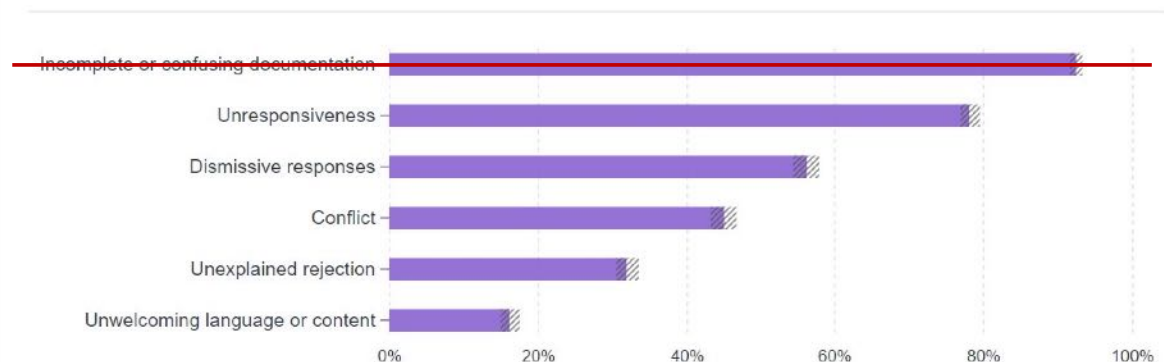
embarc.de

102

102

Fig1. - Problems encountered in open source

Source: opensourcesurvey.org

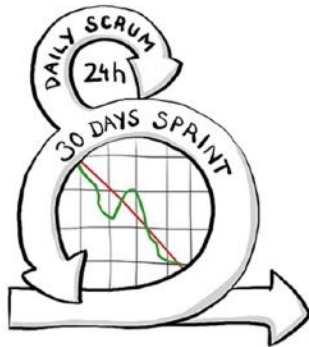


Documentation-as-Code

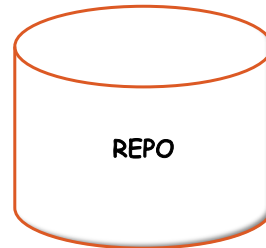
embarc.de

103 **103**

103



Continuous Documentation



Documentation as Code



Documentation-as-Code

embarc.de

104

104

Moderne (Architektur-)Dokumentation



Textformate



Single Source of Truth



Inhalte generieren



Bilder/Grafiken



Docs-as-Code



Continuous Documentation



Ausführen/Validieren



Kümmerer



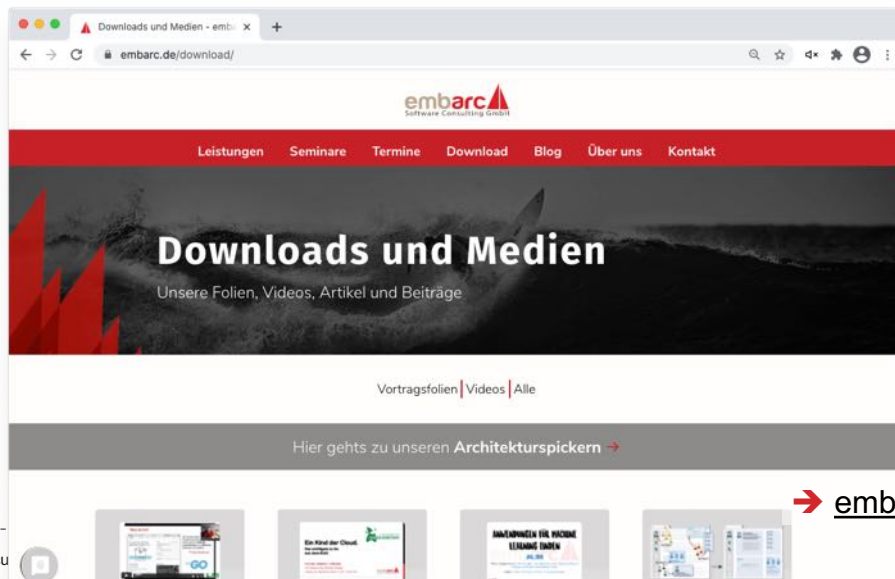
Documentation-as-Code

embarc.de

105

105

Folien von heute als PDF zum Download



→ embarc.de/download/

106

106

Flyer zum CWA Architekturüberblick



Wir schicken Ihnen gerne ein gedrucktes Exemplar dieses Überblicks als Flyer Einfach eine E-Mail senden an

info@embarc.de

mit Betreff „CWA-Flyer“ und Eurer Postadresse im Text, dann geht es los ...

<https://www.embarc.de/architektur-ueberblicke/#cwa>



Documentation-as-Code

embarc.de

107

107

Vielen Dank.

Ich freue mich auf Eure Fragen!



Falk Sippach



fs@embarc.de



@sippsack



→ [xing.to/fsi](https://www.xing.to/fsi)

