



Mandanten Factsheet GSB11 1.5

Bereichsname: Projekt GSB 11 Entwicklung



Inhaltsverzeichnis

1	Ziel dieses Dokuments.....	12
2	Technische Rahmenbedingungen.....	13
3	Komponenten des GSB11	14
4	Mandantenentwicklungsoption 1 (MEO1) - Anpassung über Konfiguration	15
4.1	Was ist ein Mandant?	15
4.2	Konfigurationselemente.....	15
4.3	Wie werden diese Einstellungen vorgenommen?.....	15
4.4	Formularkonfiguration (z.B. Kontaktformular/Passwort vergessen).....	15
4.5	Rahmenbedingungen MEO1.....	16
4.6	Vorteile MEO1	16
5	Mandantenentwicklungsoption 2 (MEO2) - Anpassung über ein Sitepackage	17
5.1	Abstimmung der geplanten Anpassungen.....	17
5.2	Aktueller Inhalt der Allow-List.....	17
5.3	Anforderung an MEO2 Mandanten, die eigene Extensions entwickeln	19
5.4	Start der Entwicklung	19
5.5	Einlieferung, Prüfung und Deployments.....	20
5.6	Anpassungsmöglichkeiten.....	21
5.6.1	Anpassung über Konfiguration.....	21
5.6.1.1	Beispiel BKM kulturstaatsministerin.de	22
5.6.2	Erweiterung Funktionsumfang.....	22
5.6.2.1	PHP Packages/TYPO3 Extension	22
5.6.2.2	PHP Classes.....	22
5.6.2.3	Beispiel BKM kulturstaatsministerin.de	22
5.6.3	Erweiterung Templates & Design	23
5.6.3.1	Ziele & Rahmenbedingungen.....	23
5.6.3.2	Komponenten des GSB11 Frontends.....	23



5.6.3.3	Beispiel BKM kulturstaatsministerin.de	24
5.7	Informationen für Entwickler:innen	26
5.7.1	Allgemeine Infos	26
5.7.2	GSB11 im Clusterbetrieb	26
5.7.3	Caching im GSB11.....	26
5.7.3.1	Technische Details.....	26
5.7.3.2	Performance-Metriken	27
5.7.3.3	Sicherheitsaspekte	27
5.7.3.4	Best Practices und Empfehlungen.....	27
5.7.4	TBD: Nutzung von Web-Components im GSB11	27
6	Multisite-Fähigkeit	28
6.1	Was bedeutet Multisite?	28
6.2	Wann ist das Management mehrerer Webauftritte in einem Mandanten sinnvoll?	28
6.2.1	Welche Auftritte kann man als Multisite aufsetzen?	28
6.3	Multisite oder nicht? Wie soll ich mich entscheiden?.....	29
6.4	Eine Behörde = ein Mandant?	29
7	Entwicklungsprozess.....	30
7.1	Entwicklungsumgebung und Hosting ITZBund.....	30
7.2	GitLab Organisation	31
7.3	Qualitätsüberprüfung & Gatekeeping.....	31
7.3.1	GitLab als Gatekeeper für Sitepackages	31
8	Infrastruktur.....	35
8.1	Deployment auf REF und PROD	35
8.1.1	Übernahme einer GSB7 Website in GSB11.....	35
8.1.1.1	Wir klären.....	35
8.1.1.2	Wir bauen auf.....	35
8.1.1.3	Onboarding des Kunden	35



8.1.1.4	Go Live des Kunden.....	36
8.1.1.5	Inbetriebnahme Testumgebung REF.....	36
8.2	Domains und Domain Aliases	36
9	Mitgeltende Dokumente	37
10	Referenzierte / ergänzende Dokumente.....	38



Tabellenverzeichnis

Es wurden keine Einträge für ein Tabellenverzeichnis gefunden



Abbildungsverzeichnis

Abbildung 1: Beispiel BKM kulturstaatsministerin.de.....	23
Abbildung 2: Beispiel BKM kulturstaatsministerin.de.....	25
Abbildung 3: Frontendarchitektur	25
Abbildung 4: Drei Umgebungen der Mandantenentwicklung.....	30
Abbildung 5: Prozess der Qualitätsprüfung & Gatekeeping	32
Abbildung 6: Prozess der Qualitätsprüfung & Gatekeeping (inkl. Kundenbenachrichtigung)	33

i Das Factsheet beschreibt aktuell technische und prozessuale Maßnahmen, die noch nicht alle vollständig umgesetzt sind. Details zum Umsetzungsstatus kann man sich im [entsprechenden Epic](#) ansehen.

- Ziel dieses Dokuments
- Technische Rahmenbedingungen
- Komponenten des GSB11
- Mandantenentwicklungsoption 1 (MEO1) - Anpassung über Konfiguration
 - Was ist ein Mandant?
 - Konfigurationselemente
 - Wie werden diese Einstellungen vorgenommen?
 - Formularkonfiguration (z.B. Kontaktformular/Passwort vergessen)
 - Rahmenbedingungen MEO1
 - Vorteile MEO1
- Mandantenentwicklungsoption 2 (MEO2) - Anpassung über ein Sitepackage
 - Abstimmung der geplanten Anpassungen
 - Aktueller Inhalt der Allow-List
 - Anforderung an MEO2 Mandanten, die eigene Extensions entwickeln
 - Start der Entwicklung
 - Einlieferung, Prüfung und Deployments
 - Anpassungsmöglichkeiten
 - Anpassung über Konfiguration
 - Beispiel BKM kulturstaatsministerin.de
 - Erweiterung Funktionsumfang
 - PHP Packages/TYPO3 Extension
 - PHP Classes
 - Beispiel BKM kulturstaatsministerin.de
 - Erweiterung Templates & Design
 - Ziele & Rahmenbedingungen
 - Komponenten des GSB11 Frontends
 - Beispiel BKM kulturstaatsministerin.de
 - Informationen für Entwickler:innen
 - Allgemeine Infos
 - GSB11 im Clusterbetrieb
 - Caching im GSB11
 - Technische Details
 - Performance-Metriken
 - Sicherheitsaspekte
 - Best Practices und Empfehlungen
 - TBD: Nutzung von Web-Components im GSB11
- Multisite-Fähigkeit
 - Was bedeutet Multisite?
 - Wann ist das Management mehrerer Webauftritte in einem Mandanten sinnvoll?
 - Welche Auftritte kann man als Multisite aufsetzen?
 - Multisite oder nicht? Wie soll ich mich entscheiden?
 - Eine Behörde = ein Mandant?
- Entwicklungsprozess
 - Entwicklungsumgebung und Hosting ITZBund
 - GitLab Organisation
 - Qualitätsüberprüfung & Gatekeeping
 - GitLab als Gatekeeper für Sitepackages

- Infrastruktur
 - Deployment auf REF und PROD
 - Übernahme einer GSB7 Website in GSB11
 - Wir klären
 - Wir bauen auf
 - Onboarding des Kunden
 - Go Live des Kunden
 - Inbetriebnahme Testumgebung REF
 - Domains und Domain Aliases
- Mitgeltende Dokumente
- Referenzierte / ergänzende Dokumente

Änderungshistorie

Version	Datum	Änderung	bearbeitet von
0	08.01.2024	Initiale Erstellung	Externes Entwicklungsteam GSB11 in Abstimmung mit IV B 2
0.1	11.01.2024	Redaktionelle Überarbeitung	Silke Spielkamp
0.2	25.01.2024	Redaktionelle & Inhaltliche Überarbeitung	Thomas Bussmeyer
0.3	05.02.2024	Ergänzung diverser Links, Hinweise für externe Hosts & Erweiterungsmöglichkeit 2	Thomas Bussmeyer & Kai Ole Hartwig
0.4	06.02.2024	Ergänzung GitLab User Profileeinstellungen	Thomas Bussmeyer
0.6	08.02.2024	Beschreibungen zu Abbildungen & Anforderungsmanagement GSB11	Sanaa Wahabzada
0.7	05.03.2024	Mandantenentwickler als External users & Nutzernamen Restriktion in GitLab ergänzt	Thomas Bußmeyer

Version	Datum	Änderung	bearbeitet von
0.9	06.03.2024	Erläuterung GSB Extensions & Ergänzungen Option 1	Thomas Bußmeyer
0.11	19.03.2024	gsb_core in Liste der Erweiterungen aufgenommen	Patrick Schrinier
0.12	25.03.2024	Option 1: Konfigurationselement angepasst	Thomas Bußmeyer
0.13	15.04.2024	Option 1: Konfigurationselemente ergänzt	Thomas Bußmeyer
0.14	27.05.2024	Option 1: Userstories an Konfigurationselemente ergänzt	Daniel Auer
0.15	03.06.2024	siteconfig_impexp in Liste der Erweiterungen aufgenommen	Willi Wehmeier
0.16	03.06.2024	gsb_csp in Liste der Erweiterungen aufgenommen	Patrick Schrinier
0.17	17.06.2024	Proxy-Cache aufgenommen	Kai Ole Hartwig
1.0	11.07.2024	Redaktionelle Überarbeitung	Sanaa Wahabzada
1.1	05.11.2024	Ergänzung: • Erweiterung Funktionsumfang: Kontaktformular/Passwort vergessen • Domains und Domain Aliases: Fremdhoster	Manuel Kirchner
1.2	27.01.2025	Ergänzung Multisite-Fähigkeit	Sanaa Wahabzada

Version	Datum	Änderung	bearbeitet von
1.3	10.03.2025	Strukturierung angepasst. Klarere Aufteilung nach MEO1 und 2	Thomas Bußmeyer
1.4	11.03.2025	Ergänzungen von Beispielen MEO2	Thomas Bußmeyer
1.5	04.06.2025	Ergänzung MEO2-Anforderungen	Sanaa Wahabzada

Prüfungshistorie

Version	Datum	Anmerkung	geprüft von
0.2	26.01.2024	Qualitätssicherung/Review	Matthias Schmidt
0.6	08.02.2024	Qualitätssicherung/Review	Sanaa Wahabzada
0.12	25.03.2024	Qualitätssicherung/Review	Sanaa Wahabzada
1.0	11.07.2024	Qualitätssicherung/Review	Sanaa Wahabzada

Freigabehistorie

Version	Datum	Anmerkung	freigegeben von
0.12	25.03.2024	Freigabe	Silke Spielkamp

Abnahme

Version	Datum	Anmerkung	abgenommen von
0.12	26.03.2024 - 27.03.2024	Abnahme BMI	Claudia Pülicher (BMI), Katja Helfert (BMI), Christoph Dahlen (Extern BMI)

1 Ziel dieses Dokuments

Ziel des Mandanten Factsheets ist es, Entwickler:innen von GSB11 Mandanten und technisch versierten Personen aller Zielgruppen eine erste Informationsquelle bezüglich der Bereitstellung, Konfiguration und Individualisierung des GSB11 zu bieten. Das Factsheet zeigt die Konfigurations- und Anpassungsmöglichkeiten des GSB11 auf und weist auf etwaige Einschränkungen hin. Über die referenzierten Dokumente werden weitergehenden Informationen zur Verfügung gestellt.

Die Zielgruppen des Mandanten Factsheets sind:

- Technische Ansprechpartner:innen beim Kunden (v.a. Behörden)
- Technische Ansprechpartner:innen beim Dienstleister der Kunden (Agenturen)
- ITZBund-Betrieb (IV B 2)
- Schnittstellen-Partner (Stellenanzeigen, Newsletter-Dienste etc.)

2 Technische Rahmenbedingungen

Im Rahmen dieser Dokumentation werden Mandantenprojekte betrachtet, die im ITZBund Hosting betrieben werden. Extern betriebene GSB11 Mandanten müssen viele der beschriebenen Prozesse selbständig definieren und implementieren.

Weitere wertvolle und zu beachtende Hinweise:

- Der GSB11 basiert auf TYPO3 v12. Die TYPO3 spezifische Core Version wird in "MAJOR.MINOR" Definition über die [GSB11 TYPO3 Extension gsb_core](#) eingebunden.
- In der externen Entwicklungsumgebung des ITZBundes (Public Cloud Development Solution - kurz PCDS) und im ITZBund Hosting ist es dem GSB11 (Software) nicht möglich zur Laufzeit Bestandteile aus dem Internet nachzuladen (Bibliotheken, Container, etc.).

3 Komponenten des GSB11

Der GSB11 basiert auf dem TYPO3 Core, Community- und GSB-spezifischen-Extensions. Diese bilden zusammen den sog. GSB11 Standardmandanten. Alle eingebundenen Community Extensions können in den composer.jsons der GSB-Extensions eingesehen werden ([Beispiel gsb_core](#)). Alle [GSB-spezifischen Extensions](#) finden sich im GitLab.

Der GSB besteht derzeit aus den folgenden Extensions

- [a11y_backend](#) (BITV Verbesserungen des Backend)
- [gsb_clustered_caching](#) (Anpassungen des Caching, damit Cache Flushs an Frontend- und Backend-Systeme verteilt werden)
- [gsb_consent](#) (Consent Management)
- [gsb_core](#) (Grundsätzliche Features, Inhaltsmodule und Templates)
- [gsb_csp](#) (CSP Header basierend auf Hashs)
- [gsb_distribution](#) (Redaktionelle Beispielinhalte)
- [gsb_feusermanagement](#) (Websitenutzerlogin)
- [gsb_metadata_cleaner](#) (Bereinigung von Dateimetadaten beim Upload im Redaktionssystem)
- [gsb_nginx_proxy](#) (Anpassung des Cachings, damit Cache Flushs an den Proxy Cache geschickt werden)
- [gsb_public_frontend](#) (Frontendtemplates, Styles, freie Fonts)
- [gsb_private_frontend](#) (Fonts und Icons des Bundes)
- [gsb_sitepackage](#) (Extension mit deren Hilfe man einen GSB11 Mandanten an kundenindividuelle Anforderungen anpassen kann)
- [gsb_solr](#) (Extension zur Anpassung der Solr-Suche an die Anforderungen im Hosting des ITZBund)
- [gsb_widgets](#) (Hilfstoools/Übersichten für das Redaktionssystem)
- [siteconfig_impexp](#) (Extension zum Importieren und Exportieren der Site-Konfiguration)

i Im TYPO3-Kontext bezeichnet das Backend das allgemeine Redaktionssystem. Im GSB11-Kontext umfasst es das TYPO3-Backend sowie die Nebensysteme wie Solr. In der IT-Architektur ist das Backend allgemein als System ohne direkte Benutzerinteraktion definiert.

4 Mandantenentwicklungsoption 1 (MEO1) - Anpassung über Konfiguration

Über die Option 1 erhalten Kund:innen eine schnelle und einfache Möglichkeit, Konfigurationen und Individualisierungen direkt im Backend des GSB11 vorzunehmen. Kund:innen können die Konfiguration sowohl importieren und exportieren. Damit ist möglich, Anpassungen bspw. auf einem DEV-System vorzunehmen, diese zu exportieren und anschließend in ein PROD-System zu importieren.

4.1 Was ist ein Mandant?

- Ein GSB11 Mandant ist eine Installation des GSB11 für einen oder mehrere Webauftritte.
- Ein Mandant besteht aus einem Redaktionssystem mit einer gemeinsam genutzten Datenbank und einigen ergänzenden technischen Komponenten für die Suche, Zwischenspeicherung von Daten.
- Alle Komponenten werden für alle Webauftritte, die in einem Mandanten verwaltet werden, gemeinsam genutzt und sind stets für alle Webauftritt einheitlich.
- Technisch gesprochen: Es gibt pro Mandant **ein** Deploymentpaket, dass auf der Mandantenumgebung installiert wird.

4.2 Konfigurationselemente

Über die grafische Benutzeroberfläche des GSB11 können folgende Elemente konfiguriert werden (die Liste ist nicht abschließend):

- URL/Domain
- Logo/ 2. Logo im Header
- Favicon
- Social Media Kanäle
- Website Titel
- Copyrighttext im Footer
- Page IDs (PIDs) von Menüs
- Farben der Website
- Toggle: Sprachumschalter
- Toggle: Suche

Der Umfang der Konfigurationselemente wird fortlaufend erweitert und entsprechend hier ergänzt.

4.3 Wie werden diese Einstellungen vorgenommen?

Das Backend-Modul "Site Management → Sites" ist eine Funktion innerhalb der Benutzeroberfläche (UI), die es Nutzern ermöglicht die oben genannten Elemente zu konfigurieren.

4.4 Formularkonfiguration (z.B. Kontaktformular/Passwort vergessen)

Bei der Administration des Kontaktformulars muss zum einen die Empfänger-Email-Adresse und zum anderen die Absender-Email-Adresse konfiguriert werden.

Empfänger-Adresse: Hierbei kann der Nutzer frei wählen. Das CMS verschickt die Nachricht exakt an die dort vergebene E-Mail Adresse. Wichtig ist nur, dass der Empfänger diese auch abrufen kann.

So kann es sein, dass die Firewall der empfangenden Behörde diese freischalten muss oder extra dafür einzurichten hat. Das ITZBund hat auf diese keine Einfluss.

Absender-Adresse: Hierbei ist auf die folgende Notation zu achten: *@noreply.gsb.bund.de. Der Kunde steht hier vor dem "@". In dem Fall das "*" -Symbol, welches als Platzhalter dient. In der Praxis sieht das beispielsweise wie folgt aus: itzbund@noreply.gsb.bund.de. Der Nutzer des Kontaktformulars erhält eine Kopie seiner Mail nun vom Absender: itzbund@noreply.gsb.bund.de

Im Standard ist ein Mailrelay aktiviert, welches über die "Passwort vergessen"-Funktion eine E-Mail an einen bereits registrierten Nutzer verschickt. An dieser Stelle sind keine weiteren Anpassungen erforderlich.

4.5 Rahmenbedingungen MEO1

- Nutzung der vorhandenen Module und ihres Funktionsumfangs
- Keine Eigenentwicklungen in Funktion und Design
- Nutzung des barrierefreien, getesteten Designs mit fest definierten Individualisierungsmöglichkeiten

4.6 Vorteile MEO1

- **Kosteneffizient** - keine Entwicklungsleistung nötig
- **Automatische Updates** vom GSB 11
- Styleguidekonform, barrierefrei, responsive
- BSI Webcheck und Pentest inklusive

5 Mandantenentwicklungsoption 2 (MEO2) - Anpassung über ein Sitepackage

Über die Option 2 erhalten Kund:innen erweiterte Möglichkeiten, Konfigurationen und Individualisierungen des GSB11 vorzunehmen. Alle Möglichkeiten der Option 1 sind enthalten. Darüber hinaus ist es möglich, den Funktionsumfang sowie das Template & Designsystem zu erweitern.

5.1 Abstimmung der geplanten Anpassungen

Alle über die oben beschriebenen Möglichkeiten hinausgehenden Anpassungen und neuen Anforderungen **müssen** mit dem Anforderungsmanager des GSB11 im BMI kommuniziert und abgestimmt werden (governmentsitebuilder@bmi.bund.de). Genehmigte Anpassungen werden in einer "Allowlist" dokumentiert und erlaubt. Weitere Informationen hierzu sind im Anforderungskonzept (*derzeit in Ausarbeitung*) zu finden.

5.2 Aktueller Inhalt der Allow-List

apache-solr-for-typo3/solr	dkd / community	Solr-Integration für TYPO3	
b13/container	b13 GmbH	Container-Element für strukturierte Inhalte	
b13/host-variants	b13 GmbH	Hostabhängige Konfigurationen	z. B. für Multidomain-Umgebungen
b13/permission-sets	b13 GmbH	Vorlagen für Redakteursrechte	
cweagans/composer-patches	Community	Composer-Patches via JSON anwenden	Für TYPO3-Patches nötig
derhansen/form_crshield	Torben Hansen	Spamschutz für TYPO3 Form (CSRF)	
ext-zip	PHP Extension	ZIP-Unterstützung	PHP-Modul erforderlich

featdd/dpn-glossary	Daniel Dorndorf	Glossar-Funktion mit automatischer Verlinkung	
friendsoftypo3/*	Friends of TYPO3	Offizielle TYPO3-Extensions	Platzhalter für mehrere Pakete
georgringer/news	Georg Ringer	Nachrichten-/Blog-Extension	Standard für News
georgringer/numbered-pagination	Georg Ringer	Paginierung mit Zahlen	Add-on zu <code>news</code> möglich
helhum/config-loader	Helmut Hummel	YAML-basierte Konfiguration für TYPO3	
helhum/typo3-console	Helmut Hummel	CLI-Tool zur Verwaltung von TYPO3	Ideal für Automatisierungen
ichhabrecht/content-defender	Alexander Kellner	Einschränkung von Content-Zugriff	z. B. nach IP oder Usergruppen
kitzberger/form-mailtext	Martin Kitzberger	E-Mail-Text für TYPO3 Form anpassbar	
leuchtfeuer/secure-downloads	Leuchtfeuer Digital	Geschützte Downloads mit Rechtesteuerung	
phpseclib/phpseclib	phpseclib developers	PHP-Implementierung von SSH/SFTP/etc.	z. B. für verschlüsselte Verbindungen
symfony/*	Symfony	Symfony-Komponenten	TYPO3 verwendet viele davon

typo3/cms-*	TYPO3 Core Team	TYPO3 CMS Core-Komponenten	Platzhalter für mehrere Pakete
wapplersystems/multisite-belogin	WapplerSystems	Backend-Login nach Site unterscheiden	Für Multisite-Instanzen

Aktuell einbindbare GSB11-Composer-Pakete (itzbund/itzbund-mandatory suggests bzw. composer suggest):

- itzbund/gsb-feusermanagement: Contains the front-end user administration.
- itzbund/gsb-solr: Search incl. specially customised indexer. Is required if a search or dynamic list is to be used on the page.
- itzbund/gsb-widgets: Widgets for the backend dashboard.
- itzbund/siteconfig_impexp: Export and import of the site configuration.

5.3 Anforderung an MEO2 Mandanten, die eigene Extensions entwickeln

Eigene Extensions werden in einem dedizierten Repository des GSB 11 GitLabs entwickelt und versioniert. Um die ständige Releasefähigkeit des Mandanten Sitepackage zu garantieren, werden in diesem nur stabile Release Versionen akzeptiert. Entwicklungsstände (Release Kandidat Versionen) eigener Extensions können somit erst auf der TEST sichtbar werden, wenn diese released wurden. Daher ist zur Entwicklung eine lokale Entwicklungsumgebung erforderlich.

Ein Release der eigenen Extension erfolgt durch den Merge des **main** Branches in den **release** Branch.

5.4 Start der Entwicklung

- Jeder Mandant startet seine Entwicklung mit dem [Sitepackage Kickstarter](#).
- Dieses Sitepackage wird als Git-Repository zentral über das GSB11 angelegt und Mandantenentwicklern zur Verfügung gestellt.
- Jeder Mandant, der im ITZBund hostet, versioniert seinen Code auf dem [GSB11 GitLab](#).
- Alle mandantenspezifischen Konfigurationen und Entwicklungen erfolgen in dieser Extension.
- Das Repository liefert eine lokale Entwicklungsumgebung auf der Basis von [ddev](#) mit. ddev ermöglicht die einfache Erstellung einer lokalen Entwicklungsumgebung auf allen gängigen Betriebssystemen. Die Umgebung wird in Dockercontainern bereitgestellt.
- Nach der initialen Bereitstellung durch das GSB11 Team steht folgendes zur Verfügung
 - das Repository
 - passende Pipelines
 - eine voll lauffähige Testinstanz mit dem GSB11 Standard Frontend und Beispielcontent
- Die Entwicklung kann nun lokal starten.

 Mehr zu TYPO3 Sitepackages findet man in der offiziellen Dokumentation.

5.5 Einlieferung, Prüfung und Deployments

Entwicklungsschritte erfolgen über lokal erzeugte Featurebranches und können regelmäßig gepusht werden. Über Merge Request werden die Branches mit der Hauptentwicklungslinie (main) zusammengeführt.

Eine GSB11-seitig zur Verfügung gestellte zentrale Pipeline gewährleistet ein flexibles, aber standardisiertes Angebot für alle mandantenspezifischen Individualisierungen. Diese umfasst die Prüfung der mandantenspezifischen Anpassungen, regelt aber auch das Deployment des Mandanten.

Deployments

Nach Integration in den main-Branch erfolgen weitere Test, der Build des neuen Mandanten-Images und das automatische Deployment auf Test. Das Deployment dauert max. 30 min.

Kontrolle: `/api/version` gibt die container-version zurück. Nach dem erfolgreichen Deployment verändert sich die Version des Containers.

Nach einem Deployment werden **bei einer Versionsänderung** mehrere TYPO3-Commands über ein Helm-Hooks automatisch ausgeführt. Dies sind unter anderem:

```
typo3 database:updateschema -vvv safe
gsbcore:configuration:import
typo3 upgrade:run -vvv
```

([gsb11-job-upgrade-wizard.yaml](#) [gsb11-job-upgrade-db-schema.yaml](#))

Technisch wird eine Anpassung über ein Sitepackage des Mandanten entweder erlaubt (wenn die aus dem Commit bzw. Pull-Request / Merge-Request resultierenden Jobs erfolgreich sind) oder führt zu einem Scheitern der CI-Pipeline mit entsprechender Benachrichtigung per E-Mail.

Bei erfolgreicher Integration gelten festgelegte Verantwortlichkeiten bezüglich Performance, Lauffähigkeit, Verfügbarkeit, Sicherheit und Behebung von Problemen.

Geteilte Verantwortung

Bei Individualisierung mittels Sitepackage gilt das Prinzip der geteilten Verantwortung. Für Erweiterungen oder Anpassungen des GSB11 sind grundsätzlich die Mandantenentwickler:innen (Kund:innen) selbst verantwortlich. Dies betrifft insbesondere Aspekte im Bereich Wartung, Updatefähigkeit, Sicherheit und Performance, die das Gesamtsystem negativ beeinflussen. Das GSB11 Team des ITZBund ist für die Sicherstellung und Überwachung der Leistungsfähigkeit des Grundsystems zuständig. Bei negativer Beeinflussung des GSB11 hinsichtlich Performance, Lauffähigkeit, Verfügbarkeit, Sicherheit

etc. eines Mandanten aufgrund von mandantenspezifischer Anpassungen, sind die Mandantenentwickler:innen (Kund:innen) selbst für die Behebung des Problems verantwortlich.

Die Grundbausteine des GSB11 werden in regelmäßigen Abständen einem **BSI Webcheck** unterzogen. Diese Bestandteile können somit als geprüft angesehen werden. Insbesondere Anpassungen/Ergänzungen von PHP Classes und Javascript innerhalb eines Mandanten Sitepackages können jedoch erneuten Prüfbedarf (vereinfachter Teilwebcheck) auslösen. Diese Ergänzungen zu werden bei der Anlieferung erkannt und Mandantenentwickler:innen werden ggf. innerhalb der GitLab Pipeline oder auch per Mail darüber informiert.

5.6 Anpassungsmöglichkeiten

5.6.1 Anpassung über Konfiguration

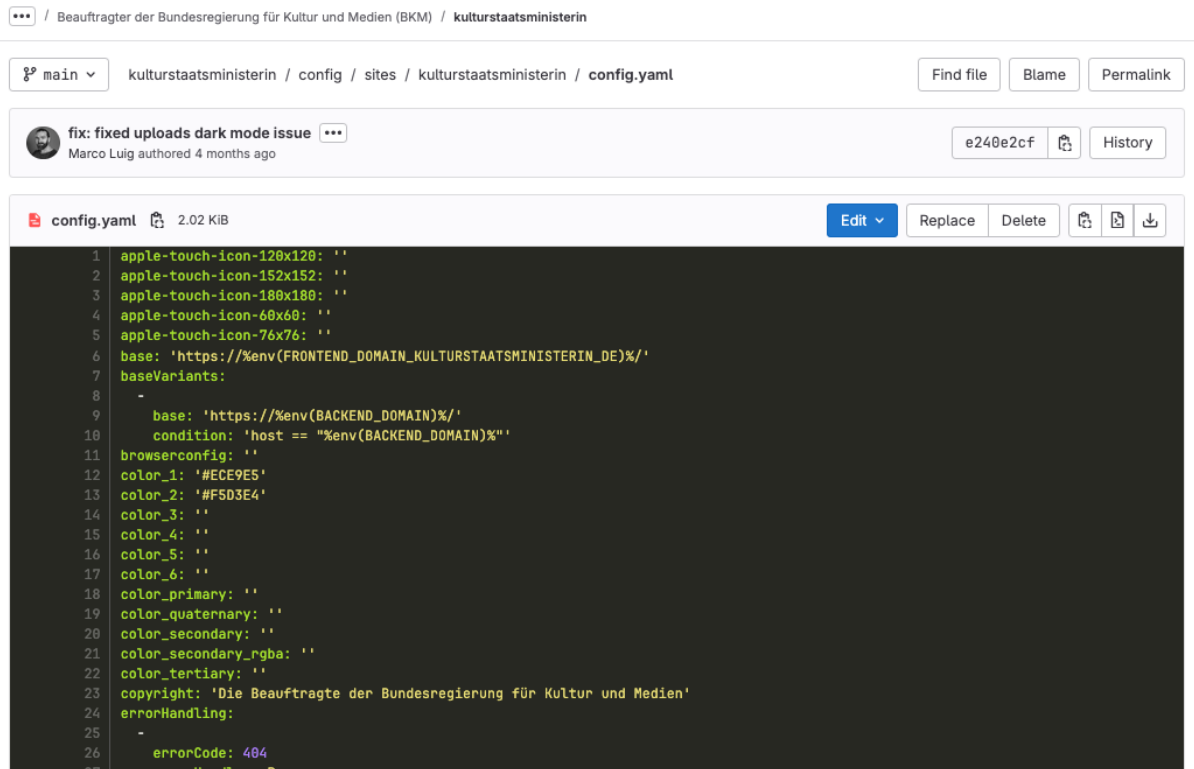
Wie und wo können Mandantenentwickler:innen diese Konfigurationen aus MEO 1 vornehmen, wenn ein Sitepackage vorhanden ist?

- Option 2 beinhaltet alle Möglichkeiten der Option 1.
- Die Konfiguration kann über die Sitepackage Extension in Form einer [TYPO3 Site Config](#) vorgenommen werden.
- Namenskonvention: Site Identifier muss identisch sein zum Pfad der Configdatei.
- Ort: [config.yaml](#)
- Die config der Sitepackage Extension kann in der /config/sites Config inkludiert werden, um doppelte Pflege zu vermeiden.
- Die config.yaml enthält für alle konfigurierbaren Element konkrete Beispiele.

Hinweise

- Der Site Package Config Ordner (/config) wird für das Deployment auf Test- und Produktivsysteme nicht verarbeitet, sondern dient nur der lokalen Entwicklungsumgebung.
- Die Konfiguration der Test- und Produktivsysteme wird durch das GSB11 Team vorgenommen.
- Änderungen, die Anpassungen an der settings.php für die lokale Entwicklung bedürfen, werden über Releasebegleitblätter des GSB11 bekannt gemacht.

5.6.1.1 Beispiel BKM [kulturstaatsministerin.de](#)



5.6.2 Erweiterung Funktionsumfang

5.6.2.1 PHP Packages/TYPO3 Extension

- Zusätzliche PHP Packages können über Composer eingebunden werden.
- Es werden nur genehmigte Pakete zugelassen (Allowlist). Diese werden bei jedem Commit überprüft.
- Pakete, welche nicht in der Allowlist enthalten sind, müssen mit dem Anforderungsmanagement BMI abgesprochen werden (governmentsitebuilder@bmi.bund.de).

5.6.2.2 PHP Classes

- PHP-Klassen können in den Ordner `Classes/` der Sitepackage Extension gelegt werden.
- Werden dabei geeignete Namensräume für die Klassen verwendet, kann das System diese Dateien automatisch laden.
- Siehe auch <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ExtensionArchitecture/FileStructure/Classes/Index.html>.
- Es werden **ausschließlich ViewHelper** zugelassen. Die Einhaltung dieser Beschränkung werden bei jedem Commit überprüft.
- Anforderungen die darüber hinausgehen müssen mit dem Anforderungsmanagement BMI abgesprochen werden (governmentsitebuilder@bmi.bund.de).

5.6.2.3 Beispiel BKM [kulturstaatsministerin.de](#)

Anpassungen durch mandantenspezifische ViewHelper.

... / Beauftragter der Bundesregierung für Kultur und Medien (BKM) / kulturstaatsministerin

main kulturstaatsministerin / Classes / ViewHelpers / +

Find file Edit Code

fix: added file reference viewhelper
Marco Luig authored 4 months ago

2b8faf4e History

Name	Last commit	Last update
..		
FallbackImageViewHelper.php	fix: added optional subtitle file upload for video files	5 months ago
FileObjectViewHelper.php	fix: added fallback image to media library	6 months ago
FileReferenceViewHelper.php	fix: added file reference viewhelper	4 months ago
InlineSvgViewHelper.php	fix: fixed inline SVG viewhelper	6 months ago
SubtitleViewHelper.php	fix: added optional subtitle file upload for video files	5 months ago

Abbildung 1: Beispiel BKM kulturstaatsministerin.de

Hinweis: Ein Beispiel für PHP Packages/TYPO3 Extension fehlt aktuell noch.

5.6.3 Erweiterung Templates & Design

5.6.3.1 Ziele & Rahmenbedingungen

- Der GSB11 strebt einen effizient zu aktualisierenden Frontendstack an, d.h. dass nur wenige npm Packages verwendet werden.
- Das Standard Design- und Templatesystem des [GSB11 \(GSB11 Frontend\)](#) basiert auf [Bootstrap 5](#). Das Markup des GSB11 folgt somit den Bootstrapvorgaben und wurde durch TYPO3 spezifische Markups und Fluidvariablen ergänzt. Die [offizielle Bootstrap Dokumentation](#) ist folglich bei Anpassungen im Template- und Designsystem sehr hilfreich.
- Alle GSB11 Extensions und damit auch die Bestandteile des GSB11 Frontends verwenden [SemVer 2.0](#). Daher können Minor- und Patchreleases der gsb_core und gsb_public_frontend und gsb_private_frontend bedenkenlos für alle Mandanten eingespielt werden.
- Für jede Frontend-Komponente erhalten Mandantenentwickler ein Changelog, anhand dessen eingeschätzt werden kann,
 - ob eine Komponente von einem Change betroffen ist.
 - ob es ein Breaking Change für diese Komponente ist.
 - welche Bestandteile der Komponente von dem Change betroffen sind, z.B. Styles, Assets, JavaScript, Settings, Markup.
- Das CSS und Javascript der gsb_public_frontend wird vorkompiliert mit ausgeliefert. Mandantenentwickler:innen können über das Sitepackage eigenes CSS (insb. CSS Variablen) und Javascript ergänzend hinzufügen.
- Es ist möglich, einen eigenen Buildprozess innerhalb seines Sitepackages zu erstellen.
- Das GSB11 Markup enthält keine visuellen Definitionen (insb. Inline CSS) und ermöglicht so eine hohe Wiederverwendbarkeit, d.h. Überschreibungen von Fluidtemplates sollte nur selten nötig sein.

5.6.3.2 Komponenten des GSB11 Frontends

Die Definition des mitgelieferten Template- und Designsystems des GSB11 findet sich in den beiden Extension [gsb_core](#) (TYPO3 Fluid Templates) und [gsb_public_frontend](#) ((S)CSS, JavaScript).

In der Extension [gsb_private_frontend](#) befinden sich der Bundes-Font und einige Icons, die nur für berechtigte Mandanten genommen werden dürfen.

Um mandantenspezifische Anpassungen vornehmen zu können, ist **keine** Anpassung dieser beiden Extension nötig. **Alle Anpassungen werden in der mandantenspezifischen Sitepackage Extension vorgenommen.**

Der GSB11 liefert ein barrierefreies Template- und Designsystems im CD Styleguide des Bundes mit. Um in der Zukunft von diesem Aspekt des GSB11 profitieren zu können, müssen mandantenspezifische Anpassungen nach definierten Regeln vorgenommen werden. Nur dann kann sichergestellt werden, dass Updates, insbesondere Patches nach SemVer 2.0 (siehe dazu auch das Patchkonzept GSB11), fehlerfrei in die Mandantenversion des GSB11 eingespielt werden können.

Durch Anpassungen an den folgenden Stellen und Verzeichnissen können Änderungen am Template- und Designsystem vorgenommen werden:

- [\(S\)CSS in /Resources/Public/StyleSheets](#)
 - Mandantenspezifische CSS-Definitionen, die die GSB11 Stylesheet aus der *gsb_public_frontend* Extension ergänzen und/oder überlagern
 - Erwarteter Output Pfad: `/Resources/Public/StyleSheets`
 - Definition des Pfades über TypoScript in `/Configuration/TypoScript/setup.typoscript`
 - Buildprozess kann mandantenspezifisch definiert werden, solange der Outputpfad eingehalten wird
- JavaScript in `/Resources/Public/JavaScripts`
 - Mandantenspezifische JavaScript-Definitionen, die die GSB11 Stylesheets ergänzen, überlagern
 - Output Pfad: `/Resources/Public/JavaScripts`
 - Buildprozess kann mandantenspezifisch definiert werden, solange der Outputpfad eingehalten wird
- `package.json`
 - Es kann ein mandantenspezifischen nodebasierter Buildstack definiert werden
 - Bei einem mandantenspezifischen nodebasierten Buildstack MUSS eine [.nvmrc-Datei](#) auf Rootebene des Projekts mitgeliefert werden, die die eingesetzte Node Version enthält
 - MUSS: `npm run build` ist definiert, um das Frontend zu bauen
- Fluid Templates
 - Fluid Templates sollten **nicht** überschrieben werden, da dies die Updatefähigkeit insbesondere der *gsb_core* Extension gefährdet
 - Das Überschreiben von Templates/Partials ist aber möglich und erfolgt nach Standard-TYPO3-Vorgehen.

5.6.3.3 Beispiel BKM [kulturstaatsministerin.de](#)

In diesem MEO2 Projekt wurden eigene Funktionserweiterungen aber auch umfangreiche Anpassungen an Templates und Design mit einem node-basierten Frontendbuildstack vorgenommen.

... / Beauftragter der Bundesregierung für Kultur und Medien (BKM) / kulturstaatsministerin

main kulturstaatsministerin / Build + Find file Edit Code

chore(deps): update dependency autoprefixer to v10.4.21 Renovate Bot authored 1 day ago 7ce8ff85 History

Name	Last commit	Last update
..		
Assets	fix: stage image width	3 days ago
reports	feat: added sitepackage with global config and assets	8 months ago
.eslintrc.cjs	feat: extended page top container	7 months ago
.gitignore	feat: added sitepackage with global config and assets	8 months ago
eslint.config.js	feat: extended page top container	7 months ago
npm-shrinkwrap.json	chore(deps): update dependency autoprefixer to v10.4.21	1 day ago
npm-shrinkwrap.json.license	feat: added sitepackage with global config and assets	8 months ago
package.json	chore(deps): update dependency autoprefixer to v10.4.21	1 day ago
package.json.license	feat: added sitepackage with global config and assets	8 months ago
postcss.config.js	feat: build mandanten image and helm chart	8 months ago
stylelint.config.js	fix: fixed stylelint config after renovate auto update	1 month ago
webpack.config.development.js	fix: fixed figcaption spacing problems in top container	3 months ago
webpack.config.production.js	fix: fixed figcaption spacing problems in top container	3 months ago

Abbildung 2: Beispiel BKM kulturstaatsministerin.de

Die folgende Abbildung illustriert die oben genannten Komponenten und Artefakte in der Frontendarchitektur und wie sie partiell überlagert und ergänzt werden.

Frontendarchitektur

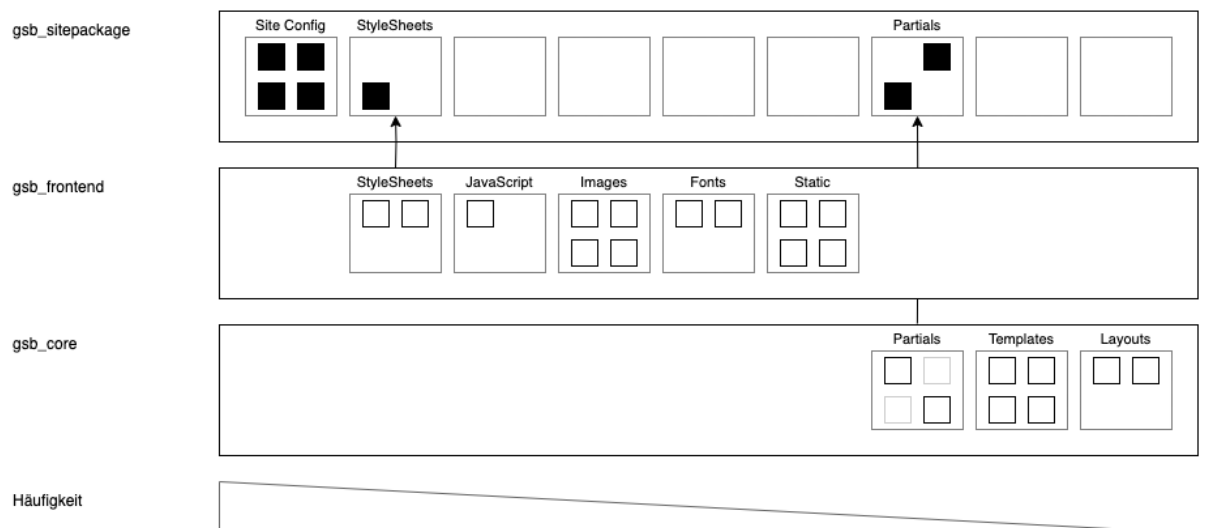


Abbildung 3: Frontendarchitektur

5.7 Informationen für Entwickler:innen

5.7.1 Allgemeine Infos

- Im ITZBund Hosting wird der GSB11 auf [Alpine Linux](#) betrieben.

5.7.2 GSB11 im Clusterbetrieb

- Der GSB11 im ITZBund wird als Cluster ausgeführt, daher muss beachtet werden:
 - typo3temp wird nicht verwendet
 - die Optionen CONFIG compressCss, compressJs, concatenateCss, concatenateJS werden nicht verwendet
 - es werden keine SQL-Queries verwendet, für alle DB Operationen wird Doctrine DBAL / Extbase verwendet
 - Datenbank Caches werden in Redis abgelegt, dazu wird `\ITZBund\GsbCore\Cache\Backend\SentinelCapableRedisBackend::class` verwendet.
 - Alle Verbindungen im Cluster werden verschlüsselt und die Services müssen sich authentifizieren
 - Alle Verbindungen werden persistent und mit Connection Pooling aufgebaut
 - Es gibt einen Cache-Proxy vor den HTTPD Pods.

5.7.3 Caching im GSB11

Der GSB11 verfügt über ein mehrstufiges, verteiltes Caching, das durch die Erweiterungen `gsb_clustered_caching` und `gsb_nginx_proxy` verwaltet wird und durch die Erweiterung `gsb_csp` ermöglicht wird.

Grundsätzlich werden mehrere Pods für das Redaktionssystem und für die Webseite(n) ausgeführt. Das Redaktionssystem und die Webseite(n) greifen auf verschiedene Pods zu. Alle dateibasierten Caches werden nicht zwischen den verschiedenen Pods geteilt. Datenbank- bzw. Redis-basierte Caches werden zentral gespeichert.

Zur Steigerung der Performance setzt der GSB11 in der Content Security Policy die Methode *Hash* ein und validiert die Integrität der Skripte und CSS-Dateien. Dies ermöglicht dem System, die Seiten insgesamt zu cachen. Zusätzlich setzt der GSB11 einen Caching-Proxy ein, der die Seiten wie ein Full-Page-Cache zwischenspeichert. Dieser Cache wird automatisch geleert, sobald TYPO3 ein `clear-cache` im Caching-Framework verarbeitet.

5.7.3.1 Technische Details

- **Erweiterungen:** Die Caching-Erweiterungen `gsb_clustered_caching` und `gsb_nginx_proxy` sind speziell für die Verwaltung und Optimierung des Cache-Systems entwickelt. `gsb_csp` sorgt dafür, dass Content Security Policies korrekt implementiert werden.
- **Datenbanken:** Für die zentralen Caches werden in der Regel Redis und relationale Datenbanken wie MySQL verwendet.

- **Versionen:** Es ist wichtig sicherzustellen, dass die verwendeten Erweiterungen auf dem neuesten Stand sind, um Sicherheitslücken zu vermeiden und von Leistungsverbesserungen zu profitieren. Bitte beachten Sie die Hinweise im Patch- und Releasekonzept des GSB11.

5.7.3.2 Performance-Metriken

- **Ladezeiten:** Durch die Implementierung des mehrstufigen Caching-Systems konnten die Ladezeiten der Webseiten erheblich reduziert werden.
- **Cache-Hitrate:** Die Cache-Hitrate wird durch diese Methode deutlich erhöht, sodass die meisten Anfragen erfolgreich aus dem Cache bedient werden können. Für Formulare, Suchergebnisse und geschützte Bereiche gibt es Einschränkungen in der Cachebarkeit der Seiten.

5.7.3.3 Sicherheitsaspekte

- **Content Security Policy (CSP):** Die Verwendung der Hash-Methode in der CSP stellt sicher, dass nur die erwarteten Skripte und CSS-Dateien geladen werden. Dies schützt vor Angriffen wie Cross-Site Scripting (XSS).
- **Integritätsprüfung:** Durch die Integritätsprüfung der Dateien wird sichergestellt, dass keine manipulierten Dateien ausgeführt werden, was die Sicherheit weiter erhöht.

5.7.3.4 Best Practices und Empfehlungen

- **Konfiguration:** Eine saubere und vollständige Konfiguration der Caching-Einstellungen ist essenziell. Es sollte regelmäßig überprüft werden, ob alle Caches korrekt ins [Caching-Framework](#) eingebunden sind.
- **Cache-Invalidierung:** Ein effektives Cache-Invalidierungskonzept ist entscheidend, um sicherzustellen, dass Benutzer stets aktuelle Inhalte sehen. Bitte beachten Sie dies bei der Konfiguration von eigenen Caches. Der GSB11 setzt bei der Invalidierung auf die [Konzepte von TYPO3](#).

Der Caching-Proxy verarbeitet ausschließlich Webseitenaufrufe und trägt damit maßgeblich zur Reduzierung der Serverlast und zur Verbesserung der Antwortzeiten bei.

5.7.4 TBD: Nutzung von Web-Components im GSB11

Voraussetzungen und wichtige Informationen:

1. Durch die Bereitstellung der Möglichkeit der Einbindung von Web Components im GSB 11 (diese können barrierefrei gestaltet werden) können theoretisch externe Inhalte, Skripte oder Funktionen geladen werden. Hieraus können Sicherheitsrisiken entstehen (z.B. XSS oder Datenabflüsse etc.). Die Verantwortung hinsichtlich Sicherheitsrisiken der eingebundenen Inhalte obliegt hierbei allein der jeweiligen Behörde. Somit übernimmt das ITZBund-Hosting hierfür keine Verantwortung (hinsichtlich Schadsoftware, Sicherheitslücken o.ä.).
2. Anpassung der CSP-Konfigurationen im Backend, Typoscript & erweiterte Nutzerrechte (Maintainer-Rechte)

6 Multisite-Fähigkeit

Der GSB11 bietet grundsätzlich die Möglichkeit mehrere Webseiten / Domains über einen Mandant zu administrieren (Multisite). Diese Funktion sollte bei Erwägung aufgrund von folgenden Themen ausgiebig diskutiert und evaluiert werden.

6.1 Was bedeutet Multisite?

- Multisite bedeutet, dass mehrere Webauftritte (Z.B. www.kulturstaatsministerin.de und deutscher-kurzfilmpreis.de) aus einem Mandanten, also aus einem Redaktionssystem heraus betrieben werden.
- Es gibt also ein Deploymentpaket/ eine Codebasis mit Hilfe dessen mehrere Webauftritte verwaltet und ausgeliefert werden.
- Ein Mandant = ein gemeinsames technisches Setup = ein Redaktionssystem = eine Datenbank = eine Codebasis

6.2 Wann ist das Management mehrerer Webauftritte in einem Mandanten sinnvoll?

Es gibt eine einfache grundsätzliche Aussage dazu: sind die **Anforderungen an alle Webauftritte in allen Dimensionen nahezu identisch**, dann kann sich ein Multisitesetup anbieten.

6.2.1 Welche Auftritte kann man als Multisite aufsetzen?

- **Alle Webauftritte sind langfristig MEO1 Setups:** MEO1er haben immer die gleiche Codebasis. Es gibt per Definition keine individuellen Entwicklungen. Viele MEO1 Auftritte kann man gut in einem Mandanten zusammenfassen.
- **Alle Auftritte haben das gleiche Admin- und Redaktionspersonal:** Wenn ein globaler Admin alle Webseiten verwaltet (eine Separierung von globalen Admins für verschiedene Webseiten ist nicht möglich) und es zentrale oder viele Inhalte, die über Webauftritte hinweg wiederverwendet werden sollen, gibt.
- **Alle Webauftritte werden von dem gleichen Dienstleister entwickelt:**
 - Eine Codebasis bedeutet, dass alle Entwicklungen in einem Git Repository zusammenfließen und gemeinsame deployed werden.
 - Ein Versionsupdate wird grundsätzlich für alle darin betriebenen Webseiten gelten.
 - Ein Deploymentpaket bedeutet, dass man Änderungen eines Dienstleisters nicht unabhängig von Änderungen eines anderen Dienstleisters deployen kann. Deployments müssten Webauftritt und dienstleisterübergreifend koordiniert werden. Immer. Das ist nicht sinnvoll möglich.
 - Interne Behördenkommunikation: Bei Verwaltung mehrerer Webseiten über einen Mandanten müssten die interne Behördenkommunikation zwischen den unterschiedlichen Stakeholdern durch einen zentralen Ansprechpartner gewährleistet werden.
- **Gleiche Anforderungen an Service und Verfügbarkeit:** Wenn die SLA für alle Mandanten identisch ist, können sie gemeinsam betrieben werden. Es wird grundsätzlich nur ein gesamtes Backup erstellt und kein Webseiten-spezifisches Backup. Bei einem Recovery-Fall kann somit nicht gewährleistet werden, dass alle Webseiten den erwarteten Stand widerspiegeln (wenn Webseite A zum Zeitpunkt X zurückgesetzt wird, wird gleichzeitig Webseite B auf den Zeitpunkt X zurückgesetzt)
- Wenn es sich um MEO2er handelt:
 - **Das Featureset der Webauftritte ist identisch:** Features. können nicht webauftrittspezifisch aktiviert werden. Alle Features sind stets für alle Mandanten aktiv.
 - Das stark angepasste Frontend ist über die Webauftritte hinweg identisch

6.3 Multisite oder nicht? Wie soll ich mich entscheiden?

Wenn einer der obigen Punkte nicht erfüllt ist, dann ist ein Multisitesetup nicht sinnvoll umzusetzen.

Sprechen Sie in jedem Fall das GSB11 Produktentwicklungsteam (Mail?) und lassen sich individuell beraten.

6.4 Eine Behörde = ein Mandant?

- Auch wenn es naheliegend erscheint, so geht diese Gleichung aus oben genannten Gründen nicht auf. Es kann aber möglich und sinnvoll sein bestimmte Webauftritte einer Behörde in einem Mandanten als Multisite zu betreiben. Wenn viele MEO1er innerhalb einer Behörde betrieben werden sollen, dann man u.U. diese Auftritt in einem Mandanten zusammenfassen und speziellere Auftritt, die MEO2er sind als separaten Mandanten aufsetzen.

7 Entwicklungsprozess

Die Organisation des Entwicklungsprozesses liegt in der Verantwortung des GSB11 Produktentwicklungsteams. Die folgenden Details sind informativ und zum besseren Verständnis des Gesamtsystems aufgeführt.

7.1 Entwicklungsumgebung und Hosting ITZBund

Die Mandantenentwicklung verläuft über drei getrennte Umgebungen:

- 1. Mandantenentwicklung Kunde:** Auf der Kundenseite bzw. bei der beauftragten Agentur findet die eigentliche Mandantenentwicklung statt.
Entwickler:innen können dazu aus dem GSB11 GitLab eine lokale GSB11 Entwicklungsumgebung auf Basis von `ddev` beziehen. Damit wird die GSB-Entwicklung auf einem lokalen Rechner/Laptop ermöglicht.
- 2. Externe Entwicklungsumgebung ITZBund (PCDS):** Über das GSB11 GitLab kann der Mandaten-Code angeliefert werden.
Der Mandaten-Code wird automatischen Qualitätschecks unterzogen, in ein Mandanten-Image integriert, auf eine DEV-Umgebung deployed und kann dort geprüft und getestet werden.
- 3. Hosting ITZBund:** In der Hosting Umgebung des ITZBunds wird der Liefergegenstand erneut auf Sicherheitslücken sowie Lauffähigkeit geprüft und dann in die Referenz- und Produktionsumgebung eingespielt. Wichtig: Die Referenzumgebung (REF) läuft nur innerhalb der Zone A und ist somit nur über eine SINA erreichbar. Die Produktionsumgebung (PROD) ist auch aus dem Internet erreichbar.

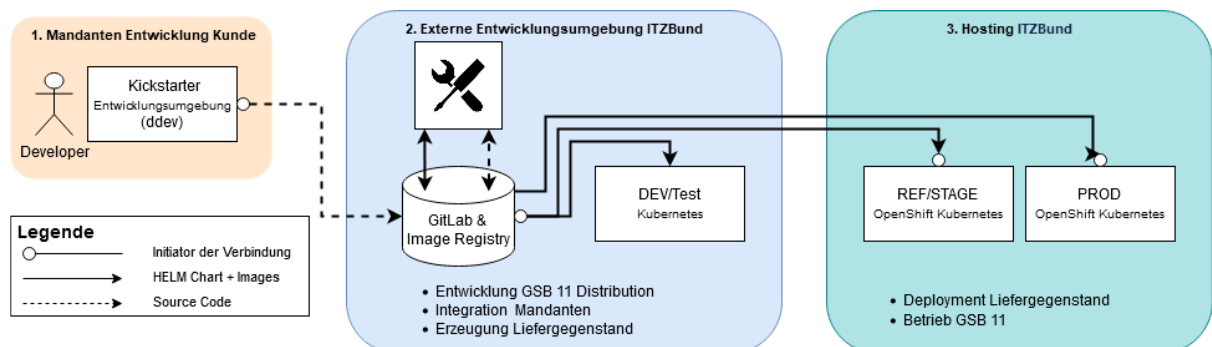


Abbildung 4: Drei Umgebungen der Mandantenentwicklung

Für die GSB11 Mandantenentwicklung wird vom ITZBund die Public Cloud Development Solution (PCDS) Umgebung zur Verfügung gestellt. Daneben wird auf dieser externen Entwicklungsumgebung, basierend auf GitLab, die Infrastruktur für die Entwicklung der Mandanten zur Verfügung gestellt.

Die externe Entwicklungsumgebung und das GSB11 Hosting beim ITZBund sind miteinander verknüpft. Die ITZBund Hosting Umgebung greift auf die über GitLab bereitgestellten Liefergegenstände in der externen Entwicklungsumgebung über einen Pull-Prozess zu. Die Liefergegenstände bestehen aus Images und Helm Charts. Die Erstellung dieser Liefergegenstände und der gesamte Build-, Deployment- und Releaseprozess ist nicht Teil der Mandantenentwicklung und werden zentral über das GSB11 Entwicklungsteam zur Verfügung gestellt. Die Dimensionierungen der Ressourcen für Mandanten werden somit auch zentral aus Erfahrungswerten der Entwicklung und des Betriebs abgestimmt.

7.2 GitLab Organisation

- Institution, Mandanten-Repository, User und Berechtigungen werden wie im Betriebshandbuch unter [Mandanten-Entwicklung](#) beschrieben, angelegt.
- Build und Deploy Pipelines (nicht in der Verantwortung der Mandantenentwickler:innen)
 - Das Mandanten Sitepackage enthält eine .gitlab-ci.yml, die die zentrale Standardpipeline einbindet.
 - Grundsätzlich erhalten die Artefakte Versionsnummer nach SemVer 2.0 in folgendem Schema: SITEPACKAGE-MANDATENID.
 - SITEPACKAGE ist eine SemVer-Versionsnummer.
 - MANDATENID ist ein String zur Identifizierung des Mandaten.
 - Die Pipeline baut
 - Composer Package der Sitepackage Extension. Ablage in [Mandantenrepository](#).
 - httpd Container mit Composer Package der Sitepackage Extension. Ablage in [zentralem Container Repository](#).
- HelmCharts & Images
 - Es wird ein Token mit Lese-Berechtigung auf das HelmCharts Registry erstellt.
 - Images (HTTPD) werden im Mandaten Repo gespiegelt.

7.3 Qualitätsüberprüfung & Gatekeeping

ITZBUNDPHP-2622

- Qualität und Konformität wird kontinuierlich in einem automatischen Prozess sichergestellt. Entwickler:innen werden bestmöglich unterstützt und frühzeitig über Qualitäts- und Konformitätsprobleme, in der Regel per Mail aus den GitLab Pipelines, informiert.
- Die wichtigsten Regeln werden schon lokal überprüft. Umgesetzt wird dies durch Githooks (pre-commit).
- Nach dem Commit und Push in das GSB11 GitLab werden in Sitepackage Pipelines weitere Checks und Tests durchgeführt.
- Die Ergebnisse dieser Tests werden dem Mandantenentwicklungsteam fortlaufend zur Verfügung gestellt.
- Die Quality Gates richten sich nach den für TYPO3 12 geltenden Best Practice Empfehlungen <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/CodingGuidelines/CglPhp/CodingBestPractices/Index.html>

7.3.1 GitLab als Gatekeeper für Sitepackages

ITZBUNDPHP-2622

Jedes Sitepackage wird nach jedem Commit Checks und Tests unterzogen:

- Typoscript Linter
- Conventional Commits werden geprüft
- Jeder Commit auf *main* löst ein Release auf das Testsystem aus (siehe auch [Releasekonzept GSB11#BranchModel&Workflow](#))
- Jedes Release erfolgt durch Merge auf den branch "release" - dies löst der Veröffentlichungsprozess Richtung ITZBund aus (siehe auch [Releasekonzept GSB11#BranchModel&Workflow](#))
- Die zentrale Sitepackage Build Pipeline erwartet einen ggf. vorhandenen Frontend Build unter dem Befehl *npm run build*
- Nur gelistete Composer Packages sind erlaubt, insb. TYPO3 Pakete
- Node Packages werden nicht reguliert

Das Fehlschlagen der CI Pipeline führt zur Benachrichtigung via E-Mail oder weiterer selbst konfigurierter GitLab-Anbindungen/Integrationen.

Im Folgenden wird der oben beschriebene Prozess kurz illustriert:

Quality Gates High Level

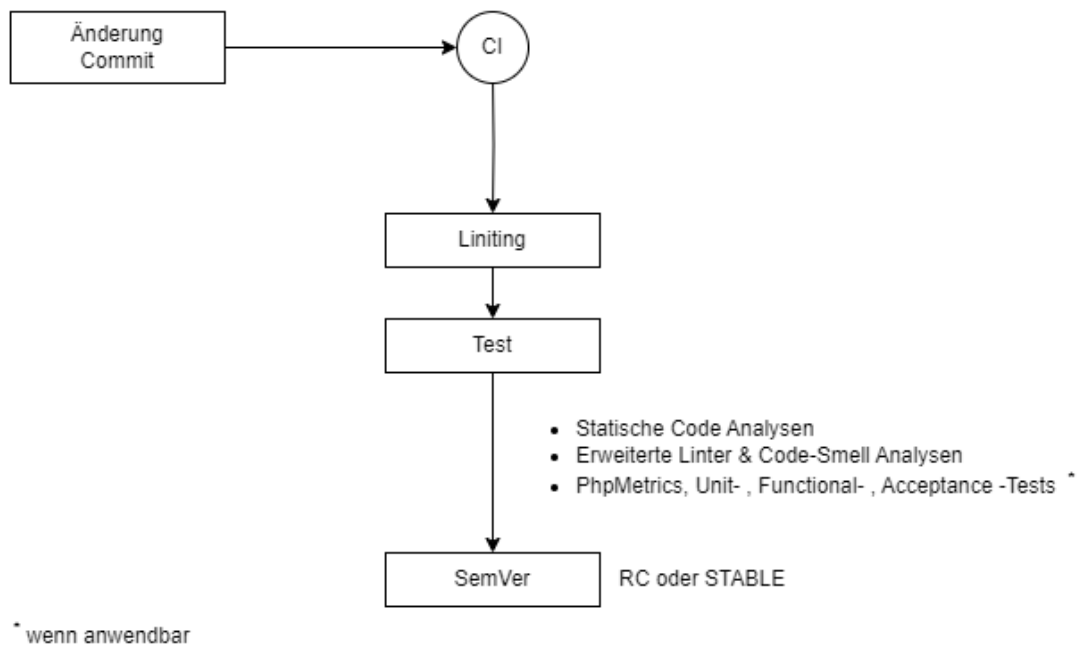
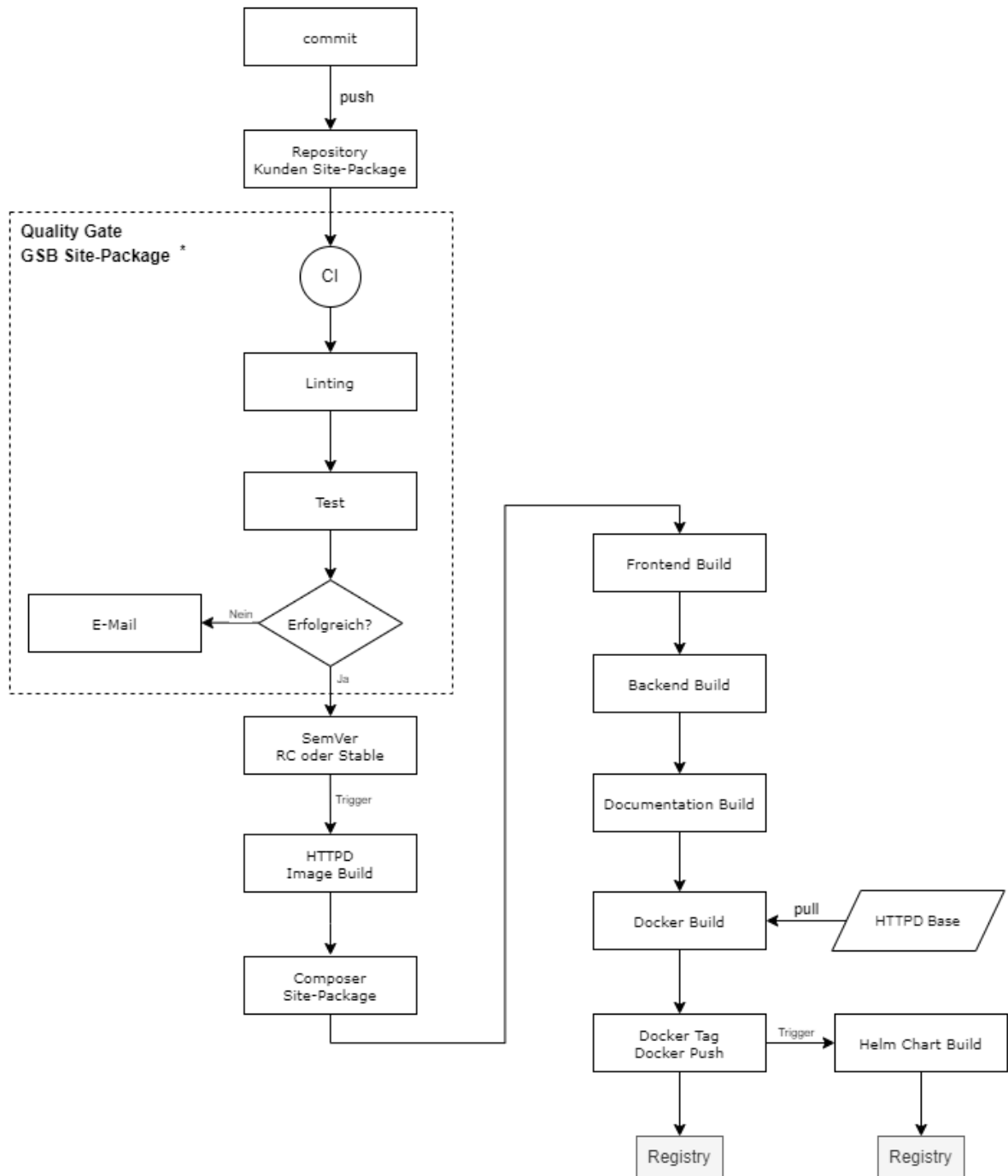


Abbildung 5: Prozess der Qualitätsprüfung & Gatekeeping

Das nächste Schaubild illustriert den gleichen Prozess auf einer detaillierteren Art (inkl. dem Prozess der Kundenbenachrichtigung):

Workflow Kunden Image GSB11



Die Version aus SemVer ist die Versionsnummer des Images + Kunden Prefix.
Das Image beymt als Label die TYPO3 Version und die GSB-Extenson Version.

* Verwaltet durch das GSB11 Team

Abbildung 6: Prozess der Qualitätsprüfung & Gatekeeping (inkl. Kundenbenachrichtigung)

8 Infrastruktur

8.1 Deployment auf REF und PROD

Für das Deployment auf die REF und PROD Umgebungen innerhalb der ITZBund Betriebsumgebung (CaaS) ist ein Kunden Onboarding seitens des ITZBund notwendig. Dieses Onboarding besteht aus einem fachlichen Teil (AKZ, Service Schein etc.) und aus einem betrieblich technischen Teil. Hier wird nur der betrieblich technische Vorgang beschrieben.

Beim betrieblichen Onboarding einer Website in die ITZBund Betriebsumgebung unterscheiden wir die Übernahme einer bereits im ITZBund gehosteten Website auf Basis des GSB7 und einer komplett neuen Website.

8.1.1 Übernahme einer GSB7 Website in GSB11

Wenn die fachlichen Voraussetzungen erfüllt sind (AKZ, Service Schein etc. vorhanden) kann das technische Onboarding in die GSB11 Betriebsumgebung (CaaS) erfolgen. Die in GSB7 gehostete Website bleibt weiterhin aktiv. Alle Vorbereitungen für den Umzug auf den GSB11 können unabhängig betrieben werden.

8.1.1.1 Wir klären

- Kunde meldet FQDN (URLs) die auf GSB11 umgezogen werden sollen
- Wir prüfen ob das alle FQDN (URLs) eines GSB7 Mandaten sind, oder ob es sich nur um einen Teilumzug handelt und bestimmte FQDN (URLs) auf dem GSB7 Mandanten verbleiben.
- Wir prüfen ob es FQDN (Aliase) gibt die von externen Dienstleistern auf die im ITZBund gehostete öffentliche IP Adresse zeigen
- Wir prüfen ob die GSB7 Websites im gleichen Rechenzentrum wie der GSB11 im ITZBund gehostet wird
- Auf Basis der obigen Prüfungen wird entschieden wie der Umzug GSB7 zu GSB11 technisch umzusetzen ist

8.1.1.2 Wir bauen auf

- Kunden GSB11 Betriebsumgebung in der PROD
- Einspielen Standard Mandat (entspricht zunächst immer der Option MEO1), für eine funktionsfähige Website mit Backend in der PROD Umgebung
- Einrichten eines Kunden Zugang auf das Backend für die neue GSB11 PROD Umgebung via NdB (Netzwerk des Bundes)
- Setup der öffentlichen IP (Internetverfahren), so dass zu einem vereinbarem Zeitpunkt die FQDN URLs von GSB7 auf GSB11 geschwenkt werden können
- Aufbau und Zugang REF Umgebung wird erst nach dem Live Gang benötigt, da die Ersteinpielung direkt auf der PROD vollzogen wird

8.1.1.3 Onboarding des Kunden

- Kunde erhält Zugangsdaten zum PROD Backend und PROD Backend-Preview (zunächst mit einem Standard Mandat MEO1)
- Kunde testet Zugang

- Einspielen eines kundenspezifischen Mandanten (MEO2) falls der Kunde sich für die Mandanten Entwicklung Option 2 entschieden hat
- Kontinuierliches Einspielen von Updates zu kundenspezifischen Mandanten (MEO2) auf PROD Umgebung bis zur Abnahme gewährleistet

8.1.1.4 Go Live des Kunden

- Kunde gibt nach Test in PROD Umgebung Mandant frei: Freigabe GoLive
- Nach Kundenfreigabe ermittelt ITZBund Betrieb nächst möglichen GoLive Termin Datum / Uhrzeit und benachrichtigt Kunde und Projektleitung
- Schenk der FQDN (URLs) die auf GSB11 umgezogen werden sollen zum GoLife Termin Datum / Uhrzeit
- **Der Kunde ist LIVE!**
- Die GSB7 Website bleibt 2 Wochen in Stand By Modus erhalten, rückschwenken jederzeit möglich
- Das GSB7 Backend Redaktion bleibt 3 Monate als Recherche Möglichkeit erhalten
- Im Falle eines Teilumzug verbleibt ein Teil der FQDN URLs auf dem GSB7 Mandanten. Dabei keine Änderung des Zugangs und der GSB7 Infrastruktur

8.1.1.5 Inbetriebnahme Testumgebung REF

- Nach GoLive auf PROD erfolgen alle Test und Abnahmen vom Kunden ausschließlich auf der REF Umgebung.
- Kunde erhält Zugangsdaten zum REF Backend und REF Backend-Preview
- REF Umgebung wird automatisch mit der jeweils aktuellen Version GSB Basis (MEO1)und GSB Kunden Mandanten (MEO2) aktualisiert.
- Patches und Release Updates sind für den Kunden verpflichtend. Abnahme und Freigabe von Major Patches nach Test auf der REF
- Details Patches und Release Updates in den Dokumenten Patch-Konzept bzw. Release-Konzept

8.2 Domains und Domain Aliases

Für das Produktivsystem stellt das ITZBund eine Hauptdomain und Subdomains zur Verfügung. In der maximalen Ausprägung verwaltet das ITZBund 6 Domains in diesem Kontext.

Aliases: können durch Mandantenentwickler:innen bei externem Dienstleister (Sub)Domains beantragt werden. Diese können dann selbständig konfiguriert werden und einen Redirect auf die Hauptdomain im ITZBund auslösen.

Prozess wenn über Fremdprovider gehostet werden soll:

- a) Anlage AKZ und Betriebsnamen (IZTBund)
- b) Neues Internetverfahren beantragen. (ITZBund)
- c) Nach Umsetzung dem Kunden die öffentliche IP mitteilen. Diese muss der Kunde an seinen Provider weitergeben, sodass dieser zum Zeitpunkt des Livegangs auf die neue IP verweist.

9 Mitgeltende Dokumente

Dokument	Bereich	Status
Releasekonzept GSB11		
Patchkonzept GSB11		

10 Referenzierte / ergänzende Dokumente

Dokument	Bereich	Status
Einstieg in die Mandantenentwicklung mit dem GSB Sitepackage Kickstarter (englisch)		
Dokumentation TYPO3 Sitepackages		