



Informations
Technik
Zentrum Bund

GSB 11 Mandanten-Factsheet

GSB 11 Produktentwicklung

Inhalt

1	Ziel und Zweck des Dokuments	3
2	GSB11-Überblick und technische Rahmenbedingungen	5
2.1	Hosting und Betriebsumgebung	5
2.2	Komponenten des GSB11	5
3	Mandanten, Mandantenoptionen und Multisite.....	7
3.1	Was ist ein Mandant?	7
3.2	Mandantenoptionen (MEO1 vs. MEO2)	7
3.3	Multisite-Fähigkeit.....	8
4	Zwingende Voraussetzungen und Rollen bei Mandantenentwicklungsoption 2 (MEO2)	10
4.1	Technische Voraussetzungen	10
4.2	Sitepackage und eigene Extensions	10
4.3	Geteilte Verantwortung („Shared Responsibility“).....	11
5	Merge-Requests, Renovate-Bot, Major-Releases	12
5.1	Automatisierte Merge-Requests (Renovate-Bot)	12
5.2	Major Releases (z. B. TYPO3-Upgrade)	12
6	Umgebungen, Infrastruktur und Zugriffe.....	13
6.1	Umgebungen im Lebenszyklus	13
6.2	Formularkonfiguration und Mail-Adressen	13
6.3	Zugriff auf Referenz- und Produktivumgebung	13
6.4	Synchronisation der Datenbanken	13
6.5	Backend-Nutzer-Regeln	14
6.6	Domain- und DNS-Management	14
7	Release-Prozess und GitLab-Workflow	16
7.1	Branch-Struktur	16
7.1.1	Ablauf.....	16
7.2	Release Candidate (RC)	16
7.3	Stable Release	16
8	Hinweise zu Template-, Design- und Frontend-Anpassungen.....	18
9	Caching, Performance und Sicherheit.....	19
9.1	Caching-Architektur	19
9.2	Content Security Policy (CSP) und Hashes	19
9.3	Performance-Metriken und Best Practices.....	19
10	Qualitätssicherung, BSI-Webcheck und Findings	20
11	Organisatorische Hinweise und Agenturverantwortung.....	21
12	Weiterführende Dokumentationen.....	22

Version	Datum	Änderung	bearbeitet von
2.0	25.06.2026	Gesamtüberarbeitung des Mandanten- Factsheets	Dominik Jodenschwager

1 Ziel und Zweck des Dokuments

Dieses Dokument richtet sich an:

- Entwickler:innen von GSB11-Mandanten (MEO1 und MEO2)
- Entwickler:innen von Migrationsdienstleistern der Behörden
- Technische Ansprechpartner:innen bei Behörden und deren Dienstleistern (Agenturen)
- ITZBund-Betrieb (insb. IV B 2)
- Schnittstellen-Partner (z. B. Stellenanzeigen, Newsletter-Dienste)

Es bietet:

- eine erste technische Informationsquelle zur Bereitstellung, Konfiguration und Individualisierung des GSB11,
- einen Überblick über Verantwortlichkeiten, Möglichkeiten und Prozesse – insbesondere bei Nutzung des GSB11 außerhalb des von MEO1,
- eine Grundlage für grobe Aufwandsschätzungen in GSB11-Migrationsprojekten.

Für Detailfragen wird auf das Anwendungshandbuch, die Release-Dokumentationen sowie das Release- und Releasemanagementkonzept verwiesen.

2 GSB11-Überblick und technische Rahmenbedingungen

2.1 Hosting und Betriebsumgebung

Dieses Dokument betrachtet grundsätzlich Mandantenprojekte, die im ITZBund-Hosting betrieben werden. Extern betriebene GSB11-Mandanten müssen viele der beschriebenen Prozesse (CI/CD, Monitoring, Caching, Security etc.) selbst definieren und implementieren.

Rahmenbedingungen:

- GSB11 basiert auf TYPO3 v13.4; die TYPO3-Core-Version (MAJOR.MINOR) wird über die GSB11-Extension `gsb_core` eingebunden.
- In der externen Entwicklungsumgebung des ITZBund (PCDS – Public Cloud Development Solution) und im ITZBund-Hosting kann die GSB11-Software zur Laufzeit keine Bestandteile aus dem Internet nachladen (Bibliotheken, Container, etc.).
- Durch die Betriebsumgebung des ITZBund bestehen Einschränkungen im Vergleich zu Self-Hosting-Szenarien (Details siehe Softwarearchitektur / Verteilungssicht).

2.2 Komponenten des GSB11

Der GSB11 basiert auf:

- TYPO3 Core
- Community-Extensions
- GSB-spezifischen Extensions

Die GSB-Extensions bilden zusammen die Mandantenentwicklungsoption 1 (MEO1). Alle eingebundenen Community-Extensions sind in den `composer.json`-Dateien der GSB-Extensions einsehbar (z. B. `gsb_core`). Alle GSB-spezifischen Extensions befinden sich im GSB11-GitLab.

Beispiele für GSB11-Extensions:

- [a11y_backend](https://git.gsb-itzbund.de/gsb11/extensions/a11y_backend) (https://git.gsb-itzbund.de/gsb11/extensions/a11y_backend) (BITV Verbesserungen des Backend, jedoch aktuell nicht mehr relevant)
- [gsb_clustered_caching](https://git.gsb-itzbund.de/gsb11/extensions/gsb_clustered_caching) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_clustered_caching) (Anpassungen des Caching, damit Cache Flushs an Frontend- und Backend- Systeme verteilt werden)
- [gsb_consent](https://git.gsb-itzbund.de/gsb11/extensions/gsb_consent/-/releases/v1.1.0) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_consent/-/releases/v1.1.0) (Consent Management)
- [gsb_core](https://git.gsb-itzbund.de/gsb11/extensions/gsb_core) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_core) (Grundsätzliche Features, Inhaltsmodule und Templates)
- [gsb_csp](https://git.gsb-itzbund.de/gsb11/extensions/gsb_csp) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_csp) (CSP Header basierend auf Hashs)
- [gsb_distribution](https://git.gsb-itzbund.de/gsb11/extensions/gsb_distribution) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_distribution) (Redaktionelle Beispieldaten)
- [gsb_feusermanagement](https://git.gsb-itzbund.de/gsb11/extensions/gsb_feusermanagement) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_feusermanagement) (Websitenutzerlogin)
- [gsb_metadata_cleaner](https://git.gsb-itzbund.de/gsb11/extensions/gsb_metadata_cleaner) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_metadata_cleaner) (Bereinigung von Dateimetadaten beim Upload im Redaktionssystem)

- [gsb_nginx_proxy](https://git.gsb-itzbund.de/gsb11/extensions/gsb_nginx_proxy) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_nginx_proxy) (Anpassung des Cachings, damit Cache Flushs an den Proxy Cache geschickt werden)
- [gsb_public_frontend](https://git.gsb-itzbund.de/gsb11/extensions/gsb_public_frontend) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_public_frontend) (Frontendtemplates, Styles, freie Fonts)
- [gsb_private_frontend](https://git.gsb-itzbund.de/gsb11/extensions/gsb_private_frontend) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_private_frontend) (Fonts und Icons des Bundes)
- [gsb_sitepackage](https://git.gsb-itzbund.de/gsb11/extensions/gsb_sitepackage) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_sitepackage) (Extension mit deren Hilfe man einen GSB11 Mandanten an kundenindividuelle Anforderungen anpassen kann)
- [gsb_solr](https://git.gsb-itzbund.de/gsb11/extensions/gsb_solr) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_solr) (Extension zur Anpassung der Solr-Suche an die Anforderungen im Hosting des ITZBund)
- [gsb_widgets](https://git.gsb-itzbund.de/gsb11/extensions/gsb_widgets) (https://git.gsb-itzbund.de/gsb11/extensions/gsb_widgets) (Hilfstoools/Übersichten für das Redaktionssystem)
- [siteconfig_impexp](https://git.gsb-itzbund.de/gsb11/extensions/siteconfig_impexp) (https://git.gsb-itzbund.de/gsb11/extensions/siteconfig_impexp) (Extension zum Importieren und Exportieren der Site-Konfiguration)

Begriffsklärung „Backend“:

Im TYPO3-Kontext: Redaktionssystem (TYPO3-Backend).

Im GSB11-Kontext: TYPO3-Backend plus Nebensysteme (z. B. Solr).

In der IT-Architektur allgemein: System ohne direkte Benutzerinteraktion.

3 Mandanten, Mandantenooptionen und Multisite

3.1 Was ist ein Mandant?

Ein GSB11-Mandant ist eine Installation des GSB11 für einen oder mehrere Webauftritte.

- Ein Mandant umfasst:
 - ein Redaktionssystem (TYPO3-Backend),
 - eine gemeinsam genutzte Datenbank,
 - ergänzende technische Komponenten (Suche, Caching etc.).
- Alle Komponenten werden von allen Webauftritten innerhalb des Mandanten gemeinsam genutzt und sind technisch einheitlich.
- Pro Mandant gibt es ein Deployment-Paket (eine Codebasis), das auf der Mandantenumgebung installiert wird.

„Eine Behörde = ein Mandant?“

Nein, nicht zwingend. Innerhalb einer Behörde können:

- mehrere MEO1-Auftritte als Multisite in einem Mandanten betrieben werden,
- stärker individualisierte Auftritte (vormals MEO2) als eigene Mandanten ausgeprägt werden.

3.2 Mandantenooptionen (MEO1 vs. MEO2)

Im ITZBund-Hosting wird der GSB11 in zwei Mandantenooptionen bereitgestellt:

1. Mandantenentwicklungsoptionen 1 (MEO1)

- Anpassung ausschließlich über Konfiguration (keine eigenen Extensions / kein individuelles Design außerhalb der vorgesehenen Möglichkeiten).
- Nutzung der vorhandenen Module und ihres Funktionsumfangs.
- Nutzung des geprüften, barrierefreien Standarddesigns mit definierten Individualisierungsmöglichkeiten.
- Automatische GSB11-Updates, Styleguide-Konformität, responsive Darstellung, BSI-Webcheck und Pentest inklusive.
- Kosteneffizient, da keine Individualentwicklung erforderlich.

2. Mandantenentwicklungsoptionen 2 (MEO2)

- Enthält alle Möglichkeiten der MEO1.
- Zusätzlich: Erweiterung von Funktionsumfang, Templates und Design über ein eigenes Sitepackage und ggf. weitere eigene Extensions.
- Alle Anpassungen jenseits von MEO1 sind mit dem Anforderungsmanagement des BMI (Regelkommunikation via governmentsitebuilder@bmi.bund.de (mailto:governmentsitebuilder@bmi.bund.de)) abzustimmen.
- Genehmigte Anpassungen werden in einer „Allowlist“ dokumentiert (Details in den Contribution Guide).
- Erhöhte Anforderungen an Entwicklungs- und Betriebsverantwortung.

Die Entscheidung für MEO1-Mandant oder MEO2-Mandant wird im Vorfeld eines Migrationsprojekts anhand technischer und fachlicher Kriterien getroffen; Anforderungsmanagement und Rollout-Team unterstützen bei dieser Entscheidung.

Die Abweichung vom Standard führt zu Unterschieden im Service und kann erhebliche Mehraufwände nach sich ziehen

 Im Standard (MEO 1) erhalten Sie durch das ITZBund einen umfänglichen Service über alle Serviceebenen	 Die Abweichung steigert langfristig Ihren Aufwand, bedingt i.d.R. die kontinuierliche Einbindung u. Abhängigkeit von Dienst-leistern (DLs)¹ und kann u.U. die Funktions-fähigkeit im Zuge neuer Releases gefährden
✓ Technischer Support GSB 11 (Fokus Betrieb)	
✓ Fragen zur Anwendung des GSB 11 (das Standardprodukt betreffend)	Keine Beantwortung mandantenspezifischer Fragen → nur durch Entwicklungs-DL
✓ Behebung GSB 11-spezifischer Bugs (das Standardprodukt betreffend)	Keine Behebung konfigurationsbedingter Bugs/Fehler → nur durch Entwicklungs-DL
✓ Sichergestellte Performance, Lauffähigkeit, Verfügbarkeit und Sicherheit über alle non-funktionalen und funktionalen Releases	Eigenständige Sicherstellung Performance, Lauffähigkeit, Verfügbarkeit u. Sicherheit über alle Sprint- u. funktionalen Releases → kontinuierliche Prüfung durch Behörde u. DL je Release (2-wöchentlich) Bearbeitung von Merge Request ² durch DL bei Major Upgrade → Ggf. Blocker für weitere Updates u. Ursache für Release-Verzögerung



¹ Ein begleitender DL muss zwingend vorhanden sein bei einem Webauftritt in der Abweichung
² Techn. Prozess, um neue Code-Anforderungen (z.B. TYPO3-Versionsupgrade) transparent und kontrolliert mit eigenem Code zusammenzubringen

Dienstleistungsabteilung | 45. Nutzendenbeirat | 24.02.2026 | Seite 21

Die MEO1 umfasst ausschließlich die vorgesehenen Konfigurationsmöglichkeiten. Dauerhafte oder wiederkehrende individuelle Anpassungen, die über diesen Rahmen hinausgehen und insbesondere Aufbau, Layout, Logik oder Funktionalität der Website verändern, sind der MEO2 vorbehalten.

Dies gilt unabhängig davon, ob die Anpassungen über Templates, Extensions, TypoScript, fileadmin-Dateien oder andere technische Wege erfolgen.

Ein Beispiel wäre das Abändern von Typoscript-Pfaden für Layout, Template oder Partial auf eigene Dateien.

Dieses stellt im eigentlichen Sinne die einfachste Version einer MEO2 dar (eigenes Template).

3.3 Multisite-Fähigkeit

Der GSB11 ermöglicht grundsätzlich die Administration mehrerer Webauftritte / Domains in einem Mandanten („Multisite“).

Definition Multisite:

- Mehrere Webauftritte (z. B. www.kulturstaatsministerin.de (<http://www.kulturstaatsministerin.de/>) und deutscher-kurzfilmpreis.de (<http://deutscher-kurzfilmpreis.de/>)) werden aus EINEM Redaktionssystem mit EINER Datenbank und EINER Codebasis betrieben.
- Ein Mandant = ein gemeinsames technisches Setup = ein Redaktionssystem = eine Datenbank = eine Codebasis.

Wann ist Multisite sinnvoll?

- Anforderungen aller Webauftritte sind in allen Dimensionen (Funktionen, Design, Dienste, SLA) nahezu identisch.
- Typischerweise:

- Alle Auftritte sind langfristig für MEO1-Mandanten ohne individuelle Entwicklungen.
- Gleiche Admin- und Redaktionsteams über alle Auftritte hinweg.
- Alle Auftritte werden vom gleichen Dienstleister entwickelt und betreut.
- Gleiche SLAs und gleiches Backup-/Recovery-Verhalten (ein gemeinsames Backup für alle Sites).
- Bei MEO2-Setups: identische Featuresets und ein stark angepasstes Frontend für alle Webauftritte.

Wann ist Multisite eher nicht sinnvoll?

- Unterschiedliche Dienstleister mit unabhängigem Deploymentbedarf.
- Unterschiedliche SLAs oder stark unterschiedliche Feature-/Designanforderungen.
- Unterschiedliche interne Stakeholder-Strukturen ohne klaren zentralen Ansprechpartner.

Im Zweifel sollte frühzeitig das GSB11-Produktentwicklungsteam für eine individuelle Beratung eingebunden werden.

4 Zwingende Voraussetzungen und Rollen bei Mandantenentwicklungsoption 2 (MEO2)

Die Entscheidung für die MEO2 geht mit zusätzlichen Verantwortlichkeiten und Verpflichtungen für Behörden und ihre Dienstleister einher. Diese ergeben sich insbesondere

- beim Merge-Request-Management,
- bei der Sicherstellung der Kompatibilität des Sitepackages (das die die Behördenspezifika enthält),
- mit der Produktentwicklung,
- aber auch in Bezug auf die laufenden Sprint-Releases im zweiwöchentlichen Rhythmus.

4.1 Technische Voraussetzungen

Für MEO2 ist zwingend erforderlich:

- Fundierte Kenntnisse in der TYPO3-Entwicklung
- GIT-Kenntnisse
- grundlegendes Verständnis der CI/CD-Pipelines, inkl. Release-Trigger und Release-Arten

Falls diese Kompetenzen nicht in der Behörde vorhanden sind, muss eine Agentur / ein Migrationsdienstleister beauftragt werden, der:

- die Migration durchführt,
- das Sitepackage und ggf. eigene Extensions entwickelt,
- anschließend den laufenden Support, das Merge-Request-Management und die Release-Begleitung übernimmt.

Hinweis

Auch wenn zunächst keine individuellen Anpassungen im Sitepackage vorgenommen werden, gilt ein als MEO2 angelegter Mandant aus Sicht des Betriebs als MEO2, sobald die Option gewählt wurde.

4.2 Sitepackage und eigene Extensions

Sitepackage (GSB11-Kontext):

- Enthält mandantenspezifische Anpassungen:
 - Styling / Corporate Design (CSS, Fonts, Farbvariablen etc.)
 - Templates / HTML-Struktur im Frontend
 - zusätzliche Felder für Inhaltselemente (z. B. Tags, Metadaten)
 - ggf. eigener Build-Prozess für CSS/JS innerhalb des Sitepackages
- Für das Sitepackage sind ausschließlich ViewHelper zulässig; dies wird bei jedem Commit automatisiert geprüft.
- Alle Anpassungen eines MEO2-Mandanten erfolgen über das Sitepackage und ggf. zusätzliche Extensions in einem dedizierten GitLab-Repository des GSB11.

Eigene Extensions (MEO2-Kontext):

- Eigene Extensions werden in einem dedizierten Repository im GSB11-GitLab entwickelt und versioniert.
- Im Sitepackage werden nur stabile Release-Versionen eigener Extensions referenziert (keine RCs).
 - Entwicklungsstände müssen lokal oder in separaten Entwicklungsumgebungen getestet werden.
- Ein Release der eigenen Extension erfolgt durch Merge des main-Banches in den release-Branch.

4.3 Geteilte Verantwortung („Shared Responsibility“)

Bei Individualisierung mittels Sitepackage gilt das Prinzip der geteilten Verantwortung:

- Mandantenentwickler:innen (Behörde bzw. deren Dienstleister)
 - sind verantwortlich für alle Anpassungen/Erweiterungen über MEO1 hinaus,
 - tragen die Verantwortung für Wartung, Updatefähigkeit, Sicherheit und Performance ihrer Individualentwicklungen,
 - müssen sicherstellen, dass das Gesamtsystem durch Eigenentwicklungen nicht negativ beeinflusst wird,
 - sind für die Behebung von Problemen verantwortlich, die durch ihre Anpassungen entstehen (Performance, Lauffähigkeit, Verfügbarkeit, Sicherheit, etc.).
- GSB11-Team (ITZBund)
 - stellt Stabilität und Updatefähigkeit des Basissystems sicher,
 - überwacht die Leistungsfähigkeit des Grundsystems,
 - unterstützt bei der Integration von Erweiterungen, die mandantenübergreifend nachnutzbar gestaltet werden sollen.

Alle Entwicklungen müssen den definierten Richtlinien entsprechen (Technische Rahmenbedingungen, Komponenten des GSB11, Allowlist, Schulungsgrundlagen, Nutzungsbedingungen). Anpassungen sollten so konzipiert werden, dass sie möglichst in die MEO1 überführt werden können.

5 Merge-Requests, Renovate-Bot, Major-Releases

5.1 Automatisierte Merge-Requests (Renovate-Bot)

Im Git-Repository des MEO2-Mandanten entstehen regelmäßig automatisierte Merge-Requests durch den Renovate-Bot (z. B. für Dependency-Updates).

Aufgaben der Dienstleister/Behörden:

- MR sichten
- Relevante Änderungen:
 - prüfen,
 - testen,
 - bei Erfolg mergen
- Irrelevante Änderungen:
 - schließen

Die Bewertung der Relevanz obliegt der Behörde bzw. dem betreuenden Dienstleister.

5.2 Major Releases (z. B. TYPO3-Upgrade)

Bei Major-Releases (z. B. Upgrade von TYPO3 v12 auf v13) entsteht zusätzlicher Aufwand:

- Auflösung von Konflikten im Sitepackage und in eigenen Extensions
- Sicherstellung der Kompatibilität aller Sitepackage-Komponenten mit der neuen Version
- Bearbeitung und Merge der vom Produktteam bereitgestellten Merge-Requests (z. B. Versions-Upgrade MR)

Fallbeispiel:

- Das Produkt GSB11 wird von TYPO3 v12 auf v13 hochgestuft; das Produktteam erstellt einen MR.
- Mandant XY hat eigene Extensions entwickelt, die nicht kompatibel mit v13 sind.
- Anpassungen im eigenen Code zur Kompatibilität liegen in der Verantwortung der Mandantenentwickler:innen.
- Erst nach Umsetzung kann der MR akzeptiert werden.

Für MEO2-Mandanten gilt: MRs des Produktteams (insb. bei Versions-Upgrades) müssen von den Mandantenentwickler:innen akzeptiert und bis zum nächsten Sprint-Release bearbeitet werden.

6 Umgebungen, Infrastruktur und Zugriffe

6.1 Umgebungen im Lebenszyklus

Vor Livegang:

- Es existiert nur die Mandantenumgebung (PCDS).
- Auf PCDS werden Release Candidates und Stable Releases getestet.

Projektlaufzeit / Vorbereitung Livegang:

- Referenzumgebung (REF) und Produktivumgebung (PRD) im Rechenzentrum des ITZBund werden aufgebaut.

Nach Livegang:

- Es existieren drei Umgebungen:
 - Mandantenumgebung (PCDS)
 - Referenzumgebung (REF)
 - Produktivumgebung (PRD)

Stable Releases werden entlang dieser Umgebungen ausgerollt (Details siehe Release-Prozess).

6.2 Formularkonfiguration und Mail-Adressen

- Auf der PCDS-Umgebung:
 - Für Formulare darf ausschließlich no-reply@gsb-itzbund.de (mailto:no-reply@gsb-itzbund.de) als Absenderadresse verwendet werden.
- Auf REF und PRD:
 - Es kann nur eine Absenderadresse im Format whatever@noreply.gsb.bund.de (mailto:whatever@noreply.gsb.bund.de) verwendet werden.

Konsequenz: Nach Übertragung der PCDS-Datenbank auf PRD müssen im Backend die Absenderadressen aller Formulare manuell auf die PROD-konforme Adresse angepasst werden.

Die Formularkonfiguration unterscheidet sich zwischen den Umgebungen!

6.3 Zugriff auf Referenz- und Produktivumgebung

- Zugriff auf REF und PRD ist nur aus dem Netz des Bundes möglich (SINA-Gerät erforderlich).
- Ohne SINA-Gerät ist kein Zugriff auf die Backend-Systeme dieser Umgebungen möglich.

6.4 Synchronisation der Datenbanken

- Die Datenbank der PRD-Umgebung wird regelmäßig (alle paar Tage) mit der REF-Datenbank synchronisiert.

- Die REF-Datenbank wird dabei gelöscht und aus PRD neu aufgebaut.
- PCDS wird **nicht** mit REF oder PRD synchronisiert.
- Inhalte sollen nach Livegang ausschließlich auf PRD gepflegt werden.

6.5 Backend-Nutzer-Regeln

- Dienstleister-Accounts (ohne Behörden-E-Mail, also kein @BehoerdeXYZ.de (http://behoerdexyz.de/)) müssen für REF und PRD explizit gewhitelistet werden.
 - Whitelisting erfolgt durch formlose E-Mail der Behörde an gsb11-hosting@itzbund.de (mailto:gsb11-hosting@itzbund.de).
- Automatische Bereinigung:
 - Konten ohne E-Mailadresse und nicht gewhitelistete Konten werden automatisch gelöscht.
 - Behörden-Accounts mit Behörden-Domäne sind hiervon ausgenommen.

Besondere TYPO3-Rechte:

- Für bestimmte Systemkonfigurationen (z. B. CSP-Settings, Einbindung externer Inhalte) ist das TYPO3-Maintainer-Recht erforderlich.
- Dieses Recht wird nicht standardmäßig vergeben, sondern nur auf Anfrage (Details im Anwendungshandbuch, z. B. Kapitel „Externe Inhalte“).

Ausgeblendete TYPO3-Module:

- Aus Sicherheitsgründen sind einige TYPO3-Module im GSB11-Backend standardmäßig ausgeblendet (Liste siehe [Anwendungshandbuch](https://produkt.gsb.bund.de/gsb11/downloads) (https://produkt.gsb.bund.de/gsb11/downloads)).

Inhaltspflege:

- Inhalte MÜSSEN grundsätzlich auf der bereitgestellten PCDS-Umgebung gepflegt werden.

Grundsätzlich kann keine Daten-Migration von Lokalen Systemen auf die PCDS erfolgen. Grund dafür sind unter anderem, dass Metadaten nicht DSGVO gerecht geprüft werden können und nicht auf Viren geprüft werden können.

6.6 Domain- und DNS-Management

- Für das Produkktivsystem stellt das ITZBund eine Hauptdomain plus Subdomains bereit, maximal 6 Domains.
- Aliases:
 - Mandantenentwickler:innen können (Sub)Domains bei externen Providern beantragen.
 - Diese können selbstständig konfiguriert und per Redirect auf die Hauptdomain im ITZBund geführt werden.

Wenn das Hosting über einen Fremdprovider erfolgen soll:

1. Anlage AKZ und Betriebsnamen im ITZBund

2. Beantragung eines neuen Internetverfahrens beim ITZBund
3. Nach Umsetzung erhält der Kunde die öffentliche IP und gibt diese an seinen Provider weiter (für Umstellung zum Livegang).

Für Deployments in REF und PRD in der ITZBund-CaaS-Umgebung ist ein ITZBund-Kunden-Onboarding erforderlich (fachlicher und betrieblich-technischer Teil).

7 Release-Prozess und GitLab-Workflow

Für MEO2-Mandanten wird für jede Behörde ein eigenes Sitepackage-Repository bereitgestellt (z. B. mit Behördenkürzel). Dieses Repository vereint:

- Individualentwicklung (Sitepackage, ggf. eigene Extensions als Dependencies)
- GSB11-Produktversion (über Merge-Requests aus der Produktentwicklung)

7.1 Branch-Struktur

- `main`-Branch (Default-Branch):
 - Entwicklungszweig, auf dem Release Candidates entstehen.
- `release`-Branch:
 - Release-Zweig für Stable Releases.

`main` und `release` sind **protected** und werden nur über Merge Requests aktualisiert (siehe Beschreibungen oben). Entwicklung läuft lokal auf einem eigenen Feature-Branch, der vom `main` abgezweigt wird und nicht automatisch mitwächst.

7.1.1 Ablauf

1. **Feature-Branch anlegen** – vorher lokal `main` aktualisieren, dann davon abzweigen.
2. **Entwickeln** – Änderungen lokal committen und auf den eigenen Feature-Branch pushen, so oft man will.
3. **Vor dem Merge Request: Rebase** – `main` lokal aktualisieren, dann den Feature-Branch auf den aktuellen `main` rebasen. Konflikte dabei lokal auflösen. Anschließend mit Force-with-lease pushen (Rebase schreibt die History um).
4. **Merge Request öffnen** – Ziel `main`, Pipeline grün abwarten, Fast-Forward-Merge.
5. **Aufräumen** – `main` ziehen, Feature-Branch lokal löschen.

Achtung: Wenn Sie eigene Branches im Git erstellen, werden diese nicht von uns aktualisiert! Folgender Feature Branch Workflow mit Rebase vor Merge wird dann notwendig:

7.2 Release Candidate (RC)

Alle RCs werden jeden zweiten Donnerstag nach Sprint-Review zu Stable Releases (zweiwöchentlich!)

- Entstehen durch Merge-Requests in den `main`-Branch.
- Werden automatisch auf der Mandantenumgebung (PCDS) ausgerollt (Deployment ~ 30 Minuten).
- Dienen als Teststand und gehen zunächst **nicht** auf REF oder PRD.

7.3 Stable Release

- Stable Releases bilden die Grundlage für produktive Deployments.

Eigenschaften:

- werden automatisch alle zwei Wochen erzeugt:
 - Zeitpunkt: Donnerstag nach Sprintende (Sprint-Review-Tag)
- können jederzeit vorzeitig erzeugt werden, indem ein MR vom `main` in den `release`-Branch erstellt und gemerged wird.
- Stable Releases werden:
 - auf PCDS und
 - (wenn vorhanden) auf REF deployed,
 - und bilden anschließend die Basis für das Deployment auf PRD.
 - Sobald Sie einen MR in ihrem release-branch mergen, wird ein stable Release getriggert (sollte nur bei Bedarf getriggert werden, siehe Punkt 3. Stable Release) und in etwa 2 Stunden auf der REF-Umgebung automatisiert deployed sein.

Zusammenfassung:

- RCs: nur PCDS, jederzeit durch MRs in `main`.
- Stable Releases:
 - manuell jederzeit (MR `main` → `release`)
 - automatisch alle zwei Wochen nach Sprint-Review.

Empfehlung:

- Das jeweils neue Stable Release sollte im Sichtprüfungszeitraum (siehe Release-Kalender) auf REF geprüft werden.
- Wird es nicht gesondert behandelt, wird es – sobald der Mandant live ist – automatisch in PRD ausgerollt.

8 Hinweise zu Template-, Design- und Frontend-Anpassungen

- Ziel des GSB11-Frontends ist ein effizient aktualisierbarer Frontend-Stack mit möglichst wenigen npm-Paketen.
- Das Standard-Design- und Templatesystem basiert auf Bootstrap 5.
 - Das Markup folgt Bootstrap-Konventionen, ergänzt um TYPO3-Markups und Fluid-Variablen.
 - Die offizielle Bootstrap-Dokumentation ist für Anpassungen sehr hilfreich.

Versionierung & Updates:

- Alle GSB11-Extensions (inkl. Frontend-Bestandteile) folgen SemVer 2.0.
- Minor- und Patch-Releases
von `gsb_core`, `gsb_public_frontend` und `gsb_private_frontend` können grundsätzlich mandantenweit eingespielt werden.
- Für jede Frontend-Komponente steht ein Changelog zur Einschätzung:
 - ob eine Komponente betroffen ist,
 - ob ein Breaking Change vorliegt,
 - welche Bestandteile (Styles, Assets, JS, Settings, Markup) betroffen sind.

CSS & JavaScript:

- CSS und JS von `gsb_public_frontend` werden vorkompiliert ausgeliefert.
- Mandantenentwickler:innen können über das Sitepackage eigenes CSS (insb. CSS-Variablen) und JS ergänzen.
- Ein eigener Build-Prozess im Sitepackage ist möglich.
- Das GSB11-Markup enthält keine Inline-CSS-Definitionen, um eine hohe Wiederverwendbarkeit zu ermöglichen; Fluid-Template-Overrides sollten selten nötig sein.

9 Caching, Performance und Sicherheit

9.1 Caching-Architektur

Der GSB11 verfügt über ein mehrstufiges, verteiltes Caching:

- Mehrere Pods für Redaktionssystem und Website-Frontend
- Dateibasierte Caches sind nicht pod-übergreifend geteilt.
- Redis- und DB-basierte Caches werden zentral gespeichert.
- Caching-Extensions:
 - `gsb_clustered_caching` und `gsb_nginx_proxy` steuern und optimieren das Cache-System.
 - Ein vorgeschalteter Caching-Proxy arbeitet wie ein Full-Page-Cache für Webseitenaufrufe.
 - Cache-Flushes in TYPO3 (clear-cache) invalidieren automatisch den Proxy-Cache.

9.2 Content Security Policy (CSP) und Hashes

- GSB11 setzt in der CSP Hash-Methoden ein, um die Integrität von Skripten und CSS sicherzustellen.
- Dies erlaubt, ganze Seiten (inkl. JS/CSS) sicher zu cachen und schützt vor XSS-Angriffen.
- `gsb_csp` sorgt für die korrekte Implementierung der Content Security Policy.

9.3 Performance-Metriken und Best Practices

- Mehrstufiges Caching reduziert Ladezeiten und erhöht die Cache-Hitrate deutlich.
- Für Formulare, Suchergebnisse und geschützte Bereiche gelten Einschränkungen in der Cachebarkeit.

Empfehlungen:

- Sorgfältige Konfiguration der Caching-Einstellungen; regelmäßige Kontrolle, ob alle Caches korrekt in das TYPO3-Caching-Framework eingebunden sind.
- Berücksichtigung der Cache-Invalidierung bei eigenen Caches; GSB11 folgt den TYPO3-Konzepten.
- Sicherstellen, dass alle verwendeten Extensions aktuell sind (Sicherheits- und Performance-Updates).

10 Qualitätssicherung, BSI-Webcheck und Findings

- Die Grundbausteine des GSB11 werden regelmäßig BSI-Webchecks unterzogen und gelten als geprüft.
- Anpassungen/Ergänzungen von PHP-Klassen und JavaScript im Mandanten-Sitepackage können einen erneuten Prüfbedarf (vereinfachter Teil-Webcheck) auslösen.
 - Solche Anpassungen werden in der GitLab-Pipeline erkannt; Mandantenentwickler:innen werden ggf. per Pipeline-Output oder E-Mail informiert.

Umgang mit Findings aus Sichtprüfungen:

- Festgestellte Findings sind im unmittelbar folgenden Sprint zu beheben.
- Meldung der Findings soll spätestens am Donnerstag der zweiten Sprintwoche erfolgen.
- Spätere Meldungen werden erst im darauffolgenden Sprint berücksichtigt.

11 Organisatorische Hinweise und Agenturverantwortung

- Für Betrieb und Weiterentwicklung eines Webauftritts ist eine dauerhaft verantwortliche Agentur erforderlich.
- Es sollte eine Agentur gewählt werden, die:
 - nicht nur die initiale Einrichtung begleitet,
 - sondern den Webauftritt langfristig administriert, betreut und weiterentwickelt.

12 Weiterführende Dokumentationen

Für vertiefende Informationen werden u. a. folgende Dokumente benötigt (nur für Behörden des Bundes bestimmt):

- Anwendungshandbuch (z. B. Kapitel zu externen Inhalten, Backend-Modulen, Rechtekonzept)
- [Releasekonzept GSB11](#)
- [Patchkonzept GSB11](#)
- [Releasemanagementkonzept](#)
(<https://social.intranet.bund.de/inhalte/yUKtAI7OueLnOnoMRBtZyw>)
- Release-Kalender

Zugang zu diesen Dokumenten kann über die zuständigen Kontakte in der Behörde angefragt werden.