

Room Server, Bots & Dienste

MeshCore Hessen – Room Server, Bots & Dienste

- [Dienste](#)
 - [Bot-Architektur auf dem Raspberry Pi](#)
 - [Status- und Informationsbot](#)
 - [Room Server oder Gruppenkanal?](#)
 - [Automatischer Testdienst für Empfangsprüfungen](#)
 - [Room Server verstehen](#)
 - [Weitere sinnvolle MeshCore-Hessen-Dienste](#)
 - [Room Server konfigurieren](#)
 - [Bot-Monitoring und Healthchecks](#)
 - [Bot-Sicherheit und Missbrauchsschutz](#)
 - [Was ist ein MeshCore-Bot?](#)

Dienste

Dienste

Bot-Architektur auf dem Raspberry Pi

Für MeshCore Hessen kann ein Raspberry Pi als lokaler Host für automatisierte MeshCore-Dienste dienen.

Companion

Ein dediziertes MeshCore-Companion-Gerät wird dauerhaft mit dem Pi verbunden. Für einen stationären Aufbau ist USB-Serial oft einfacher zu überwachen als eine drahtlose Hostverbindung.

Botprozess

Ein eigener Prozess liest MeshCore-Ereignisse, prüft Befehle und erzeugt Antworten. Die Implementierung kann auf einer vorhandenen Bibliothek oder einer stabilen CLI-/Protokollschnittstelle aufbauen.

Dienstmanager

Der Bot läuft als systemd-Dienst und startet nach einem Reboot automatisch neu.

MQTT-Client

Optional veröffentlicht ein separater Teil ausgewählte Statusereignisse an MQTT. Die Funkfunktion bleibt davon entkoppelt.

Lokale Daten

Konfiguration, Rate-Limit-Zustand und gegebenenfalls notwendige Betriebsdaten werden lokal gespeichert. Chatverläufe werden nicht ohne definierten Zweck dauerhaft gesammelt.

Logs

Logs enthalten technische Fehler und Zustandswechsel. Nachrichteninhalte werden nur dann protokolliert, wenn dies für den konkreten Dienst erforderlich und datenschutzrechtlich vorgesehen ist.

Updates

Botcode, Abhängigkeiten und MeshCore-Firmware werden versioniert dokumentiert. Änderungen werden zuerst mit einem Testgerät geprüft.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Dienste

Status- und Informationsbot

Neben einem Pingdienst kann ein Bot bewusst ausgewählte Betriebsinformationen bereitstellen.

Status

Ein `status`-Befehl kann beispielsweise melden, dass der Bot läuft und wann seine Anwendung gestartet wurde.

Hilfe

Mit `help` beziehungsweise einem kurzen Hilfebefehl werden ausschließlich die tatsächlich unterstützten Funktionen angezeigt.

Infrastrukturinformationen

Ein Bot kann öffentliche Informationen wie Wiki-Adresse, Hinweise zur Einrichtung oder den Namen eines Testkanals ausgeben.

Keine Geheimnisse

Interne IP-Adressen, Passwörter, exakte nicht öffentliche Repeaterstandorte oder Administrationsdaten werden niemals über einen öffentlichen Funkbefehl ausgegeben.

Dynamische Daten

Wetter, Warnmeldungen oder andere Internetdaten wären zusätzliche Dienste und müssen jeweils hinsichtlich Quelle, Aktualität, Lizenz und Ausfallverhalten separat geplant werden.

Antwortlänge

LoRa ist kein Webbrowser. Botantworten bleiben kurz und werden nicht für lange Wikiartikel oder umfangreiche Listen missbraucht.

Priorität

Der Testdienst bleibt technisch einfach. Zusatzfunktionen dürfen nicht dazu führen, dass eine externe API-Störung den grundlegenden Empfangstest blockiert.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Room Server oder Gruppenkanal?

Room Server und Gruppenkanal lösen unterschiedliche Aufgaben und sollten nicht als austauschbare Begriffe verwendet werden.

Gruppenkanal

Ein Gruppenkanal ist Teil der Nachrichtenkommunikation zwischen Clients. Teilnehmer mit der passenden Kanalkonfiguration können Gruppenmeldungen austauschen.

Room Server

Ein Room Server ist dagegen ein eigener erreichbarer MeshCore-Knoten mit serverseitiger Raumfunktion.

Persistenz

Der zentrale Unterschied liegt in der Serverfunktion: Der Room Server kann Posts für den Raum verwalten, während ein normaler Gruppenkanal nicht einfach ein zentraler BBS-Speicher ist.

Infrastrukturbedarf

Ein Kanal benötigt keinen zusätzlichen Room-Server-Knoten. Ein Room Server benötigt dagegen eigene unterstützte Hardware und laufende Firmware.

Anwendungsfall

Für spontane oder reguläre Gruppenkommunikation ist ein Kanal passend. Für einen bewusst betriebenen gemeinsamen Raum mit Serverfunktion kann ein Room Server sinnvoll sein.

MeshCore Hessen

Ein allgemeiner regionaler Kommunikationskanal und ein MeshCore-Hessen-Room-Server können parallel existieren. Im Wiki werden beide Funktionen getrennt erklärt, damit Nutzer wissen, womit sie interagieren.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Automatischer Testdienst für Empfangsprüfungen

Ein automatischer Testdienst kann Nutzern helfen zu prüfen, ob ihre Nachricht das Mesh erreicht und eine Antwort zurückkommt.

Ziel

Der Dienst soll keine allgemeine Chat-KI sein. Seine Kernaufgabe ist eine reproduzierbare technische Rückmeldung.

Einfacher Befehl

Ein definierter Befehl wie `ping` kann eine kurze Antwort auslösen. Die Antwort sollte eindeutig zeigen, dass sie automatisch erzeugt wurde.

Was ein erfolgreicher Test beweist

Eine empfangene Botantwort zeigt, dass die Testnachricht den Bot erreicht hat und die Antwort den Weg zurück zum Nutzer geschafft hat.

Sie beweist nicht, dass jeder andere Knoten im Netz erreichbar ist oder dass die Verbindung dauerhaft dieselbe Qualität besitzt.

Sinnvolle Antwortdaten

Der Bot kann – soweit die verwendete Schnittstelle diese Informationen verlässlich bereitstellt – Zeit, Dienstname und technische Empfangsinformationen zurückgeben.

Es werden keine Werte erfunden, die der Bot tatsächlich nicht messen kann.

Rate Limit

Ein Nutzer soll nicht beliebig viele automatische Antworten pro Sekunde auslösen können. Eine kurze Sperrzeit schützt Airtime und Botprozess.

Schleifenschutz

Automatische Antworten anderer Bots dürfen nicht wiederum neue Antworten auslösen. Botnachrichten werden daher erkannt beziehungsweise nur explizite Benutzerbefehle verarbeitet.

Offlinebetrieb

Für MeshCore Hessen sollte die grundlegende Ping-Antwort lokal auf dem Raspberry Pi funktionieren. MQTT und VPS können ergänzend Statistiken erhalten, sind aber nicht Voraussetzung für die Funkantwort.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Room Server verstehen

Ein Room Server ist eine eigene MeshCore-Rolle für einen gemeinsamen Nachrichtenraum. Das offizielle Projekt beschreibt ihn als einfachen BBS-Server für gemeinsam genutzte Posts.

Eigene Rolle

Ein Room Server ist nicht bloß ein Kanalname und auch kein Internet-Chatserver. Er läuft als MeshCore-Firmware auf unterstützter LoRa-Hardware und erscheint als eigener Knotentyp.

Zugriff

Clients können mit einem Room Server interagieren und dort Nachrichten beziehungsweise Posts austauschen. Die konkreten Clientfunktionen hängen vom aktuellen Softwarestand ab.

Administration

Room Server können lokal seriell und über unterstützte Remote-Management-Funktionen administriert werden. Die offizielle CLI-Referenz dokumentiert gemeinsame und roomspezifische Kommandos.

Read-only

Die Firmware besitzt eine Option `allow.read.only`. Damit kann ein Room Server Lesezugriff ohne normales Postrecht ermöglichen, wenn dies entsprechend konfiguriert wird.

Repeaterfunktion

Die Room-Server-Firmware besitzt die Option `set repeat on|off`. Ein Room Server kann damit zusätzlich die Repeaterfunktion aktivieren.

Das bedeutet nicht, dass Room Server und Repeater in jeder internen Funktion identisch sind. Bei Planung und Diagnose bleibt der Knotentyp relevant.

Einsatz bei MeshCore Hessen

Ein Room Server eignet sich für einen persistenten gemeinsamen Treffpunkt im Funknetz. Für automatisierte Testantworten ist dagegen ein Host-basierter Bot oft flexibler.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Weitere sinnvolle MeshCore-Hessen-Dienste

Nach einem stabilen Testbot können weitere Dienste ergänzt werden. Jeder Dienst sollte einen klaren Nutzen besitzen und sparsam mit Airtime umgehen.

Echo-Test

Ein einfacher Echo-Befehl kann einen kurzen vom Nutzer vorgegebenen Testwert zurückgeben. Die Eingabelänge wird streng begrenzt.

Zeitdienst

Ein Bot kann die Zeit seines Hosts ausgeben. Dabei wird klar angegeben, welche Zeitzone verwendet wird.

Infrastrukturstatus

Öffentliche Repeater können mit einem kurzen Statusdienst zusammengefasst werden, sofern die dafür verwendeten Daten zuverlässig vorliegen.

Wiki-Hilfe

Kurze Befehle können auf passende Wiki-Themen hinweisen. Lange Inhalte werden nicht über LoRa übertragen.

Störungsmeldung

Ein standardisierter Befehl könnte eine technische Störungsmeldung an das Betreiberteam weitergeben. Vor Umsetzung müssen Missbrauchsschutz, Speicherung und Datenschutz geklärt

werden.

Warninformationen

Amtliche Warninformationen wären technisch möglich, benötigen aber eine verlässliche offizielle Datenquelle, Aktualitätsprüfung, Gebietsauswahl und eine klare Kennzeichnung. Sie werden nicht ohne eigenes Konzept in den Testbot eingebaut.

Kein Bot-Zoo

Mehrere Dienste sollten möglichst auf einer gemeinsamen, gut gewarteten Plattform laufen, statt für jede kleine Funktion einen zusätzlichen Funkknoten zu betreiben.

Priorität

Für MeshCore Hessen lautet die sinnvolle Reihenfolge: stabiler Pingdienst, Monitoring, klare Betriebsdokumentation und erst danach zusätzliche Komfortfunktionen.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Room Server konfigurieren

Die Konfiguration eines Room Servers ähnelt in Teilen der Repeaterkonfiguration, besitzt aber zusätzliche Einstellungen für den Raumzugriff.

Name und Funkparameter

Der Server erhält einen eindeutigen Namen und dieselben kompatiblen Radioeinstellungen wie das gewünschte Mesh.

Administrationspasswort

Ein produktiver Room Server erhält ein eigenes starkes Administrationspasswort. Dieses wird nicht mit Gastzugängen oder Botpasswörtern wiederverwendet.

Gastzugang

Die offizielle CLI-Referenz bietet Einstellungen für Gastpasswort und Lesezugriff. Welche Rechte ein Gast benötigt, wird nach Einsatzzweck festgelegt.

Read-only

Mit der roomspezifischen Read-only-Option kann ein Zugriff ohne Schreibrecht ermöglicht werden. Das ist beispielsweise für einen öffentlichen Informationsraum interessant.

Repeat

Soll derselbe Knoten zusätzlich Pakete weiterleiten, wird die Repeaterfunktion bewusst aktiviert. Zuvor werden Standort, Funklast und Netzplanung betrachtet.

Flood und Adverts

Advert-Intervalle und Flood-Grenzen werden wie bei anderer Infrastruktur an das regionale Netz angepasst.

Dokumentieren

Firmwareversion, Radio-Preset, Zugriffsmodell und Repeaterstatus werden dokumentiert. Zugangsdaten selbst gehören nicht ins öffentliche Wiki.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Bot-Monitoring und Healthchecks

Ein Bot kann als Prozess laufen und trotzdem keine Funkverbindung mehr besitzen. Monitoring muss deshalb mehrere Ebenen unterscheiden.

Prozessstatus

Der erste Healthcheck prüft, ob der Dienstprozess läuft.

Companion-Verbindung

Zusätzlich wird geprüft, ob der Companion erreichbar ist und erwartete Statusinformationen liefert.

Funkaktivität

Ein weiterer Indikator kann sein, ob innerhalb eines normalen Zeitraums MeshCore-Ereignisse empfangen wurden. In einem sehr ruhigen Netz ist fehlender Verkehr jedoch nicht automatisch ein Fehler.

MQTT

Wenn MQTT verwendet wird, veröffentlicht der Gatewaydienst einen eigenen Online-/Heartbeatstatus. Ein MQTT-Ausfall wird separat von einem MeshCore-Ausfall gemeldet.

Selbsttest

Ein vollständiger Ende-zu-Ende-Test über LoRa benötigt einen zweiten unabhängigen Knoten. Der Bot kann nicht allein beweisen, dass seine eigene Antenne tatsächlich erfolgreich sendet und empfängt.

Alarmierung

Störungen können an Administratoren gemeldet werden. Der Alarmweg sollte nicht ausschließlich über denselben MeshCore-Knoten laufen, dessen Ausfall gerade erkannt werden soll.

Historie

Uptime, Neustarts und Verbindungsabbrüche werden als technische Betriebsdaten gespeichert. Dadurch lassen sich sporadische USB- oder Stromprobleme erkennen.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Bot-Sicherheit und Missbrauchsschutz

Ein öffentlich erreichbarer Bot verarbeitet Eingaben fremder Teilnehmer. Er muss deshalb so gebaut werden, dass Nachrichten nicht direkt zu gefährlichen Systembefehlen werden können.

Keine Shell aus Chattertext

Benutzereingaben werden niemals ungeprüft in eine Shell, SQL-Abfrage oder Dateipfadoperation eingesetzt.

Whitelist

Der Bot akzeptiert eine kleine Liste definierter Befehle. Unbekannte Eingaben werden ignoriert oder mit einer kurzen Hilfe beantwortet.

Längenbegrenzung

Eingaben und Antworten erhalten feste Größenlimits.

Rate Limit

Pro Absender und zusätzlich global wird die Antwortfrequenz begrenzt. Damit kann ein einzelner Nutzer nicht unbegrenzt automatischen Funkverkehr erzeugen.

Rechte

Der Botprozess läuft auf Linux nicht als root, sofern dies technisch nicht erforderlich ist. Er erhält nur Zugriff auf die benötigte serielle Schnittstelle und seine eigenen Dateien.

MQTT

MQTT-Zugangsdaten liegen in geschützten Konfigurationsdateien beziehungsweise Umgebungsvariablen und werden nicht in ein öffentliches Repository eingchecked.

Remote-Administration

Botbefehle und Repeater-Administrationsbefehle werden getrennt. Ein öffentlicher Chatbefehl darf nicht indirekt beliebige Repeater-CLI-Kommandos ausführen.

Updates

Abhängigkeiten werden gepflegt. Sicherheitsupdates des Betriebssystems und verwendeter Bibliotheken gehören zum Betriebsplan.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026

Was ist ein MeshCore-Bot?

Ein Bot ist keine eigene offizielle MeshCore-Firmware-Rolle. Gemeint ist eine externe Anwendung, die über einen Companion beziehungsweise eine Host-Schnittstelle am Mesh teilnimmt und automatisch auf Ereignisse reagiert.

Host

Der Bot kann beispielsweise auf einem Raspberry Pi, Linux-Server oder anderen dauerhaft laufenden Rechner ausgeführt werden.

Funkzugang

Der Host benötigt einen Weg ins Mesh. Dafür kann ein unterstützter Companion per Serial, BLE oder TCP verwendet werden.

Software

Die offizielle meshcore-cli kann Befehle aus Skripten ausführen. Für eigene Programme stehen außerdem Bibliotheken und die dokumentierten Companion-Protokolle als Grundlage zur Verfügung.

Identität

Ein Bot sollte als eigener Dienst erkennbar sein und nicht unbemerkt die Identität einer privaten Person nachahmen.

Reaktionen

Bots können auf definierte Nachrichten reagieren, Statusinformationen liefern oder technische Tests ausführen.

Grenzen

Automatisierung erzeugt Funkverkehr. Ein schlecht programmierter Bot mit Antwortschleifen kann den gemeinsamen Kanal unnötig belasten.

MeshCore-Hessen-Regel

Bots werden zunächst in einem Testbereich erprobt, erhalten Rate Limits und reagieren nur auf klar definierte Befehle.

Quellen, Rechte und redaktionelle Prüfung

- [MeshCore – offizielle Dokumentation](#)
- [meshcore-cli – offizielles Projekt](#)
- [Bundesnetzagentur – Allgemeinzuteilungen](#)
- [Bundesnetzagentur – Vfg. 91/2025](#)
- [MeshCore Wiki DE – ergänzende Communityquelle](#)

Redaktioneller Hinweis: Diese Seite ist eine eigenständige Darstellung für MeshCore Hessen. Sachinformationen wurden anhand der oben genannten Quellen geprüft. Texte, Bilder und die Gliederung des MeshCore Wiki DE wurden nicht übernommen; die Seite wird nur als ergänzende Communityquelle zur Faktenkontrolle genannt. Für technische Details hat die aktuelle offizielle MeshCore-Dokumentation Vorrang. Deutschlandbezogene Funkrechtsangaben werden an den Veröffentlichungen der Bundesnetzagentur geprüft.

Urheberrecht: Technische Begriffe, Protokollnamen, Befehlsnamen, Zahlenwerte und andere notwendige Sachangaben werden unverändert verwendet, soweit dies für fachliche Richtigkeit erforderlich ist. Es werden keine längeren fremden Textpassagen oder Bilder reproduziert.

Prüfstand: 18. August 2026