



20. Mai 2026

17. Bechtle
IT-Forum
Thüringen
Steigerwald Stadion Erfurt

20
26

Digitale Verwaltung

Digitale Souveränität umsetzen statt
diskutieren: Vom Stack zur echten
Verwaltungsmodernisierung

Digitale Souveränität entsteht
nicht allein. **Sie entsteht
im Zusammenspiel.**



Warum digitale Souveränität jetzt praktisch werden muss

Digitale Souveränität ist kein politisches Schlagwort mehr, sondern eine operative Notwendigkeit

Wir stehen vor **drei** gleichzeitigen Herausforderungen:

- **EU-Regulierung**
 - **Fachkräftemangel**
 - **Modernisierungsdruck**
- ▶ **Die Frage ist nicht ob, sondern wie wir souverän werden**

Technische Ausgangslage

Herausforderungen in Landes- und Kommunalarchitekturen:



Heterogene Infrastruktur, historisch gewachsene Fachverfahren

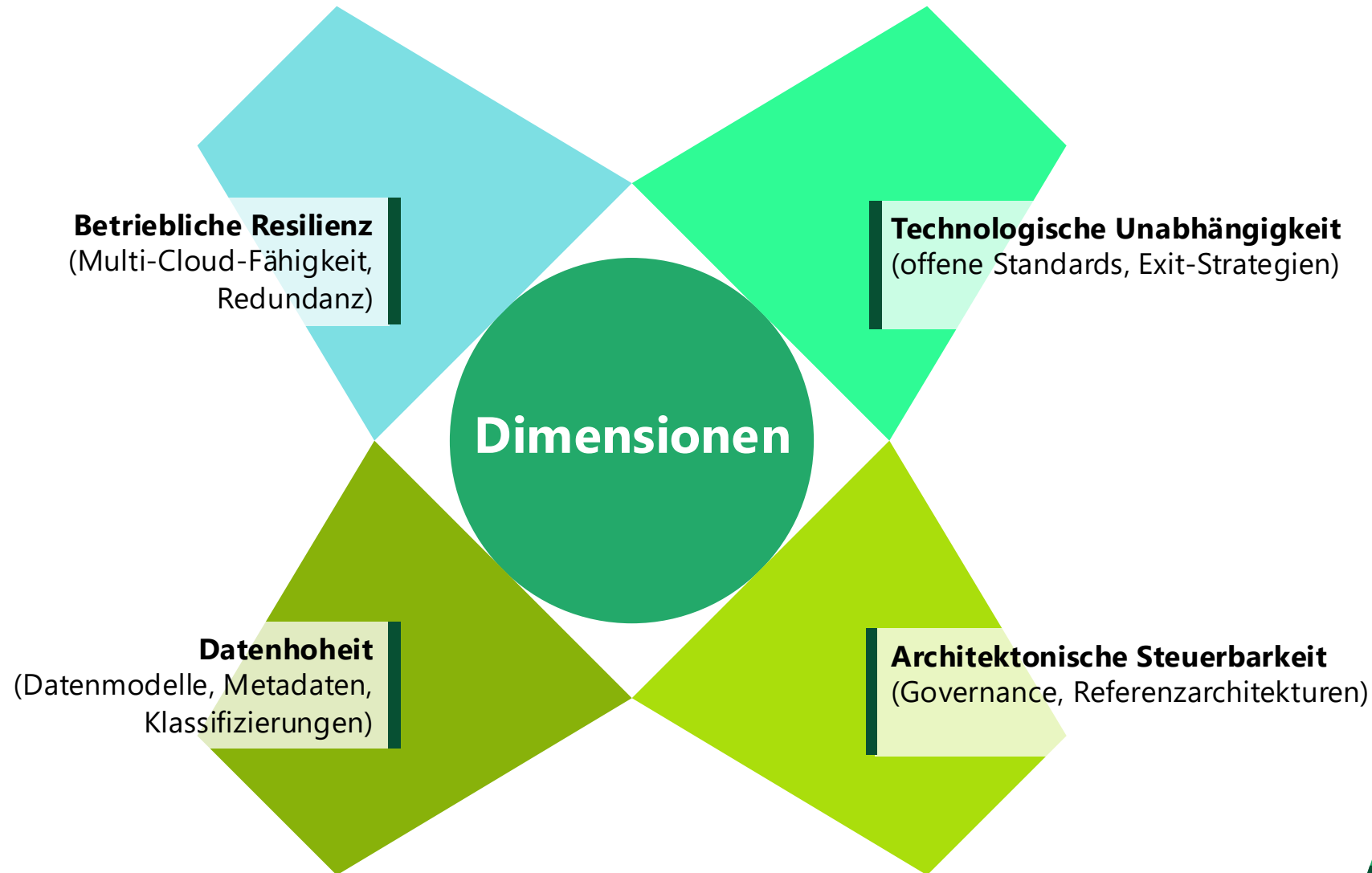
Unterschiedliche Betriebsmodelle
(On-Prem, Private Cloud, Public Cloud)

Proprietäre Schnittstellen, fehlende Interoperabilität

Hohe Integrationskosten, geringe Wiederverwendbarkeit

Sicherheitsanforderungen steigen
(Zero Trust, KRITIS, NIS2)

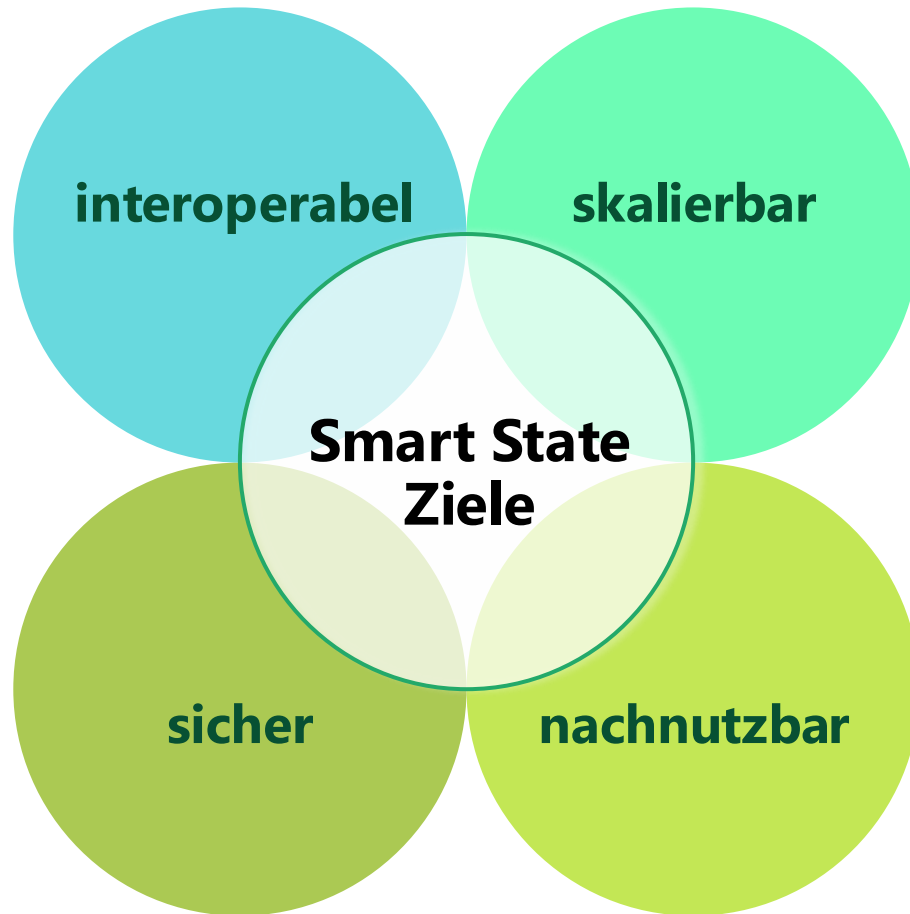
Digitale Souveränität technisch definiert



Vom Leitbild zur Umsetzung – Smart State Thüringen

Mit Smart State haben wir ein strategisches Zielbild.

Doch Strategie allein verändert keine Verwaltung.



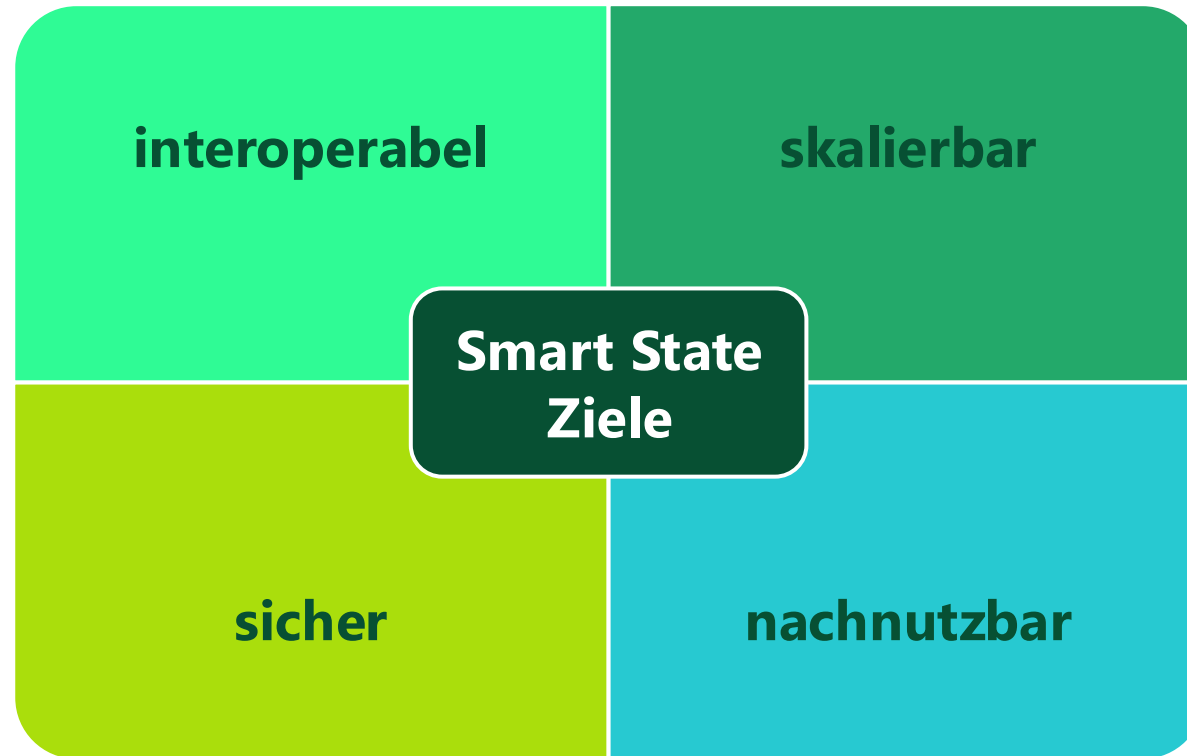
Entscheidend ist die Operationalisierung:

Welche Standards setzen wir? Welche Technologien?

Welche Governance? Wie bringen wir das in die Fläche?"

Vom Leitbild zur Umsetzung – Smart State Thüringen

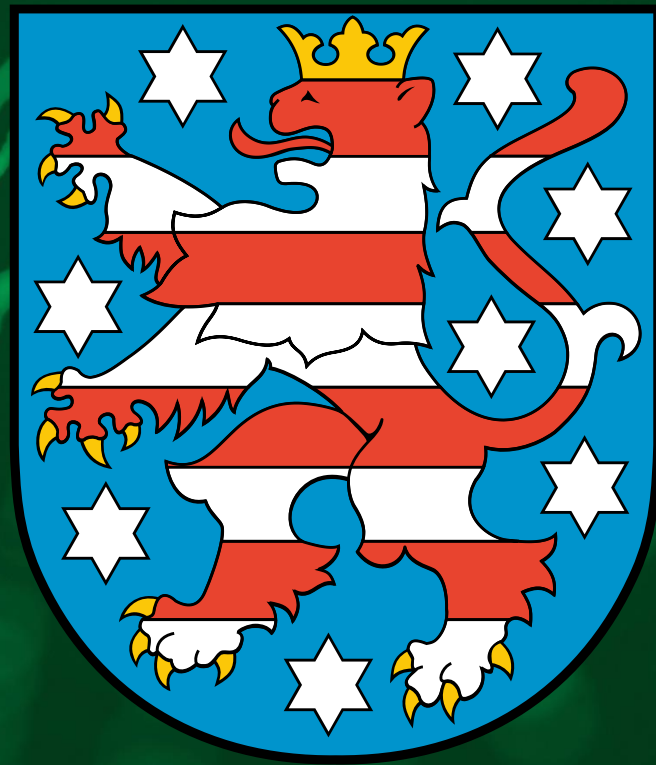
Mit Smart State haben wir ein strategisches Zielbild.
Doch Strategie allein verändert keine Verwaltung.



Entscheidend ist die Operationalisierung:

**Welche Standards setzen wir? Welche Technologien? Welche Governance?
Wie bringen wir das in die Fläche?"**

Thüringen-Stack



Der Thüringen Stack: Technische Zielarchitektur 1/4

Layer-Modell:

1. Infrastruktur-Layer

- **Kubernetes-basierte Betriebsplattform**
- **Automatisiertes Provisioning (IaC)**
- **Netzwerksegmentierung nach Zero-Trust-Prinzipien**

Der Thüringen Stack: Technische Zielarchitektur 2/4

Layer-Modell:

2. Plattform-Layer

- **API-Gateway, Identity-Provider, Event-Bus**
- **Low-Code-Plattform, Workflow-Engine**

Der Thüringen Stack: Technische Zielarchitektur 3/4

Layer-Modell:

3. Applikations-Layer

- **Modularisierte Fachverfahren**
- **Wiederverwendbare Komponenten**

Der Thüringen Stack: Technische Zielarchitektur 4/4

Layer-Modell:

4. Prozess- und Daten-Layer

- **FIM-Bausteine, BPMN-Modelle**
- **Standardisierte Datenmodelle** (XÖV, OZG-Standards)

Anforderungen an Architekturprinzipien des Thüringen-Stacks

- **API-First**
- **Modularisierung statt Monolithen**
- **Event-Driven-Architecture**
- **Security-by-Design**
- **Cloud-agnostisch**
- **Konfigurationsgetrieben statt Code-getrieben**

Einordnung in den Deutschland-Stack

Deutschland Stack – Thüringen Stack – Schnittmengen

Der Thüringen Stack ist kein Alleingang.

Er fügt sich in die bundesweiten Entwicklungen ein.

- **Kompatibilität zu FIT-Connect**
 - **Nutzung von XÖV-Standards**
 - **OZG-Kompatibilität**
 - **Interoperabilität über föderale Schnittstellen**
- **Damit wird Thüringen anschlussfähig – und trägt gleichzeitig zur Weiterentwicklung des Deutschland Stacks bei.**

Architektur und Governance



Thüringer Ansatz – Integration in Landesstrukturen 1/2

Ressortübergreifende Governance (Freistaat Thüringen)

- **Steuerung über bestehende Strukturen**
(CIO-Stab, Koordinierungsstelle, Lenkungsausschuss, Arbeitskreis, etc.)
- **Beispiel: Verzahnung von Digitalisierungsprojekten über zentrale Steuerungsgremien**

Idee: Landes-IT-Architekturboard Thüringen

- **Konsolidierung der E-Government-Zielarchitektur**
- **Beispiel: Bewertung und Steuerung von Technologien**
(z. B. Plattform- und Architekturentscheidungen)

Thüringer Ansatz – Integration in Landesstrukturen 2/2

Anbindung an Bund & Nachnutzung (EfA/OZG)

- **Integration von IT-Planungsrat-Beschlüssen in Landeslösungen**
- **Beispiel: OZG-Umsetzung und EfA-Nachnutzung**

Konkrete Themenintegration: Daten & KI

- **Erweiterung bestehender Gremien um Datenmanagement- und KI-Themen**
- **Beispiel: landesweite Verankerung von Datenkompetenz und KI-Ansätzen**

- ▶ **Durchgängige, interoperable Landesarchitektur für Verwaltung und Kommunen**

Fazit

Governance ist kein Hindernis – sondern Voraussetzung

- **Enabler für Innovation und Umsetzung**
- **Sicherstellung von Qualität und Skalierbarkeit**
- **Grundlage für digitale Souveränität**

► **Erfolgsfaktor:**

**„Kein perfekter Masterplan –
aber klare Governance + funktionierender Stack“**

Technologie



Technologieeinführung: Low Code als Plattformkomponente

Technische Anforderungen:

- **API-Integration** (REST, GraphQL)
- **Event-Integration** (Kafka, AMQP)
- **Versionierung & Deployment-Automatisierung**
- **Rollen- & Berechtigungskonzepte**
- **Mandantenfähigkeit**
- **Logging, Monitoring, Telemetrie**

Bewertungsmethodik für Plattformtechnologien

Technische Kriterien:

- **Architekturkompatibilität**
- **Erweiterbarkeit** (Custom Code, Plugins, SDKs)
- **Datenhaltung & Exportfähigkeit**
- **Integrationsfähigkeit** (APIs, Events, Connectors)
- **Sicherheitsarchitektur** (Zero Trust, MFA, Audit Logs)
- **Betriebsmodelle** (On-Prem, Cloud, Hybrid)

Lessons Learned aus der technischen Einführung

- **Low Code benötigt harte Governance, sonst entstehen Schatten-IT**
- **Bausteinbibliotheken müssen versioniert & automatisiert ausgerollt werden**
- **Styleguide + Komponentenbibliothek = konsistente UX**
- **Technische Schulung ist entscheidend**
(DevOps, API-Design, Security)
- **Integration in CI/CD ist Pflicht, nicht Kür**

Fazit

Digitale Souveränität ist ein Architektur- und Plattformthema.

Wer sie erreichen will, braucht:

- **einen funktionierenden Stack,**
- **klare technische Governance und**
- **konsequente Umsetzung.**

**Digitale Souveränität ist erreichbar –
wenn wir sie nicht nur diskutieren, sondern umsetzen.**

**Mit einem klaren Stack, verbindlicher Governance und
pragmatischer Technologieeinführung schaffen wir eine moderne,
sichere und interoperable Verwaltung.**

Vielen Dank!



Die Zukunft
gemeinsam
im Fokus

Anhang

Zero Trust



Zero Trust Architektur 1/2

Zero Trust als Prinzip souveräner Architekturen

Souveränität braucht Sicherheit.

Zero Trust ist kein Produkt, sondern ein Architekturprinzip.

Wir verankern Zero Trust im Stack:

- **Identität als neue Perimeter**
- **durchgängige Authentifizierung**
- **Mikrosegmentierung**
- **kontinuierliche Überprüfung statt implizitem Vertrauen**

Zero Trust Architektur 2/2

Technische Umsetzung:

- **Identity-zentrierte Zugriffskontrolle**
- **Mikrosegmentierung**
- **Continuous Verification**
- **Least-Privilege-Prinzip**
- **Policy-as-Code** (OPA, Kyverno)
- **Integration in Kubernetes-Cluster**

Nachnutzung

Nachnutzung technisch umgesetzt 1/2

Bausteinbibliothek – FIM-Import – Styleguide

Nachnutzung ist der Hebel für echte Modernisierung.

Wir haben in Niedersachsen gezeigt, wie es geht:

- **automatisierter Import der FIM-Bausteine in die Low-Code-Plattform**
- **landesweiter UI-Styleguide**
- **Entwicklungsrichtlinien für Low-Code**
- **gemeinsame Fachverfahrenslandschaft statt Insellösungen**

Nachnutzung technisch umgesetzt 2/2

Bausteinbasierte Architektur:

- **Automatisierter Import von FIM-Bausteinen in Low-Code**
- **Versionierte Prozessmodelle** (BPMN 2.0)
- **Standardisierte Datenmodelle** (JSON-Schema, XÖV)
- **UI-Komponentenbibliothek als NPM-Package**
- **DevOps-Pipelines für Baustein-Rollouts**

