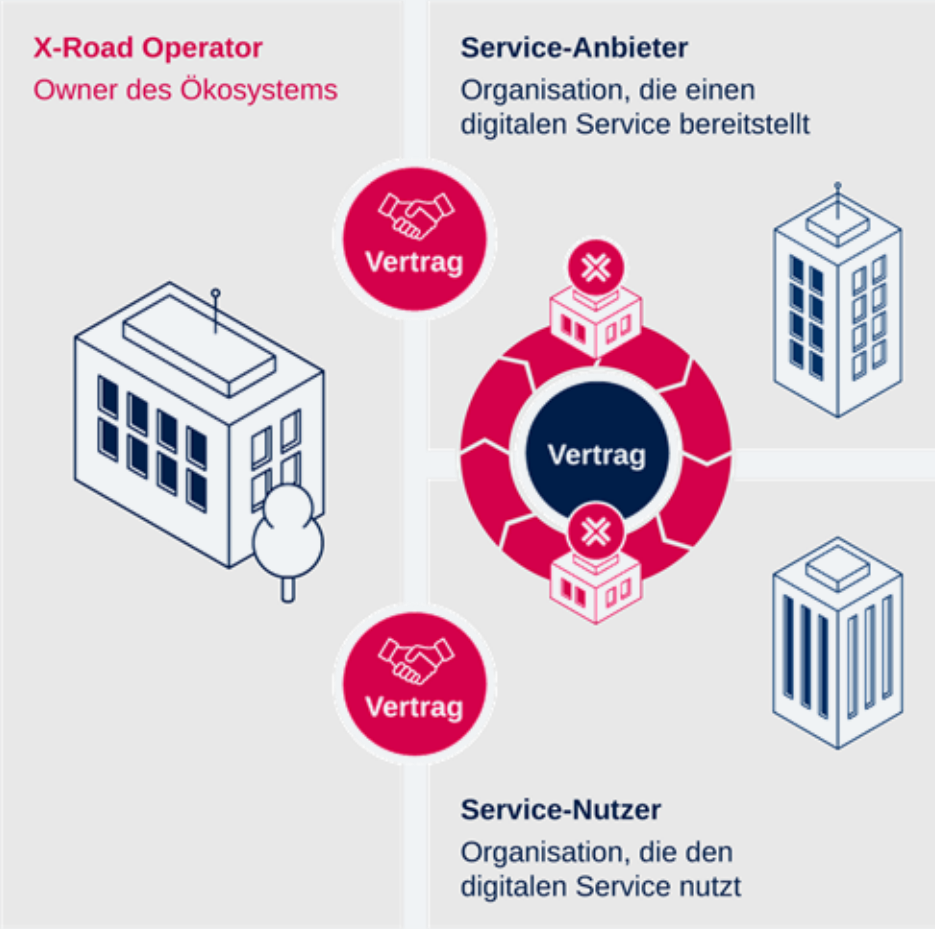
X-Road Architektur

Inhalte

1. Warum X-Road
2. Architektur
3. Komponenten und Konzepte
4. Sandbox
5. Nächste Schritte

Governance des Datenaustauschs



Warum Kommunen jetzt einen Security Server aufbauen sollten

Handlungsfähigkeit sichern: Ein eigener Security Server schafft die technische Voraussetzung, um künftig standardisiert, sicher und nachvollziehbar Daten mit Land, Kommunen und weiteren Partnern auszutauschen.

Anschlussfähigkeit frühzeitig herstellen: Kommunen, die jetzt einsteigen, können Anforderungen, Rollen und Prozesse aktiv mitgestalten – statt später unter Zeitdruck bestehende Strukturen nachziehen zu müssen.

Grundlage für digitale Verwaltungsleistungen schaffen: Der Security Server ermöglicht automatisierte, wiederverwendbare Datenaustauschprozesse und unterstützt damit Effizienz, Entbürokratisierung und bessere Services für Bürger*innen und Unternehmen.

Behörde A möchte Daten von Behörde B

Heute ohne X-Road

- Viele verschiedene Kommunikationsstandards wie z.B. OSCI/XTA, FIT-Connect, E-Mail, SFTP, …
- Keine einheitliche Identität, kein einheitliches Audit
- Jeder neue Partner = neues Projekt

Mit X-Road

- **Ein Ökosystem** für alle Partner
- **Gemeinsame PKI** (CA + Zertifikate)
- **Revisionssichere Logs** standardmäßig
- **Dezentrale Verantwortung** weiterhin möglich wenn gewünscht

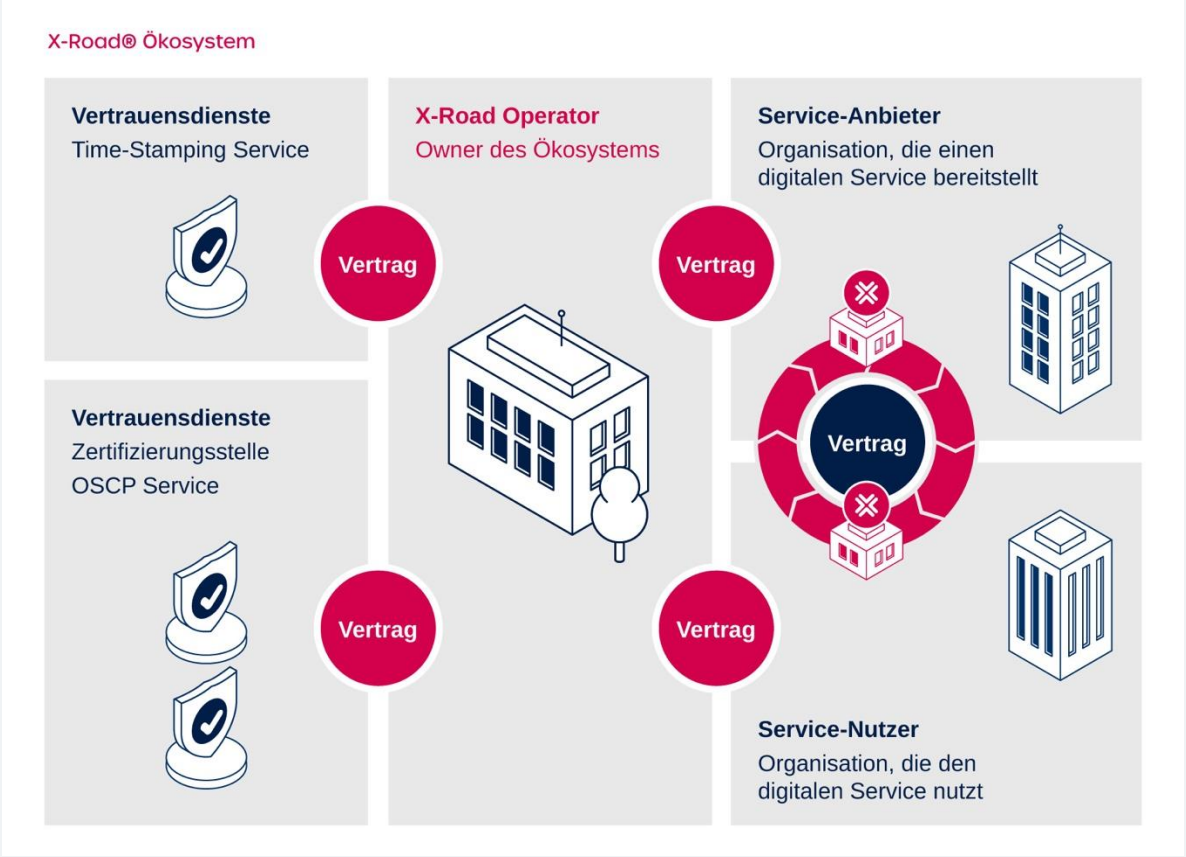
Was das alles genau bedeutet und wie es funktioniert, schauen wir uns gemeinsam an!

**nicht nur für Behörden*

Architektur

Aufbau der X-Road

- Vertrauensdienste werden durch X-Road Operator verfügbar gemacht
- **Mitglieder**
  - sind über Sicherheitsserver mit der X-Road verbunden
  - können als **Service Anbieter** verschiedene Dienste (REST/SOAP) anbieten und/oder als **Service Nutzer** die Dienste von anderen Mitgliedern aufrufen
  - bestimmen selbst mit welchen Mitgliedern sie welche Dienste austauschen
- Daten werden direkt zwischen den Sicherheitsservern ausgetauscht und sind verschlüsselt



Architektur & Rollen

Operator (Schleswig-Holstein)

- Betreibt Central Services, verwaltet Policies, Trust Services und Zugang für Mitglieder

Trust-Services

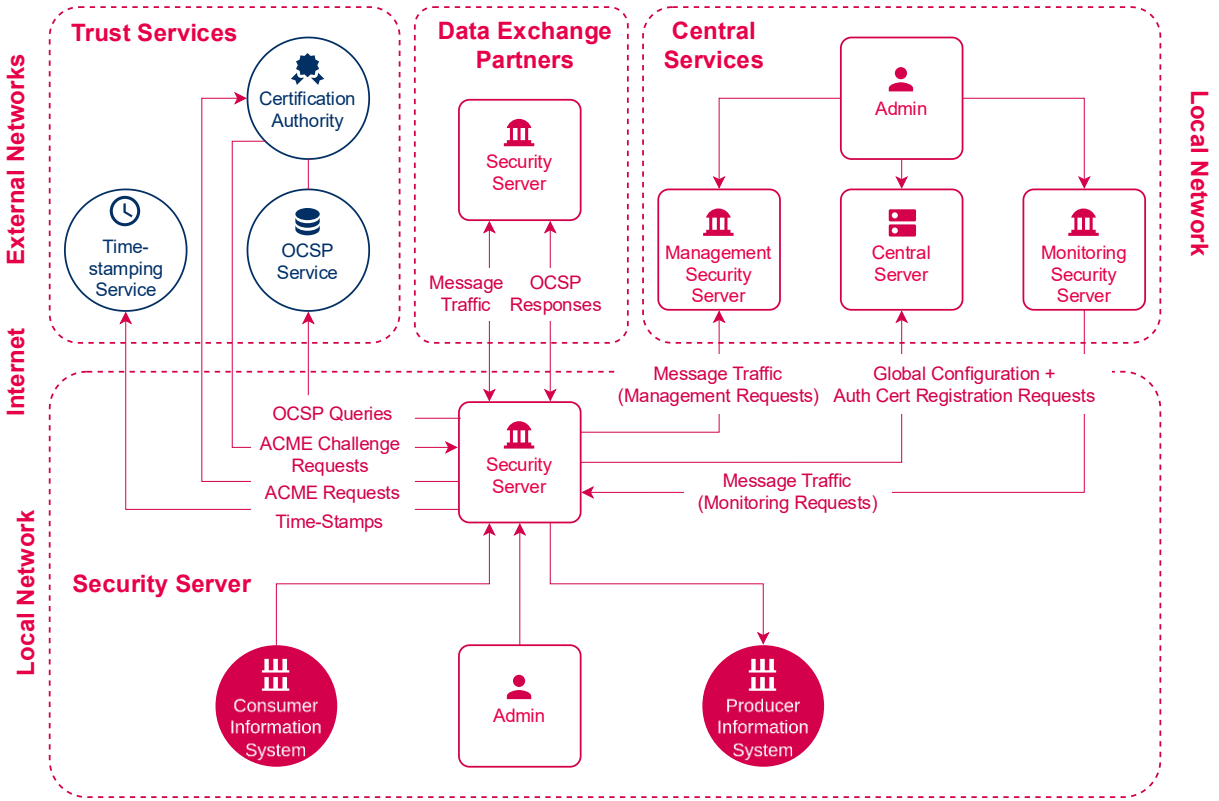
- CA+OCSP & TSA zur Absicherung der Kommunikation

Mitglieder

- Behörden, Kommunen, Unternehmen
- Security Server ist mandantenfähig

Information System

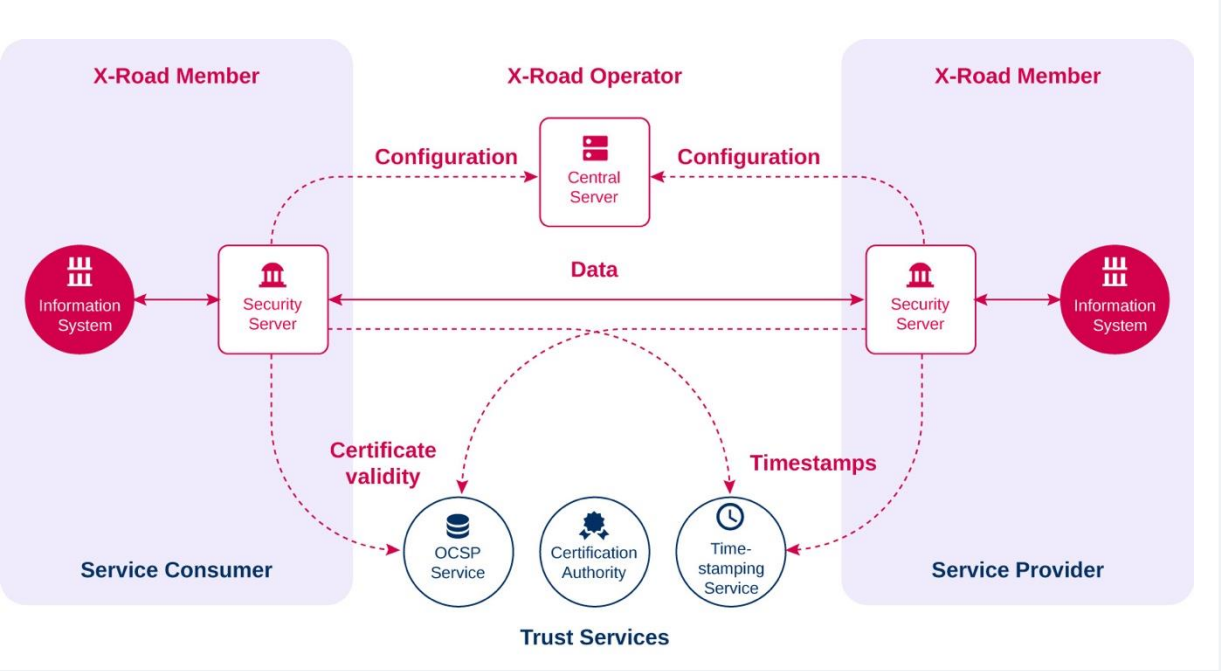
- Fachliche Anwendungen der Mitglieder (Bereitstellung oder Aufruf von REST/SOAP Schnittstellen)



Wie läuft eine Kommunikation ab?

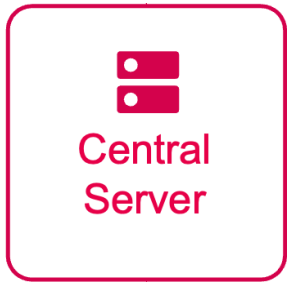
- Die Security Server fragen regelmäßig Konfiguration (enthält Mitglieder, Services, Endpunkte, …) vom Central Server ab
- Der Service Provider behält die **volle Kontrolle**, wer auf Daten und Services zugreifen darf (ACL)
- Transport findet **direkt** zwischen den **Security Servern** der Mitglieder statt
  - Nachrichten werden signiert und protokolliert
  - Verbindung ist per **mTLS** gesichert
- **Central Server** hat **keine aktive Rolle** im Datenaustausch

  → Ökosystem bleibt auch bei Störungen lang funktionsfähig



Komponenten und Konzepte

Central Server & Mitgliedschaft



ITV.SH

Der Central Server (CS) verwaltet:

- **GlobalConf:** beinhaltet Mitgliederverzeichnis, Trust Anchors, Konfiguration (Pull alle 60 s)
- **Member-Registry:** wer ist wer

Erster Anbindungs-Schritt:

Operator legt euch im CS als Member an. Dafür:

- **Member-Class:** GOV (Land), MUN (Kommunen), COM (Unternehmen)
- **Member-Code:** eindeutig je Member-Class

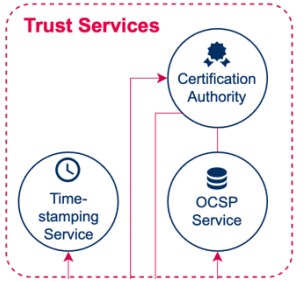
Member-ID-Format

`Instanz / Member-Class / Member-Code`

Beispiele

- `DE-SH/GOV/01-STK` (Staatskanzlei SH)
- `DE-SH/MUN/01002-KI` (Stadt Kiel)
- `DE-SH/COM/DE123456789` (Unternehmen)

Trust Services & Zertifikate



Drei Trust Services

1. **CA** (Certificate Authority)
   - stellt Zertifikate aus
2. **OCSP** (Online Certificate Status Protocol)
   - prüft, ob ein Zertifikat noch gültig ist
3. **TSA** (Time Stamping Authority)
   - liefert vertrauenswürdige Zeitstempel (Non-Repudiation)

Zwei Zertifikatstypen

Sign Cert

- gebunden an den **Member**
- Signiert jede einzelne Nachricht
- Sicherheit auf **Nachrichten-Ebene**

Auth Cert

- gebunden an den **Security Server**
- Authentifiziert den Security Server via mTLS gegenüber anderen Security Servern
- Sicherheit auf **Kanal-Ebene**

Security Server

Wie er ins Netz kommt

- **Configuration Anchor** = signierte „Geburtsurkunde" vom Operator
- **Soft Token** = PIN-geschützter Schlüsselspeicher (Sign + Auth Keys)

Was er tut – pro Nachricht

- Signiert (Sign Key)
- Baut mTLS (Auth Key)
- Lookup in GlobalConf
- Routet zum richtigen Ziel
- Schreibt revisionssicheren Log

Warum nicht direkt in der Fachanwendung?

Defense in Depth

Wir wollen nicht, dass jede Fachanwendung selbst Krypto, ACL und Audit macht. Der Security Server macht das zentral, einmal richtig.

Mandantenfähig

Ein Security Server kann mehrere Member hosten. Praktisch z.B. für kommunale Rechenzentren, die mehrere Kommunen bedienen.

Subsystem: Identität pro Fachverfahren

- **Member = Behörde** (juristischer Träger)
- **Subsystem = Anwendung** / Fachverfahren / Namespace

Volle X-Road-Identität

`Instanz/Member-Class/Member-Code/Subsystem`

Beispiel

`DE-SH/GOV/01-STK/MELDEWESEN`

Taucht später wieder im **X-Road-Client**-Header auf.

Warum die Trennung?

- ACLs greifen auf **Subsystem-Ebene** — ein Fachverfahren darf nicht automatisch, was ein anderes darf.

Beispiele

- `MELDEWESEN`
- `KFZ-REGISTER`
- `DEMO-CONSUMER` *(unsere Hands-on-Übung gleich)*

Services werden **immer am Subsystem** registriert, nie direkt am Member.

Service Description & Endpoints

So macht ihr einen Service bekannt

1. **OpenAPI-Spec** (REST) oder **WSDL** (SOAP) ins SS-Admin-UI hochladen
2. Security Server extrahiert die Endpunkte → jeder bekommt einen **Service Code** (z. B. isopenapi)
3. Mapping zur **Fachanwendung** in der Service Description (interner URL/Endpoint)

Internal Connection (Security Server ⇄ Fachanwendung)

- **HTTPS mit mTLS** — empfohlen, auch im internen Netz
- **HTTP** nur für ersten Verbindungstest

> **Wichtig** Die Fachanwendung selbst muss kein “X-Road“ können — sie spricht ganz normal REST/SOAP mit „ihrem" Security Server.

Access Control List & Local Groups

Granularität

- Pro **Service** × pro **konsumierendem Subsystem**.
- Beispiel: `STK/MELDEWESEN` darf `STADT-KIEL/MELDEDATEN/queryByName` aufrufen.

Local Groups

- Bündeln Konsumenten lokal beim Provider. Praktisch: „alle Konsumenten in Behörde X" oder „alle Pilot-Anwendungen"

Wer pflegt das?

- Der **Service-Provider** — volle Kontrolle bleibt bei dem, der die Daten bereitstellt.

Global Groups

- Gibt es auch — vom Operator zentral gepflegt (z. B. „alle SH-Behörden"). Selten genutzt, gut zu wissen dass es das gibt.

Merke ACL-Check passiert am Provider-Security Server, nachdem die Nachricht durchgekommen ist.

Message Logging & Auditierbarkeit

Was wird geloggt

Jede Nachricht wird **gehasht, signiert, mit Zeitstempel** abgelegt.
(Technisch: ein signierter ASiC-Container, prüfbar mit `asicverifier`.)

Zwei Modi

- **Full Logging** *(Default)* — Header + Body
- **Metadata only** — nur Header

Wo es eingestellt wird

- Admin-UI → Settings → System Parameters → Message Log
- Oder `/etc/xroad/conf.d/local.ini`, `[message-log]`, `message-body-logging=false`

Wo der Admin schaut

- Logs: `/var/log/xroad/` (VM) / Docker-Logs (Container)
- Admin-UI zeigt Status pro Zertifikat/Service

Empfehlung **Prod: metadata only.** Datensparsamkeit, DSGVO-Konformität.
Entscheidung trifft jede Mitgliedsorganisation selbst — wer revisionssicheren Inhaltsnachweis braucht, behält Full Logging.

Sandbox

Unsere Sandbox

Infrastruktur

- **Instanz:** DE-SH-TEST
- **Central Server:** cs.sandbox.x-road-sh.de
- **TestCA / OCSP:** ocsp.test.x-road-sh.de
- **Betrieb**: IONOS K8s, gemanaged durch Nortal

Demodienst (für Hands-on)

- **Member:** SH
- **Subsystem:** DEMO-PROVIDER
- **Service-Code:** isopenapi
- **Aktive Member:** *live in der Demo*

Endpunkt-Beispiel

```
GET /r1/DE-SH-TEST/GOV/SH/DEMO-PROVIDER/isopenapi/api/members
```

Sandbox vs. Produktionsumgebung

Anders in der Sandbox

- **TestCA** statt echter CA
- **MD5-Hash-Member-Codes** statt ARS / W-IdNr
- **Softtoken** Pflicht (kein HSM)
- **Keine echten personenbezogenen Daten** erlaubt
- **Full Logging ok** (keine echten Daten)

Schon „echt"

- ✓ X-Road-Protokoll identisch
- ✓ Signaturen, mTLS, Logs identisch
- ✓ SS-Software identisch (gleicher NIIS-Build)
- ✓ Onboarding-Schritte identisch

Übergang nach Prod

→ **Organisatorisch:** Beitrittsprozess (Nutzungsbedingungen, Vertretungsnachweis)

→ **Technisch:** Schutzbedarf-Frage, neuer Member-Code

Unsere Sandbox

Wenn Sie die Sandbox ausprobieren wollen,

melden Sie sich unter xroad.sh@nortal.com

bei uns – wir helfen Ihnen gerne weiter!

Vielen Dank für Ihre Aufmerksamkeit!

Hier finden Sie weitere Informationen:

https://x-road-sh.de

xroad.sh@nortal.com

https://x-road-sh.de/newsletter