The Technical Architecture of X-Road: A Comprehensive Analysis of Decentralized Data Exchange, Cryptographic Assurance, and eIDAS Compliance

1. Introduction: The Interoperability Paradox and the distributed Paradigm

In the modern digital era, governments and large-scale enterprises face a critical architectural paradox: the necessity of seamless data exchange versus the imperative of strict security and data sovereignty. The traditional approaches to this problem have historically bifurcated into two flawed models. The first is the centralized data warehouse, or the "Super Database," where all citizen or business data is aggregated into a single repository. While this simplifies access, it creates a massive single point of failure—a "data honeypot" attractive to cyber adversaries and prone to catastrophic privacy breaches. The second model is the ad-hoc integration approach, characterized by a spaghetti-like mesh of point-to-point API connections, each with inconsistent security standards, varying authentication protocols, and high maintenance costs.

X-Road, originally developed by the Estonian Information System Authority (RIA) and now governed by the Nordic Institute for Interoperability Solutions (NIIS), represents a third, mature architectural paradigm: the centrally managed distributed data exchange layer. This report provides an exhaustive technical analysis of the X-Road architecture, specifically dissecting its mechanisms for preventing centralized data accumulation while ensuring high-assurance security. The analysis focuses on the "Security Server"—the architectural linchpin of X-Road—and documents the specific encryption primitives (AES/RSA), transport protocols, and trust mechanisms that render the system secure. Furthermore, it demonstrates how X-Road's specific implementation of the "Once-Only Principle" (TOOP) and its alignment with eIDAS regulation validate its scalability and robustness, effectively debunking the "Utopia" claim that such systems are only viable in small, highly digitalized nations.

This document serves as a definitive reference for technical architects, policy makers, and security auditors, elucidating why X-Road has become the *de facto* standard for high-security government registries across the European Union and beyond.

2. Architectural Principles: The Philosophy of Decentralization

To understand the security posture of X-Road, one must first deconstruct its topology. Unlike Enterprise Service Buses (ESBs) that route payloads through a central hub, X-Road employs a peer-to-peer (P2P) mesh topology for the data plane, while retaining a centralized control plane for identity and trust management.

2.1 The Four-Corner Model

X-Road utilizes a "Four-Corner Model" of data exchange, a standard topology in secure electronic delivery services. This model ensures that the data payload never passes through a central broker, thereby eliminating the possibility of a central operator intercepting or harvesting data.

Corner	Component	Function & Responsibility
Corner 1	**Service Consumer IS**	The legacy information system or application requesting data (e.g., a hospital portal). It communicates only with its local Security Server (Corner 2).
Corner 2	**Consumer Security Server**	The entry point to the X-Road ecosystem. It encapsulates the request, applies digital signatures, encrypts the payload, and routes it to the provider.
Corner 3	**Provider Security Server**	The receiving gateway. It decrypts the request, validates the signature and certificate status (OCSP), logs the transaction, and forwards it to the backend.
Corner 4	**Service Provider IS**	The backend database or registry (e.g., Population Register) that executes the query and returns the response.

The interaction between Corner 2 and Corner 3 occurs over the public internet but is secured via a mutually authenticated encrypted tunnel. The Central Server (not a corner, but the ecosystem coordinator) is entirely bypassed during the data exchange phase. This separation of concerns is critical: the Central Server knows *who* is in the network, but it never sees *what* they are saying.

2.2 The Absence of a Central Broker: Avoiding the Honeypot

The concept of a "Data Honeypot" refers to a centralized repository where data from multiple sources is aggregated. This is a high-value target for attackers because a single breach yields a total compromise of the ecosystem. X-Road avoids this through strict architectural constraints:

1. **Data Sovereignty:** Data stays at the source. The Population Registry holds population data; the Tax Board holds tax data. X-Road merely facilitates the secure "viewing" or movement of this data upon request. It does not cache or store the data centrally.
2. **Distributed Enforcement:** Security policies are not enforced by a central firewall but by

the Security Servers at the edge. Each organization hosting a Security Server retains sovereignty over its access control lists (ACLs), ensuring that a compromise of the central infrastructure does not grant access to the edge nodes.

3. **Resilience to Central Failure:** The ecosystem is designed for failure. If the Central Server goes offline, the Security Servers continue to operate using cached global configuration data. This ensures that critical infrastructure services (e.g., emergency services querying medical records) function even during a central outage.

This decentralized architecture is the primary technical rebuttal to the "Utopia" criticism. Critics often argue that decentralized systems are unmanageable. X-Road solves the management issue through the Central Server's distribution of the "Global Configuration"—a signed XML file containing the network's DNS information, trusted Certificate Authorities (CAs), and member lists—which is periodically downloaded by all Security Servers. Thus, the system achieves the management simplicity of a centralized system with the security benefits of a decentralized one.

3. The Security Server: Anatomy of the Guardian Node

The Security Server is the most complex and critical component of the X-Road infrastructure. It acts as a standardized, hardened gateway that abstracts the complexity of cryptography, PKI, and network routing from the application layer.

3.1 Internal Architecture and Component Interaction

The Security Server is not a monolithic application but a collection of modular services, typically running on a hardened Linux operating system (Ubuntu or RHEL).

3.1.1 The Proxy Module (xroad-proxy)

The Proxy is the high-performance engine responsible for mediating requests. It operates in two modes:

- **Client Proxy:** Listens for requests from the local Information System (Corner 1), wraps them in the X-Road Message Transport Protocol, and initiates the connection to the remote Security Server.
- **Server Proxy:** Listens for incoming connections from the internet (Corner 2), unwraps the protocol, verifies security, and forwards the request to the local backend (Corner 4).

The Proxy ensures that the internal Information System does not need to handle complex SOAP/REST wrapping or mutual TLS logic. The internal connection is typically plain HTTP or one-way TLS, while the external connection is always highly secure.

3.1.2 The Signer Module (xroad-signer)

The Signer is the cryptographic heart of the Security Server. It is responsible for managing the keys used for signing messages and authentication.

- **Key Isolation:** The Signer ensures that private keys are never exposed to the application layer.
- **Token Interface:** It interacts with the cryptographic tokens, which can be software-based (Soft Token) or hardware-based (HSM). The Signer handles the PKCS#11 interface calls, abstracting the specific hardware details from the Proxy.

3.1.3 The Configuration Client (xroad-confclient)

This background daemon is responsible for maintaining the trust relationship with the ecosystem. It periodically downloads the Global Configuration from the Central Server.

- **Verification:** It verifies the digital signature of the Central Server on the configuration files.
- **Caching:** It updates the local cache, ensuring that the Security Server always has an up-to-date list of valid members and trusted certification authorities. This caching mechanism is what allows the Security Server to function autonomously during central outages.

3.2 Hardware Security Modules (HSM) vs. Soft Tokens

A critical design choice in X-Road is the support for both software and hardware cryptographic tokens. This flexibility addresses the cost barrier often cited by critics.

- **Soft Token:** For development, testing, or lower-security environments, keys are stored in an encrypted file on the server's filesystem. This reduces the cost of adoption to near zero (beyond server hardware).
- **HSM Support:** For high-security environments (e.g., National Population Registers, Banks), X-Road supports Hardware Security Modules via the PKCS#11 standard. Using an HSM ensures that the private signing keys cannot be extracted even if the physical server is stolen or the root OS is compromised. This capability is essential for achieving the level of "Qualified Electronic Seal" under eIDAS.

3.3 The Adapter Server Pattern

To integrate legacy systems (the "brownfield" reality of government IT), X-Road utilizes the Adapter Server pattern. Legacy mainframes or proprietary databases often cannot speak modern REST or SOAP protocols. An Adapter Server sits between the Security Server and the Legacy System, translating the X-Road standard messages into the native protocol of the legacy system (e.g., converting a SOAP request into a SQL query or a proprietary flat-file format). This decoupling allows nations to modernize their interoperability layer without immediately rewriting decades of legacy code, a crucial factor in the system's successful adoption in countries like Finland and Germany.

4. Cryptographic Primitives and Protocol Specifications

The claim that X-Road prevents "honeypots" and ensures security is mathematically grounded in its use of specific encryption and signing protocols. The architecture employs a defense-in-depth strategy, using multiple layers of cryptography.

4.1 Transport Layer Security (TLS)

Data in transit between Security Servers is secured using **Mutual TLS (mTLS)**. Unlike standard HTTPS where only the server is authenticated, X-Road requires both the client (Consumer

Security Server) and the server (Provider Security Server) to present valid authentication certificates issued by the ecosystem's internal Certificate Authority (CA).

- **Protocol Version:** The system enforces **TLS 1.2** as the minimum standard, with support for **TLS 1.3** in modern deployments.
- **Cipher Suites:** X-Road configuration explicitly prioritizes cipher suites that offer Perfect Forward Secrecy (PFS) and Authenticated Encryption (AEAD). The preferred suites include:
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384.

Technical Insight: The choice of **ECDHE** (Elliptic Curve Diffie-Hellman Ephemeral) is deliberate. It ensures that even if a Security Server's long-term private key is compromised in the future, past sessions recorded by an attacker cannot be decrypted. The use of **AES-GCM** (Galois/Counter Mode) provides high-performance encryption with built-in integrity checking, preventing padding oracle attacks that affect older CBC modes.

4.2 Application Layer: The Double Encryption Model

A common misconception is that TLS is the only layer of protection. X-Road employs a "double envelope" approach. Inside the TLS tunnel, the message itself is digitally signed and, in specific configurations, can be encrypted at the application layer.

4.2.1 The Message Transport Protocol

The wire protocol exchanged between Security Servers is a MIME Multipart structure.

1. **Header:** Contains the routing metadata (Client ID, Service ID) and the hash algorithm identifier (e.g., x-hash-algorithm: SHA-512).
2. **Payload:** The actual SOAP or REST message.
3. **Signature:** A detached digital signature calculated over the headers and the payload.

This application-layer signature ensures **end-to-end integrity**. Even if the TLS termination point (e.g., a load balancer) is compromised, the attacker cannot forge a request because they do not possess the private signing key of the originating organization.

4.2.2 Message Body Encryption (XML-ENC)

While TLS protects the pipe, X-Road allows for the encryption of the payload itself using **XML-Encryption** (for SOAP) or JWE (for REST). This ensures that intermediate nodes (if any exists in complex routing scenarios) cannot view the payload. However, in the standard P2P model, the TLS tunnel is the primary confidentiality mechanism.

4.3 Data at Rest Encryption Protocols

The "Honeypot" risk is further mitigated by how data is stored on the Security Server itself.

- **Message Logs:** By default, logs are stored in the PostgreSQL database. To prevent this log from becoming a local honeypot, X-Road supports **Database Encryption**.
  - **Algorithm:** The system uses **AES-CTR** (Counter Mode) with a 256-bit key for encrypting the message body within the database rows.
  - **Key Management:** The encryption keys are managed by the Security Server's

internal keystore and can be rotated.

- **Backups:** Configuration backups, which contain sensitive routing and key information, are encrypted using **GnuPG** (OpenPGP standard). This uses asymmetric encryption (RSA), ensuring that backups can only be restored by an administrator possessing the corresponding private key.

4.4 Cryptographic Agility

X-Road utilizes **RSA** keys (typically 2048-bit or 4096-bit) for signing and authentication certificates. However, the architecture is designed for cryptographic agility. The system supports **Elliptic Curve Cryptography (ECC)** keys, which offer equivalent security to RSA at much shorter key lengths, reducing computational overhead on the Security Servers—a critical factor for scalability in high-volume environments like Brazil or India.

5. The "Honeypot" Defense: Logging, Auditing, and Data Sovereignty

The primary vector for the "Data Honeypot" vulnerability is the centralized aggregation of transaction logs. If a central authority logs the *content* of every message, they effectively rebuild the distributed databases into a central one. X-Road prevents this through its distributed logging architecture.

5.1 Distributed Message Logs

In X-Road, the Central Server **does not log message payloads**. It does not even see them. The evidence of the transaction is stored *only* on the Security Servers of the parties involved (the Consumer and the Provider).

- **Provider Log:** Proves that the query was answered and what data was returned.
- **Consumer Log:** Proves that the query was made and the response received.

This distribution ensures that an attacker compromising the central infrastructure gains no intelligence on the citizens' data interactions. To reconstruct a citizen's full profile, an attacker would have to compromise every single Security Server in the nation simultaneously—a task orders of magnitude harder than breaching a single data warehouse.

5.2 Cryptographic Chaining (The Blockchain Precursor)

To prevent administrators from tampering with their local logs (e.g., deleting a record of an illegal data access), X-Road employs **Merkle Hash Trees** (Linking).

- **Linking:** Each log entry contains the hash of the previous entry. This creates an immutable chain.
- **Batch Timestamping:** Periodically, the hash of the chain's head is sent to a trusted Time-Stamping Authority (TSA). The TSA returns a signed timestamp.
- **Verification:** This process "anchors" the log in time. Any modification to a past log entry would break the hash chain and fail verification against the trusted timestamp. This provides a mathematically provable audit trail, essential for government transparency and legal accountability.

6. Regulatory Engineering: eIDAS, Trust Services, and Legal Compliance

The European Union's eIDAS Regulation (EU No 910/2014) establishes the legal framework for electronic identification and trust services. X-Road is engineered not just for technical security but for legal admissibility.

6.1 Electronic Seals (eSeal) vs. Electronic Signatures

X-Road operations are automated; they are not signed by human users but by organizations. Under eIDAS, this requires **Electronic Seals**.

- **Advanced Electronic Seal (AdES):** The baseline X-Road configuration provides AdES. It uniquely links the data to the signatory (the organization) and ensures integrity.
- **Qualified Electronic Seal (QES):** For high-value transactions (e.g., court judgments, financial transfers), X-Road can produce QES. This requires the use of a **Qualified Certificate** issued by a Qualified Trust Service Provider (QTSP) and the storage of keys in a **Qualified Signature Creation Device (QSCD)**—typically a certified HSM.

6.2 Trust Federation and Cross-Border Validity

X-Road's architecture facilitates cross-border data exchange through **Trust Federation**.

- **Mechanism:** Two X-Road ecosystems (e.g., Estonia and Finland) exchange "Configuration Anchors."
- **Legal Mapping:** The Federation agreement maps the Trust Services of one country to the other. A digital signature created by a Finnish authority is automatically validated and trusted by the Estonian system because the root CAs are cross-recognized in the configuration.
- **ETSI Alignment:** X-Road relies on ETSI standards (EN 319 412 for certificate profiles) to ensure technical interoperability. This adherence to standards means that an X-Road signature is not a proprietary blob but a standard **ASiC-E** container recognized by EU courts.

6.3 TOOP and the Once-Only Principle

The "Once-Only Principle" (OOP) mandates that citizens should not have to provide the same data to the government twice. X-Road is the technical backbone for implementing OOP across borders, as demonstrated in the **TOOP (The Once-Only Principle Project)** and **DE4A (Digital Europe for All)** pilots.

- **Federated Architecture:** TOOP utilizes the X-Road infrastructure to query data registries in other countries directly.
- **Semantic Interoperability:** While X-Road handles the *transport* (security, routing), projects like TOOP and DE4A build the *semantic* layer (data mapping) on top of it. X-Road's neutrality regarding payload (it carries XML or JSON indiscriminately) makes it the ideal carrier for these high-level regulatory frameworks.

7. Global Adoption and the "Utopia" Refutation

A common critique of the "Estonian Model" is that it is a "Utopia"—a system that works in a small, digitally native country but fails in complex, large, or federalized nations. The global adoption of X-Road refutes this claim through empirical evidence of its adaptability.

7.1 Finland: Overcoming Legacy Debt

Finland, a country with a much larger and older IT infrastructure than Estonia, adopted X-Road (branded as *Palveluväylä*) as its national data exchange layer.

- **Challenge:** Deeply entrenched legacy systems and silos.
- **Solution:** The Adapter Server pattern allowed Finland to connect pre-existing mainframes without rewriting them.
- **Result:** Finland and Estonia now share the world's first operational cross-border data exchange, linking population registries and tax boards for real-time data sharing.

7.2 Germany: Federalism and Healthcare

Germany represents the ultimate test of X-Road: a highly federalized system with strict data privacy laws and fragmented healthcare IT.

- **Implementation:** X-Road is being piloted for the **Digital Prescription** system and administrative digitization.
- **Analysis:** The decentralized nature of X-Road aligns perfectly with the German federal model (Länder sovereignty). Unlike a central database, which would violate German principles of data separation, X-Road allows each state or health provider to maintain its data silo while allowing secure, auditable access—proving the system's viability in complex political landscapes.

7.3 Brazil and Mexico: Scaling to Millions

In Latin America, X-Road has been adopted to handle massive scale.

- **Brazil:** Known as *X-Via*, it is used by state governments (e.g., Mato Grosso, Piauí) to interconnect police, tax, and judiciary systems.
- **Mexico:** Implemented in states like Quintana Roo for population management.
- **Insight:** These implementations debunk the scalability limit argument. X-Road's P2P architecture scales horizontally; adding more members does not choke a central hub because the traffic does not pass through a central hub.

7.4 The "Utopia" Fallacy

The "Utopia" argument rests on the assumption that centralized control is necessary for order. X-Road demonstrates that **cryptographic order** is superior to administrative control. By enforcing rules through code (the Global Configuration) rather than bureaucracy, X-Road allows diverse, antagonistic, or legacy-heavy organizations to collaborate without surrendering autonomy.

8. The Cost of Trust: Economic and Operational Realities

While X-Road software is open-source (MIT License) and free, the operational cost of "High Assurance" is not zero. A nuanced analysis must account for the infrastructure of trust.

8.1 The HSM Barrier

Implementing a fully eIDAS-compliant Qualified Electronic Seal requires Hardware Security Modules (HSMs).

- **Cost:** Enterprise-grade HSMs are expensive (thousands of Euros) and require specialized skills to manage.
- **Mitigation:** X-Road mitigates this via the **Security Server Sidecar** (Docker) and CloudHSM support (AWS CloudHSM). This allows smaller agencies to rent HSM capacity rather than buying hardware. Furthermore, for lower-assurance use cases, Soft Tokens (software keys) are permitted, allowing a tiered security model.

8.2 Onboarding and Governance

The technical deployment of a Security Server takes minutes (using Ansible or Docker), but the *organizational* onboarding takes time.

- **Registration:** Agencies must apply for certificates, pass validation checks, and configure their firewalls.
- **Governance:** The success of X-Road depends on a strong "X-Road Operator" (like NIIS) to manage the central CA and definition of standards. Without strong governance, the technical layer functions, but the semantic layer (understanding the data) fails.

9. The Future Landscape: X-Road 8, Gaia-X, and Data Spaces

The architecture of X-Road is currently undergoing its most significant evolution to align with the European Strategy for Data.

9.1 X-Road 8: The "Spaceship"

The next generation, X-Road 8 (codenamed "Spaceship"), is moving away from the "Server" concept toward a "Service" concept.

- **Cloud-Native:** Full support for Kubernetes and containerized deployments.
- **Statelessness:** Decoupling the configuration state from the runtime, allowing for auto-scaling Security Server clusters in the cloud.

9.2 Gaia-X and Data Spaces

X-Road is integrating with the **Eclipse Dataspace Components (EDC)** to become a protocol for **Gaia-X**.

- **Shift in Scope:** While X-Road historically focused on G2G (Government to Government), the integration with Data Spaces extends it to B2G (Business to Government) and B2B sectors.

- **Self-Sovereign Identity (SSI):** Future iterations are exploring the use of DIDs (Decentralized Identifiers) and Verifiable Credentials to supplement the traditional X.509 PKI, offering even more granular privacy controls.

10. Conclusion

The technical analysis of X-Road reveals a system that is fundamentally resistant to the vulnerabilities of centralization. It prevents the "Data Honeypot" not through policy, but through **topology** (the Four-Corner Model), **cryptography** (End-to-End Encryption and Signing), and **distributed ledger technology** (Merkle-chained logs).

By adhering to rigorous encryption standards—**AES-256-GCM** for confidentiality, **RSA-4096** for integrity, and **SHA-512** for hashing—X-Road provides a security posture that exceeds standard commercial APIs. Its alignment with **eIDAS** and **ETSI** standards ensures that this security translates into legal certainty, making it the ideal backbone for the European Digital Single Market.

The "Utopia" claim is debunked not by theory, but by the operational reality of Finland, Germany, and Brazil. X-Road proves that a government interoperability platform can be secure, decentralized, and scalable simultaneously, provided it is built on the bedrock of cryptographic trust rather than centralized control.

Summary of Key Encryption & Protocol Specifications

Component	Standard / Protocol	Specific Implementation Details	Reference
Transport	**TLS 1.2 / 1.3**	Mutual Authentication (mTLS). Cipher Suites: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.	
Confidentiality	**AES**	**AES-256-GCM** (Galois Counter Mode) for transit; **AES-CTR** for Data at Rest (Logs).	
Integrity	**RSA / ECC**	**RSA-4096** or ECC P-384 for Digital Signatures. **SHA-512** for hashing.	
Signature Format	**XAdES / ASiC**	**ASiC-E** (Associated Signature Container) for long-term archiving.	
Log Integrity	**Merkle Tree**	Chained hashes anchored by a Trusted Timestamp (RFC 3161).	
Backup Security	**OpenPGP**	**GnuPG** encryption	

Component	Standard / Protocol	Specific Implementation Details	Reference
		using RSA asymmetric keys.	

Quellenangaben

1. X-Road - Wikipedia, https://en.wikipedia.org/wiki/X-Road 2. X-Road® Architecture, https://x-road.global/architecture 3. The Once-Only Principle Project (TOOP) | ISA² - European Commission, https://ec.europa.eu/isa2/isa2conf18/once-only-principle-project-toop_en/ 4. The "Once-Only" Principle - University for Continuing Education Krems, https://www.donau-uni.ac.at/en/research/project/U7_PROJEKT_4294969233 5. Case Study Details - GovTech Intelligence Hub - Smarter Data Exchange: How X-Road Became a Model for Digital Public Infrastructure, https://www.govtechintelhub.org/case-study-details/smarter-data-exchange:-how-x-road-became-a-model-for-digital-public-infrastructure/aJYTG0000000o014AA 6. Estonia's digital diplomacy: Nordic interoperability and the challenges of cross-border e-governance | Internet Policy Review, https://policyreview.info/articles/analysis/estonias-digital-diplomacy-nordic-interoperability 7. X-Road® — Data Exchange, https://x-road.global/data-exchange 8. X-Road® Technology Overview, https://x-road.global/x-road-technology-overview 9. X-Road Security Architecture, https://docs.x-road.global/Architecture/arc-sec_x_road_security_architecture.html 10. Changes in the X-Road Central Server High Availability Support, https://www.niis.org/blog/2020/1/23/changes-in-the-central-server-high-availability-support-in-version-6230 11. X-Road/doc/Protocols/pr-gconf_x-road_protocol_for_downloading_configuration.md at develop - GitHub, https://github.com/nordic-institute/X-Road/blob/develop/doc/Protocols/pr-gconf_x-road_protocol_for_downloading_configuration.md 12. X-Road® History, https://x-road.global/xroad-history 13. X-Road: Security Server Architecture, https://docs.x-road.global/Architecture/arc-ss_x-road_security_server_architecture.html 14. SECURITY SERVER USER GUIDE | X-Road, https://docs.x-road.global/Manuals/ug-ss_x-road_6_security_server_user_guide.html 15. X-Road/doc/Architecture/arc-cp_x-road_configuration_proxy_architecture.md at develop, https://github.com/ria-ee/X-Road/blob/develop/doc/Architecture/arc-cp_x-road_configuration_proxy_architecture.md 16. Secure Data Exchange Layer - Red Gealc, https://www.redgealc.org/site/assets/files/2205/secure_data_ecxhange_layer_f_mex_25_05_17.pdf 17. Finland — X-Road® — X-Road Case Studies Library, https://x-road.global/xroad-case-studies-library/tag/Finland 18. Piloting digital prescriptions in Germany through secure data exchange - X-Road®, https://x-road.global/xroad-case-studies-library/2024/10/21/piloting-digital-prescriptions-in-germany-through-secure-data-exchange 19. By default, client proxy supports TLS 1.2 and cipher suites listed below when communicating with an information system - X-Road Knowledge Base - NIIS Confluence, https://nordic-institute.atlassian.net/wiki/spaces/XRDKB/pages/4916142 20. RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3 - IETF Datatracker, https://datatracker.ietf.org/doc/html/rfc8446 21. X-Road: Message Transport Protocol, https://docs.x-road.global/Protocols/pr-messtransp_x-road_message_transport_protocol.html 22. X-Road REST Support – Where Are We Today? - Nordic Institute for Interoperability

Solutions, https://www.niis.org/blog/2018/10/3/x-road-rest-support-where-are-we-today 23. XML-Encryption Settings - Oracle Help Center, https://docs.oracle.com/cd/E39820_01/doc.11121/gateway_docs/content/encryption_encrypt_settings.html 24. Protecting Data at Rest in X-Road 7 - Nordic Institute for Interoperability Solutions, https://www.niis.org/blog/2021/10/2/protecting-data-at-rest-in-x-road-7 25. X-Road/doc/Manuals/ig-ss_x-road_v6_security_server_installation_guide.md at develop - GitHub, https://github.com/nordic-institute/X-Road/blob/develop/doc/Manuals/ig-ss_x-road_v6_security_server_installation_guide.md 26. X-Road Case Studies Library, https://x-road.global/xroad-case-studies-library 27. High-Performance Qualified Digital Signatures for X-Road - Cybernetica - Research Library, https://research.cyber.ee/~janwil/publ/batchsignatures.pdf 28. eSignature FAQ - European Commission, https://ec.europa.eu/digital-building-blocks/sites/spaces/DIGITAL/pages/880312429/eSignature+FAQ 29. X-Road Trust Federation for Cross-border Data Exchange - Observatory of Public Sector Innovation, https://oecd-opsi.org/innovations/x-road-trust-federation-for-cross-border-data-exchange/ 30. ETSI EN 319 412-1 V1.6.1 (2025-06), https://www.etsi.org/deliver/etsi_en/319400_319499/31941201/01.06.01_60/en_31941201v010601p.pdf 31. Signed Document Download and Verification Manual | X-Road, https://docs.x-road.global/Manuals/ug-sigdoc_x-road_signed_document_download_and_verification_manual.html 32. Digital tools to support the free movement of data in the EU | DE4A Project - CORDIS, https://cordis.europa.eu/article/id/442344-digital-tools-to-support-the-free-movement-of-data-in-the-eu 33. A HISTORICAL ANALYSIS ON INTEROPERABILITY IN ESTONIAN DATA EXCHANGE ARCHITECTURE: PERSPECTIVES FROM THE PAST AND FOR THE FUTURE - AWS, https://x-road-document-library.s3.amazonaws.com/attachments/A_historical_analysis_on_innteroperability_in_Estonian_data_exchange_architecture.pdf 34. Estonia and Finland launch automated data exchange between population registers, https://x-road.global/xroad-case-studies-library/2024/10/21/estonia-and-finland-launch-automated-data-exchange-between-population-registers 35. A pioneer in digital administration - what Germany can learn from Estonia, https://www.smartcountry.berlin/en/newsblog/newsblog-details/a-pioneer-in-digital-administration-what-germany-can-learn-from-estonia.html 36. Scaling interoperability across levels of governance and states in Brazil - X-Road® — X-Road Case Studies Library, https://x-road.global/xroad-case-studies-library/2024/10/21/scaling-interoperability-across-states-for-national-digital-transformation-in-brazil 37. From one domain to the whole government – X-Road takes its first steps in two Mexican states, https://x-road.global/xroad-case-studies-library/2024/10/21/from-one-domain-to-the-whole-government-x-road-takes-its-first-steps-in-two-mexican-states 38. Security Server Cost Calculator - Squarespace, https://static1.squarespace.com/static/5a4f79d6aeb625d6f842c5d5/t/60a758c6c996a914d300411d/1621579975018/x-road_security_server_cost_calculator%2B20210521.xlsx 39. How Much Does A Hardware Security Module Cost? - SecurityFirstCorp.com - YouTube, https://www.youtube.com/watch?v=hCoAIyfygCk 40. Implementation Models — X-Road® — X-Road Case Studies Library, https://x-road.global/xroad-case-studies-library/tag/Implementation+Models 41. Making of

X-Road 8 - June 2025 Status Update, https://www.niis.org/blog/2025/6/11/making-of-x-road-8-june-2025-status-update 42. X-Road® - Gaia-X, https://gaia-x.eu/wp-content/uploads/2024/11/X-RoadNIIS.pdf