Deep Science Publishing, 2026
https://doi.org/10.70593/978-81-69589-32-1_2

DeepScience
Open Access Books

Chapter 2: Designing Trusted Data Platforms for Enterprise Infrastructure

2.1. Introduction

To many stakeholders, enterprise infrastructure means core services that are trusted, high performance, cost-effective, and resilient. Trusted Data Platforms support enterprise infrastructure by enforcing security, privacy, governance, and business continuity requirements. To consider such platforms truly trusted, the formal definition of trust must therefore be examined. It is commonly stated that trust must be earned. However, simply having the trust of stakeholders is not sufficient to communicate that data platforms are indeed capable of providing the required levels of confidentiality, integrity, availability, and resilience. Stakeholders must also be able to judge both the level of risk associated with the platform and whether the measures in place are appropriate when compared against the level of trust being afforded. A trusted data platform is able to demonstrate that its level of risk is acceptable to all relevant stakeholders and that it has taken adequate measures to protect the data it processes.

Trust can be considered as an attribute assigned to an entity in a relationship by a judge, where that attribute has been earned through a series of actions performed by that entity. While the judge deciding whether to trust may, for example, be a colleague deciding whether to share sensitive information with another or a more abstract stakeholder, the entity in the relationship is frequently an organisation that is a custodian of data. In more general terms, trust is the expectation that the entity, on any future interaction, will act in accordance with a declared set of principles, thereby establishing an integrity profile for itself. For enterprise infrastructure, these principles usually span the areas of security, privacy, governance, and business continuity. When any of these principles fail, trust is accordingly broken, sometimes with devastating or catastrophic consequences.

2.1.1. Security and Privacy by Design

Security and Privacy by Design advocates for built-in protection of the assets entrusted to a technology solution. Core architectural patterns, such as layered and defence-in-depth, facilitate the application of protective controls. Conducting a threat modelling exercise identifies vulnerabilities and outlines adequate protection measures. Safekeeping of assets also requires ongoing vigilance, which is achieved through assessments of risk, risk appetite, and the application of appropriate risk mitigation measures.

Security and Privacy by Design are used to embed appropriate protective controls into an enterprise's trusted data platform, similar to those that would fortify any solution using sensitive data. Assets assigned to the various forms of protective controls—that provide defence against different forms of cyber-attack—are continuously monitored and adjusted according to changing threats, security requirements, and risk appetite, updated with current threat intelligence, and promptly deployed to minimize exposure to security incidents and breaches. These threat-control layers should include such security mechanisms as perimeter, entry point, application security, monitoring, identity-management, and database-security tools.

Fig 2.1: Designing Trusted Data Platforms for Enterprise Infrastructure

2.1.2. Privacy-Preserving Analytics

Privacy-Preserving Analytics (PPA) enables the sharing of metadata, and analytics models built on data shared with privacy-preserving properties. Sharing raw data is often not allowed, or at least the information shared must be minimized. Privacy-preserving

techniques such as pseudonymization, differential privacy, and secure enclaves help enable analytics on this private information while trying to preserve privacy, but they introduce an overhead cost on utility.

Pseudonymization replaces sensitive data attributes with pseudonyms, where sensitive information can be disclosed in the future (for example, authorized audit). Pseudonymization is an identified risk mitigation technique in GDPR (recital 28) but does not satisfy GDPR since it can be reversed when the secret key is stolen or leaked. The privacy-preserving property is based on the Key Management and the need for availability, since pseudonyms cannot be renewed. Differential privacy independently randomizes the output of a query, enabling information about a group to be shared while trying to keep the presence or absence of a individual respondent private. With any randomness it cannot be precise, and many queries can violate privacy due to the "cumulative effect"; the accuracy is affected too. Differential privacy is a security requirement identified in several standards, including NIST SP800-537, and it helps evaluate the risk enforcement on Big Data Risk Management Framework of NIST SP800-53.

Secure enclaves help protect sensitive data not encrypted in storage and memory and make computations on plaintext possible; only the result is revealed, when the secret key is protected. The privacy-protecting property depends on the Security of the enclave ecosystem. With the increased usage of models in Business Intelligence and Data Science, the auditability also gains importance. Pseudonymization relies on the Key Management to limit the perceived risk. Differential privacy addresses this requirement with a parameter quantifying the risk, but for what if the user does not set the delta? Secure enclaves totally hide the data and the query from the algorithm owner. All the mentioned techniques violate utility, especially for Data Science models dealing with large datasets, because normally a dataset is used only once.

2.2. Foundations of Trusted Data Platforms

Data platforms help an organization collect, store, and share its data across various silos, applications, and actors. When these platforms are built as Trusted Data Platforms (TDPs), they facilitate the designing of trust-related requirements and applied measures while complying with existing Data Governance and Compliance policies. Data Governance and Compliance are complemented by Data Provenance and Lineage, which facilitate the capture, storage, and verification of the origin, history, and evolution of the data. Acquiring full knowledge about the data source and the processes undergone by the data increases the overall trustworthiness of the data.

2.2.1. Data Governance and Compliance

Data governance defines policies, roles, responsibilities, and stewardship expectations for an organization's information assets. Effective governance ensures that data is managed properly throughout its lifecycle, from creation and collection to dissemination and disposal. Data governance is closely linked to compliance, which involves ensuring adherence to rules, laws, and regulations governing data usage. Privacy laws establish principles around information collection, use, and access. Regulation specifies how well-defined and implementable the established principles are. Compliance policies detail who owns the data, who can access it, how it must be protected, and the penalties associated with violations.

Governance requirements stem from the intended use of information in data products, as well as the associated legal, regulatory, and ethical obligations. Mapping compliance requirements to data governance frameworks ensures that the required compliance roles and controls are established within the governance framework. Mapping regulatory requirements to data governance definitions ensures that the relevant, mandatory, and enforceable policies and procedures are defined within the context of the data product to support compliance. Automated controls that align the infrastructure and operation of the trusted data platform with compliance requirements help maintain continuous compliance with reduced cost and effort.

Fig 2.2: Data Governance and Compliance

2.2.2. Data Provenance and Lineage

Data Provenance and Lineage ensure the integrity and trustworthiness of data, especially as it traverses multiple sources, consumers, and transformations within the enterprise. While provenance captures how the data arrives, lineage displays its journey. A complete, trusted, and visual lineage is critical for audit purposes and consequently trust. The processes involve the recording of what, who, when, where, why, and how. In addition to capturing provenance at each transformation step, it must also be stored and made easily available for automatic verification.

By validating data and its lineage against technical and business rules, threats to trust can be identified and corralled. This is important because data is susceptible to attacks that can corrupt, infect, and introduce uncertainty. Critical or sensitive data such as dummy banking details deserves primary importance. For all data, especially AI-generated data, the origin must always be verified. Data that is out of date also needs to be checked and corrected. An understanding of how individuals generate data can help pinpoint anomalous behavior. Data can mislead if one does not know its limitations, confidence levels, and possible biases.

2.3. Architectural Principles for Enterprise Platforms

The distinct characteristics of enterprise platforms dictate certain architectural decisions and principles when designing a Trusted Data Platform. Enterprise platforms are generally modular, can be creatively combined with other platforms to form larger systems, and are scalable. Modular systems are composed of components (or services) connected through interfaces, meaning that they encapsulate some part of the functionality of the system and "know how to communicate with the rest of the system". Components have defined responsibilities or service boundaries, including functionality such as data ingestion that cannot be separated. Scaling in an enterprise platform context refers not only to scaling up or down (i.e. the size of the incidentals of the system) but also to scaling out in volume. Furthermore, modularity can improve resilience, since different components can fail independently.

The architectural design pattern for many Trusted Data Platforms has been a centralized Data Lake approach, although other solutions are emerging. A Data Lake is a data repository for storing large volumes of structured and unstructured data in its native format until needed. Unlike Data Warehouses, Data Lakes hold a wider array of data sources without predefined schemas. However, Data Lakes have limitations. It is difficult to establish a common data governance regime in a centralized Data Lake. Issues related to data ownership and stewardship become blurred, making it hard to determine responsibility for data discoverability, availability, quality, and usability.

Consequently, a Data Mesh approach is becoming an appealing alternative: a portal that enables different teams to publish and consume the data they need; places responsibility for data quality in the hands of the teams that generate the data; and channels funds toward data as a product. Nevertheless, the Data Mesh approach raises questions about data discoverability and interoperability across internal teams.

2.3.1. Modular and Scalable Architecture

A modular and scalable architecture enables labour distribution, improves agility, and balances commonality and variability to accommodate growth. Businesses can engage different vendors for specialized components with defined interfaces, separating product and operational lifecycles while leaving service boundaries open in face of changing demand and risk scenarios. Such decoupling also bolsters resilience: disparate systems withstand individual outages, while intra-platform dependencies are shielded by quality gates in metadata-managed pipelines. Moreover, partitioned data stores and computation allow uneven scaling and lower exposure.

Enterprise-wide scale, however, raises governance and control challenges. Data Mesh advocates distributed ownership, with platform services reducing demand on central teams. Governance adapts to a "Data product mindset". Individual domains define discoverability, but must support cross-domain API contracts, security, and cataloging to ensure frictionless data combination. Existing platforms are considered Data Lakes; Data Mesh is debated by those seeking a Lake's centralization, discoverability, and integrity. Data Lakes offer unified ingestion, cleaning, and access, using metadata to onboard heterogeneous sources, enable product-centric APIs, and enforce quality in integrated virtual views.

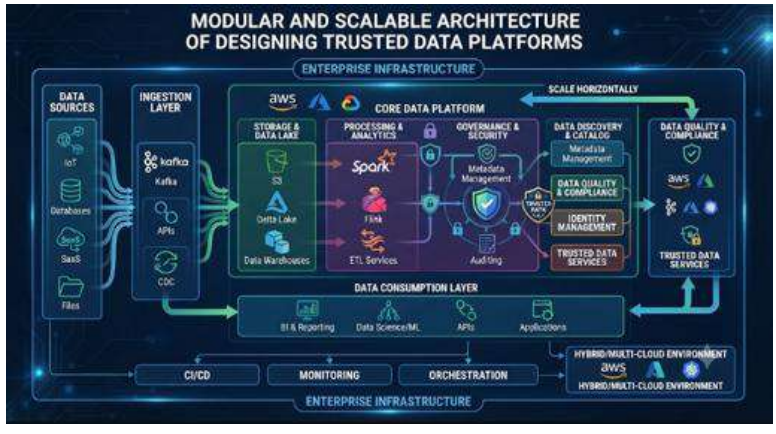

Fig 2.3: Modular and Scalable Architecture

2.3.2. Data Mesh versus centralized Data Lake

Comparing Data Mesh and a centralized Data Lake clarifies how these approaches fulfill the requirements of modular and scalable architecture and reveal different strengths and weaknesses regarding trust aspects beyond security and privacy. Decentralization through data mesh and clear responsibilities for discoverable data products enhance trust by supporting usability and resulting quality, but weaken risk-mitigation capabilities during development and operation. Centralization through a data lake simplifies control, visibility, and risk management, but hinders usability and often leads to a "garbage in, garbage out" situation. Several aspects must remain centrally governed.

The requirements of modular design, scalability, and trust hinge not only on the architectural separation into logical components, their decoupled growth, and resilience to isolated failures, but also on the nature of service definitions and interactions. A decentralized approach builds a Data Mesh, in which data is treated as a product developed and maintained by specialized, federated teams and metadata serves as a common basis for discovery, for collaboration with other products, and for orchestrating access and transformation processes. In contrast, a centralized Data Lake provides a consolidated repository for all own and third-party data, with use-case-oriented ingestion, transformation, and access flows.

2.4. Data Integration and Interoperability

Enterprise data platforms integrate constantly generated data products from various sources through a well-defined ingestion pipeline. The data ingestion process normalizes the ingested data, seamlessly integrating it into the underlying data structures and making it comprehensible for end-users. Data quality gates, based on predefined business rules, validate that the ingested data meets the organization's quality standards for accuracy, integrity, and completeness. Metadata associated with data products moving into the domain of the data platform provides a comprehensive overview of the data's origin and nature. Data lineage tracking mechanisms continue to follow the data across the platform, enabling users to trace the data's journey, understand its context, and assess its reliability.

Interoperability within an enterprise data platform guarantees easy access and usage of data in a consistent and automated manner. APIs simplify, automate, and standardize access to data products with minimal effort from data product publishers and consumers, providing a contract that stipulates the acceptable content and form of requests, responses, and error codes. Implementing API specifications (REST, GraphQL) and design rules (naming conventions, URL structure, query layout, error handling) releases the burden of implementation from API consumers, helping them to quickly adopt an

API-driven approach. API versioning is strategically included in the API contract, enabling a smooth transition to accommodate changes in the API. API security controls ensure appropriate data access by internal or external consumers and protect sensitive information from unintended exposure. Cataloging the enterprise data platform's APIs improves discoverability and encourages internal and cross-organization reuse.

2.4.1. Ingestion, Transformation, and Orchestration

Data integration is the process of aggregating data from multiple sources into a unified form for analysis, enabling enterprises to deliver valuable insights. Data-enabled decision making relies on high-quality data, derived from structured, unstructured, and semi-structured sources spread across multiple silos. The process of collecting, cleansing, and aggregating data is part of the data ingestion stage. When developing pipelines for data ingestion, enterprises face challenges relating to schema evolution, quality assurance, and metadata management. Defining a data quality gate on the ingestion pipeline can help mitigate poor quality issues. Orchestration is the process of managing the data pipelines, defining the workflow execution across the different ingestion pipelines and applying the required transitions in the data flow.

Data sets should be ingestible in parallel, using a dedicated ingestion pipeline. Data pipelines must be defined such that they do not affect the reliability or performance of the source systems. Ingestion from source systems should not mandate online availability because of SLAs between the source and destination system. Ingestion from all data sources must be managed via a metadata repository, which tracks the progress of the ingestion and stores comprehensive statistics about the status of each data set. Once the data enter the data platform, lineage information must be preserved so they can be certified for consumption.

2.4.2. API-first Data Access

Data delivery in a Trusted Data Platform relies on APIs. APIs evolve to be the backbone of the digital economy by enabling global business networks. An API provides the information required not only for a production, logistics and supply chain management partnership but also for the data an organization is willing to share that business partners and customers consider worthwhile. Similar to the supply of information, an API is the only data that is distributed using an expensive yet essential new data type – a data product.

An API is an application program interface that presents a formally defined interface through which one system or application can interact with another by invoking specific

services (usually a resource such as an event, a product, or an offer) and receiving a response. A well-designed API simplifies the integration process, hiding the complexity from the developer and enabling them to focus on shaping a great user experience rather than implementing error handling or connection management.

Well-designed APIs are easy to consume, provide all relevant information, and manage the API contract online in a catalogue. It encompasses all crucial aspects of API design such as security (authentication/authorization), secure access (using rate limiting, throttling, SSL links) , contract definition (using tools such as Swagger/OpenAPI) and intelligent versioning management. API versioning comes into play whenever a developer makes a change to an existing API. Versioning is critical because running multiple iterations of an API at the same time, or at least supporting deprecated ones for a transition period, ensures that existing consumers are not affected. The catalogue supports discoverability via API documentation and provides an engaging on-boarding mechanism for API users.

Fig 2.4: Ingestion, Transformation, and Orchestration

2.5. Conclusion

The principles and processes addressed throughout these sections converge to describe the fundamental requirements for trusted data platforms for enterprise infrastructure. By balancing data security, privacy, and regulatory compliance with data accessibility, utility, and reusability—while ensuring all controls are auditable and verifiable—acceptable levels of trust can be achieved and their satisfying operational characteristics demonstrated. Central to these principles are Security and Privacy by Design, Data

Governance and Compliance, Data Provenance and Lineage, and modular and scalable architectures. Security and Privacy by Design ensure that sensitive data is never exposed to risk without safeguards, that users can never abuse their privileges, and that the consequences of any breach are mitigated. Data Governance and Compliance establish the foundation of policies and responsibilities that allow data to be used in compliance with internal and external legal requirements. Data Provenance and Lineage ensure that any hidden knowledge encoded in the data's vectors, semantic structure, and processing history can be uncovered. A modular and scalable architecture allows all these principles to be applied incrementally, enabling risks to be managed within acceptable limits even during rapid growth.

The remainder of the discussion situates these foundational principles within an enterprise context with additional emphasis on data integration, interoperability, and discoverability. Domain-specific ontologies on which enterprise AI solutions build are inevitably expressed within disparate labels and representations; cross-domain knowledge exploitation therefore necessitates a common point of access to all data, and API-first policies for services and data products ensure consistent, discoverable contracts that renounce source-specific bindings and dependencies. Data ingestion, transformation, and orchestration segregate data collection from consumption, enabling disparate datasets to be assembled, aggregated, and correlated without requiring engineering effort for each new analysis. The resulting architecture resembles a Data Mesh. Data Meshes eliminate management bottlenecks by shifting the onus of data governance and presenting data assets as products—not by scattering ownership across silos, but by assigning domain expertise wherever possible while ensuring custodianship, discoverability, and interoperability are central considerations.

References

Abadi, D. J., Boncz, P. A., Harizopoulos, S., Idreos, S., & Madden, S. (2013). The design and implementation of modern column-oriented database systems. Foundations and Trends in Databases, 5(3), 197–280.

Inala, R., Kaulwar, P. K., Nagabhyru, K . C., Adusupalli, B., & Arun Raj, S. R. (2026). Leveraging IEC 61850 for Interoperable and Resilient Smart Grid Communication Architecture. In Lecture Notes in Electrical Engineering(pp.549–566).SpringerNatureSwitzerland. https://doi.org/10.1007/978-3-032-20533-9_47.

Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., Meng, X., Kaftan, T., Franklin, M. J., Ghodsi, A., & Zaharia, M. (2015). Spark SQL: Relational data processing in Spark. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 1383–1394.

Bernstein, P. A., & Newcomer, E. (2009). Principles of transaction processing (2nd ed.). Morgan Kaufmann.

Dehghani, Z. (2022). Data mesh: Delivering data-driven value at scale. O'Reilly Media.

Ghemawat, S., Gobioff, H., & Leung, S.-T. (2003). The Google file system. Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, 29–43.

Garapati, R. S., Aitha, A. R., Yandamuri, U. S., Gottimukkala, V. R. R., Nagubandi, A. R., & Kolla, S. H. (2026, March). Cloud-Native Orchestration of Multi-Counterparty Derivatives and Collateral in Manufacturing Enterprises via AI-Assisted Financial Audit Engines. In 2026 IEEE International Conference on AI Engineering and Innovations (AIEI) (pp. 1-6). IEEE.

Balamurugan, J., Bhuvaneswari, M., Aitha, A. R., Nagaraju, S., R.Viswanathan, & Dhasarathan, N. (2025). Explanatory Transformer-Based Sequential Recommendation: Assessing BERTRec with SASRec for Customer Behaviour Prediction. In 2025 International Conference on Emerging Engineering Technologies and Applications (IC-EETA) (pp. 470–475). IEEE. 2025 International Conference on Emerging EngineeringTechnologiesandApplications(IC-EETA). https://doi.org/10.1109/ic-eeta66496.2025.11548015

Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media.

Nagabhyru, K. C., Inala, R., Jayakumar, S., & Raj, S. R. (2026). Communication Resilience in Smart Grid Nans: Challenges, Solutions and Future Outlook. In International Conference on Microelectronics, Electromagnetics and Telecommunication (pp. 318-329). Springer, Cham.

Lakshman, A., & Malik, P. (2010). Cassandra: A decentralized structured storage system. ACM SIGOPS Operating Systems Review, 44(2), 35–40.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. National Institute of Standards and Technology, Special Publication 800-145.

Rani, S., Thavara, S. S., Kr, A., Naguband, A. R., Garg, A., & Vajpayee, A. Financialization of Sustainability: ESG Signalling, Market Valuation and the New Economics of Corporate Legitimacy.

Kummari, D. N., Aitha, A. R., Kumar, M. V. K., Pandiri, L., Gottimukkala, V. R. R., & Nagubandi, A. R. (2026, March). Real-Time AI-Driven Audit and Compliance Framework for Smart Manufacturing Financial Workflow. In 2026 IEEE International Conference on AI Engineering and Innovations (AIEI) (pp. 1-5). IEEE.

Kumar, M. V. K., Kolla, S. H., Pamisetty, V., Pandiri, L., Yandamuri, U. S., & Valiki, D. (2026, May). Enterprise-Scale Generative AI Agents for Secure and Governed Automation in Insurance and Public Financial Management. In 2026 International Conference on Computational Robotics, Testing and Engineering Evaluation (ICCRTEE) (pp. 1-6). IEEE.

Stonebraker, M., Madden, S., Abadi, D. J., Harizopoulos, S., Hachem, N., & Helland, P. (2007). The end of an architectural era (It's time for a complete rewrite). Proceedings of the 33rd International Conference on Very Large Data Bases, 1150–1160.

Vassiliadis, P. (2009). A survey of extract-transform-load technology. International Journal of Data Warehousing and Mining, 5(3), 1–27.

Kummari, D. N., Burugulla, J. K. R., Malempati, M., Amistapuram, K., Garapati, R. S., & Nagabhyru, K. C. (2025, December). Enhancing Audit Compliance and Operational Efficiency in Manufacturing and Commercial Insurance Through Agentic AI and Data Engineering Frameworks. In 2025 IEEE International Conference on Communication Networks and Computing (CNC) (pp. 714-720). IEEE.

Sanku, R., Kolla, S. H., Karri, S., & AS, Y. (2026, February). An Intelligent Analytics-Aware Cloud-Cluster Framework for Large-Scale Data Analytics. In 2026 3rd International Conference on Integrated Intelligence and Communication Systems (ICIICS) (pp. 1-7). IEEE.

Nagabhyru, K. C., Gadi, A. L., Seenu, A., Davuluri, P. S. L. N., Segireddy, A. R., & Pamisetty, V. (2026). Towards Automated Financial Risk Scoring in Automotive Financing with Explainable Machine Learning. In 2026 IEEE International Conference on AI Engineering and Innovations (AIEI) (pp. 1–6). IEEE. 2026 IEEE International Conference on AI Engineering and Innovations (AIEI). https://doi.org/10.1109/aiei69164.2026.11496822