ACL Digital
an ALTEN group company

Secure Architecture and Solution Design

Industry best practices for architectural considerations for **secure by design, privacy by design and security compliance by design**

Phase 02 of Software Lifecycle Management

Digital Product Engineering Services

> Gartner reports, by 2025, 60% of large enterprises worldwide would adopt privacy-enhancing computations (PEC) in untrusted environments and multi-party data analytics use cases.

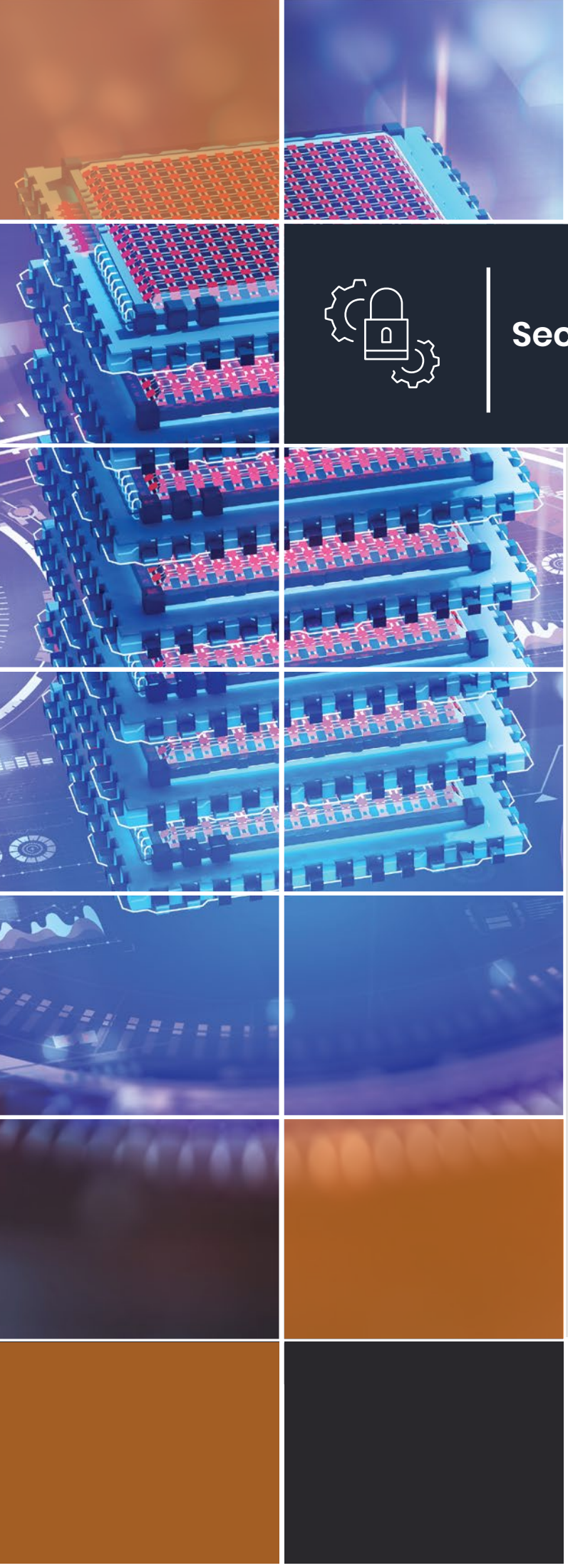
Table of contents

Introduction

In the wake of web 3.0 technologies paving the way for higher functional adoption, it is imperative for every enterprise to employ the highest level of cybersecurity and data privacy norms. Gartner reports, by 2025, 60% of large enterprises worldwide would adopt privacy-enhancing computations (PEC) in untrusted environments and multiparty data analytics use cases. By 2026, 40% of data repositories such as data lakes, will include native capabilities for PEC up from less than 5% today.

With the latest development in building enterprise grade software applications, there is a large influx of newer libraries, newer technologies, and newer ways of shortening the application development lifecycles.

While adopting various cybersecurity related aspects and protecting data privacy of human entities, three architectural patterns that strike at the foundation of all software application development are ‘secure by design’, ‘privacy by design’ and ‘compliance by design’.

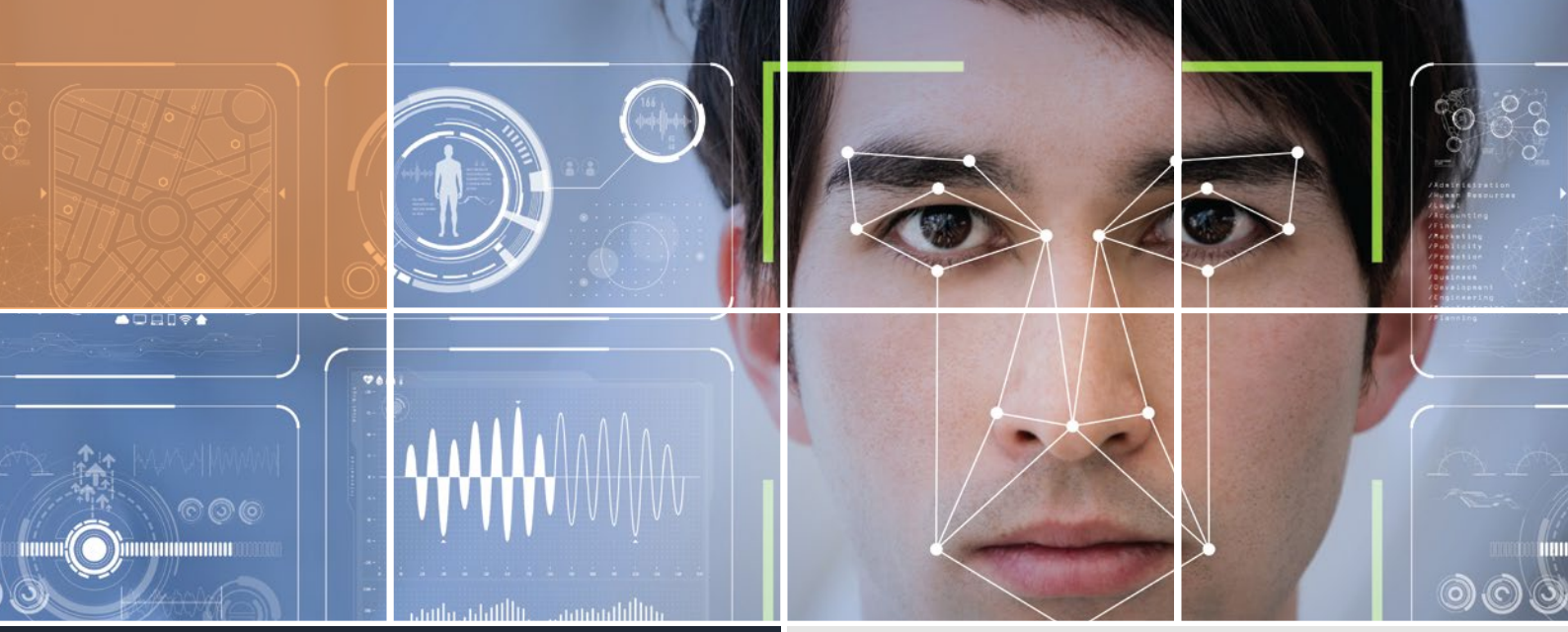


Secure by design

Secure architectural design decisions are based on well-known strategies, tactics, and patterns defined as reusable techniques for achieving specific quality concerns. Security tactics and design patterns provide solutions for enforcing the necessary authentication, authorization, confidentiality, data integrity, privacy, accountability, availability, safety, and non-repudiation requirements, even when the system is under attack.

There are different architectural considerations and secure practices that can be applied at a foundation level while implementing applications.

The Open Web Application Software Project (OWASP) top 10 offers good coverage towards AppSec for designing web applications.



Secure Identity and Access Management

A strong identity management is key to effective user management functions in any application that provides granular control over authentication and authorization.

Federated Identity Management

It would be highly recommended to make use of an external identity provider (IdP) which federates user identities across multiple applications and authorizes access to services offered by a Service Provider (SP) via authenticated tokens.

Multi-factor authentication

For password-based authentication, it is often recommended to introduce another layer of user identification for securing access. This second level of authentication could be in the form of a captcha, a One Time Password (OTP) sent over email or phone, authenticator apps providing a time-based code, or use of biometric interfaces such as fingerprint, face recognition or iris scans.

Password-less authentication

Alternately, password-less authentication techniques do not require multi-factor authentication. These methods include biometric based authentication, usage of magic links (one-time authentication links sent on emails), or authentication notification received on authenticator apps. Password-less authentication saves more time, money and energy thereby reducing maintenance overhead of remembering multiple passwords and the overhead of changing them over time, as well.

Securing Application Design

One must begin designing all applications with the most restrictive access control model. The application must exude **secure by default** principles from its very foundation.

1. All users by default should have no access to the application data or functions that manage the application data. Any need for such access must be sanctioned through a permission-based model which allows access for a timebound period usually through a token-based approach.

2. Any other external user (e.g., system administrator, database administrator) should have no access ever to application data. No special privilege can be accorded to enable access to these users ever. All access must be controlled and restricted only through the application.

3. Apply encryption-at-rest algorithms with customer-managed keys for all application data at rest. Most cloud providers offer such services with RSA 2048-bit customer-managed keys.

4. Apply strong field-level encryption to all application data such that any inadvertent external unauthorized user will not be able to decipher.

5. Apply pseudonymization algorithms to entities sourced from other enterprise systems. For e.g., if your application is connected to the enterprise CRM, any CRM entity stored within your application must use a pseudonymized identifier instead of the actual data. This will preserve the data integrity without exposing the actual details.

6 Any deletes of the application data entities must anonymize such entities instead of actual removal. Sometimes, removal of entities impact business functions that are dependent on such data and hence, anonymization obfuscates the entities without impacting the data integrity.

7 Use secure key vaults for storing all secrets, keys, or any other configuration parameter values. The application is made more secure by using an application configuration service with parameter value retrieval from key vaults.

8 Utilize a sidecar design pattern with microservices while deploying application logging, exception logging, auditing, and monitoring services. This way, the application lives in its own container without exposure of any data outside its own boundary.

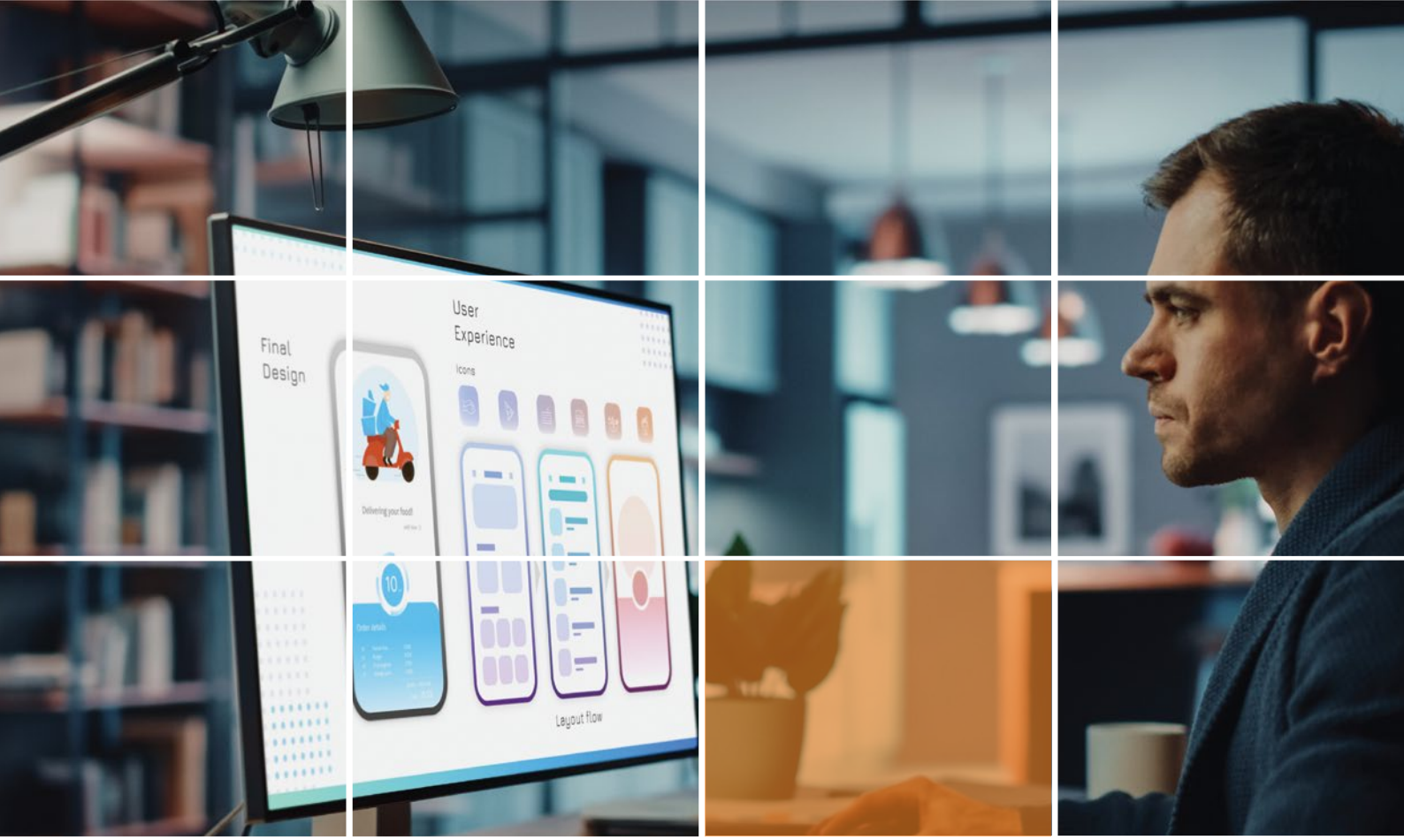
9 Hardening of containers and virtual machines further enable vulnerability scanning of open-source software (OSS) to contain data exposure issues. COBIT 5 offers readily available hardened virtual machines and appliances for most operating systems.

10 Vulnerability scanning of source code through static, dynamic, and mobile ensure threat impact Regular use of source code analysis tools like Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST) and Mobile Application Security Testing (MAST) help reduce common weakness enumeration (CWE).

11 Generative AI offering synthetic data, instead of original data for testing purposes. This way, PII data need not be exposed to any unauthorized user.

12 Secure service endpoints by introducing a security profile and token-based access for all inbound and outbound service access.

13 Introduce secured API Management interface for managing all service endpoints with well-defined service level agreements (SLA) and adequate security profiles. The SLAs help evade unauthorized access and allow enterprise access to only authorized personnel.



Privacy by design

Often enterprises create applications based on business needs without proper planning for dealing with enterprise data; every new application starts capturing and storing personal data freely without following an enterprise-wide strategy for dealing with data privacy. This leads to overloaded data stores with tons of personal data in them, adding privacy risk and liability. Due to missing organization strategy and lack of early architectural considerations for data privacy, bolting on privacy measures after applications are deployed, only further increases infrastructure complexity and therefore operational costs.

For enterprises to be better equipped with data privacy principles, they should employ 'privacy by design' to remain within audit boundaries and for minimizing privacy related risks.

It often pays to conduct a privacy impact assessment and create a baseline privacy criterion which considers all cases of possible privacy violation. Later this baseline is used for laying the foundational architectural principles for all digital adoption within the enterprise and create a more matured baseline by building on it further.

Privacy-enhancing computation techniques (PEC)

Privacy-enhancing computation techniques allow protection of data in use in addition to in-transit and at-rest protections. It supports the CIA triad (Confidentiality, Integrity, and Availability) of enterprise data in use by multiple applications. Privacy-enhancing computation is named as one of the factors for instilling Engineering Trust in the Top Strategic Technology Trends for 2022 by Gartner.

These are used for identifying a variety of technologies and approaches that protect data while it is used mainly regarding confidentiality and privacy. These can be segregated into the following areas –

- **Differential privacy** can transform data on the fly while keeping the source data intact. Here synthetic data can create entire new repositories instead and homomorphic encryption transforms the source data. Synthetic data is produced through generative AI algorithms by preserving all attributes of the original source data. Such synthetic data is very useful for testing near-real data scenarios without exposing or sharing confidential personal data

- Federated machine learning enables decentralized usage of knowledge without transferring the actual identifiable data. **Secure Multiparty Computation** (SMPC / SPC) enables analysis of data while remaining in encrypted form.

- Trusted execution environments (TEE) in IaaS cloud models are enabled by **confidential computing** and are useful for containing and protecting personal data while being used by applications.

- The broad adoption of **Zero-knowledge proofs** (ZKPs) in blockchain technology offers superior transparency, immutability, and privacy. They allow one party to prove the knowledge of certain transaction to the other party without revealing the content of the transaction. Access to the data is allowed only if all nodes conduct a secure handshake amongst themselves. This approach ensures that a malicious actor cannot gain access to the secured transaction.

In addition to privacy-enhancing computation techniques, certain additional architectural considerations may help protect data privacy:

- Masking of personally identifiable information (PII) across all logging data will help protect the application from the risk of exposure.

- Minimizing PII data capture and processing, and limiting to only intended use, subject to the legal basis helps protect the spread of PII data.

- Pseudonymizing PII data helps in reducing overuse or unnecessary storage.

- Having a check on the retention period of PII data will help control for how long data stores will continue to host PII data.

- Backup, disaster recovery and archival sites should be probed for how long PII data will continue to be stored.

Security Compliance by design

If an enterprise makes a conscious effort of introducing a strategic approach to integrate cybersecurity and data protection related regulatory requirements into their organization's processes, it will help reduce the impact of any non-conformances early in the lifecycle. To cope up with the ever-changing landscape of regulations, procedures, policies and standards around cybersecurity and data protection, it is becoming indispensable for all enterprises to adopt them at the earliest without waiting for a risk to occur.

Depending upon the business domain for which applications are being built, appropriate cybersecurity regulations are also needed to be complied with. Common regulations include –

- Sarbanes-Oxley Act (SOX)
- General Data Protection Regulation (GDPR)
- ISO 27001: Information Security Management System
- Service Organization Controls 2 (SOC 2)
- National Institute of Standards and Technology (NIST) standards
- Health Insurance Portability and Accountability Act (HIPAA)
- Federal Risk and Authorization Management Program (FedRAMP)
- The Payment Card Industry Data Security Standard (PCI DSS)
- Federal Information Security Management Act (FISMA)
- Family Educational Rights and Privacy Act (FERPA)
- Basel or Gramm-Leach-Bliley Act (GLBA)
- California Consumer Privacy Act (CCPA)
- Authority to Operate (ATO)
- Center for Internet Security (CIS)

The complexities and high overhead of data security and compliance measures require a good balance of processes and tools that would facilitate smooth software delivery. Compliance measures need to be applied appropriately to the different functional domains of the Software Lifecycle Management process

- Database compliance
- Application code compliance
- Infrastructure compliance
- Open-source software (OSS) compliance

Recommended Reading

Whitepaper on Phase-01: Requirement Lifecycle Management

References

- https://docs.microsoft.com/en-us/azure/security/fundamentals/encryption-atrest
- https://docs.microsoft.com/en-us/azure/architecture/patterns/sidecar
- Gartner Report: Predicts 2022: APIs Demand Improved Security and Management
- Gartner Report: Top Strategic Technology Trends for 2022
- OWASP Top 10: https://owasp.org/Top10/