Response to Public Consultation

EDPB Template [2026] for Personal Data Breach Notification

Technical Contribution on Pre-Disclosure Prevention, Effectuation Status and Verifiable Mitigation

Respondent: Sangam Kumar Das
Independent inventor and researcher
Educational Qualification - Bachelor in Technology in Electronics and Telecom Engineering

Capacity: Individual technical respondent. This submission does not represent any client, employer, institution or commercial party.

Date: 30 July 2026

Executive Summary

I am grateful to the European Data Protection Board for the opportunity to comment on the Template [2026] for Personal Data Breach Notification.

The proposed template is comprehensive and should materially improve the consistency of breach notifications across supervisory authorities. It captures the nature and cause of an incident, the affected data, protections already in place, likely consequences, risk assessment, mitigation, preventive measures and communication to data subjects.

My comments are confined to one technical issue: the template should more clearly distinguish between:

1. an attempted disclosure or harmful operation that was prevented before personal data became externally usable;
2. a partially completed disclosure;
3. a completed confidentiality, integrity or availability effect;
4. an event whose technical completion cannot yet be determined; and
5. a completed event whose downstream use was later constrained or revoked.

This distinction is increasingly important in automated, cloud-based and AI-mediated systems. Conventional incident descriptions often focus on whether an attacker entered a system, accessed a database or initiated an export. Those facts are important, but they do not always establish whether personal data crossed the final operational boundary and became usable by an unauthorised recipient.

Conversely, the absence of a conventional file export does not necessarily mean that no breach occurred. An AI model, automated agent or analytics service may expose personal data through generated outputs, derived inferences, API responses, tool calls or repeated low-resolution disclosures.

I therefore recommend that the template expressly capture:

- whether the unauthorised operation was attempted, partially completed, completed or prevented;
- the first boundary at which the personal data became externally usable;
- whether a technical control prevented that boundary from being crossed;
- whether the recipient, destination, purpose and permitted output were technically verified;
- whether the controller possesses independently verifiable evidence of the release or prevention decision;
- whether repeated outputs or linked systems created cumulative exposure;
- whether protective measures remained effective throughout the incident; and
- whether future disclosures can be revoked, constrained or prevented.

The proposed additions would not change the legal definition of a personal data breach. They would help controllers and supervisory authorities determine more accurately what technically occurred and what risk actually materialised.

1. Introductory Remarks

The template is primarily intended for implementation by supervisory authorities through structured IT tools. Its predefined values, conditional fields and tooltips are therefore especially important. They will influence how controllers describe incidents and how supervisory authorities compare notifications across organisations and jurisdictions.

The template already distinguishes confidentiality, integrity and availability breaches and asks whether personal data was exfiltrated, likely exfiltrated, not exfiltrated or impossible to assess. It also asks whether data remained unintelligible or whether individuals remained unidentifiable.

Those distinctions are useful. However, modern digital systems contain several intermediate technical states between an attempted operation and a completed external effect.

For example:

- malicious code may create an export request, but an egress control may prevent transmission;
- an employee may initiate an unauthorised download, but a protected release boundary may refuse to provide a usable file;
- an AI agent may generate a candidate response containing personal data, but an output control may suppress it before delivery;
- data may be transmitted but remain cryptographically unusable to the recipient;
- a cloud service may receive data, but the applicable authority may prohibit decryption, reconstruction or further release;
- a model may reveal personal information through inference even though no original database record was exported; or
- repeated individually low-risk outputs may cumulatively expose an individual.

A technically precise notification should therefore identify not only what operation was initiated, but also whether and where it became effective.

2. Scope of This Response

This response addresses six points:

1. separating attempted operations from completed breaches;
2. identifying the final disclosure or effectuation boundary;
3. separating unintelligibility from non-identifiability;
4. recording purpose, recipient and destination violations;
5. capturing cumulative and AI-mediated disclosure;
6. recording verifiable prevention and mitigation evidence.

A technical architecture supporting these distinctions is then presented, followed immediately by clarification addressing latency and legacy-system objections.

3. Comment 1 — Distinguishing Attempted, Prevented, Partial and Completed Events

Relevant fields

This comment principally concerns:

- Field 55: Nature of the personal data breach;
- Field 56: Type of confidentiality breach;
- Field 60: Nature of the incident;
- Field 62: Further description of the incident; and
- Field 89: Outcome of the risk assessment.

The issue

Field 56 currently allows a controller to indicate that personal data was:

- exfiltrated or disclosed;
- likely exfiltrated or disclosed;
- not exfiltrated or disclosed, with reasonable evidence;
- impossible to assess; or
- still to be determined.

This is a useful starting point. However, "not exfiltrated or disclosed" may describe materially different situations:

- no unauthorised operation was initiated;
- an operation was initiated but failed accidentally;
- an operation was detected and blocked by a preventive control;
- data was copied internally but prevented from crossing an external boundary;
- encrypted material left the system but no usable reconstruction authority was available;
- an output was generated but suppressed before delivery; or
- part of the requested data was released while the remainder was blocked.

These situations may differ substantially in:

- the effectiveness of the controller's safeguards;
- the likelihood of future recurrence;
- the credibility of the forensic evidence;
- the residual risk to data subjects; and

- the remedial action required.

Proposed additional field

I respectfully suggest adding a structured field titled:

Operational completion status

Possible values:

a. Attempted operation prevented before unauthorised access or disclosure
b. Attempted operation partially completed
c. Unauthorised access occurred, but no external disclosure was established
d. External disclosure or availability to an unauthorised recipient occurred
e. Disclosure occurred, but the data remained technically unusable
f. Downstream use or further transmission was subsequently prevented
g. Completion status cannot presently be determined
h. Other, with explanation

The tooltip could state:

Indicate the furthest technical stage reached by the incident. Where an operation was prevented, identify the control and boundary that prevented completion. This field does not determine by itself whether the event constitutes a personal data breach under the GDPR.

This final sentence is important. The template should collect technical facts without suggesting that every prevented attempt is independently notifiable.

4. Comment 2 — Identifying the Point at Which the Breach Became Effective

The issue

A controller may know that an attacker accessed a system but still be uncertain whether personal data became externally usable.

Traditional incident descriptions often use infrastructure events as proxies for actual consequences:

- a database query was executed;
- an archive was created;
- a download button was activated;
- a network connection was established;
- an AI tool was invoked; or

- an email was queued.

These events do not necessarily establish that the personal data reached an unauthorised recipient in usable form.

The opposite problem also exists. A system may not record a conventional “download,” yet disclose personal data through:

- a generated AI response;
- a retrieval-augmented model;
- an API response;
- a screen display;
- a voice assistant;
- an automated decision;
- an agentic tool invocation;
- a transformed dataset;
- a statistical output; or
- an inferred profile.

The template would benefit from identifying the first boundary at which the personal data or derived information became usable outside the authorised processing context.

Proposed field

First externally effective boundary

Possible values:

a. User interface or screen display
b. File download or export service
c. Email or messaging transmission
d. API response
e. Cloud storage or synchronisation boundary
f. Network egress boundary
g. AI or automated-system output
h. External tool, plugin or agent action
i. Database or application access by an unauthorised person
j. Physical document or device transfer
k. Other
l. Unknown

A related narrative field could ask:

Describe the first point at which the affected personal data became available for unauthorised use, disclosure, alteration, destruction or denial of access. If the operation was prevented, describe the boundary at which prevention occurred.

This would help supervisory authorities distinguish:

- compromise of an internal component;
- preparation for disclosure;
- attempted disclosure; and
- completed external effect.

5. Comment 3 — Separate Unintelligibility from Non-Identifiability

Relevant field

Field 57 asks:

Was the data unintelligible to anyone who was not authorised to access it or were individuals not identifiable?

This combines two technically and legally distinct conditions.

The issue

Unintelligibility

Data may be unintelligible because it is:

- strongly encrypted;
- fragmented;
- tokenised;
- protected by inaccessible cryptographic material; or
- encoded in a form that the unauthorised recipient cannot reconstruct.

The information may still be personal data for the controller or another entity possessing the reconstruction means.

Non-identifiability

Information may instead be intelligible but not relate to an identified or identifiable natural person from the relevant perspective.

These conditions require different evidence.

For encrypted data, relevant questions include:

- where the keys were stored;
- whether the keys were compromised;
- whether decryption authority existed;
- whether the encryption remained intact; and
- whether the unauthorised party had realistic means of obtaining the keys.

For allegedly anonymous information, relevant questions include:

- whether records could be isolated;
- whether the information could be linked;
- whether specific information could be inferred;
- what auxiliary information was available; and
- whose perspective was assessed.

Combining the two concepts in one field may produce ambiguous answers.

Proposed restructuring

Field 57 should be divided into two separate questions.

57A — Was the affected data technically unintelligible?

Possible values:

a. Yes, and the protection remained intact
b. Protection was applied, but may have been subverted
c. No
d. Cannot presently be assessed
e. To be determined

57B — Could the affected information relate to identified or identifiable individuals from the unauthorised recipient's perspective?

Possible values:

a. No, based on a documented anonymisation assessment
b. Identification was unlikely but could not be excluded
c. Yes
d. Cannot presently be assessed
e. To be determined

Suggested tooltip

Encryption, pseudonymisation and anonymisation should not be treated as interchangeable. If relying on unintelligibility, describe the protection and whether reconstruction means were compromised. If relying on non-identifiability, describe the perspective assessed and the basis on which identification, linkage and inference were considered unlikely.

6. Comment 4 — Capturing Recipient, Purpose and Destination Violations

Relevant fields

This issue relates particularly to:

- Field 60: Nature of the incident;
- Field 62: Further description;
- Field 63: Systems and infrastructure;
- Field 79: Confidentiality consequences;
- Field 92: Mitigation; and
- Fields 94–95: Preventive measures.

The issue

A disclosure may be unauthorised even where the receiving system was technically approved for some other operation.

Examples include:

- disclosure to an authorised processor for an unauthorised purpose;
- routing to an approved cloud provider in an unapproved region;
- use of an approved AI model with an unapproved retrieval source;
- release to the correct organisation but the wrong employee or account;
- disclosure at a precision or granularity exceeding the approved level;

- use by a recipient after revocation or expiry;
- transfer through an unauthorised plugin or agent tool; or
- release to an authorised endpoint under a substituted identity.

The template currently includes options such as incorrect permissions, wrong recipient, misconfiguration and unauthorised access. These are helpful, but a more structured account of the violated condition would improve root-cause analysis.

Proposed field

Authorisation condition that failed

Multiple selections should be permitted:

a. Recipient identity
b. User or account identity
c. Destination endpoint
d. Jurisdiction or processing region
e. Processing purpose
f. Data category
g. Data precision or resolution
h. Volume or frequency limit
i. Model, software or runtime version
j. Tool, plugin or external service
k. Expiry or revocation state
l. Cumulative disclosure threshold
m. Other
n. Unknown

This would help separate:

- authentication failure;
- authorisation failure;
- purpose limitation failure;
- destination-control failure;
- disclosure-volume failure; and
- policy-enforcement failure.

7. Comment 5 — Cumulative and AI-Mediated Disclosure

The issue

The template mainly describes a breach as an incident affecting identifiable records or systems. However, modern AI and analytics systems may expose information through a sequence of outputs rather than through one direct database extraction.

Examples include:

- repeated prompts extracting fragments of personal information;
- multiple aggregate queries permitting differencing;
- a sequence of coarse location outputs revealing a movement pattern;
- retrieval tools combining information from several protected sources;
- an agent accumulating outputs in persistent memory;
- model responses revealing membership in a dataset;
- derived profiles revealing health, political, religious or behavioural information; and
- automated tools sending individually limited outputs to a common destination.

No single output may appear sufficient to identify or harm a person. The complete sequence may do so.

Proposed clarification

Field 63 or a new dedicated field should ask:

Did repeated outputs, queries or linked systems contribute to the breach?

Possible values:

a. Yes, repeated disclosures increased the exposed information
b. Yes, multiple systems or datasets were combined
c. Yes, an AI model or autonomous agent generated or transmitted the information
d. No
e. Unknown
f. To be determined

A narrative field should request:

- the number and period of relevant outputs;
- whether the same recipient received them;
- whether model or agent memory was involved;

- whether external data sources were consulted;
- whether multiple recipients could combine their holdings; and
- whether the controller maintained cumulative disclosure state.

The template should also allow "AI-generated or inferred personal data" as an additional type of breached information under Fields 70–71.

This would not create a new category of personal data. It would identify the technical means through which existing personal-data categories were disclosed or inferred.

8. Comment 6 — Verifiable Evidence of Prevention and Mitigation

Relevant fields

This comment principally concerns:

- Fields 75–76: Measures in place;
- Fields 89–90: Risk assessment;
- Field 92: Mitigation;
- Fields 94–95: Measures preventing recurrence; and
- Field 125: Attachments.

The issue

The template asks controllers to describe technical measures, but it does not expressly distinguish between:

- a policy stating that disclosure was prohibited;
- a system configuration intended to prevent disclosure;
- a log showing that a control was evaluated;
- evidence that the control refused the operation; and
- evidence that no usable external release occurred.

For supervisory analysis, those levels of evidence are materially different.

A firewall rule, access policy or contractual prohibition may have existed but failed. Conversely, a release control may have produced verifiable evidence that an unauthorised output was prevented before it became usable.

Proposed additional evidence fields

Was the prevention or release decision technically evidenced?

Possible values:

a. Yes, independently verifiable evidence confirms that the operation was blocked
b. Yes, evidence confirms that only a reduced or protected representation was released
c. Evidence indicates that the operation was allowed
d. Logs exist but are incomplete or alterable
e. No reliable evidence exists
f. To be determined

Types of evidence available

Multiple selections:

a. Tamper-evident event log
b. Cryptographically protected validation record
c. Recipient and endpoint verification record
d. Output digest or integrity record
e. Policy-version record
f. Revocation or expiry record
g. Network egress evidence
h. Decryption or reconstruction-authority record
i. Cumulative disclosure record
j. Independent forensic report
k. Other

Field 125 could include:

Dated copy or export of technical evidence showing whether the affected operation was authorised, prevented, partially completed or completed.

The template should not require controllers to disclose sensitive security secrets. Evidence may be summarised, redacted or provided through an appropriate confidential channel.

9. Proposed Technical Solution — Protected Pre-Effectuation Release Control

The following architecture is offered as a technology-neutral example of how a controller may prevent an attempted breach from becoming an externally effective disclosure.

It is not proposed as a mandatory design and does not determine whether a personal data breach has occurred as a matter of law.

9.1 Candidate operation

A proposed data access, export, AI response, tool invocation, file transmission or system action is first represented as a candidate operation.

Relevant parameters may include:

- requesting identity;
- intended recipient;
- destination endpoint;
- processing purpose;
- affected data class;
- requested precision;
- volume;
- jurisdiction;
- software, model or runtime;
- policy version;
- validity period;
- previous related outputs; and
- revocation status.

9.2 Non-effective state

The candidate operation is prevented from becoming externally effective merely because an application, employee, model or agent has requested it.

Computation and external effect are separated.

For example, an AI system may calculate a candidate answer, but the answer is not deliverable until the release conditions are verified.

9.3 Protected enforcement domain

A protected component evaluates whether the operation is permitted.

It may verify:

- recipient identity;
- destination;

- purpose;
- jurisdiction;
- applicable policy;
- data class;
- permitted precision;
- model or tool identity;
- cumulative exposure;
- expiry;
- revocation; and
- whether the operation is associated with a known incident or compromised credential.

The protected component should be resistant to alteration by the ordinary application or model requesting the operation.

9.4 Adaptive response

Where the requested operation is not fully authorised, the system may:

- block it;
- suppress fields;
- lower the geographic or temporal precision;
- release an aggregate representation;
- reduce the number of records;
- remove sensitive attributes;
- substitute synthetic or masked values;
- require additional authorisation; or
- quarantine the operation for review.

9.5 Scoped, non-transferable release authority

Following validation, the system may issue a narrowly scoped authority bound to:

- one operation;

- one output;
- one recipient;
- one endpoint;
- one purpose;
- one policy state;
- one validity interval; and
- one release boundary.

Possession of a copied authority should not enable a different recipient or system to reuse it.

9.6 Final-boundary verification

The first point at which the personal data would become externally usable independently verifies the release authority.

If verification fails, the effect does not complete.

Relevant boundaries may include:

- an API gateway;
- download service;
- email gateway;
- cloud egress service;
- AI-output gateway;
- external-tool connector;
- database access proxy;
- device interface; or
- physical actuation boundary.

9.7 Validation and denial evidence

The system records whether the operation was:

- authorised;
- denied;

- reduced in scope;
- redirected;
- quarantined;
- attempted repeatedly;
- blocked due to revocation; or
- completed.

The evidence should identify the applicable policy and technical conditions without unnecessarily duplicating the affected personal data.

9.8 Incident-response use

After detection of an incident, the controller may:

- revoke affected release authorities;
- invalidate compromised identities;
- restrict destinations;
- lower permitted precision;
- block a model, plugin or tool;
- prevent further outputs;
- freeze cumulative exposure state;
- require human approval; or
- place the affected system in a non-disclosing mode.

This architecture can therefore contribute both to:

- preventing an attempted disclosure; and
- limiting further consequences after an initial breach.

10. Addressing Latency and Legacy-System Objections

10.1 Latency

The proposed architecture should not be understood as requiring a complete legal, forensic or privacy-risk assessment to be performed synchronously for every packet, record or transaction.

A practical implementation may separate:

Policy and assessment layer

This layer performs more intensive work, including:

- defining authorised recipients;
- classifying data;
- establishing permitted purposes;
- determining geographic and jurisdictional restrictions;
- setting output-resolution rules;
- evaluating model and tool risks;
- defining cumulative exposure limits; and
- approving transformations.

Release-verification layer

This layer checks the limited set of facts that must remain true when the operation occurs, such as:

- whether the recipient matches;
- whether the endpoint is approved;
- whether the purpose remains valid;
- whether the policy has expired;
- whether access has been revoked;
- whether the requested precision is permitted;
- whether the output matches the approved operation; and
- whether a cumulative threshold has been reached.

The release layer therefore need not rerun the complete risk assessment for every event.

More computationally intensive analysis can be performed:

- when a policy is established;
- when a system or model changes;
- when a new recipient is added;
- when a new data source becomes available;
- following a security event; or
- when a predefined risk trigger occurs.

Local verification, protected cached policy state and narrowly scoped predicates can reduce operational overhead. However, the actual latency will depend on the deployment, hardware, policy complexity, transaction volume and location of the enforcement function.

This submission therefore does not propose a universal latency figure. Any numerical performance claim should be supported by reproducible testing under identified conditions.

The relevant principle is proportionality:

The control should be sufficiently fast for the intended operation while remaining sufficiently independent and reliable to prevent a non-conforming release from completing.

10.2 Legacy systems

The proposal also does not require organisations to replace their databases, applications or cloud platforms.

A controller may introduce release-time controls incrementally at an existing boundary, including:

- an API gateway;
- reverse proxy;
- database access proxy;
- email security gateway;
- file-download service;
- data-loss-prevention system;
- analytics export function;
- cloud egress gateway;
- AI-output filter;

- model tool connector; or
- identity and access-management enforcement point.

The legacy application may continue performing its existing computation.

What changes is that the application no longer possesses unrestricted authority to make every computed output externally effective.

Deployment may begin with:

- high-risk data;
- sensitive exports;
- privileged administrator actions;
- AI-agent tool use;
- cross-border transfers;
- large-volume queries;
- location data;
- health data;
- biometric data; or
- systems with a history of misdelivery.

Legacy compatibility should therefore be treated as an architectural integration question, not as a requirement for wholesale infrastructure replacement.

The template could recognise this by including “release-boundary or egress enforcement” among the preventive measures in Field 94 and by allowing controllers to explain phased implementation in Field 95.

11. Recommended Amendments to Specific Template Fields

Field 56 — Type of confidentiality breach

Add:

- attempted disclosure prevented before external usability;
- partial disclosure;
- completed disclosure;

- data transmitted but technically unusable;
- downstream use prevented after initial disclosure.

Field 57 — Protective status

Split into:

- whether the data remained unintelligible; and
- whether individuals remained non-identifiable.

Field 60 — Nature of the incident

Add:

- AI or autonomous-agent output disclosure;
- unauthorised tool or plugin action;
- purpose-boundary violation;
- jurisdiction or destination-policy violation;
- cumulative query or repeated-output disclosure;
- release-authority or final-boundary control failure.

Field 62 — Incident description

Request:

- furthest stage reached by the operation;
- first externally effective boundary;
- whether a preventive control blocked or reduced the operation;
- evidence supporting that conclusion; and
- root cause of any enforcement failure.

Field 63 — Systems and infrastructure

Add:

- AI models and model versions;
- agent frameworks;
- plugins and external tools;

- retrieval systems;
- API gateways;
- output filters;
- cloud egress controls;
- policy-enforcement points; and
- final release boundaries.

Fields 75–76 — Measures in place

Add:

- recipient-bound release control;
- destination-bound release control;
- purpose-bound processing control;
- output-bound authorisation;
- cumulative disclosure monitoring;
- policy expiry and revocation;
- tamper-evident validation evidence;
- AI-output and tool-use controls; and
- fail-closed release enforcement.

Field 79 — Confidentiality consequences

Add:

- personal data may be inferred from repeated or combined outputs;
- AI systems may have retained or propagated the information;
- the disclosure may enable automated profiling or consequential action;
- the recipient may combine the data with information previously received.

Field 90 — Risk-assessment methodology

Request information on:

- whether exposure was assessed per individual output or cumulatively;

- the recipient's actual capabilities;
- model and agent involvement;
- available auxiliary datasets;
- ability to revoke or prevent further release; and
- reliability of evidence establishing whether disclosure completed.

Fields 92–95 — Mitigation and recurrence prevention

Add:

- revocation of release authority;
- recipient or endpoint blocking;
- output-resolution reduction;
- agent or plugin suspension;
- model-version restriction;
- cumulative exposure limits;
- release-boundary verification;
- protected policy state;
- independent validation evidence; and
- fail-closed behaviour.

Field 100 — Article 34(3) protective measures

Where a controller relies on measures that rendered the data unintelligible, the form should separately ask:

- whether the unauthorised recipient obtained reconstruction authority;
- whether keys or equivalent protected states were compromised;
- whether future access has been revoked;
- whether the protection remained effective throughout the incident; and
- what evidence supports those conclusions.

Field 125 — Attachments

Add:

- release or denial validation record;
- recipient and endpoint verification record;
- output-integrity record;
- relevant policy-version record;
- cumulative disclosure report;
- technical record of revocation or containment.

12. Suggested New Structured Fields

For ease of implementation, the following compact fields could be added without materially lengthening the form.

A. Furthest operational stage reached

- Request initiated
- Data accessed internally
- Output generated
- Release attempted
- Release partially completed
- Release completed
- Downstream use detected
- Unknown

B. Preventive control outcome

- Control prevented completion
- Control reduced the disclosed data
- Control detected but did not prevent
- Control was bypassed
- Control was misconfigured

- No relevant control existed
- Unknown

C. First effective boundary

- User interface
- API
- Download
- Email or message
- Network egress
- Cloud synchronisation
- AI output
- Agent tool
- Physical transfer
- Other
- Unknown

D. Evidence quality

- Independently verifiable
- Tamper-evident internal evidence
- Ordinary application logs
- Partial evidence
- No reliable evidence
- Under investigation

E. Cumulative exposure

- Single isolated event
- Repeated outputs to one recipient
- Multiple datasets combined
- Multiple recipients under common control

- AI or agent accumulation
- Unknown

13. Technical Disclosure and Transparency

The architectural concepts described in this contribution are further developed in my published international patent materials, including PCT/IB2026/053385 and related technical disclosures available through WIPO PATENTSCOPE.

I am the inventor and applicant associated with those materials. This is disclosed for transparency.

The references are provided only to demonstrate that the proposed technical measures have been described at an implementation level.

This submission:

- does not request endorsement of any patent application;
- does not request procurement, licensing or funding;
- does not ask the Board to adopt proprietary terminology;
- does not claim that implementation of a patented architecture establishes GDPR compliance; and
- does not suggest that the template should mandate any particular commercial product.

The Board may express any relevant principle in neutral and technology-independent language.

14. Closing Remarks

The proposed template is a strong foundation for harmonising personal data breach notifications.

My recommendation is not to increase the reporting burden through lengthy technical narratives. It is to collect a small number of additional structured facts capable of answering four fundamental questions:

1. What operation was attempted?
2. How far did it progress?
3. At what boundary did it become—or fail to become—externally effective?

4. What reliable evidence supports that conclusion?

These facts matter because a compromised component is not always equivalent to a completed disclosure, and the absence of a conventional download is not always equivalent to the absence of disclosure.

AI systems, automated agents, APIs and distributed cloud services increasingly separate:

- access from use;
- computation from disclosure;
- disclosure from reconstruction;
- and initial release from downstream effect.

A modern breach-notification template should be capable of representing those distinctions.

The template should also recognise that technical controls can do more than detect and document a breach. Properly designed controls may prevent an attempted operation from becoming effective, reduce the information released, revoke future authority, or contain the incident before additional consequences occur.

Capturing those facts would help supervisory authorities:

- assess actual risk more accurately;
- distinguish preventive controls from paper policies;
- evaluate the effectiveness of mitigation;
- compare incidents across organisations;
- identify recurring architectural weaknesses; and
- encourage technically enforceable data-protection measures.

I am grateful to the Board for its work and for the opportunity to contribute.

Sangam Kumar Das
Independent inventor and researcher