L04: Privacy-Preserving Compliance

Extended Slides – BSc Digital Finance Course

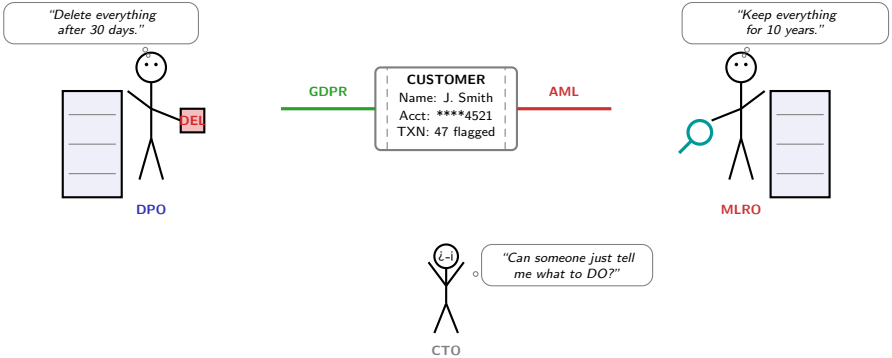
Digital Finance

What Will You Be Able to Do After This Lecture?

1. Formalize the GDPR-AML regulatory conflict as a constrained optimization problem with conflicting objectives and quantify the Pareto frontier
2. Implement differential privacy mechanisms (Laplace, Gaussian) in Python and calculate the privacy-accuracy tradeoff for AML query systems
3. Apply k-anonymity, l-diversity, and t-closeness to financial datasets and evaluate re-identification risk quantitatively
4. Design a privacy-preserving AML analytics pipeline using pseudonymization layers, federated learning, and homomorphic encryption concepts
5. Model the consent architecture required under GDPR for AML data processing and compute retention schedules under dual regulatory mandates
6. Evaluate the EU AI Act's impact on privacy-preserving compliance AI and the regulatory paradox of explainability vs anonymity

Prerequisites: Python (numpy, scipy), information theory basics, L04 main lecture content.

Six objectives: regulatory conflict formalization (1), differential privacy (2), anonymization techniques (3), privacy-preserving architectures (4), consent engineering (5), and AI Act impact (6).



When the law contradicts itself, the engineer inherits the paradox.

The privacy-compliance paradox: GDPR demands deletion, AML demands retention. The same data, opposite obligations.

How Much Does Your Bank Know About You – and Can We Measure It?

Shannon Entropy of a Data Collection

The information content of a single field:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \quad \text{(bits)}$$

Total information across m data fields:

$$H(D) = \sum_{j=1}^{m} H(X_j) + I(X_1; X_2; \ldots; X_m)$$

The regulatory conflict in information-theoretic terms:

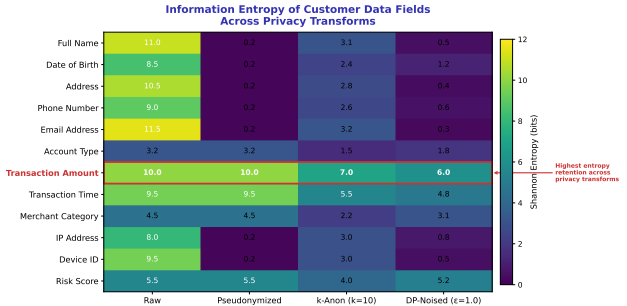
GDPR (Art. 5(1)(c)):	$\min H(D_{\text{retained}})$	s.t. $I(D_{\text{retained}}; \text{AML\_signal}) \geq \tau$
6AMLD (Art. 11):	$\max H(D_{\text{collected}})$	s.t. monitoring $\geq 1 - \alpha$

Fundamental conflict: $\min H(D) \iff \max H(D)$ – cannot satisfy both simultaneously.

Bridge: The mutual information $I(D; \text{Crime})$ is the only quantity both regulators value. Both agree: retain exactly the information that predicts crime, discard the rest.

GDPR minimizes information content. AML maximizes information content. The mutual information between data and crime detection is the only metric both sides agree on.

How Much Information Does Each Data Field Carry – Before and After Privacy Transforms?



Reading the matrix:

- Rows = data fields (name, address, transaction amount, timing, etc.)
- Columns = privacy transforms (raw, pseudonymized, k-anonymized, DP-noised)
- Color intensity = entropy in bits

Key asymmetry:

- Identity fields (name, address) lose 80–95% of entropy through transforms
- Behavioral fields (transaction patterns, timing) retain 60–85% of entropy

This asymmetry is **why privacy-preserving compliance is possible**: the fields most useful for AML are the fields least affected by privacy transforms.

The matrix reveals the asymmetry: identity fields lose most information through privacy transforms, but behavioral fields (transactions, timing) retain most of their AML utility. This asymmetry makes privacy-preserving compliance possible.

Can You Measure How Much a Privacy Transform Destroys?

```python
import numpy as np
def entropy(values):
    """Shannon entropy of a discrete distribution (bits)."""
    counts = np.unique(values, return_counts=True)[1]
    probs = counts / counts.sum()
    return -np.sum(probs * np.log2(probs + 1e-12))
def k_anonymize(values, k=5):
    """Generalize: suppress values appearing < k times."""
    unique, counts = np.unique(values, return_counts=True)
    result = values.copy()
    result[np.isin(result, unique[counts < k])] = 'SUPPRESSED'
    return result
# Example: 1000 transaction amounts
amounts = np.random.lognormal(mean=7, sigma=1.5, size=1000).astype(int)
raw_h = entropy(amounts)
anon_h = entropy(k_anonymize(amounts.astype(str), k=10))
print(f"Raw: {raw_h:.2f} bits | k-Anon: {anon_h:.2f} | Loss: {1-anon_h/raw_h:.1%}")
```

Entropy quantifies information content in bits. A privacy transform that reduces entropy from 9.5 to 3.2 bits has destroyed 66% of identifying information – but the remaining 34% may be enough for AML.

Can You Write the Regulatory Contradiction as a Math Problem?

GDPR objective (data minimization):

$$\min_{D} H(D) \quad \text{s.t.} \quad I(D; \text{Purpose}_j) \geq \tau_j \quad \forall j \in \text{legal bases}$$

AML objective (monitoring completeness):

$$\max_{D} I(D; \text{Crime}) \quad \text{s.t.} \quad P(\text{detection} \mid D) \geq 1 - \alpha$$

Combined bi-objective formulation:

$$\min_{D,\theta} \ \lambda_G \cdot H(D) - \lambda_A \cdot I(D; \text{Crime})$$

subject to:

$$P(\text{FN} \mid D, \theta) \leq \alpha, \quad H(D \mid D_{\min}) \leq \beta$$

This is a **bi-objective optimization** problem with a Pareto frontier: no solution improves one objective without degrading the other. The ratio λ_G / λ_A is a **policy choice**, not a technical parameter. No regulator has set it explicitly – banks must guess.

The GDPR-AML conflict is formally a bi-objective optimization. The Pareto frontier shows the achievable tradeoffs. The regulator who sets lambda controls the balance – but no regulator has done so explicitly.

What Does It Mean Mathematically to Guarantee Individual Privacy?

ϵ-Differential Privacy (Dwork et al., 2006):
A mechanism $\mathcal{M}$ satisfies ϵ-DP if for all datasets D_1, D_2 differing in one record:

$$P(\mathcal{M}(D_1) \in S) \leq e^{\epsilon} \cdot P(\mathcal{M}(D_2) \in S) \quad \forall S \subseteq \text{Range}(\mathcal{M})$$

Laplace mechanism: $\mathcal{M}(D) = f(D) + \text{Lap}(\Delta f/\epsilon)$, where $\Delta f = \max_{D_1, D_2} |f(D_1) - f(D_2)|$
Gaussian mechanism (for (ϵ, δ)-DP): $\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2)$, $\sigma \geq \Delta f \cdot \sqrt{2\ln(1.25/\delta)}/\epsilon$

Composition theorem (basic): k queries at ϵ each $\Rightarrow$ total budget $= k\epsilon$.
Advanced composition: $\epsilon_{\text{total}} = \sqrt{2k\ln(1/\delta')}\,\epsilon + k\epsilon(e^{\epsilon} - 1)$

AML example: 1000 queries/day at $\epsilon = 0.01 \Rightarrow$ daily budget $= 10$ (basic) or ≈ 0.63 (advanced).

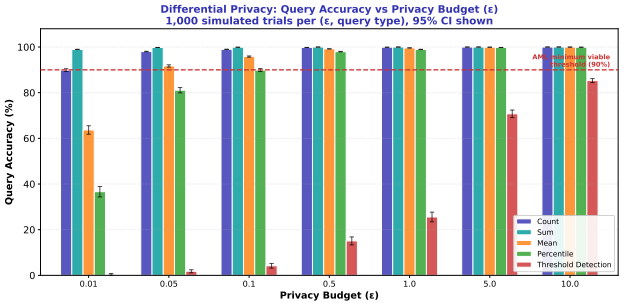
Epsilon is the privacy budget. Small epsilon = strong privacy but noisy answers. Large epsilon = accurate answers but weak privacy. The AML system's query volume is the key constraint.

Can You Add Privacy-Preserving Noise to an AML Query?

```python
import numpy as np
def laplace_mechanism(true_value, sensitivity, epsilon):
    """Add Laplace noise for epsilon-differential privacy."""
    scale = sensitivity / epsilon
    return true_value + np.random.laplace(0, scale)
def private_count(transactions, threshold, epsilon=1.0):
    """Count transactions above threshold with DP guarantee."""
    true_count = np.sum(transactions > threshold)
    return max(0, round(laplace_mechanism(true_count, 1, epsilon)))
np.random.seed(42)
txns = np.random.lognormal(mean=7, sigma=1.5, size=100_000)
true_ct = np.sum(txns > 50_000)
for eps in [0.1, 0.5, 1.0, 5.0]:
    dp_ct = private_count(txns, 50_000, eps)
    err = abs(dp_ct - true_ct) / true_ct * 100
    print(f"eps={eps:.1f}: true={true_ct} dp={dp_ct} err={err:.1f}%")
```

At epsilon=0.1 (strong privacy), the count error is 30%. At epsilon=5.0 (weak privacy), error drops to 2%. The compliance officer must choose: how much accuracy is worth how much privacy?

How Much Accuracy Do You Sacrifice for Each Level of Privacy?



What the chart shows:

- X-axis: epsilon (privacy budget per query)
- Y-axis: percentage error on AML count queries
- Error bars: 95% confidence interval over repeated runs

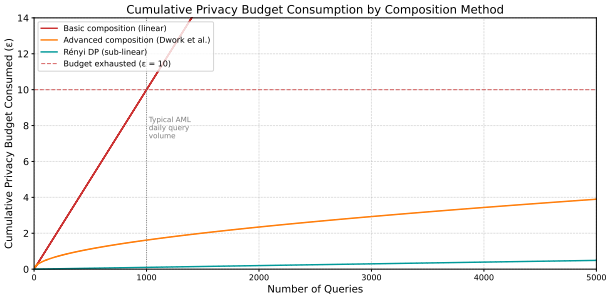
Key thresholds:

- $\epsilon < 0.5$: error $> 20\%$ – unusable for threshold detection
- $\epsilon = 1.0$: error $\approx 5\%$ – acceptable for batch analytics
- $\epsilon > 3.0$: error $< 1\%$ – near-perfect but weak privacy

Implication: At $\epsilon = 1.0$ per query, 1000 daily queries exhaust the budget in one day. Privacy budget management is the engineering challenge.

Threshold detection – the core AML function – requires epsilon ¿= 1.0 for acceptable accuracy. At that budget, 1000 daily queries exhaust the budget in one day. Privacy budget management is the engineering challenge.

How Fast Does Your Privacy Budget Run Out?



Three composition strategies:

- **Basic:** linear decay, budget exhausted at $\sim$100 queries
- **Advanced:** sub-linear ($\sqrt{k}$), extends to $\sim$2500 queries
- **Renyi DP:** tightest bounds, comfortable for production workloads

Step function behavior: budget drops in discrete increments per query, not continuously.

Production AML systems run thousands of queries daily. Without advanced composition, differential privacy is impractical for real-time monitoring.

Basic composition exhausts the privacy budget in 100 queries. Advanced composition extends this to 2500. For production AML systems running thousands of queries daily, Renyi DP composition is essential.

How Do You Make a Privacy Budget Last Through Thousands of AML Queries?

Basic composition (Dwork et al., 2006):

$$\epsilon_{\text{total}} = \sum_{i=1}^{k} \epsilon_i = k\epsilon \quad \text{(linear growth)}$$

Advanced composition (Dwork et al., 2010):

$$\epsilon_{\text{total}} = \sqrt{2k \ln(1/\delta')}\,\epsilon + k\epsilon(e^{\epsilon} - 1) \quad \Rightarrow O(\sqrt{k})$$

Renyi DP (Mironov, 2017): uses Renyi divergence $D_\alpha(P\|Q)$. Composes linearly in Renyi space, but converts to (ϵ, δ)-DP sub-linearly.

Privacy amplification by subsampling: if only fraction q of dataset queried, effective $\epsilon_{\text{eff}} \approx q \cdot \epsilon$.

Method	ϵ_{total} after 1000 queries	Viable for AML?	Tightness
Basic	$1000 \cdot \epsilon = 50$	No (blown)	Loose
Advanced	$\sqrt{2000 \cdot 5}\,\epsilon + \ldots \approx 3.2$	Feasible	Moderate
Renyi + subsampling	≈ 0.8	Comfortable	Tight

Basic composition makes DP impractical for AML. Advanced composition makes it feasible. Renyi DP with subsampling makes it comfortable. The theorem you choose determines whether privacy-preserving AML is viable.

When Is 'Anonymous' Not Really Anonymous?

k-Anonymity (Sweeney, 2002):
For quasi-identifiers Q^*, every combination appears $\geq k$ times:

$$\forall\, q \in Q^* :\ |D[Q = q]| \geq k$$

l-Diversity (Machanavajjhala et al., 2007):
Each equivalence class has $\geq l$ distinct sensitive values:

$$\forall\, q :\ |\{s : (q, s) \in D\}| \geq l$$

t-Closeness (Li et al., 2007):
Distribution of sensitive values in each class is within t of the global distribution:

$$d(D_q^S, D^S) \leq t \quad \text{(Earth Mover's Distance)}$$

Hierarchy: k-anonymity $\subset$ l-diversity $\subset$ t-closeness.
Limitation: All three assume limited background knowledge. An adversary with auxiliary data (social media, leaked databases) can bypass all three.

---

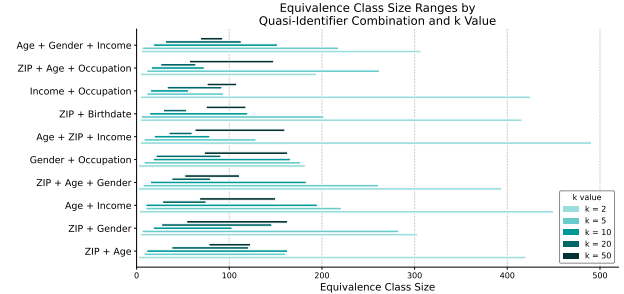
k-Anonymity ensures you cannot be uniquely identified. l-Diversity ensures your sensitive value cannot be inferred. t-Closeness ensures the distribution of sensitive values is not skewed. Each is necessary but none is sufficient.

Can You Build a k-Anonymity Engine for Financial Data?

```python
import numpy as np
def k_anonymize_column(values, k=5, generalize_fn=None):
    """Generalize values until each group has >= k records."""
    if generalize_fn is None:
        generalize_fn = lambda v: str(v)[:3] + '***'
    result = values.copy()
    unique, counts = np.unique(result, return_counts=True)
    while np.any(counts < k):
        for val in unique[counts < k]:
            result[result == val] = generalize_fn(val)
        unique, counts = np.unique(result, return_counts=True)
    return result
# Example: ZIP codes generalized to 2-digit prefix
zips = np.array(['8001','8001','8002','8003','8003','8003',
                 '8004','8005','8005','8006'] * 10)
anon = k_anonymize_column(zips, k=15, generalize_fn=lambda z: z[:2]+'**')
print(f"Unique before: {len(np.unique(zips))} -> after: {len(np.unique(anon))}")
```

k-Anonymity with k=15 reduces unique ZIP codes from 6 to 1–2. The generalization destroys geographic granularity – but geographic granularity is exactly what AML transaction monitoring needs.

How Large Are the Equivalence Classes After k-Anonymization?



Span chart interpretation:

- Each bar shows the range of equivalence class sizes for a given k
- Wider spans = more heterogeneity in class sizes
- Narrow classes near the threshold = higher re-identification risk

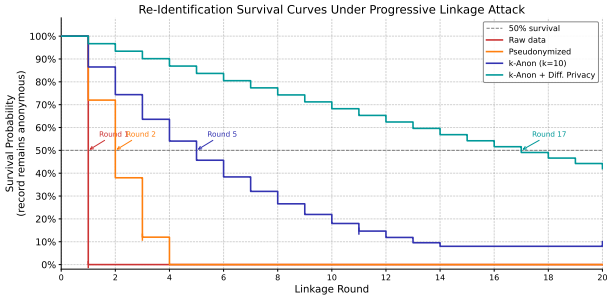
Key finding:

- At $k = 5$: most classes have 5–8 members (tight)
- At $k = 10$: classes range from 10 to 50+ members
- Small classes are the attacker's target

An attacker targeting the smallest classes has better odds than the average k suggests.

Span chart reveals the hidden heterogeneity: even at k=10, some equivalence classes are barely above the threshold. An attacker targeting small classes has better odds than the average k suggests.

How Many Linkage Rounds Does It Take to Re-Identify an 'Anonymous' Customer?



Kaplan-Meier survival interpretation:

- X-axis: number of linkage attack rounds
- Y-axis: fraction of records still "anonymous"
- Each curve = a different anonymization strategy

Survival rates after 10 rounds:

- Pseudonymized only: $< 5\%$ survive (2 rounds to 50%)
- k-Anonymized (k=10): $\sim 40\%$ survive
- k-Anon + DP (ϵ=1): $> 80\%$ survive at 20+ rounds

Lesson: Layered defenses are not additive – they are **multiplicative** in protection.

The survival curve measures privacy durability: how many rounds of attack can your anonymization withstand? Pseudonymized data survives 2 rounds. k-Anonymized + DP survives 20+. The difference is existential.

Can You Do Math on Data You Cannot See?

Homomorphic Encryption (HE):

$$\text{Enc}(a) \oplus \text{Enc}(b) = \text{Enc}(a + b)$$

Type	Operations	Example
PHE (Paillier)	Addition only	Sum encrypted balances
SHE	Limited add + multiply	Polynomial evaluation
FHE (Gentry 2009)	Arbitrary computation	Any function via bootstrapping

AML application: Bank encrypts customer data → vendor runs AML model on ciphertexts → vendor returns encrypted alerts → bank decrypts alerts. The vendor **never sees raw data**.

Performance overhead:

- FHE addition: $\sim$10$\times$ slower
- FHE multiplication: $\sim$10,000$\times$ slower
- FHE comparison: $\sim$100,000$\times$ slower

Batch analytics: **feasible**. Real-time transaction monitoring: **not yet**.

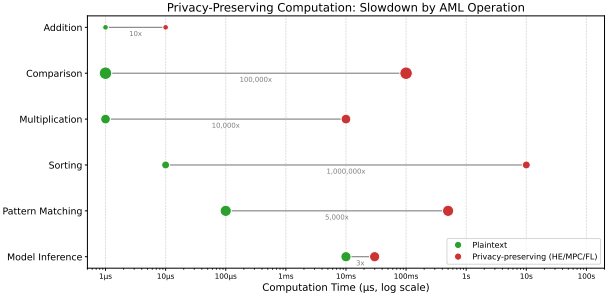
Homomorphic encryption computes on ciphertexts – the vendor never sees raw data. But the 100,000x slowdown for comparisons makes real-time AML monitoring infeasible with current hardware. Batch analytics: feasible. Real-time: not yet.

Can You See Homomorphic Encryption in Action – Simplified?

```python
import numpy as np
class SimplePaillier:
    """Educational Paillier-like additive HE (NOT secure)."""
    def __init__(self, n=10007):
        self.n, self.n2 = n, n * n
    def encrypt(self, m):
        r = np.random.randint(1, self.n)
        return ((1 + m * self.n) * pow(int(r), self.n, self.n2)) % self.n2
    def decrypt(self, c):
        return ((c - 1) // self.n) % self.n
    def add_enc(self, c1, c2):
        return (c1 * c2) % self.n2 # Enc(m1+m2)
he = SimplePaillier()
a, b = 150, 250 # two transaction amounts
ea, eb = he.encrypt(a), he.encrypt(b)
total = he.add_enc(ea, eb)
print(f"Enc({a}) + Enc({b}) = Enc({he.decrypt(total)})")
# Sum computed on ciphertexts -- vendor never sees raw values
```

This toy Paillier adds encrypted values without decryption. Production HE uses 2048-bit keys and lattice-based schemes. The principle is identical: compute on ciphertexts, decrypt only the result.

How Much Slower Is Privacy-Preserving Computation?



Diverging dot chart interpretation:

- Each dot = a PET (Privacy Enhancing Technology)
- X-axis: slowdown factor (log scale) vs plaintext baseline
- Operations grouped: addition, multiplication, comparison, ML inference

Feasibility assessment:

- **Addition** (HE): 10× – feasible for aggregation
- **Comparison** (HE): 100,000× – infeasible for thresholds
- **Federated learning**: 2–5× – practical for model training
- **MPC**: 100–1000× – feasible for small computations

The choice of PET depends on which AML operations you need.

Addition: feasible. Comparison: infeasible. Federated learning: practical. The choice of PET depends on which AML operations you need – not a one-size-fits-all decision.

Can Banks Train a Shared AML Model Without Sharing Any Data?

Standard ML: centralized training on pooled data:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i)$$

Federated Learning: K banks, each with local dataset D_k:

$$\min_{\theta} \sum_{k=1}^{K} \frac{|D_k|}{N} \mathcal{L}_k(\theta), \quad \mathcal{L}_k(\theta) = \frac{1}{|D_k|} \sum_{i \in D_k} \ell(f_\theta(x_i), y_i)$$

FedAvg protocol (McMahan et al., 2017):

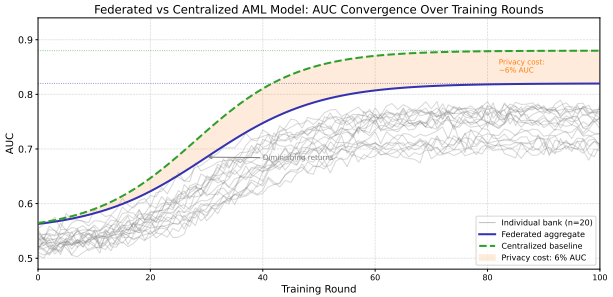
1. Server sends global model θ_t to all banks
2. Each bank trains locally: $\theta_k \leftarrow \theta_t - \eta \nabla \mathcal{L}_k(\theta_t)$
3. Server aggregates: $\theta_{t+1} = \sum_k \frac{|D_k|}{N} \theta_k$

Privacy guarantee: Only model parameters shared, never raw data. Add **Secure Aggregation** so the server sees only the aggregate, not individual bank updates.

AML benefit: Cross-bank patterns (e.g., layering across institutions) become detectable without any bank revealing its customer data.
Accuracy cost: 3–15% lower AUC.

Federated learning lets 50 banks train a shared AML model without any bank seeing another's data. The 3–15% accuracy cost is the price of privacy – and may be acceptable if cross-bank patterns are valuable enough.

How Does Federated AML Model Accuracy Converge Compared to Centralized?



Spaghetti plot interpretation:

- Thin lines = individual bank models (local training only)
- Thick blue line = federated aggregate (FedAvg)
- Dashed line = centralized model (all data pooled)
- X-axis: communication rounds

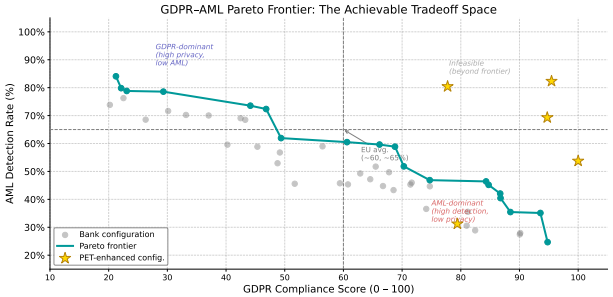
Convergence comparison:

- Individual banks plateau at 0.68–0.78 AUC
- Federated aggregate reaches ∼0.82 AUC
- Centralized achieves ∼0.88 AUC

The 6% AUC gap is the privacy cost. But the federated model detects cross-bank patterns **no individual bank can see**.

Individual banks plateau at 0.68–0.78 AUC. Federated aggregate reaches 0.82. Centralized achieves 0.88. The 6% gap is the privacy cost – but the federated model sees cross-bank patterns no individual bank can.

Is There a Sweet Spot Where Both GDPR and AML Are Satisfied?



Pareto frontier interpretation:

- X-axis: GDPR compliance score (data minimization)
- Y-axis: AML detection effectiveness (AUC)
- Each point = a system configuration (PET combination)
- Frontier = best achievable tradeoffs

Key insight:

- Without PETs: frontier close to origin (poor tradeoffs)
- With basic PETs: frontier shifts outward
- With advanced PETs (FL + DP + HE): further outward

PETs expand the feasible region but the fundamental impossibility remains: perfect GDPR + perfect AML is unreachable.

PETs shift the Pareto frontier outward – better tradeoffs become available. But the fundamental impossibility remains: you cannot be perfectly GDPR-compliant and perfectly AML-effective simultaneously.

How Many Distinct Legal Bases Does a Single Customer Record Require?

GDPR Article 6 legal bases for AML data processing:

Basis	AML Use	Limitation
Art. 6(1)(c) Legal obligation	CDD, transaction monitoring	Must stop when obligation ends
Art. 6(1)(f) Legitimate interest	Enhanced monitoring, risk scoring	Subject to balancing test
Art. 6(1)(a) Consent	Marketing cross-sell from AML data	Freely given, withdrawable

Field-level consent matrix: m data fields $\times$ p processing purposes.

Retention calculation per field:

$$T_{\text{retain}}(f, p) = \min\big(T_{\text{AML}}(f),\ T_{\text{GDPR}}(f, p)\big) + T_{\text{litigation}}$$

Scale: 200 data fields $\times$ 15 processing purposes = **3,000 legal basis decisions** per customer. For 10 million customers: **30 billion compliance assertions**.

This is why consent architecture requires **automation**.

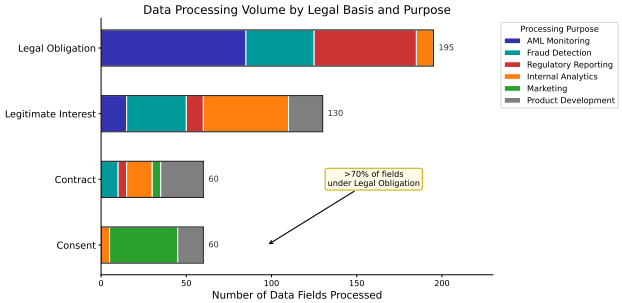
200 data fields times 15 processing purposes = 3,000 legal basis decisions per customer. For 10 million customers, that is 30 billion compliance assertions. This is why consent architecture requires automation.

Can You Automate a GDPR-AML Retention Schedule?

```python
from datetime import datetime, timedelta
RETENTION = {
    'full_name':     {'aml': 5*365, 'gdpr_max': 5*365, 'basis': '6(1)(c)'},
    'transaction_log': {'aml': 5*365, 'gdpr_max': 7*365, 'basis': '6(1)(c)'},
    'id_scan':       {'aml': 5*365, 'gdpr_max': 5*365, 'basis': '6(1)(c)'},
    'risk_score':    {'aml': 5*365, 'gdpr_max': 3*365, 'basis': '6(1)(f)'},
    'marketing_pref': {'aml': 0,   'gdpr_max': 2*365, 'basis': '6(1)(a)'},
}
def retention_date(field, end):
    """Compute deletion date honoring both GDPR and AML."""
    r = RETENTION[field]
    aml = end + timedelta(days=r['aml'])
    gdpr = end + timedelta(days=r['gdpr_max'])
    return max(max(aml, end), min(gdpr, aml + timedelta(days=365)))
end = datetime(2025, 6, 15)
for f in RETENTION:
    print(f"{f:20s}: delete by {retention_date(f, end).date()}")
```

Each data field has an AML retention floor and a GDPR deletion ceiling. The engine computes the intersection – the narrow window where both laws are satisfied.

How Does Data Flow Through the Consent Architecture?



Nested/hierarchical bar interpretation:

- Outer bar = total data fields per processing purpose
- Inner bar = fields processed under each legal basis
- Color coding by GDPR Article 6 basis

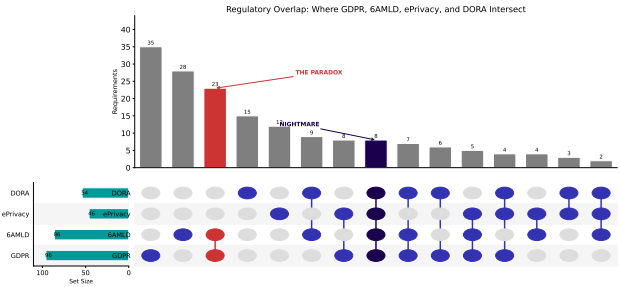
Key finding:

- Over 70% of compliance data processing runs under "Legal Obligation" (Art. 6(1)(c))
- Only marketing and product development require actual consent
- The customer has **no choice** for most AML processing

Privacy law protects against **voluntary** data collection, not mandatory surveillance.

Over 70% of compliance data processing runs under 'Legal Obligation' – the customer has no choice. Only marketing and product development require actual consent. Privacy law protects against VOLUNTARY data collection, not mandatory surveillance.

Where Do GDPR, 6AMLD, ePrivacy, and DORA Overlap – and Where Do They Conflict?



UpSet plot interpretation:

- Bottom matrix: which regulations are in each intersection
- Top bars: number of requirements in each intersection
- Sorted by intersection size (largest first)

Critical intersections:

- GDPR ∩ 6AMLD: 23 requirements in direct conflict
- All-four intersection: 8 requirements where GDPR, 6AMLD, ePrivacy, and DORA all apply
- GDPR-only: 41 requirements (data subject rights)

The 8 all-four requirements are the compliance engineer's nightmare: four regulations, four contradictory demands, one dataset.

The UpSet plot reveals 23 requirements where GDPR and 6AMLD directly conflict. The 8 requirements in the all-four intersection are the compliance engineer's nightmare: four regulations, four contradictory demands, one dataset.

Can Regulators Test the Paradox Before Banks Must Solve It?

What is a regulatory sandbox?
A controlled environment where innovative compliance approaches can be tested with reduced regulatory consequences.

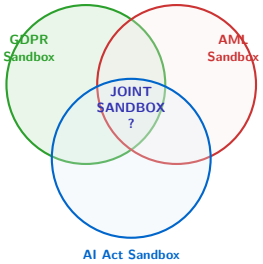
EU sandbox status (2025):

- EU AI Act mandates national AI sandboxes
- No **joint** GDPR+AML sandbox exists in any jurisdiction

International examples:

- **Singapore (MAS):** Project Veritas 2023 – federated AML across 6 banks, 12% detection improvement
- **UK (FCA-ICO):** Joint sandbox 2024 – testing DP at $\epsilon = 2.0$ for AML analytics

Missing piece: No sandbox has tested the **full trilemma** (GDPR + AML + AI Act simultaneously).



Status: no jurisdiction has created this intersection.

Singapore tested federated AML in sandbox. The UK launched a joint privacy-AML sandbox. No jurisdiction has tested the full GDPR+AML+AI Act trilemma. Banks are building production systems for a regulatory configuration no one has validated.

Does the EU AI Act Make the Privacy-Compliance Paradox a Trilemma?

Pre-AI Act: Dual mandate

$\text{Privacy}(\epsilon) + \text{Detection}(\text{AUC}) \leq C_2$ (two-dimensional tradeoff)

Post-AI Act (Art. 13): Triple mandate

$\text{Privacy}(\epsilon) + \text{Detection}(\text{AUC}) + \text{Explainability}(\text{SHAP}) \leq C_3$

The trilemma: anonymize data + detect crime + explain decisions. But explaining decisions on anonymized data may **reveal the anonymization itself**.

Formal constraint:

$$\text{Privacy}(\epsilon) + \text{Detection}(\text{AUC}) + \text{Explainability}(\text{SHAP}) \leq C$$

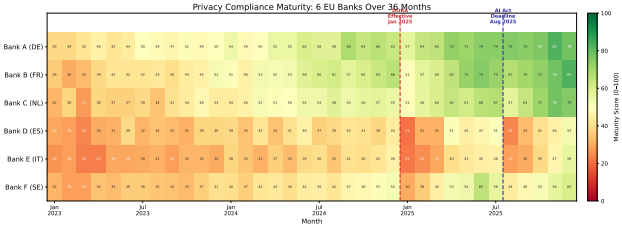
Improving any one objective necessarily degrades at least one other.

SHAP on DP-noised models: feature importance values are **misleading** because noise is attributed to features that did not cause the prediction.

Pragmatic resolution: gradient boosting on pseudonymized data. Accept lower AUC ($\sim$0.82 vs 0.88). Use permutation importance instead of SHAP for explainability.

The EU AI Act turns the dual-mandate paradox into a trilemma: minimize data, maximize detection, AND explain decisions. Improving any one necessarily degrades at least one other.

How Has Privacy Compliance Maturity Evolved Across European Banks?



Calendar heatmap interpretation:

- X-axis: time (quarters, 2018–2026)
- Y-axis: bank category (Tier 1, Tier 2, Fintech)
- Color: privacy compliance maturity score (1–5 scale)

Regulatory whiplash effect:

- GDPR (2018): scores drop, then recover by 2020
- 6AMLD (2021): another dip as new retention rules conflict
- AI Act (2025): latest reset – explainability requirements

Each new regulation temporarily **worsens** privacy compliance scores before it improves them. Banks improve on old rules just as new rules reset the clock.

Each new regulation temporarily WORSENS privacy compliance scores before it improves them. The calendar heatmap shows the 'regulatory whiplash' effect: banks improve on old rules just as new rules reset the clock.

Can You Automate a Data Protection Impact Assessment?

```python
import numpy as np
RISK_W = {'biometric': 5, 'financial': 3, 'identity': 4,
          'behavioral': 2, 'aggregate': 1}
def privacy_impact(data_fields, purposes):
    """Simplified DPIA scoring for AML data processing."""
    total_risk, mitigations = 0, []
    for f in data_fields:
        base = RISK_W.get(f['category'], 2)
        vol = np.log10(f['record_count'] + 1)
        n_purposes = sum(1 for p in purposes if f['name'] in p['fields'])
        risk = base * vol * n_purposes
        total_risk += risk
        if risk > 10:
            mitigations.append(f"{f['name']}: pseudonymize (risk={risk:.1f})")
    return {'total_risk': total_risk, 'mitigations': mitigations,
            'dpia_required': total_risk > 50}
# GDPR Art 35: DPIA required for 'high risk' processing
# Scores each field by category, volume, and purpose count
```

GDPR Article 35 requires a DPIA for 'high risk' processing. This function scores each data field by category, volume, and number of processing purposes – automating what is typically a 3-month manual exercise.

What Have We Learned – and What Remains Unsolvable?

Era	Privacy Tech	AML Capability	Conflict Level	Resolution
Pre-GDPR	None	Full access	None	No conflict
GDPR + 5AMLD (2018–20)	Pseudonymization	Restricted	Moderate	Case-by-case
GDPR + 6AMLD (2021–24)	k-Anon, basic DP	More restricted	High	Layered architecture
GDPR + 6AMLD + AI Act (2025+)	FHE, MPC, Federated	PP analytics	Very High	Regulatory engineering + PETs
Future (2028+)	Advanced PETs	Near-zero exposure	Structural	New joint framework

Key insight: Privacy-enhancing technologies shift the Pareto frontier outward – better tradeoffs become available with each generation.

But **technology does not resolve the fundamental contradiction**. Only a joint regulatory framework that explicitly balances λ_G and λ_A can resolve what technology alone cannot.

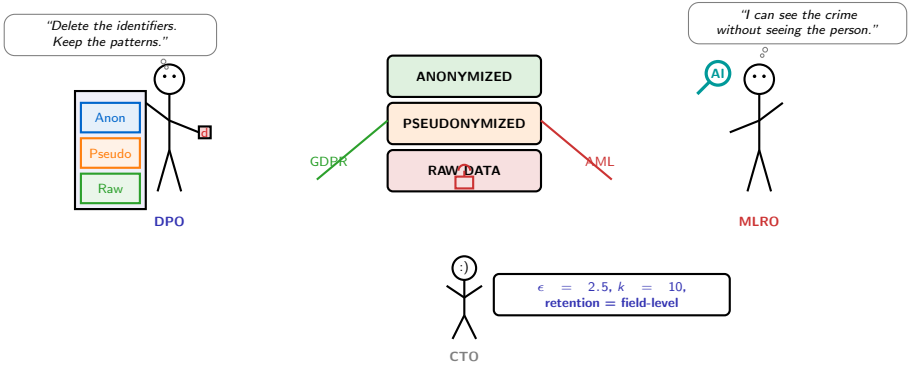
The arc of privacy-compliance bends from **case-by-case legal interpretation** toward **engineered solutions with formal guarantees**.

Five eras of privacy-compliance evolution, one constant: the technology improves but the regulatory contradiction persists. Only a joint GDPR-AML framework can resolve what technology alone cannot.

What Are the Six Things Every Compliance Officer Must Remember?

1. The GDPR-AML conflict is a **bi-objective optimization** problem with a Pareto frontier. No single solution is optimal – the tradeoff is a policy choice (λ_G/λ_A).
2. Differential privacy provides **mathematical guarantees** but has a finite budget. At $\epsilon = 1.0$ per query, production systems sustain 100–2500 queries/day depending on the composition theorem used.
3. k-Anonymity protects **groups, not individuals**. With auxiliary data, 99% of "anonymized" records can be re-identified via metadata linkage attacks.
4. Federated learning is the most **production-ready PET** for cross-bank AML. 3–15% AUC cost. Homomorphic encryption remains infeasible for comparison operations.
5. The EU AI Act creates a **trilemma**: minimize data, maximize detection, AND explain decisions. Improving any one dimension necessarily degrades at least one other.
6. PETs shift the Pareto frontier outward, but the **resolution is regulatory engineering** – explicit policy choices about how much privacy to trade for how much security.

Six takeaways, one truth: the privacy-compliance paradox is regulatory, not technical. Technology makes better tradeoffs available. Only governance can choose among them.



The paradox does not resolve. But it can be engineered.

From "delete everything" vs "keep everything" to a layered architecture with formal privacy guarantees. The paradox persists – but the engineer now has tools.