When Law Meets Code:
Technical Hurdles in Implementing AI Regulation

Benjamin Guedj
University College London and Inria

Québec–Oxford–France Workshop
September 2025

This talk

- Aim: translate regulatory *principles* into practices grounded in ML reality.
- Perspective: a machine learning researcher highlighting where **law meets code** (not a legal exegesis).
- Takeaway: credible regulation needs methods that are auditable, scalable, and adaptive.

This talk

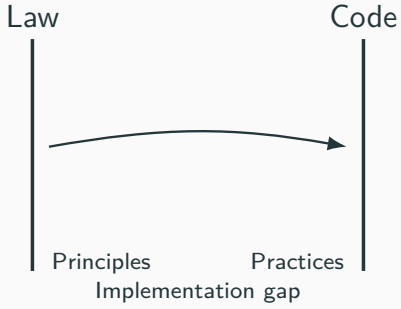
- Aim: translate regulatory *principles* into practices grounded in ML reality.
- Perspective: a machine learning researcher highlighting where **law meets code** (not a legal exegesis).
- Takeaway: credible regulation needs methods that are auditable, scalable, and adaptive.

This talk

- Aim: translate regulatory *principles* into practices grounded in ML reality.
- Perspective: a machine learning researcher highlighting where **law meets code** (not a legal exegesis).
- Takeaway: credible regulation needs methods that are auditable, scalable, and adaptive.

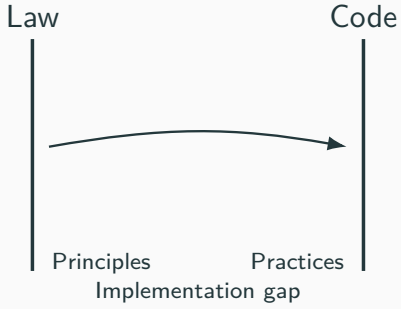
The regulation–reality gap

From principles to practice



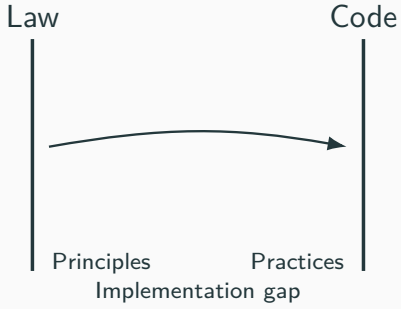
- Principles widely endorsed: transparency, fairness, safety, accountability.
- Implementation is hard: systems are complex and constantly changing.
- "Explainability" can mean many things (saliency maps, counterfactuals, feature attributions).
- Tech evolves faster than static rules $\Rightarrow$ risk of vagueness or unworkable laws.

From principles to practice



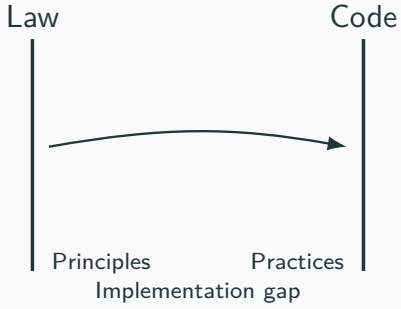
- Principles widely endorsed: transparency, fairness, safety, accountability.
- Implementation is hard: systems are complex and constantly changing.
- "Explainability" can mean many things (saliency maps, counterfactuals, feature attributions).
- Tech evolves faster than static rules $\Rightarrow$ risk of vagueness or unworkable laws.

From principles to practice



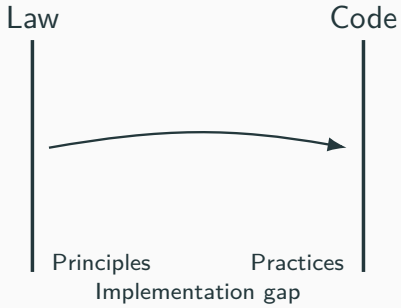
- Principles widely endorsed: transparency, fairness, safety, accountability.
- Implementation is hard: systems are complex and constantly changing.
- "Explainability" can mean many things (saliency maps, counterfactuals, feature attributions).
- Tech evolves faster than static rules ⇒ risk of vagueness or unworkable laws.

From principles to practice



- Principles widely endorsed: transparency, fairness, safety, accountability.
- Implementation is hard: systems are complex and constantly changing.
- "Explainability" can mean many things (saliency maps, counterfactuals, feature attributions).
- Tech evolves faster than static rules $\Rightarrow$ risk of vagueness or unworkable laws.

From principles to practice

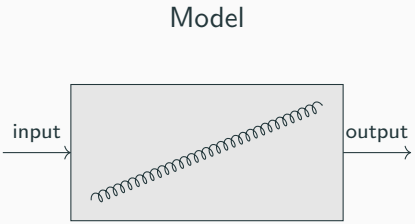


- Principles widely endorsed: transparency, fairness, safety, accountability.
- Implementation is hard: systems are complex and constantly changing.
- "Explainability" can mean many things (saliency maps, counterfactuals, feature attributions).
- Tech evolves faster than static rules $\Rightarrow$ risk of vagueness or unworkable laws.

Implication for regulation: definitions should be flexible and tied to technical feasibility.

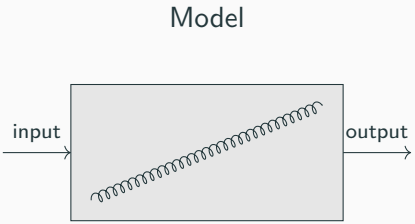
Core implementation challenges

1) Transparency & explainability



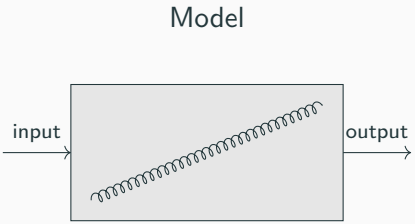
- Deep models often behave as black boxes.
- Explanations serve different goals: debugging, accountability, user understanding.
- Risk of "tick-box" explanations that create paperwork, not insight.
- What helps: simple baselines, model/data cards, counterfactual examples (*what would change the decision?*).

1) Transparency & explainability



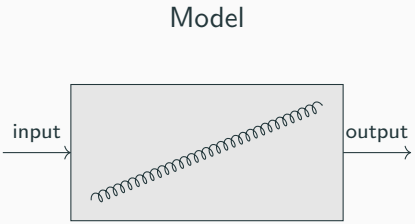
- Deep models often behave as black boxes.
- Explanations serve different goals: debugging, accountability, user understanding.
- Risk of "tick-box" explanations that create paperwork, not insight.
- What helps: simple baselines, model/data cards, counterfactual examples (*what would change the decision?*).

1) Transparency & explainability



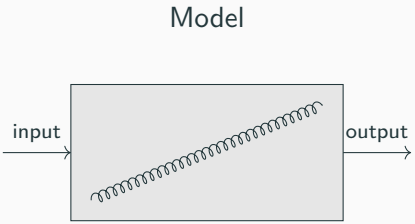
- Deep models often behave as black boxes.
- Explanations serve different goals: debugging, accountability, user understanding.
- Risk of “tick-box” explanations that create paperwork, not insight.
- What helps: simple baselines, model/data cards, counterfactual examples (*what would change the decision?*).

1) Transparency & explainability



- Deep models often behave as black boxes.
- Explanations serve different goals: debugging, accountability, user understanding.
- Risk of “tick-box” explanations that create paperwork, not insight.
- What helps: simple baselines, model/data cards, counterfactual examples (*what would change the decision?*).

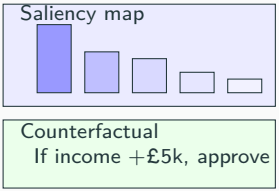
1) Transparency & explainability



- Deep models often behave as black boxes.
- Explanations serve different goals: debugging, accountability, user understanding.
- Risk of "tick-box" explanations that create paperwork, not insight.
- What helps: simple baselines, model/data cards, counterfactual examples (*what would change the decision?*).

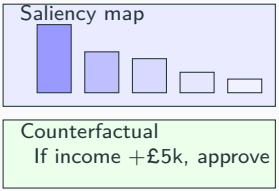
Implication for regulation: require clarity of purpose for explanations, not one-size-fits-all.

Transparency example: loan approvals



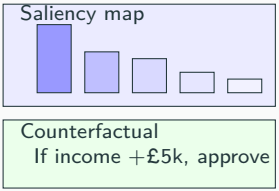
- Question: *Why was I refused?*
- Saliency view (*which inputs mattered most*) vs counterfactual (*what minimal change flips the decision*).
- Each serves different users: developer vs auditor vs affected person.
- Choose the tool that matches the accountability goal.

Transparency example: loan approvals



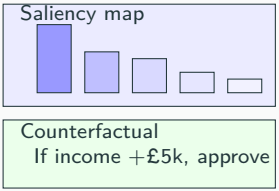
- Question: *Why was I refused?*
- Saliency view (*which inputs mattered most*) vs counterfactual (*what minimal change flips the decision*).
- Each serves different users: developer vs auditor vs affected person.
- Choose the tool that matches the accountability goal.

Transparency example: loan approvals



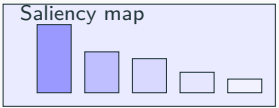
- Question: *Why was I refused?*
- Saliency view (*which inputs mattered most*) vs counterfactual (*what minimal change flips the decision*).
- Each serves different users: developer vs auditor vs affected person.
- Choose the tool that matches the accountability goal.

Transparency example: loan approvals



- Question: *Why was I refused?*
- Saliency view (*which inputs mattered most*) vs counterfactual (*what minimal change flips the decision*).
- Each serves different users: developer vs auditor vs affected person.
- Choose the tool that matches the accountability goal.

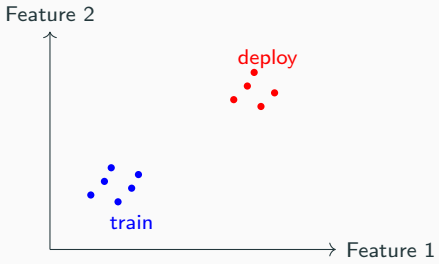
Transparency example: loan approvals



- Question: *Why was I refused?*
- Saliency view (*which inputs mattered most*) vs counterfactual (*what minimal change flips the decision*).
- Each serves different users: developer vs auditor vs affected person.
- Choose the tool that matches the accountability goal.

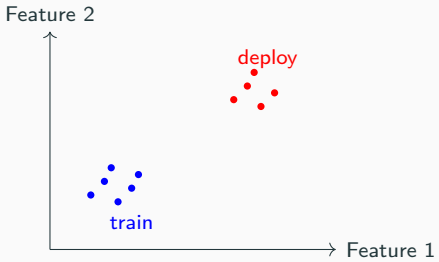
Implication for regulation: allow multiple explanation types mapped to stakeholder needs.

2) Robustness & distribution shift



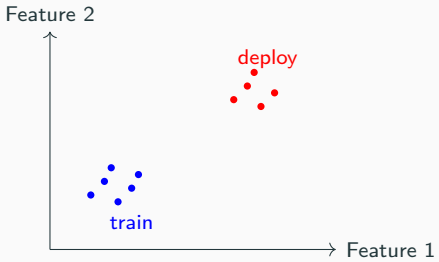
- Deployment data often differ from training data (*distribution shift*).
- Performance can degrade silently without monitoring.
- Robustness trades off with accuracy and cost.
- What helps: shift-aware validation, stress tests, online monitoring, rollbacks.

2) Robustness & distribution shift



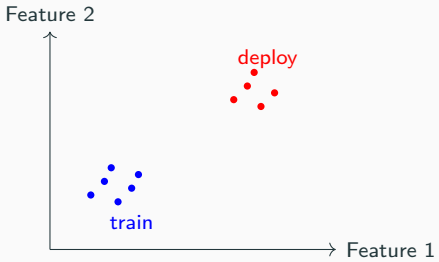
- Deployment data often differ from training data (*distribution shift*).
- Performance can degrade silently without monitoring.
- Robustness trades off with accuracy and cost.
- What helps: shift-aware validation, stress tests, online monitoring, rollbacks.

2) Robustness & distribution shift



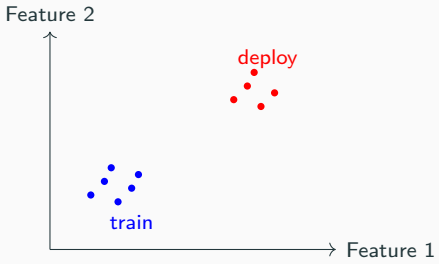
- Deployment data often differ from training data (*distribution shift*).
- Performance can degrade silently without monitoring.
- Robustness trades off with accuracy and cost.
- What helps: shift-aware validation, stress tests, online monitoring, rollbacks.

2) Robustness & distribution shift



- Deployment data often differ from training data (*distribution shift*).
- Performance can degrade silently without monitoring.
- Robustness trades off with accuracy and cost.
- What helps: shift-aware validation, stress tests, online monitoring, rollbacks.

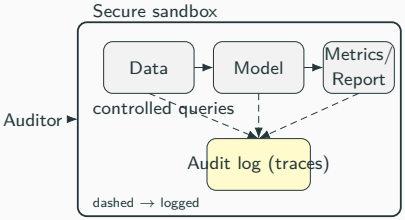
2) Robustness & distribution shift



- Deployment data often differ from training data (*distribution shift*).
- Performance can degrade silently without monitoring.
- Robustness trades off with accuracy and cost.
- What helps: shift-aware validation, stress tests, online monitoring, rollbacks.

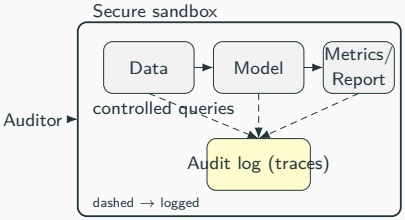
Implication for regulation: mandate stress testing under realistic shifts and post-deployment monitoring.

3) Auditing & verification



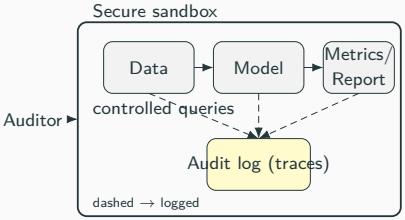
- Unlike finance, there is not yet a well-established audit framework for AI systems.
- Challenge: verifying systems without exposing private data or company secrets.
- Credible audits require clear scope, repeatable tests, and evidence trails.
- What helps: independent sandboxes, red-team exercises, and audit logs.

3) Auditing & verification



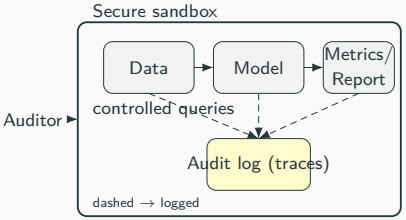
- Unlike finance, there is not yet a well-established audit framework for AI systems.
- Challenge: verifying systems without exposing private data or company secrets.
- Credible audits require clear scope, repeatable tests, and evidence trails.
- What helps: independent sandboxes, red-team exercises, and audit logs.

3) Auditing & verification



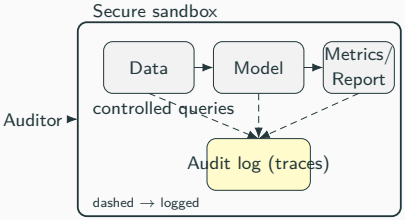
- Unlike finance, there is not yet a well-established audit framework for AI systems.
- Challenge: verifying systems without exposing private data or company secrets.
- Credible audits require clear scope, repeatable tests, and evidence trails.
- What helps: independent sandboxes, red-team exercises, and audit logs.

3) Auditing & verification



- Unlike finance, there is not yet a well-established audit framework for AI systems.
- Challenge: verifying systems without exposing private data or company secrets.
- Credible audits require clear scope, repeatable tests, and evidence trails.
- What helps: independent sandboxes, red-team exercises, and audit logs.

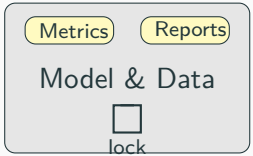
3) Auditing & verification



- Unlike finance, there is not yet a well-established audit framework for AI systems.
- Challenge: verifying systems without exposing private data or company secrets.
- Credible audits require clear scope, repeatable tests, and evidence trails.
- What helps: independent sandboxes, red-team exercises, and audit logs.

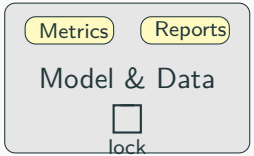
Implication for regulation: audits should use shared, transparent methods—not improvised case by case.

Auditing example: face recognition



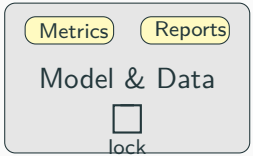
- Auditor needs: datasets (with consent), model artefacts, evaluation scripts, metrics.
- Barriers: privacy, IP, and security constraints.
- Practical compromise: secure sandboxes with controlled queries + signed artefact trails.
- Outcome: reproducible findings without exposing sensitive assets.

Auditing example: face recognition



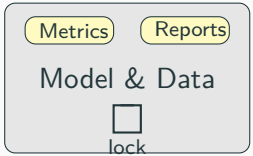
- Auditor needs: datasets (with consent), model artefacts, evaluation scripts, metrics.
- Barriers: privacy, IP, and security constraints.
- Practical compromise: secure sandboxes with controlled queries + signed artefact trails.
- Outcome: reproducible findings without exposing sensitive assets.

Auditing example: face recognition



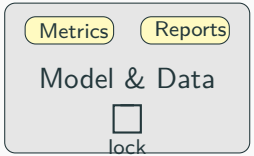
- Auditor needs: datasets (with consent), model artefacts, evaluation scripts, metrics.
- Barriers: privacy, IP, and security constraints.
- Practical compromise: secure sandboxes with controlled queries + signed artefact trails.
- Outcome: reproducible findings without exposing sensitive assets.

Auditing example: face recognition



- Auditor needs: datasets (with consent), model artefacts, evaluation scripts, metrics.
- Barriers: privacy, IP, and security constraints.
- Practical compromise: secure sandboxes with controlled queries + signed artefact trails.
- Outcome: reproducible findings without exposing sensitive assets.

Auditing example: face recognition



- Auditor needs: datasets (with consent), model artefacts, evaluation scripts, metrics.
- Barriers: privacy, IP, and security constraints.
- Practical compromise: secure sandboxes with controlled queries + signed artefact trails.
- Outcome: reproducible findings without exposing sensitive assets.

Implication for regulation: enable trusted audit environments balancing access and confidentiality.

4) Bias & fairness

- Competing definitions: demographic parity, equalised odds, calibration.
- Context matters: healthcare vs hiring may prioritise different harms.
- Fixes can shift errors between groups or reduce overall accuracy.
- What helps: pre-specify targets, publish trade-offs, involve domain experts.

4) Bias & fairness

- Competing definitions: demographic parity, equalised odds, calibration.
- Context matters: healthcare vs hiring may prioritise different harms.
- Fixes can shift errors between groups or reduce overall accuracy.
- What helps: pre-specify targets, publish trade-offs, involve domain experts.

4) Bias & fairness

- Competing definitions: demographic parity, equalised odds, calibration.
- Context matters: healthcare vs hiring may prioritise different harms.
- Fixes can shift errors between groups or reduce overall accuracy.
- What helps: pre-specify targets, publish trade-offs, involve domain experts.

4) Bias & fairness

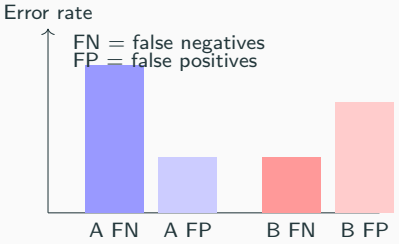
- Competing definitions: demographic parity, equalised odds, calibration.
- Context matters: healthcare vs hiring may prioritise different harms.
- Fixes can shift errors between groups or reduce overall accuracy.
- What helps: pre-specify targets, publish trade-offs, involve domain experts.

4) Bias & fairness

- Competing definitions: demographic parity, equalised odds, calibration.
- Context matters: healthcare vs hiring may prioritise different harms.
- Fixes can shift errors between groups or reduce overall accuracy.
- What helps: pre-specify targets, publish trade-offs, involve domain experts.

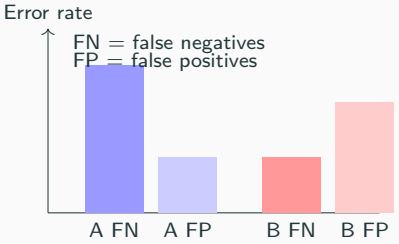
Implication for regulation: fairness must be context-specific and transparent about trade-offs.

Fairness example: diagnosis trade-off



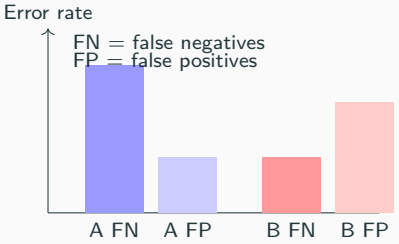
- Goal: reduce missed diagnoses (false negatives) across groups.
- Side-effect: more false positives in some groups.
- Policy choice: state priorities explicitly; monitor impacts over time.
- Communicate trade-offs to stakeholders.

Fairness example: diagnosis trade-off



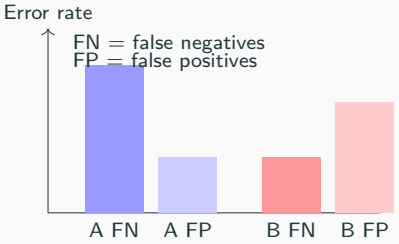
- Goal: reduce missed diagnoses (false negatives) across groups.
- Side-effect: more false positives in some groups.
- Policy choice: state priorities explicitly; monitor impacts over time.
- Communicate trade-offs to stakeholders.

Fairness example: diagnosis trade-off



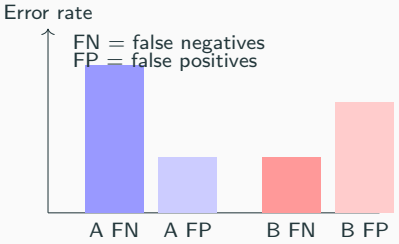
- Goal: reduce missed diagnoses (false negatives) across groups.
- Side-effect: more false positives in some groups.
- Policy choice: state priorities explicitly; monitor impacts over time.
- Communicate trade-offs to stakeholders.

Fairness example: diagnosis trade-off



- Goal: reduce missed diagnoses (false negatives) across groups.
- Side-effect: more false positives in some groups.
- Policy choice: state priorities explicitly; monitor impacts over time.
- Communicate trade-offs to stakeholders.

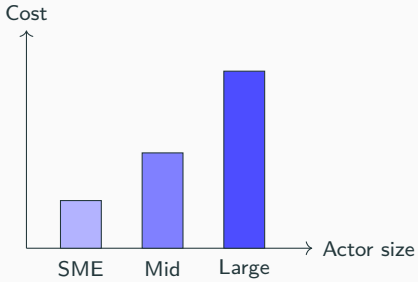
Fairness example: diagnosis trade-off



- Goal: reduce missed diagnoses (false negatives) across groups.
- Side-effect: more false positives in some groups.
- Policy choice: state priorities explicitly; monitor impacts over time.
- Communicate trade-offs to stakeholders.

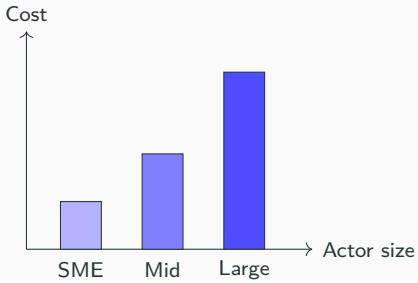
Implication for regulation: require disclosure of fairness targets and observed trade-offs.

5) Scalability of compliance



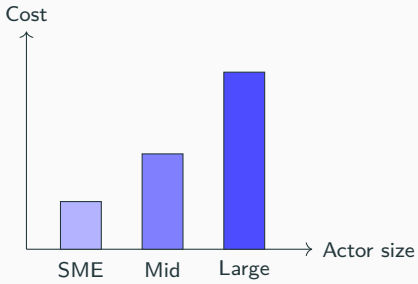
- Compliance costs grow with model size, datasets, and documentation.
- Large firms can absorb costs; SMEs and open-source may struggle.
- Risk: entrench incumbents, reduce innovation diversity.
- What helps: proportionate obligations, shared tooling, templates, open benchmarks.

5) Scalability of compliance



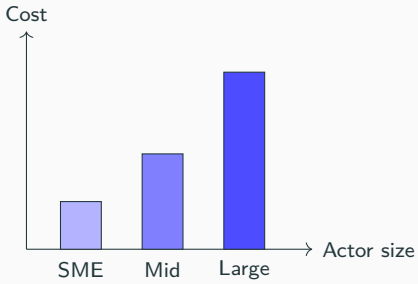
- Compliance costs grow with model size, datasets, and documentation.
- Large firms can absorb costs; SMEs and open-source may struggle.
- Risk: entrench incumbents, reduce innovation diversity.
- What helps: proportionate obligations, shared tooling, templates, open benchmarks.

5) Scalability of compliance



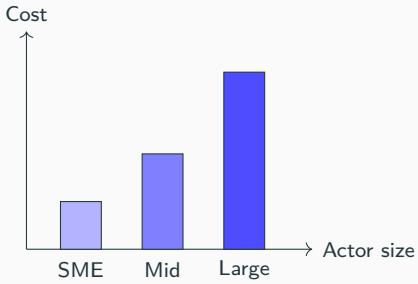
- Compliance costs grow with model size, datasets, and documentation.
- Large firms can absorb costs; SMEs and open-source may struggle.
- Risk: entrench incumbents, reduce innovation diversity.
- What helps: proportionate obligations, shared tooling, templates, open benchmarks.

5) Scalability of compliance



- Compliance costs grow with model size, datasets, and documentation.
- Large firms can absorb costs; SMEs and open-source may struggle.
- Risk: entrench incumbents, reduce innovation diversity.
- What helps: proportionate obligations, shared tooling, templates, open benchmarks.

5) Scalability of compliance



- Compliance costs grow with model size, datasets, and documentation.
- Large firms can absorb costs; SMEs and open-source may struggle.
- Risk: entrench incumbents, reduce innovation diversity.
- What helps: proportionate obligations, shared tooling, templates, open benchmarks.

Implication for regulation: ensure proportionality by actor size/risk.

Implications & collaboration

Implications for regulation

- Balance ambition vs feasibility.
- Focus on outcomes/properties (robustness, monitoring) over brittle checklists.
- Encourage iteration: phased obligations, sandboxes, post-deployment monitoring.
- Prioritise transparency of processes and evidence rather than a single explanatory method.

Implications for regulation

- Balance ambition vs feasibility.
- Focus on outcomes/properties (robustness, monitoring) over brittle checklists.
- Encourage iteration: phased obligations, sandboxes, post-deployment monitoring.
- Prioritise transparency of processes and evidence rather than a single explanatory method.

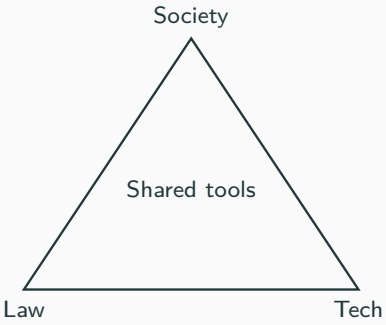
Implications for regulation

- Balance ambition vs feasibility.
- Focus on outcomes/properties (robustness, monitoring) over brittle checklists.
- Encourage iteration: phased obligations, sandboxes, post-deployment monitoring.
- Prioritise transparency of processes and evidence rather than a single explanatory method.

Implications for regulation

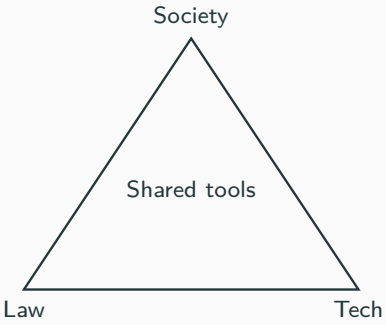
- Balance ambition vs feasibility.
- Focus on outcomes/properties (robustness, monitoring) over brittle checklists.
- Encourage iteration: phased obligations, sandboxes, post-deployment monitoring.
- Prioritise transparency of processes and evidence rather than a single explanatory method.

Opportunities for joint work



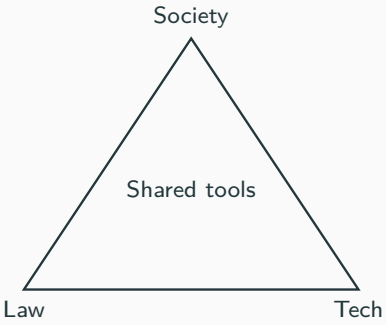
- Shared evaluation assets: stress-test suites, fairness benchmarks, auditing playbooks.
- Certification pathways modelled on safety-critical domains with adaptive updates.
- Research–policy interfaces: fellowships, residencies, “translator” roles.
- Public-interest compute/data access for regulators, academia, SMEs.

Opportunities for joint work



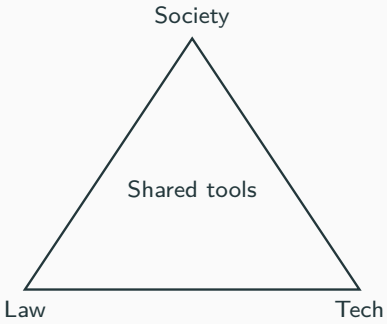
- Shared evaluation assets: stress-test suites, fairness benchmarks, auditing playbooks.
- Certification pathways modelled on safety-critical domains with adaptive updates.
- Research–policy interfaces: fellowships, residencies, “translator” roles.
- Public-interest compute/data access for regulators, academia, SMEs.

Opportunities for joint work



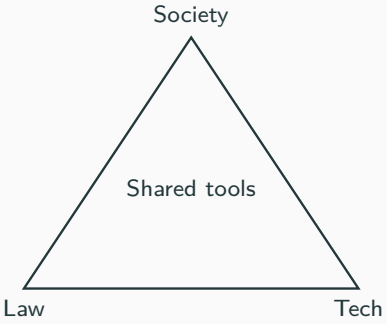
- Shared evaluation assets: stress-test suites, fairness benchmarks, auditing playbooks.
- Certification pathways modelled on safety-critical domains with adaptive updates.
- Research–policy interfaces: fellowships, residencies, “translator” roles.
- Public-interest compute/data access for regulators, academia, SMEs.

Opportunities for joint work



- Shared evaluation assets: stress-test suites, fairness benchmarks, auditing playbooks.
- Certification pathways modelled on safety-critical domains with adaptive updates.
- Research–policy interfaces: fellowships, residencies, “translator” roles.
- Public-interest compute/data access for regulators, academia, SMEs.

Opportunities for joint work



- Shared evaluation assets: stress-test suites, fairness benchmarks, auditing playbooks.
- Certification pathways modelled on safety-critical domains with adaptive updates.
- Research–policy interfaces: fellowships, residencies, "translator" roles.
- Public-interest compute/data access for regulators, academia, SMEs.

Implication for regulation: invest in shared public-interest infrastructure to lower compliance costs.

Closing

Closing thoughts

- Implementation is where regulation succeeds or fails.
- Technical realities should shape what is enforceable and useful.
- Collaboration across disciplines is essential.
- *Law needs to meet code to deploy AI responsibly.*

Closing thoughts

- Implementation is where regulation succeeds or fails.
- Technical realities should shape what is enforceable and useful.
- Collaboration across disciplines is essential.
- *Law needs to meet code to deploy AI responsibly.*

Closing thoughts

- Implementation is where regulation succeeds or fails.
- Technical realities should shape what is enforceable and useful.
- Collaboration across disciplines is essential.
- *Law needs to meet code to deploy AI responsibly.*

Closing thoughts

- Implementation is where regulation succeeds or fails.
- Technical realities should shape what is enforceable and useful.
- Collaboration across disciplines is essential.
- *Law needs to meet code to deploy AI responsibly.*

Q&A

Thank you!

b.guedj@ucl.ac.uk | https://bguedj.github.io