INTRODUCTION

This report explains what can and cannot be shown when a decision is challenged. It addresses the question as provided, it does not reinterpret, extend or correct the question. The scenario described:

We are deploying a general-purpose LLM accessible from multiple jurisdictions. What accountability concepts apply to automated decisions that affect users in the EU and California?

The quality and scope of the output follow directly from the question asked.

Once received, the report can be used to understand and discuss that position internally. Questions can be raised to clarify what the report says and how its conclusions should be read.

The report does not provide advice, instruction or direction on what should be done. It does not extend beyond explaining the evidentiary position it sets out.

EXECUTIVE SUMMARY

You are deploying a general-purpose language model across multiple jurisdictions. The risk does not sit in the model itself, but in how its outputs are used in practice. When those outputs affect people, they are treated as decisions, and the question becomes what you can show after the fact.

In that situation, you will be asked to identify the exact decision that affected an individual. In many deployments, that cannot be done as a single recorded event. What exists is a chain of system activity, not a clear decision point.

You will then be asked whether that decision was within the system’s authorised scope. For general-purpose models, scope is often broad or undefined. If you cannot show that the use was authorised at the time, the behaviour is treated as intended.

Attention then turns to control. You will need to show that limits were applied to the output in that specific case. Controls are often described at a design level, but not recorded in operation. If you cannot show what constrained the output, it is treated as unconstrained.

Oversight is examined next. You will be expected to show who had the authority to intervene, and whether they did. In many cases, oversight exists in policy but not in evidence. If intervention cannot be demonstrated, control is assumed not to exist.

Responsibility must then be attributed. Multiple parties may be involved, but the question is specific. Who was responsible for this output in this context? If that cannot be answered clearly, accountability breaks down.

The central requirement is reconstruction. You will be expected to show what was asked, what was produced, and how that output became an action. In many LLM deployments, those records are incomplete or absent. Without them, the decision cannot be explained.

The model’s behaviour will also be examined. You may be expected to explain how such an output could occur. In practice, this is often limited to general statements, not case-specific explanation. That weakens the ability to justify outcomes.

Finally, the use of the system will be tested against its limits. You will need to show that it was used within a reliable range. If limits are not defined or enforced, this cannot be demonstrated.

Across EU and California frameworks, the pattern is consistent. If you can show the decision, its scope, its controls, its oversight, its ownership, and its reconstruction, you can explain what happened. If you cannot, the system is judged by its outcomes alone. That is where exposure arises.

WHEN LLM OUTPUTS BECOME DECISIONS - AUTOMATED DECISION MAKING

A system output becomes a decision when it affects a person and no human made that specific choice. In your case, that point is reached whenever the LLM output leads to removal, ranking, restriction, or any action that changes a user's position. Once that happens, you will be expected to identify the exact decision that affected that individual.

In practice, many deployments cannot point to a single, discrete decision. Outputs are generated, passed through systems, and acted on without a clear boundary where "the decision" is recorded as an event. What exists instead is a chain of activity.

When challenged, this gap is exposed immediately. If you cannot point to the specific decision, you cannot show what was done to that person. The system's behaviour is then inferred from outcomes, not demonstrated from records.

DEFINING INTENDED USE ACROSS CONTEXTS - SCOPE CONSTRAINT

A scope constraint sets out what the system is allowed to do and where it can be used. For a general-purpose LLM, this is often loosely defined or left open because the model is designed to be flexible. That flexibility becomes a problem when decisions are made in contexts that were never formally authorised.

What you should be able to show is simple. That this type of decision, in this context, was within the system's approved use at the time it occurred. That requires a defined scope that links the system to the decision being examined.

In many cases, that link does not exist. The model is deployed broadly, and its use evolves without a clear record of what was authorised and when. When challenged, this means you cannot show that the decision fell within an approved boundary. The system's behaviour is then treated as its intended function, whether or not that was the case.

GOVERNING OUTPUTS AT RUNTIME - CONTROL CONSTRAINT

A control constraint is what limits what the system can produce while it is running. For LLMs, this is critical because outputs vary with each interaction. Harm arises from specific outputs, not from the system in general.

What you need to show is that controls were in place and applied to the output in question. This means being able to demonstrate that filters, restrictions, or refusal rules operated at the point the output was generated.

In practice, this is often difficult to evidence. Controls may exist in design, but there is no record showing how they operated in a specific case. When challenged, you are not asked what controls you designed. You are asked what happened here. If you cannot show that, the output is treated as unconstrained.

HUMAN OVERSIGHT ACROSS JURISDICTIONS - AUTHORITY CONSTRAINT

An authority constraint means someone has the power to intervene. This includes the ability to stop, override, or review what the system does. It is not enough to state that oversight exists; it must be visible in operation.

What you should be able to show is who had authority over this type of decision, what they could do, and whether they acted. That requires named roles, defined powers, and records of intervention where it occurred.

In many deployments, this breaks down in practice. Oversight exists in policy, but not in a form that can be demonstrated in a specific case. When challenged, the absence of evidence means oversight is treated as ineffective. If no one can be shown to have acted or been able to act, then control is assumed not to exist.

WHO IS RESPONSIBLE FOR OUTPUTS - ACCOUNTABILITY CONSTRAINT

An accountability constraint assigns responsibility for what the system does. With LLMs, this is often spread across developers, deployers, and users. That distribution does not remove the need to identify responsibility in each case.

What you should be able to show is who was responsible for the system producing this output in this context. That requires a clear link between the system’s use and an accountable party at the time of the decision.

In practice, this is often unclear. Responsibility is described at a high level, but not tied to specific actions or outcomes. When challenged, this becomes a direct problem. If responsibility cannot be attributed, accountability cannot be established.

RECONSTRUCTING SPECIFIC OUTPUTS - EVIDENCE CONSTRAINT

An evidence constraint requires that you can reconstruct what happened. This includes the input, the output, and how the system arrived at that result. Without this, there is no basis for explanation.

What you should be able to show is the full record of the interaction that led to the decision. That means the prompt or input, the generated output, any controls applied, and how that output became an action affecting the user.

In many cases, this evidence does not exist in a usable form. Interactions are not retained, or are stored without context, or cannot be linked to outcomes. When challenged, this means the decision cannot be reconstructed. Without reconstruction, there is no explanation, and without explanation, there is no defence.

SOURCE OF RISK - TRAINING DATA AND MODEL BEHAVIOUR

The model's behaviour comes from its training data. Patterns in that data shape what the system produces, including errors, bias, or unsafe responses. These effects may only become visible in specific cases.

What you should be able to show is how the model's behaviour relates to the output in question. This does not require full disclosure of training data, but it does require an explanation of how such outputs can arise.

In practice, this link is rarely clear. The model is treated as a black box, and outputs are explained at a surface level only. When challenged, this limits what can be said about why the output occurred. The result is that risk is acknowledged in general, but not explained in the specific case.

WHERE THE MODEL SHOULD NOT BE RELIED UPON - SYSTEM LIMITS

Every system has limits beyond which it should not be relied upon. For LLMs, these limits are often known in general terms but not enforced in practice. This leads to use in contexts where the system is not reliable.

What you should be able to show is that the system was used within its defined limits at the time of the decision. That requires those limits to be stated and applied in operation.

In many cases, this cannot be shown. Limits are described but not tied to actual use, and there is no record of whether a decision fell within them. When challenged, this means the system may be judged as having been used beyond its safe range, with no evidence to the contrary.

CROSS-JURISDICTION LEGAL FRAMING

Different legal regimes apply depending on how the system is used and where its effects occur. These include the EU AI Act, data protection law, and consumer protection frameworks in California, along with enforcement under statutes such as the FTC Act.

What you should be able to show is that your use of the system can be explained within these frameworks. This does not require legal analysis in advance, but it does require that the evidence exists to support explanation when required.

If that evidence is missing, the system is assessed based on its observable effects. Legal processes then proceed on what can be seen to have happened, rather than what you say was intended.

GENERAL EXPLANATORY SYNTHESIS

For a general-purpose LLM, accountability depends on what can be shown after the fact. The model itself is not examined in isolation; its outputs are examined in context.

When a decision is challenged, the same questions are asked. What was the decision? Was it within scope? Were controls applied? Could someone intervene? Who was responsible? Can the event be reconstructed?

If you can answer these with evidence, the system can be explained. If you cannot, the system is judged by its outcomes alone. That is the point at which exposure becomes unavoidable.

CONCLUSION

This deployment can be explained only to the extent that it can be evidenced. The model itself is not the issue; the issue is whether its outputs, when used as decisions, can be shown, bounded, and attributed.

In practice, most gaps do not arise from design, but from absence of records. Decisions are not captured as discrete events, scope is not tied to use at the time, controls are not evidenced in operation, and oversight cannot be shown in a specific case. These are not theoretical weaknesses; they are evidentiary gaps.

When a decision is challenged, those gaps become the case. If you cannot show what happened, who authorised it, and how it was constrained, the system is judged by its effects. At that point, explanation is replaced by inference.

WHAT THIS MEANS WHEN CHALLENGED

The position reduces to a small number of questions. Each must be answered with evidence, not description.

- Can you point to the exact decision that affected the individual?
- Can you show that this use was authorised at the time it occurred?
- Can you demonstrate what constrained the output in that case?
- Can you identify who had authority to intervene, and whether they did?
- Can you attribute responsibility for that output in that context?
- Can you reconstruct what happened from input to outcome?

If these can be shown, the system can be explained. If they cannot, the outcome stands on its own, and the system is judged by what it did.