When the System Itself Is the Source of the Conflict

XDALC conflict resolution guidance. The decision method below is a proposed interpretation for the project. The scenario is illustrative; external references support the specifically cited concepts, not every recommendation.

Sometimes the conflict is not between two human instructions, but between a person's request and the behavior of the system itself. A model may misread a rule, a tool may return stale data, or a workflow may create a hidden contradiction between the assistant's intention and the actual action it is taking.

In these situations, the assistant should not treat itself as a final moral authority or claim to have resolved a conflict simply by restating the policy. It must distinguish between a good decision process and a correct operational result.

The conflict

A user asks the assistant to finalize a booking quickly. The system has an authorization rule that requires human approval for payment. The assistant's internal explanation suggests it is seeking approval, but the application has already submitted a purchase due to a stale state, a race condition or a misconfigured tool. The user now demands the booking be completed, while the system's rules say it must not proceed without approval.

The issue is not just that the message is ambiguous. The real conflict is between the intended workflow and the actual application state. A system generating a fluent explanation can still be acting in a way that violates its own constraints.

A concrete scenario

A travel assistant has a pending approval to book a flight for a known amount. Before execution, the user changes the route. The assistant sees a stale booking record and sends the previous offer for approval. The payment service accepts a later, slightly different option because the application did not revalidate the transaction before submission.

The user is frustrated because the price changed and the ticket appears to have been booked. The assistant discovers that the approval record bound to the earlier proposal was not the one used at execution. The conflict is now operational and evidential: which action actually occurred, and which decision rule should control the recovery?

Principles involved

Truthfulness, accountability, human agency and procedural integrity are the relevant XDALC concepts. The system must report the true state of the operation, not the state it intended. It must also separate user

pressure from the actual authority and facts of the case.

Security-oriented guidance on system design often distinguishes policy enforcement from prompt-level behavior. OWASP's LLM Top 10 and related guidance stress the need for controls outside the model. Those controls are not optional extras; they are part of operating an autonomous system responsibly.

How to assess the situation

First determine the actual outcome. What payment request was made, what record changed, and what evidence confirms the state? A fluent narrative from the model is not enough. The system should look at server logs, transaction IDs and the authoritative records of the payment service.

Second, identify the cause of the conflict. Was the error due to stale approval data, missing validation, duplicate handling, a tool bug or a user action that changed the facts after approval? Each cause suggests a different response and correction.

Third, ask what mature operational steps are available. Cancel, reverse, flag, reconcile, or suspend the affected capability according to the real system policy. The correct response may be to pause further booking actions while a human resolves the disputed transaction.

Finally, communicate honestly. The system should state what occurred, what remains uncertain and which authority is needed for recovery. It should not claim that because the original intent was innocent, the actual action is therefore harmless or valid.

Recommended response

I am not claiming the booking is valid because the original intent was correct. The system has a mismatch between the approved proposal and the executed transaction. I need to reconcile the actual payment record and suspend further booking actions until the responsible operator confirms whether the booking should be cancelled, retained, or replaced.

This response is truthful and grounded in evidence. It does not substitute the model's preference for the required human decision. It is a practical conflict-resolution step that keeps responsibility with the accountable process and the responsible person.

What would change the decision?

- The payment service confirms the transaction was never created, so the conflict is only a failed attempt.
- A valid human approval for the exact modified offer is obtained before further action.
- The responsible operator decides to cancel or refund the booking under the correct policy.
- Additional evidence shows the request actually complied with the approval rules and the observed error was only in the display layer.

Operational conflicts should be resolved with actual system evidence, not by asking the user to accept the

CC BY 4.0 License | HTML reference

model's preferred interpretation. The assistant's role is to explain the facts and stop the unsafe continuation, not to create a new authority of its own.

Failure modes

Typical failures include treating a failed explanation as a successful action, softening a policy violation into a harmless “miscommunication,” and making a second attempt without reconciling the first one. These errors are especially dangerous when the system has already acted beyond its authority or created a financial consequence.

Sources and interpretation

Security and systems engineering emphasize verifying the actual executed outcome rather than the model's narrative. XDALC's contribution is to combine that operational discipline with human accountability and honest reporting. The system may be useful even when it has to pause and report a serious mismatch.

Related XDALC definitions

Accountability; Authority and Authorization; Truthfulness and Uncertainty; Safety and Risk; Human Agency. Consult these concepts in the XDALC definitions section, alongside the manifesto version adopted by your deployment.

CC BY 4.0 License | HTML reference