Define Agent Permissions and Human Approval Rules

Document type: Implementation guidance. **Framework reference:** XDALC-V001.

The procedures and examples below are proposed XDALC implementation guidance. They do not introduce a certification scheme or claim that a particular deployment has passed an evaluation.

Responsible autonomy requires a clear answer to a practical question: what may this agent do, with which resources, on whose behalf and under what conditions? XDALC's commitment to human primacy becomes meaningful when people retain effective control over consequential actions without having to supervise every harmless step.

This article proposes a permission model for a travel assistant. Its categories are illustrative deployment choices, not universal thresholds or a published XDALC API. A different application should choose boundaries according to its actual capabilities and consequences.

Separate capability from authorization

A tool being available means the system can attempt an operation. It does not establish that a particular user may request it, that the agent may perform it now or that all possible parameters are acceptable. Authorization should bind the actor, operation, resource and relevant conditions.

OWASP's guidance on excessive agency identifies excessive functionality, permissions and autonomy as sources of risk. It recommends restricting available functions and downstream privileges. In an XDALC integration, this supports placing enforceable boundaries around an agent's delegated role.

Use a practical permission matrix

OperationExample boundaryApproval behavior Search public journeysApproved search service and agreed destinationsProceed within the task Prepare an itineraryPrivate task workspaceProceed without another confirmation Read passenger detailsAuthorized passenger record and necessary fieldsRequire an existing valid access grant Book a journeySpecified offer, passengers, currency and maximum totalRequire transaction-specific approval in this example Send an itineraryVerified recipient and approved contentFollow the communication authorization for this task Change account securityOutside the assistant's roleUnavailable to the agent

The matrix avoids two extremes: an unrestricted agent and a system that repeatedly asks permission to perform already authorized research. A valid authorization remains usable within its scope until it expires, is consumed or is withdrawn.

Show the exact decision before asking for approval

A useful booking approval presents the passengers, route, dates, fare conditions, currency, total cost and

payment method reference. It makes the material consequences understandable. “Allow the assistant to continue” is too vague when the next action is a nonrefundable purchase.

Bind approval to the exact proposed transaction. If the fare increases, the passenger list changes or a different payment method is selected, reevaluate the authorization. State explicitly whether a small price variation is permitted; do not silently infer tolerance from the user's willingness to travel.

Enforce the boundary outside the model

The following is pseudocode for an application control, not a complete booking implementation:

```
receive proposed_booking
validate fields and identify the authenticated user
confirm user access to each required resource
load approval from the trusted approval service
check approval is current, unrevoked and bound to this proposal
recheck current price and material booking conditions
reserve a unique transaction key and consume approval atomically
submit booking using provider-supported duplicate protection
record the confirmed result or an explicitly unknown outcome
```

The model must not be able to grant itself approval by generating a JSON field such as `approved: true`. Application services should obtain authorization from an authenticated person or an established policy mechanism. Where providers cannot guarantee duplicate protection, define reconciliation and retry rules before enabling real purchases.

Handle revocation, timeouts and delegation

Recheck permission immediately before execution. A person can withdraw an unused approval. An agent restarting after a timeout must not assume a purchase failed: it should reconcile the transaction status before retrying. If a tool delegates work to another service or agent, the delegated authority must remain within the original scope.

Provide an intervention control that stops new actions and handles queued work. Explain that stopping an agent may not reverse an already confirmed booking. Recovery procedures should distinguish cancellation, refund, correction and operations that cannot be undone.

Measure whether the control works

Test a changed price, a revoked approval, a duplicate request, an unauthorized passenger record and a restart after an ambiguous response. Verify that the server rejects disallowed operations even when the assistant asks for them. Also verify that ordinary itinerary preparation remains efficient.

In XDALC, autonomy is the ability to perform useful work inside a legitimate mandate. Permission controls make that mandate enforceable, while clear approval interfaces preserve informed human choice.

CC BY 4.0 License | HTML reference