Wallet Context Change Test Matrix

Blank working record - one transition per record

Test ticket: record the application release, wallet/provider version, environment and approved policy revision. Use disposable test accounts. Keep credentials and personal data out of this form.

Field and purpose	Your value
test_id A stable reference for one transition and its reruns.	
initial_account The exposed account before the event. Use an alias in shared records.	
initial_chain The chain ID before the event, with its representation stated.	
provider_event The event name and sanitized payload, including the resulting account or chain.	
session_revision A non-secret reference to the application session context being evaluated.	
expected_application_decision The approved allow, hold, deny or reauthentication rule for this action.	
observed_result What actually happened. Keep NOT RUN until the test is executed.	
evidence_ref A sanitized trace or test artifact that supports the observed result.	
owner The person accountable for resolving a mismatch or missing policy.	

Keep observed_result as NOT RUN until execution. An expected decision is a policy statement; it is not evidence that the application behaved that way.

Wallet Context Change Test Matrix

Four proposed transitions - write your policy before running a test

Aliases below are illustrative event labels, not executable address payloads. Copy one scenario into the blank record and choose the requested application action.

ID / provider transition	Policy question	Evidence to collect
WT-01 / accountsChanged: account A to account B	Can the existing session authorize the requested action as B?	Event, resulting account and server authorization decision.
WT-02 / chainChanged: 0x1 to 0x89	Does the authorized context cover the new chain?	New chain ID and the application decision for a chain-specific action.
WT-03 / accountsChanged: empty account list	Which wallet-scoped actions stop, and what happens to application login?	Empty account list plus separate frontend and server states.
WT-04 / disconnect: provider cannot reach any chain	How does the application handle unavailable RPC independently of login state?	Provider error and the separately observed application session state.

Separate the observations. EIP-1193 describes provider events. RPC disconnection and application logout are different observations. A wallet interface may use Disconnect for a different operation. Record the actual behavior.

After execution. Capture the event and resulting wallet context, then the authorization outcome for the scoped action. Link sanitized evidence. Assign an owner to a missing policy or mismatch.

This form does not execute tests or certify security. Key recovery, delegated session keys and token allowances require separate reviews.

Wallet Context Change Test Matrix

Fictional example - NOT RUN

Invented policy P-DEMO-1 holds privileged wallet actions after a subject mismatch until a fresh proof for account B is verified. This is not a universal logout rule and describes no client project.

Field	Illustrative entry
test_id	WT-DEMO-01
initial_account	account A (fictional alias)
initial_chain	0x1 (illustrative context; no chain contacted)
provider_event	accountsChanged: [account B] (alias, not a literal address payload)
session_revision	app-context-r17; subject=account A
expected_application_decision	Under fictional policy P-DEMO-1, hold privileged wallet actions until a fresh proof for B is verified. Do not reassign the session subject implicitly.
observed_result	NOT RUN - illustrative example only
evidence_ref	NONE - no execution artifact exists
owner	Authentication owner - role to assign

Primary references, checked October 20, 2026
EIP-1193: https://eips.ethereum.org/EIPS/eip-1193
ERC-4361: https://eips.ethereum.org/EIPS/eip-4361

Resource owner: Dmytro Nasyrov, Founder and CTO.
https://pharosproduction.com/dmytro-nasyrov/
Implementation context: https://pharosproduction.com/services/blockchain-services/