Participants: User | Authenticator | Authenticator Specific Modules | FIDO Client | User Agent | Web Server | FIDO Server

User → User Agent: User clicks on any action required authentication on https://webapp?purchase

User Agent → Web Server: HTTP GET https://webapp?purchase

Web Server → FIDO Server: Start UAF Authentication/Transaction Request

FIDO Server: Generate CHALLENGE (c), Auth Policy (p), and Transaction Text (t)

FIDO Server → Web Server: Send UAF Authentication/Transaction Request $= (c, p, t)$

t is present in transaction confirmation ONLY.

Web Server → User Agent: HTTP 200 OK ($a = $ APP_ID, c, p, t)

User Agent → FIDO Client: a, c, p, t

FIDO Client: Obtain the TLS_DATA

FIDO Client → User Agent: Get FACET_ID by a

User Agent → FIDO Client: Return list of FACET_ID(s)

FIDO Client → Authenticator Specific Modules: $h, ak, t, fc = \mathsf{hash}(fcp), ...$

1. Select authenticator(s) according to p
2. Lookup the key handle (h) and access key (ak)
3. $fcp = (a, c, \text{FACET\_ID}, \text{TLS\_DATA})$

Authenticator Specific Modules → Authenticator: Send Sign Command $(h, ak, t, fc, ...)$

Authenticator → User: Trigger local user verification

User → Authenticator: User Verified

1. Unlock UAuth Key Pair $= (Auth.pub, Auth.priv)$ for this handle h by ak
2. Generate a nonce (n)
3. Generate the Sign Data $= SignData = (fc, n, cntr, t, sig = \mathsf{signature\_by\_}Auth.priv(fc, n, cntr, t))$

$cntr = $ Signature Counter.

Authenticator → Authenticator Specific Modules: $SignData$

Authenticator Specific Modules → FIDO Client: $SignData$

FIDO Client → User Agent: $SignData$

User Agent → Web Server: $SignData$

Web Server → FIDO Server: $SignData$

1. Verify the $SignData$ signature by $Auth.pub$

FIDO Server → Web Server: Return verification result

Web Server → User Agent: HTTP 200 OK (verification result) and session binding