

Systems and Methods for Device Aware Interception of Telephone Communication Requests and Cross Device Communication Handoff

[0001] Provisional patent specification for Smart Call Bridge

[0002] Smart Call Bridge

Technical field

[0003] The disclosure concerns browser event handling, communication resource selection, machine-readable optical encoding, and transfer of a website visitor's telephone communication intent from a first computing device to a second device. More particularly, it concerns conditionally intercepting activation of an eligible telephone action before native protocol dispatch and presenting a configured communication bridge while preserving a native telephone route in other input environments.

Background

[0004] Web pages often present telephone numbers as anchors having a tel URI. Selecting such an anchor requests dispatch to a browser or operating-system protocol handler. On some desktop installations this produces an application-launch prompt, opens an installed calling application, or leaves the user without a practical next step. A prompt is not necessarily an error, and desktop devices can support calling. The communication experience depends on the available software and the user's environment.

[0005] A visitor may be viewing a page on a computer while possessing a separate camera-equipped telephone. Requiring the visitor to transcribe a displayed number can interrupt the communication sequence. A website may also offer a contact form, but the intended telephone action and the form may be spatially separated. A bridge can retain the selected business destination and expose alternative actions in the current page context.

[0006] This background identifies the operational problem motivating the disclosure. It does not assert that telephone hyperlinks, QR encoding, popups, input media queries, call tracking, or cross-device transfer are individually new. The disclosure describes their coordination and implementation; patentability remains subject to examination and prior-art analysis.

Summary

[0007] A first device renders a website having an eligible telephone trigger. A runtime associates the trigger with a telephone destination and obtains an input-capability state. Upon activation, a router either allows the original native telephone action or suppresses default dispatch and renders a communication bridge. The bridge includes a human-readable telephone destination and a machine-readable code representing a telephone communication URI. A second device scans the code and invokes an appropriate communication interface, subject to user confirmation and second-device support.

[0008] In one embodiment, a coarse primary pointer permits native telephone dispatch; a non-coarse primary pointer causes presentation of a modal. A change listener updates the capability state. In further embodiments, policy combines pointer capability with actual input-event information, explicit user preference, or a configured environment override. The policy does not require reliable detection of an installed desktop calling application.

[0009] Additional embodiments provide a distinct native computer-call link, a separate copy control, a selector switching a displayed code between call and text destinations, two simultaneously displayed codes, and navigation to a contact form. A main telephone destination can remain on public page elements while a separate modal destination is used within the bridge. Web action records and provider outcomes are recorded as different classes of evidence. Direct telephone or messaging codes do not inherently notify the first device when scanned.

[0010] The runtime can be instantiated as a React component or as a page enhancement installed by a script or a platform adapter. Remote configuration, number provisioning, provider correlation, billing, and centralized reporting are optional extensions rather than prerequisites of the local bridge.

Terminology and reference numerals

[0011] An eligible telephone trigger means an interactive element whose configured activation invokes a telephone destination: for example, a linked number, tel anchor, button, icon, navigation element, or sticky call action. Unlinked text is not an eligible trigger unless an optional adapter explicitly converts it into an interactive telephone action. A generic contact-page link is not automatically a telephone trigger.

[0012] The term input-capability state describes a browser-observable input characteristic, not a definitive statement that a device is a mobile telephone or can place a call. The labels native route and bridge route describe routing decisions. The term handoff distinguishes making a communication destination available to another device from establishing a completed call or transmitting a message.

[0013] Like numerals identify like functional elements throughout FIGS. 1-19: 100 first device; 110 browser; 120 website page; 130 eligible telephone trigger; 140 original tel URI; 150 native protocol handler; 160 application-launch prompt; 200 bridge runtime; 210 trigger adapter; 220 capability detector; 230 router; 240 bridge controller; 250 number resolver; 260 URI builder; 270 QR encoder; 280 configuration; 290 fallback controller; 300 bridge interface; 310 visible number; 320 computer-call action; 330 copy action; 340 call QR; 350 text QR; 360 mode selector; 370 form action; 380 dismissal control; 390 accessibility controller; 400 second device; 410 camera or scanner; 420 dialer; 430 message composer; 440 communication service; 450 business recipient; 500 event dispatcher; 510 event collector; 520 analytics destination; 530 provider or CRM; 540 outcome record; 550 correlation store; 560 form and confirmed submission; 600 account service; 610 site configuration store; 620 hosted distribution; 630 platform adapter; 640 subscription service; 650 configuration version; 660 location mapping; 670 redirect endpoint; 680 expiring handoff token; 690 deduplication registry. Functional elements can share software or hardware rather than requiring separate physical components.

Brief description of drawings

[0014] FIG. 1 depicts a conventional telephone activation and application-launch prompt. FIG. 2 depicts the browser runtime, configuration, optical handoff, and optional analytics architecture. FIG. 3 depicts capability acquisition and reevaluation. FIG. 4 depicts eligible trigger types and exclusion of unlinked text. FIG. 5 depicts conditional interception. FIG. 6 depicts preservation and recovery of the original destination.

[0015] FIG. 7 depicts a single-call-code bridge corresponding to the supplied component's core presentation. FIG. 8 depicts the proposed call and text selector. FIG. 9 depicts proposed simultaneous call and text codes. FIG. 10 depicts four communication paths and a separate copy control. FIG. 11 depicts alternative interface placements. FIG. 12 depicts main and modal destination mapping.

[0016] FIG. 13 depicts number resolution, URI encoding, scanning, and user confirmation. FIG. 14 depicts web events and provider outcomes with distinct evidence boundaries. FIG. 15 depicts form navigation and confirmed submission. FIG. 16 depicts an optional remotely configured service. FIG. 17 depicts distribution

through alternative platform adapters. FIG. 18 depicts failure recovery and accessibility operations. FIG. 19 depicts the optional tracked token creation, validation, retrieval page, and separate evidence records. The drawings are schematic functional illustrations; arrow direction denotes the depicted operation rather than a mandatory physical connection.

Scope of embodiments and technical relationships

[0017] The core direct-code embodiment coordinates an existing telephone trigger, a capability-dependent event decision, retention of the original telephone URI, and presentation of a destination-bearing optical symbol in the page context. The disclosed examples do not require a subscription, an analytics collector, an SMS option, a contact form, a tracking number, a particular brand of browser, or a separately installed phone application. Those elements may be added in the combinations specifically described below. A page with code presentation disabled can still expose a readable number and native calling action, although that variation does not perform the illustrated optical handoff.

[0018] For direct handoff, a telephone communication URI is a tel URI or an sms URI containing a recipient telephone address. The optional tracked handoff uses an HTTPS URI that retrieves a recipient action; that retrieval URI is distinguished from the telephone communication URI eventually presented on the second device. A QR symbol is an optical code. The specification concretely describes QR encoding; substituting another symbology requires a compatible encoder and scanner that can recover the same URI payload. No ability of every barcode format or every scanner to launch a communication action is assumed.

[0019] A resolved recipient record identifies a business or location and its configured communication addresses. In the common-number embodiment, voice and SMS use the same normalized telephone number. In a further proposed embodiment, the record has a voice address and a separately provisioned SMS address for the same recipient; the respective controls explicitly label the address they use. This variation does not treat different strings as the same telephone number or silently replace the displayed call number when Text is selected.

Detailed description

Computing environment and sequence

[0020] In FIG. 2, browser 110 executes runtime 200 on first device 100. Device 100 comprises one or more processors, memory storing executable instructions, a display, an input interface, and a network interface. Page 120 may be server-rendered, dynamically rendered, or updated by a client-side application. Configuration 280 supplies a telephone destination and presentation settings. Second device 400 comprises a display, a camera or scanner 410, and software capable of resolving the encoded communication URI.

[0021] Trigger adapter 210 resolves trigger 130, detector 220 supplies capability state, and router 230 chooses the native or bridge route. For the bridge route, resolver 250 determines a destination, URI builder 260 creates a scheme-specific URI, and encoder 270 generates a QR symbol. Controller 240 renders interface 300. The runtime executes locally and need not send the telephone destination to a QR-generation service. The source imports a pinned QR library from a distribution host; local encoding does not mean the library itself is bundled offline.

[0022] The native route uses the retained tel URI 140 and handler 150. The bridge route provides number 310 and code 340 for transfer to device 400. Scanning may open dialer 420, after which the visitor confirms a call through service 440 to recipient 450. Opening the dialer, connection establishment, answer, duration, and commercial conversion are separate states.

Supplied React and Framer embodiment

[0023] The supplied SmartCall.js defines SmartCall, useIsTouchDevice, buildTelLink, QRCodeSvg, and CallModal. SmartCall renders a tel anchor with a configured label and style variant. useIsTouchDevice initializes state to false, then uses a React effect to evaluate matchMedia('(pointer: coarse)'). It subscribes to media-query changes through addEventListener or the older addListener interface and removes the subscription upon cleanup. No browser-name classifier appears in the supplied routing function.

[0024] When the stored state is not coarse, handleClick prevents the anchor's default operation and sets open to true. A portal renders CallModal under document.body when the document exists. For coarse input, the handler does not prevent default, leaving tel dispatch to the browser. Because the effect runs after rendering, an activation before capability evaluation can take the bridge route even on a coarse-pointer device. A proposed enhancement delays eligibility or reads capability synchronously at activation; it is not represented as behavior already corrected in the source.

[0025] The current component accepts one phone value and one countryCode. buildTelLink strips non-digits from each, returns tel: for an empty telephone value, recognizes certain prefixed inputs, prepends a country code for ten-digit inputs, defaults ten-digit numbers without a country code to +1, and otherwise returns tel:+ followed by the digits. This is a heuristic, not complete worldwide telephone validation. The phone is displayed as supplied while the URI contains normalized digits.

[0026] QRCodeSvg calls qrcode(0, 'M'), adds the URI, creates the symbol, and traverses its module matrix. Each dark module becomes a unit path rectangle; the path and background are rendered in an SVG with a module-count viewBox and a nominal size of 180 pixels. Encoding exceptions yield an empty path with a default count of 21. Thus, the source can display an empty QR frame rather than a meaningful recovery state. A production embodiment checks for a valid generated symbol and provides fallback instructions.

[0027] The visible telephone number in CallModal is a button that copies the supplied string. It first uses navigator.clipboard.writeText; on failure it creates a temporary input, selects it, and attempts document.execCommand('copy'). The fallback reports copied after the command call without examining its Boolean result. A proposed robust implementation distinguishes a successful copy from a rejected command. The supplied number button does not initiate a desktop call.

[0028] CallModal provides a close button, backdrop dismissal when the backdrop is the event target, and Escape dismissal. It temporarily disables body scrolling and restores the prior value at cleanup. It uses role='dialog' and aria-modal='true'; the supplied file does not implement a complete focus trap, focus restoration, or programmatic dialog heading association. Reduced-motion CSS disables the modal and pulsing-dot animations. The eyebrow text 'Call or text' does not establish an implemented SMS path: the only generated code encodes a tel URI.

[0029] The source implements per-component interception rather than general binding to arbitrary existing site links. It includes configurable heading, subcopy, QR caption and enablement, button variant, full width, pulse dot, phone, and label through Framer property controls. The code was inspected, not executed against live devices in this drafting work; implementation descriptions identify the code paths present and do not claim independently demonstrated call completion.

Capability selection and hybrid input environments

[0030] FIG. 3 shows a capability sample by detector 220 and subsequent reevaluation on a media-query change. A minimal policy routes coarse-primary-pointer environments to handler 150 and other environments to controller 240. A coarse pointer is an approximation of an input environment. Touchscreen desktops, tablets

with keyboards, and devices with multiple pointers may not match conventional desktop/mobile labels. The modal can retain an explicit computer-call action even when the bridge route is selected.

[0031] In a proposed policy, configuration 280 specifies routePolicy = capability, forceBridge, or forceNative. Capability mode may combine primary-pointer media query results with PointerEvent.pointerType, hover support, or an explicit visitor preference. Keyboard activation uses the configured default rather than pretending that a key identifies a telephone device. Policy snapshots are associated with the activation so changes during an open modal do not unexpectedly dispatch an action.

[0032] A proposed implementation evaluates capability again before interception, updates stored state on supported change notifications, and removes listeners during teardown. When signals conflict, the implementation chooses a configured conservative route and keeps both the readable number and original destination accessible. The runtime does not infer that a desktop calling app is installed merely because a browser supports tel links. Such an app check is generally outside ordinary webpage capability detection and is not needed for the disclosure.

Eligible triggers and event interception

[0033] FIG. 4 includes anchors with tel destinations and approved buttons or icons having explicit phone metadata. Adapter 210 may use a component callback, an allowlisted CSS selector, or event delegation through the page DOM. Configuration identifies the destination associated with each trigger. A linked number qualifies regardless of whether its label says Call. Plain text, a generic form link, mailto links, and unapproved interactive elements remain outside the telephone-trigger set.

[0034] For a proposed site-wide implementation, a delegated click handler examines event.composedPath or the closest eligible ancestor, checks that the original action is a telephone request, and retrieves a destination record. It excludes actions inside interface 300 so a computer-call link cannot recursively open another bridge. It avoids rebinding elements already owned by a component adapter. Repeated installation uses a registry 690 scoped by document, site, and runtime instance.

[0035] FIG. 5 separates capability routing from successful bridge preparation. The native branch leaves the event uncancelled. For the bridge branch, a production enhancement prepares validated destination data and a usable controller synchronously before preventing default. If preparation cannot complete, the native action proceeds. If a render exception occurs after cancellation, fallback controller 290 exposes a genuine link to original URI 140 and may attempt immediate dispatch while user activation is available; later asynchronous automatic dispatch may be blocked and is not guaranteed.

[0036] The runtime may preserve modified clicks, nonprimary mouse buttons, already-cancelled events, disabled controls, and browser context-menu behavior through explicit eligibility rules. Keyboard activation of semantic anchors is supported through ordinary click events; adapters for nonsemantic elements supply role and keyboard behavior. These rules are proposed extensions, since the supplied handleClick checks only its stored isTouch value.

Activation ownership and asynchronous work

[0037] In a proposed delegated adapter, eligibility is tested in this order: reject an already-prevented or noncancelable event; find an approved trigger in the composed event path; exclude elements owned by the bridge; reject disabled triggers and preserved modified or nonprimary activations; resolve the retained original tel URI; then sample routing policy. The listener is registered with passive set to false when it may cancel default. If a framework already owns the trigger, that framework callback owns the routing decision and the delegated adapter excludes that element. Event cancellation alone does not prevent all other application

callbacks, so installation assigns one owner rather than relying on preventDefault as a universal duplicate-handler lock.

[0038] Before canceling an eligible event, controller 240 has a local validated configuration snapshot, a destination, and a usable rendering path. Remote configuration is prefetched during initialization; an activation does not await a remote fetch and then retroactively cancel the original click. One proposed implementation precomputes symbols for the supported modes. Another shows the readable destination and genuine fallback link immediately and completes encoding locally; a QR error leaves those actions usable and removes the invalid symbol. In either case, the default-prevention decision occurs during the original event dispatch.

[0039] Controller 240 holds an active interaction identifier and the selected trigger reference. A repeated activation for that interaction focuses the existing bridge rather than creating a second modal. An activation for a different location either closes and replaces the current interaction or is rejected while it is open, according to an explicit setting. Closing releases the controller, removes interaction-specific listeners, restores focus to an existing eligible originating element or a configured page heading, and retires the interaction identifier. A delayed result checks the identifier before updating the UI, preventing a result for an old interaction from overwriting a later destination.

Destination resolution and safe fallback

[0040] FIGS. 6 and 12 illustrate preservation of original URI 140 separately from bridge configuration. The original href is retained rather than replaced globally with a modal tracking destination. When a script is blocked or fails to load, a server-rendered anchor can still request its original telephone action. This property requires that an actual anchor exists independently of successful enhancement; a component that never renders because its module fails to load cannot provide the same guarantee by itself.

[0041] Resolver 250 receives site, location, trigger, and number configuration. A proposed configuration includes mainPhone, modalPhone, country context, supported communication modes, contactTarget, labels, route policy, and implementation version. mainPhone is used in public page copy, coarse/native routing, and original fallback. If a valid dedicated modalPhone is configured, it is used for bridge number 310, call action 320, call code 340, and text code 350. If it is absent, an embodiment uses mainPhone or disables the bridge according to an explicit setting.

[0042] Location mapping 660 can associate a header action, office-specific card, or sticky CTA with a location identifier. A lookup returns a coherent snapshot of destination, labels, and form target. An already open modal uses that snapshot until refreshed through an explicit action; it does not silently mix a new number with an old QR. Invalid or unauthorized location mappings lead to the original destination rather than a random site's destination.

[0043] A dedicated number must be provisioned and routed to the intended recipient. SMS capability and receiving configuration must be verified before a text option is exposed. A number assigned to the modal channel provides channel-level evidence only to the extent its allocation and exclusive use are maintained. Reuse in advertisements, historical distribution, or number recycling can contaminate attribution. Assigning a modal number does not establish that every caller first saw the bridge.

Per mode destination records

[0044] A proposed recipient record contains recipient_id, location_id, retained original tel URI, main voice address, bridge voice address, optional bridge SMS address, supported modes, form target, and configuration version. Destination selection follows an explicit precedence: an approved trigger-specific record, then its location record, then a site default that the trigger is authorized to use. A partial record is validated as a whole

rather than freely mixing numbers from unrelated locations. If a trigger-specific record is invalid, fallback uses that trigger's retained original URI instead of selecting an unrelated site's default.

[0045] In the common-number embodiment, bridgeVoice and bridgeSms resolve to modalPhone. In the separate-address embodiment, bridgeVoice may resolve to +12025550102 and bridgeSms to +12025550103, both reserved example numbers assigned to recipient_id = office_a. Call displays the voice address and encodes tel:+12025550102; Text displays the SMS address and encodes sms:+12025550103. The copy action states whether it copies the selected mode address or the voice address. An SMS address is not inferred from the existence of a voice address; absent verified SMS support, the Text control is disabled or omitted and the form path remains available.

[0046] A resolved record stores the normalized address separately from its display string. For each active mode, the displayed address, clipboard value under the stated copy policy, encoded URI, and native action derive from that same record. An extension belongs to the voice address only when the chosen tel syntax and clients support it; it is not automatically appended to an SMS address. Local service codes, emergency destinations, and malformed or empty telephone requests are excluded by the production bridge adapter unless separately supported under an explicit validation policy. The retained original anchor is not rewritten merely because the bridge resolver rejects a destination.

URI construction and optical encoding

[0047] FIG. 13 separates display formatting, normalized telephone identity, URI construction, symbol generation, scanning, and communication confirmation. A proposed resolver requires a valid international number with explicit country context. It removes permitted presentation separators, rejects malformed input, and resolves national-format input using configured country rules rather than silently assuming every number belongs to North America. Extension handling is explicit, for example a validated tel extension parameter where supported.

[0048] For an illustrative reserved North American example, mainPhone = +12025550101 and modalPhone = +12025550102. The native destination is tel:+12025550101; the modal call destination is tel:+12025550102; and the modal message destination is sms:+12025550102. These examples identify data transformations and are not operational service numbers. A text body is optional, percent-encoded under the selected scheme contract, and tested against supported clients. A bare SMS recipient URI avoids reliance on inconsistent body-parameter handling.

[0049] Encoder 270 may use a QR implementation with selected error correction and automatic symbol version selection. It encodes the full selected URI, generates the matrix, and renders a vector or raster symbol with adequate contrast, square modules, and a quiet zone measured in module units. A proposed enhancement reserves at least four light modules around the code instead of depending on a fixed pixel frame. The size of the displayed code is chosen for scan distance, payload density, and responsive layout.

[0050] The scan is an optical transfer between displays and camera 410; a direct tel code does not involve a web redirect. The scanner recognizes the scheme and offers a dialer action. An sms code offers a message composer. Scanner/client support and user consent remain necessary. The QR does not itself place an irrevocable call or send a message. A production test should decode the rendered symbol and compare it byte-for-byte with the intended URI.

Bridge layouts and communication paths

[0051] FIG. 7 depicts interface 300 with dismissal control 380, visible copyable number 310/330, and call code 340. This arrangement corresponds to the core presentation supplied in SmartCall.js. Number copying

permits manual dialing without requiring an installed desktop calling application. The code path is a second-device dialing alternative.

[0052] FIG. 8 depicts an additional selector 360 with Call and Text modes. Initially the selected mode is call. Selecting Text changes the state, label, URI scheme, accessible description, and QR matrix as one update. The recipient remains the resolved modal destination. State is derived from the same configuration snapshot so the label cannot describe a text code while the symbol still represents tel. In a proposed implementation the mode-specific URI is the QR component's memoization dependency.

[0053] FIG. 9 provides simultaneous call code 340 and text code 350 with persistent distinct labels. A wide interface may place them side by side; a narrow interface stacks them. Each has its own caption and adequate quiet zone. Merely presenting two codes is not asserted to make an interaction patentable. The codes support different native second-device interfaces associated with a consistent business destination.

[0054] FIG. 10 provides four communication paths: computer calling through link 320, second-device calling through code 340, second-device messaging through code 350 or selector 360, and a website inquiry through form action 370. A copy button 330 remains independent of computer-call link 320. The visible number may itself be the call link while an adjacent control performs copying. The computer-call label explains the need for compatible configured software; it does not promise that an installed application will be found.

[0055] FIG. 11 shows modal, side panel, floating widget, and inline presentation alternatives. They use the same resolved destination and routing logic. Branding may vary without changing the communication operations. An inline bridge can replace or expand a region near trigger 130 rather than imposing a full-page overlay. The disclosure concerns technical operations across such placements, rather than exclusive ownership of every popup shape or arrangement.

Form navigation and submission context

[0056] FIG. 15 depicts form action 370 closing or suspending interface 300 and directing the user to form 560. For a same-page target, the implementation resolves an approved element identifier, scrolls to it with reduced-motion support, and focuses an appropriate heading or input. A missing anchor presents a configured alternate contact destination or a recoverable message rather than emitting a submission event.

[0057] For a contact-page target, the runtime validates an approved URL and transfers permitted source context through an allowlisted query parameter, same-origin session storage, or a form integration. An inquiry can carry source = desktop_contact_modal, component_id, site_id, location_id, and configuration version. Source information is separate from sensitive message text and caller identity. Cross-origin transfer requires an explicit integration; first-device storage is not automatically visible to another domain.

[0058] Dispatcher 500 records navigation selection separately from successful submission. Form submission success requires a confirmed response from the form adapter or backend, such as an accepted lead identifier. Merely clicking Submit, running client validation, scrolling, or displaying the form does not establish success. A submission identifier plus event_id prevents retries from creating duplicate conversions. A form adapter must document the exact success signal and cannot infer success from arbitrary DOM text alone.

Evidence based analytics and outcome correlation

[0059] FIG. 14 depicts optional event dispatcher 500, collector 510, analytics destination 520, provider/CRM 530, and outcome record 540. Canonical event names are smart_call_intent, smart_call_bridge_view, smart_call_phone_action, smart_call_message_action, smart_call_qr_handoff, smart_call_form_submit, and smart_call_fallback. The contract is a proposed production addition; the supplied component does not dispatch these events.

[0060] Intent is recorded once for an eligible activation; bridge_view is recorded after a usable bridge becomes visible. Phone action distinguishes computer_call, copy_number, and native_mobile_call. Message action distinguishes text_qr_selected and contact_form_open. An optional separate smart_call_qr_displayed records display or mode changes. That display event is not renamed handoff. Handoff requires evidence of a second-device interaction under a separately defined correlation mechanism. Verified form success and fallback each have their own event.

[0061] A record includes event_id, event_name, timestamp, site_id, component_id, placement, implementation_version, config_version, location_id when available, input_path, action, qr_mode, and page_path. Campaign parameters are included only when available and permitted. A proposed dispatcher uses a local queue and bounded retry policy; collector 510 rejects duplicate event IDs through registry 690. Analytics failure does not prevent interface operation or native calling. Phone numbers, message bodies, caller identities, and sensitive URL parameters are excluded from routine browser event payloads.

[0062] Legacy names using scbt_ or scb_ prefixes are mapped through a versioned alias table rather than emitted in parallel as extra conversions. For example scbt_call_intent and scb_phone_click may map to smart_call_intent only when their documented semantics match. An old scbt_qr_handoff record lacking verifiable evidence is retained as unknown or display evidence rather than upgraded to a confirmed scan.

[0063] The desktop page cannot observe a direct tel or sms QR scan simply because it generated a symbol. A browser click or number copy likewise does not establish an answered call. Provider records can establish an inbound call or message to the dedicated number, subject to the provider's record semantics; answered, missed, completed, and qualified lead are separate outcome definitions. Shared-number outcomes cannot by themselves distinguish manual dialing, desktop link use, or scanning.

[0064] An optional proposed tracked alternative uses redirect endpoint 670 and expiring opaque token 680. The QR encodes an HTTPS address associated with a server-side record containing site, component, mode, destination, configuration version, and expiration. A second-device request resolves the record and presents a user-confirmed call or text link. Tokens are unguessable, restricted to allowlisted destinations, and expired or revoked according to policy. Fetching the URL is only request evidence: previews, crawlers, and prefetchers can generate requests. A separate confirmed selection improves evidence but still does not prove a completed communication.

[0065] A further optional provider correlation store 550 associates a allocated tracking destination and allocation interval with a handoff session, or accepts a provider-specific correlation identifier when the provider actually supports it. Concurrent sessions, number reuse, late calls, replay, and clock differences require explicit ambiguity handling. These mechanisms remain proposed and require technical validation; the source and screenshots do not establish implemented cross-device session attribution.

Optional tracked handoff server sequence

[0066] FIG. 19 illustrates the optional tracked sequence. For the proposed HTTPS embodiment, a server creates an opaque token using a cryptographically secure random generator and stores a token digest as the lookup key. A record contains tenant/site identifier, recipient and location identifiers, selected mode, validated destination, configuration version, creation time, expiration time, and revocation state. The public URI has the form `https://bridge.example/h/<token>`; the reserved example domain represents a configured endpoint. The number and message body need not appear in that public URI. The endpoint derives the authorized tenant from a configured endpoint host or route, compares it with the stored tenant, and rejects unknown, expired, revoked, or cross-tenant records. It does not accept a destination override from request parameters.

[0067] A GET for a valid token returns a small landing page that names the recipient and selected mode and presents a genuine tel or sms link. The GET may create a retrieval record but does not consume the token as a confirmed communication. A separate user selection may send a same-origin action acknowledgment while allowing the genuine communication link to dispatch independently; failure of acknowledgment does not block the link. Records distinguish token_created, endpoint_requested, and destination_selected from provider-confirmed outcomes. A crawler can request the page, and automated clients can also generate selections, so neither signal is absolute proof of a human, an answered call, or a sent text.

[0068] One implementation allows repeated retrieval until expiration and deduplicates selections by token and action identifier. A stricter embodiment marks a selection acknowledgment once while allowing the user to recover the destination after a failed launch. Expiration is checked against server time, and a replacement token does not inherit a prior token's confirmation. If the endpoint is unavailable, the first-device bridge still exposes its readable destination and direct URI alternative; an expired second-device page requests a new scan rather than routing to an unverified number. This network-dependent embodiment differs from direct URI scanning, which requires no handoff server to resolve the address.

[0069] An optional tracking-number allocation binds recipient, allocation start/end times, and an interaction identifier to a provider destination. A provider record is accepted through an authenticated integration and deduplicated by provider plus provider record identifier. Correlation requires a matching called or messaged destination and an allocation interval under a defined timestamp and late-arrival policy. If two candidate allocations overlap, timestamps conflict, or a recycled number receives a delayed communication, the outcome remains ambiguous instead of being assigned to the latest displayed code. Token records alone cannot identify which direct telephone call corresponds to an interaction; that association additionally requires supported provider evidence or an exclusive allocation that still carries its documented limits.

Remote configuration and platform distribution

[0070] FIG. 16 depicts account service 600, configuration store 610, hosted distribution 620, subscription service 640, and configuration version 650. In an optional embodiment an authorized administrator creates a site record, supplies destination mappings and feature flags, and selects presentation options. The browser requests only public runtime configuration for an authorized origin; account credentials, billing secrets, and provider credentials remain on the server.

[0071] The hosted runtime obtains a validated versioned configuration snapshot. A timeout or invalid schema can use a valid cached snapshot under a bounded expiry policy or leave the original native anchor active. An account suspension may disable enhancements while preserving the public calling path. Subscription state enables service features but is not required to construct a local call bridge. Candidate products such as Supabase, Stripe, and Next.js are replaceable implementation choices, not proven deployed components of the invention.

[0072] FIG. 17 shows alternative adapters 630: a universal script installer, WordPress theme/plugin integration, and Framer component or plugin integration. A WordPress installer can enqueue the runtime once and supply a site identifier. A proposed Framer plugin can inject the same hosted runtime; this differs from the uploaded standalone React component. Webflow, Shopify, and other approved page systems can host an equivalent snippet. The browser behavior is described independently of distribution platform.

[0073] Page lifecycle management includes cleanup of listeners, preservation of original anchors, and avoidance of duplicate runtime instances. For single-page navigation, configuration is refreshed on a route change and newly rendered triggers are recognized through delegation or explicit adapter registration. Shadow

roots and embedded frames need specific integration because document-level selectors do not automatically traverse all boundaries. Each adapter exposes the same destination and event semantics.

Recovery and accessibility

[0074] FIG. 18 depicts failure recovery through controller 290 and accessible interaction through controller 390. A proposed runtime validates configuration and prepares usable UI before cancellation. After failure it restores a genuine original call link and a visible number, records a stable fallback reason if possible, and offers user-triggered retry. If a QR fails, the code is replaced with instructions or another supported action; a blank symbol is not presented as scannable.

[0075] A production accessibility controller moves focus into the bridge, associates the heading and instructions with the dialog, traps focus for a modal presentation, and restores focus to the originating trigger after close. It handles Escape and a labeled close button, avoids making background content interactive while modal, and supports keyboard-operable call/text selectors and copy controls. An inline presentation uses appropriate region semantics rather than falsely declaring aria-modal. Restoring body overflow must account for concurrent overlays through a shared ownership policy.

[0076] Configuration and labels are treated as data. Destinations are allowlisted to telephone, messaging, approved contact URLs, and the optional owned redirect endpoint. QR generation avoids inserting untrusted executable SVG markup. Remote settings and origin authorization are validated at both service and client boundaries. The number is intentionally visible in the direct-URI embodiment, and encoding it in a QR does not conceal it or prevent harvesting.

State model and example algorithms

[0077] A proposed state model uses UNINITIALIZED, READY_NATIVE, READY_BRIDGE, BRIDGE_OPEN_CALL, BRIDGE_OPEN_TEXT, FORM_NAVIGATION, CLOSED, and FALLBACK. UI state and communication outcome state remain distinct. Changing QR mode changes presentation state only. A native application launch or scan does not transition the first browser into CALL_COMPLETED without trusted external outcome evidence. Cancellation and dismissal return control to the page without creating a conversion.

[0078] The following pseudocode describes an additional production embodiment. It is enabling algorithmic guidance rather than a claim that these extensions are already deployed.

```

initialize(page, config):
    originalLinks = retainEligibleTelephoneDestinations(page)
    snapshot = validateConfig(config)
    detector = sampleCapabilityAndSubscribeToChanges()
    controller = prepareBridgeController(snapshot)
    installOnce(adapter, detector, controller)
onActivation(event):
    trigger = resolveApprovedTelephoneTrigger(event)
    if trigger is absent or trigger is insideBridge: return
    original = trigger.originalTelUri
    route = selectRoute(sampleCapability(), event, policy)
    emitIntentOnce(event, trigger)
    if route == NATIVE: return // leave default uncanceled
    prepared = prepareValidatedBridge(trigger, snapshot)
    if prepared is invalid: return // native default remains
    event.preventDefault()
    try:
        showBridge(prepared)
        emitBridgeViewAfterVisible(prepared)
    catch:
        exposeUserActivatedOriginalLink(original)
        attemptImmediateNativeDispatchIfPermitted(original)
        emitFallback(reason = RENDER_FAILURE)
selectQrMode(mode):
    assert mode in configuredSupportedModes
    uri = buildUri(mode, resolvedModalNumber)
    symbol = encodeQr(uri, quietZoneModules = 4)
    atomicallyUpdate(modelLabel, uri, symbol)
    emitModeSelection(mode) // never infer scan or completion
onConfirmedFormSuccess(response):
    if response.accepted and response.submissionId:
        emitOnce('smart_call_form_submit', response.submissionId)

```

[0079] A proposed configuration example supplies the following fields. A dedicated number is optional; absence does not justify inventing an available service number.

```

site_id: example_site
component_id: header_call
mainPhone: +12025550101
modalPhone: +12025550102
location_id: office_a
routePolicy: capability
qrModes: [call, text]
contactTarget: #contact-form
implementation_version: proposed_v1
config_version: 1
analyticsEnabled: false

```

[0080] For a fine-pointer activation of header_call, the resolver selects the office_a configuration, prepares interface 300, cancels tel:+12025550101 default dispatch, and displays +12025550102 with a call QR encoding tel:+12025550102. Selecting Text generates sms:+12025550102 and updates its label. A visitor scanning that symbol sees the second-device message composer and may send a message. The first page records mode selection only. A provider message record can later supply an outcome independently.

[0081] For a coarse-pointer activation with the same configuration, the native anchor remains tel:+12025550101. If analytics is enabled, a native selection event can be emitted without interrupting dispatch. For unavailable configuration or a failed preparation, the event remains uncanceled. For a rendering failure after cancellation, a visible original link supports a subsequent deliberate action rather than an unverified assumption that an asynchronous fallback always launches software.

Concrete deployment combinations

[0082] In a minimal local embodiment corresponding to FIGS. 3 and 7, a component binds its own original tel anchor, samples the coarse-pointer query, and presents one call code and a copyable number for the bridge route. No remote service, form, SMS, or analytics is required. The supplied source demonstrates the described

code paths with the limitations already identified; the proposed activation-time sample and validated QR fallback can be added without requiring the optional service architecture.

[0083] In a proposed configured direct embodiment corresponding to FIGS. 4, 8, 10, 12, and 15, a site adapter binds approved telephone triggers, resolves one recipient record, preserves the native main number, and renders voice/text mode controls and a form path. It uses direct tel and sms codes, with same-number or explicitly labeled separate-address operation. Configuration is prefetched; an open interaction uses one version. A provider or form integration is optional and its events cannot convert code display into scan evidence.

[0084] In a proposed tracked embodiment corresponding to the optional server and correlation elements of FIGS. 2 and 14, the bridge obtains a token before offering an HTTPS code, and the endpoint performs the retrieval sequence above. If a token is not ready at activation, the runtime can present the validated direct-code embodiment or a readable destination while requesting a token, according to the configured policy. It never labels an uncreated token as a usable QR. The interface replaces payload and label together when moving from a direct code to a tracked code; first-device polling, if enabled, displays only the server's defined evidence status and does not automatically launch a call on the first device.

[0085] These combinations specify how optional features interact rather than requiring every illustrated feature in every implementation. Variations in interface placement, adapter platform, mode layout, or local versus tracked encoding preserve their respective destination and event contracts. The proposed combinations and added algorithms require confirmation of human conception and are not represented as previously tested implementations.

Suggested validation procedures

[0086] The following procedures are proposed tests, not completed validation results. Test coarse, fine, keyboard, hybrid, early-before-effect, and media-query-change activation. Check that one activation opens at most one bridge and that bridge-internal native links do not recurse. Verify original anchors when script loading is blocked and after malformed configuration. Check number agreement between display, clipboard, native link, call QR, and text QR for each mapping.

[0087] Decode generated symbols on the supported second-device matrix, verify quiet zone and contrast at intended sizes, and compare decoded URIs with configuration. Validate the dialer and composer destinations without assuming automatic call or message transmission. Confirm dismissal, focus trap/return, scrolling restoration, reduced motion, missing form targets, rejected submissions, duplicate success callbacks, collector outages, and configuration refresh behavior.

[0088] For analytics, verify event deduplication and exact trigger definitions. Confirm that merely displaying or scanning a direct-URI code cannot manufacture first-device handoff evidence. Provider integration tests distinguish ringing, answered, missed, message received, and qualified lead statuses. Tracked-token tests cover expiry, reuse, bots, simultaneous sessions, and ambiguity. Test logs should record implementation version, devices, actual results, and unresolved defects.

Optional example claims for review

[0089] These claims are drafting examples for assessing disclosure support and inventorship. They are not required to file a provisional and are not a determination of allowable scope. Claims 1 and 9 address first-device direct-URI operation; claims 18 and 20 separately address tracked first-device and server operations. Second-device invocation is a purpose of the first-device code rather than a required scan or completed call

step in claim 1. This grouping provides alternative technical review positions and does not resolve infringement attribution, eligibility, novelty, or obviousness.

[0090] 1. A computer-implemented method comprising: rendering, on a first computing device, a website having an interactive telephone trigger associated with an original telephone URI; obtaining an input-capability state of the first computing device; receiving activation of the trigger; selecting, according to the input-capability state, a native route or a communication-bridge route; for the native route, permitting native dispatch of the original telephone URI; and for the communication-bridge route, preventing default dispatch and rendering an interface comprising a human-readable telephone destination and an optical machine-readable code encoding a telephone communication URI for invocation by a second computing device.

[0091] 2. The method of claim 1 wherein the input-capability state comprises a primary-pointer coarse-input media-query result and is updated upon a media-query change notification.

[0092] 3. The method of claim 1 wherein the interface comprises a selector, and selecting a call mode or a text mode changes the encoded communication URI between a tel scheme and an sms scheme while retaining a resolved recipient destination.

[0093] 4. The method of claim 1 wherein the interface concurrently presents separately labeled call and text codes associated with a common resolved recipient destination.

[0094] 5. The method of claim 1 wherein the original telephone URI addresses a main number and the rendered interface and encoded URI address a different configured modal number associated with the same business recipient.

[0095] 6. The method of claim 1 further comprising rendering a native computer-call link and a copy-number control as distinct actions and excluding activation of the native computer-call link from the interception applied to the telephone trigger.

[0096] 7. The method of claim 1 further comprising navigating from the interface to an approved contact form, retaining source context, and generating a submission event upon receipt of a confirmed successful submission response.

[0097] 8. The method of claim 1 further comprising retaining the original telephone URI as a usable native anchor when bridge preparation fails, and recording browser action evidence separately from externally verified communication outcome evidence.

[0098] 9. A system comprising one or more processors and memory storing executable instructions that, when executed, cause a first computing device to render a website telephone trigger associated with an original telephone URI, obtain an input-capability state, permit native dispatch for a native route selected according to the state, and, for a bridge route selected according to the state, prevent default dispatch and render a readable telephone destination and a QR symbol encoding a telephone communication URI for invocation by a second computing device.

[0099] 10. A non-transitory computer-readable storage medium storing executable instructions that, when executed by one or more processors of a first computing device, cause the first computing device to render a website telephone trigger associated with an original telephone URI, obtain an input-capability state, select a native route or a bridge route according to the state upon activation of the trigger, permit dispatch of the original telephone URI for the native route, and, for the bridge route, prevent default dispatch and render a readable telephone destination and a QR symbol encoding a telephone communication URI for invocation by a second computing device.

[0100] 11. The method of claim 1 wherein the optical machine-readable code is a QR symbol encoding a tel URI containing a normalized recipient telephone address, and generating the symbol occurs on the first computing device without transmitting that address to a remote QR generation service.

[0101] 12. The method of claim 1 further comprising resolving a validated configuration snapshot and preparing a usable bridge rendering path during dispatch of the activation event before preventing default dispatch, and permitting the original telephone action when the snapshot or rendering path is unavailable.

[0102] 13. The method of claim 1 wherein the telephone trigger belongs to an approved trigger set identified by a delegated event path, and activation of an element owned by the rendered interface is excluded from the delegated interception.

[0103] 14. The method of claim 1 wherein a configuration snapshot associates a location identifier, a normalized telephone address, a displayed address, a form target, and a version, and the rendered interface retains that snapshot for an open interaction until an explicit refresh or replacement.

[0104] 15. The method of claim 3 wherein a replacement QR symbol and its selected-mode label are committed together from the resolved recipient record, and failure to generate a valid replacement preserves a usable native telephone link and removes the invalid symbol.

[0105] 16. The method of claim 1 wherein a recipient record identifies a voice telephone address and a different SMS telephone address for the same recipient, and selecting a mode presents a readable address and a code encoding a scheme-specific URI derived from that mode's address.

[0106] 17. The method of claim 1 further comprising assigning an interaction identifier to the rendered interface, focusing that interface upon repeated activation for the same interaction, and rejecting a delayed update whose interaction identifier no longer matches the active interaction.

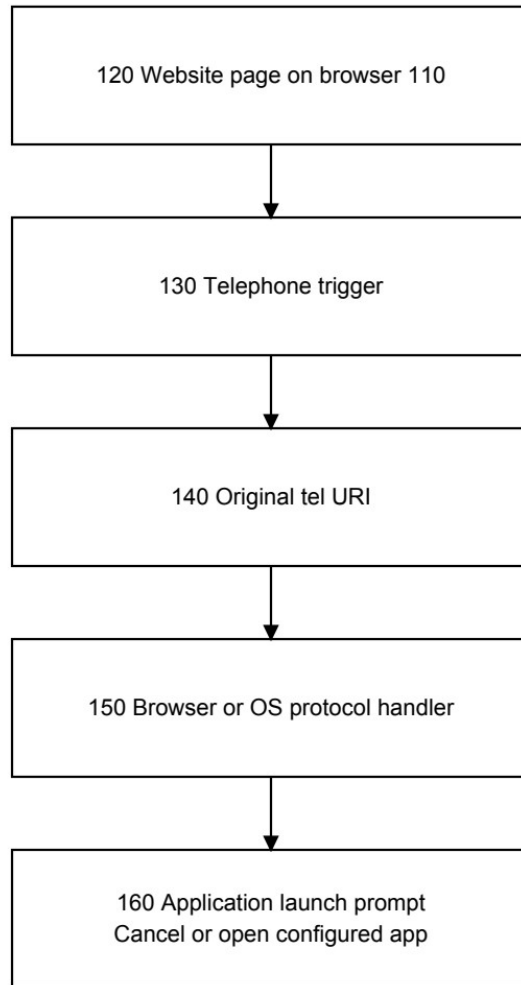
[0107] 18. A computer-implemented method performed by a first computing device comprising: retaining an original telephone URI associated with a website telephone trigger; obtaining an input-capability state; selecting a native route or a bridge route according to the state upon activation of the trigger; permitting dispatch of the original telephone URI for the native route; and, for the bridge route, preventing default dispatch and rendering a readable recipient telephone address and a QR symbol encoding an HTTPS retrieval URI containing an opaque token associated with a server record of a validated telephone destination, selected communication mode, and expiration time, the retrieval URI identifying an endpoint configured to present a user-selectable tel or sms link derived from the record on a second computing device.

[0108] 19. The method of claim 18 wherein the rendered interface exposes a direct telephone communication link independently of endpoint availability and commits a tracked QR symbol and its label together only after a token is available.

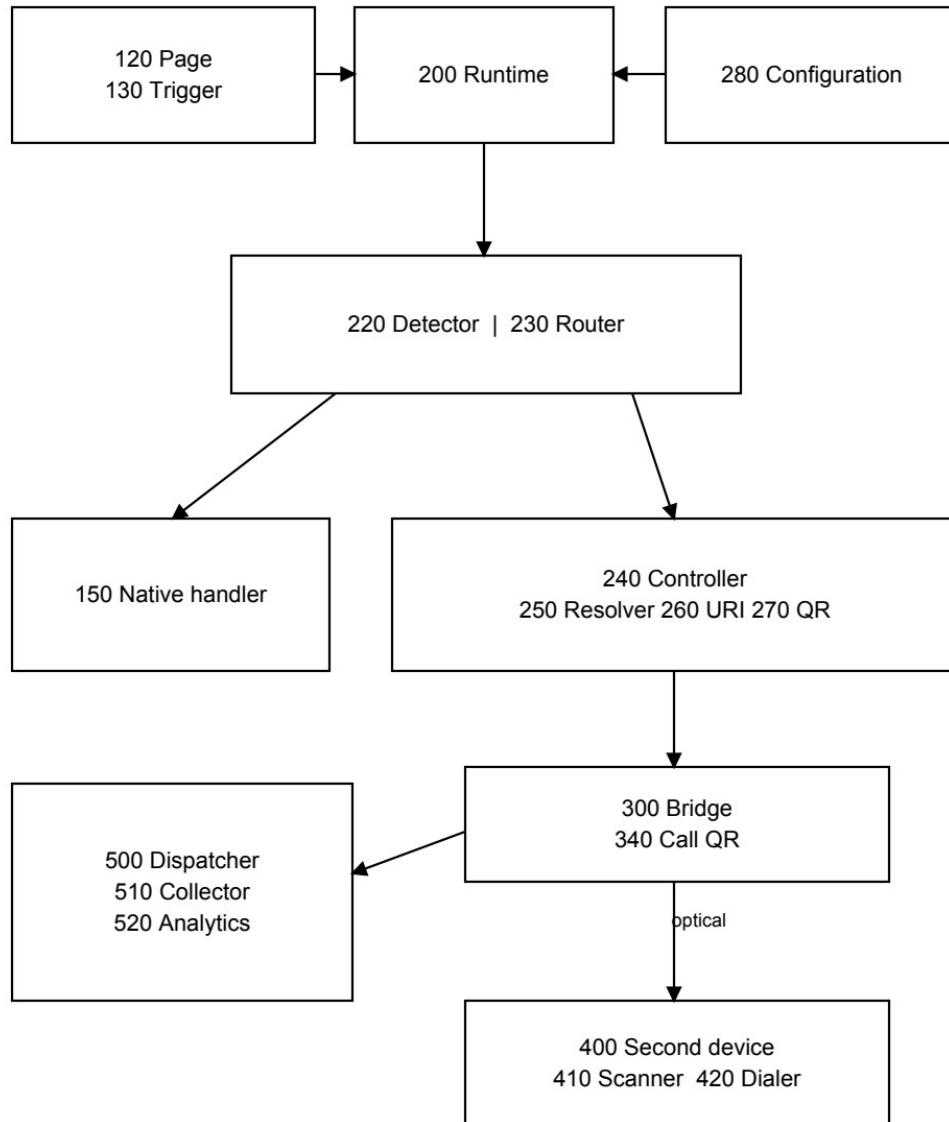
[0109] 20. A computer-implemented method performed by a handoff server comprising: storing a record associating an opaque token with a tenant identifier, validated recipient telephone destination, communication mode, and expiration time; receiving a request containing the token; rejecting an expired, revoked, unknown, or tenant-inconsistent record; and, for a valid record, returning a page comprising a user-selectable tel or sms link derived from the stored destination and mode, while recording retrieval evidence separately from link-selection evidence and externally confirmed communication outcomes.

Abstract

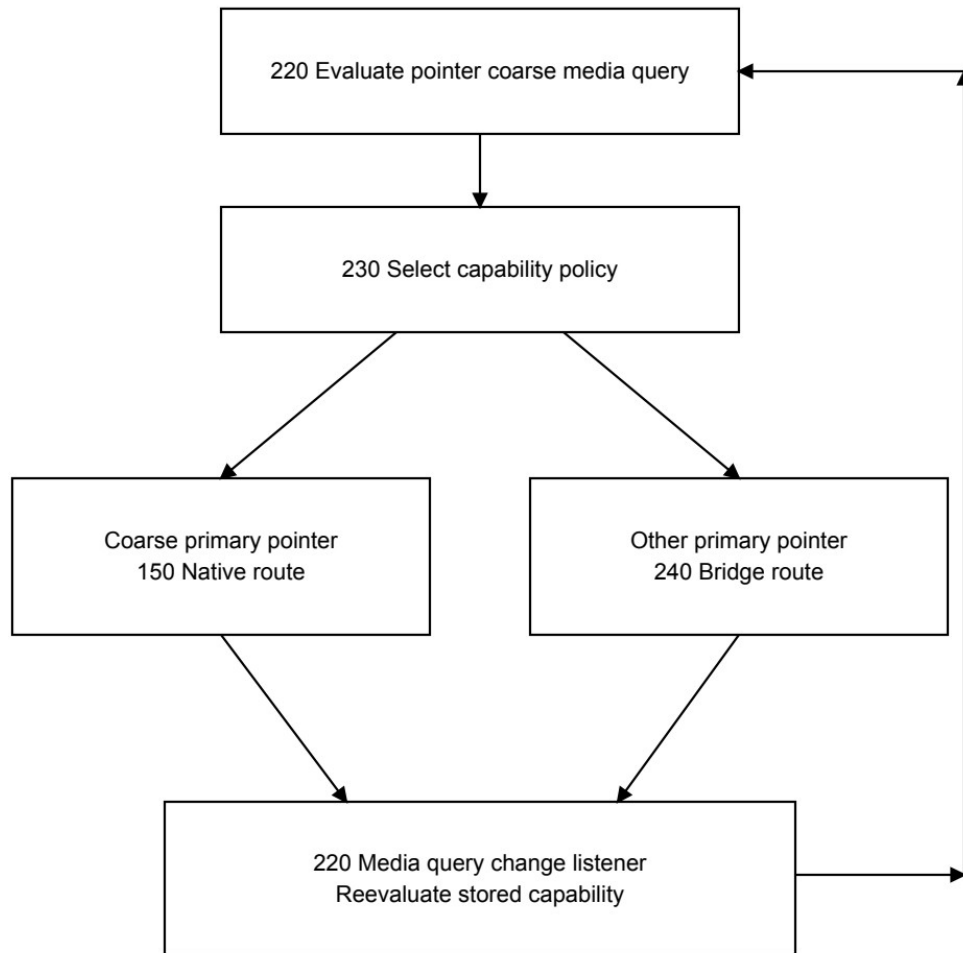
[0110] A website runtime conditionally routes activation of an interactive telephone trigger according to an input-capability state. A native route permits dispatch of an original telephone URI. A bridge route prevents default dispatch and displays a human-readable telephone destination and an optical code encoding a communication URI for use by a second device. Optional embodiments provide distinct computer-call and copy controls, selectable or simultaneous call and text codes, a contact-form path, separate main and modal destinations, remotely supplied configuration, and evidence-based event reporting. Original telephone destinations are retained for fallback. Browser action evidence is distinguished from provider communication outcomes, and direct telephone or messaging QR display is not treated as proof of scanning or completed communication.

FIG. 1 Conventional telephone dispatch

Environment illustration. A prompt may occur; it is not universally an error.

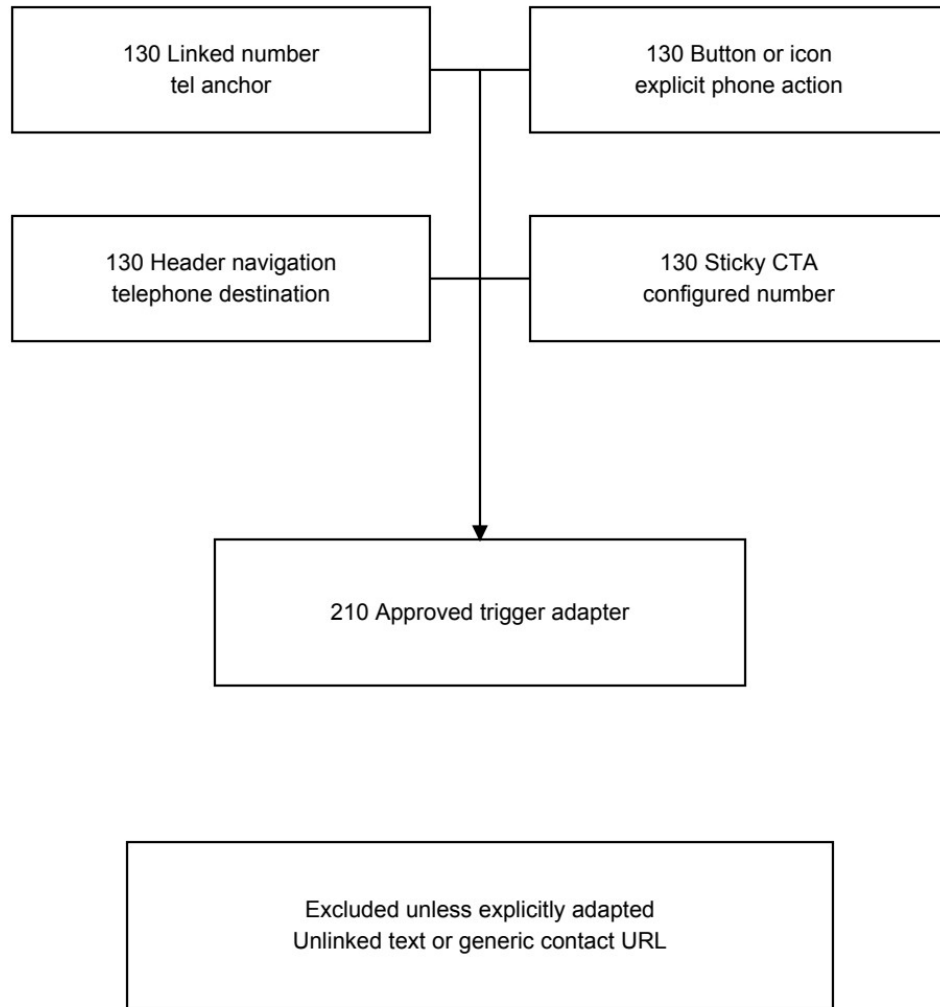
FIG. 2 Runtime and handoff architecture

Local optical handoff with optional configuration and analytics services.

FIG. 3 Capability selection and reevaluation

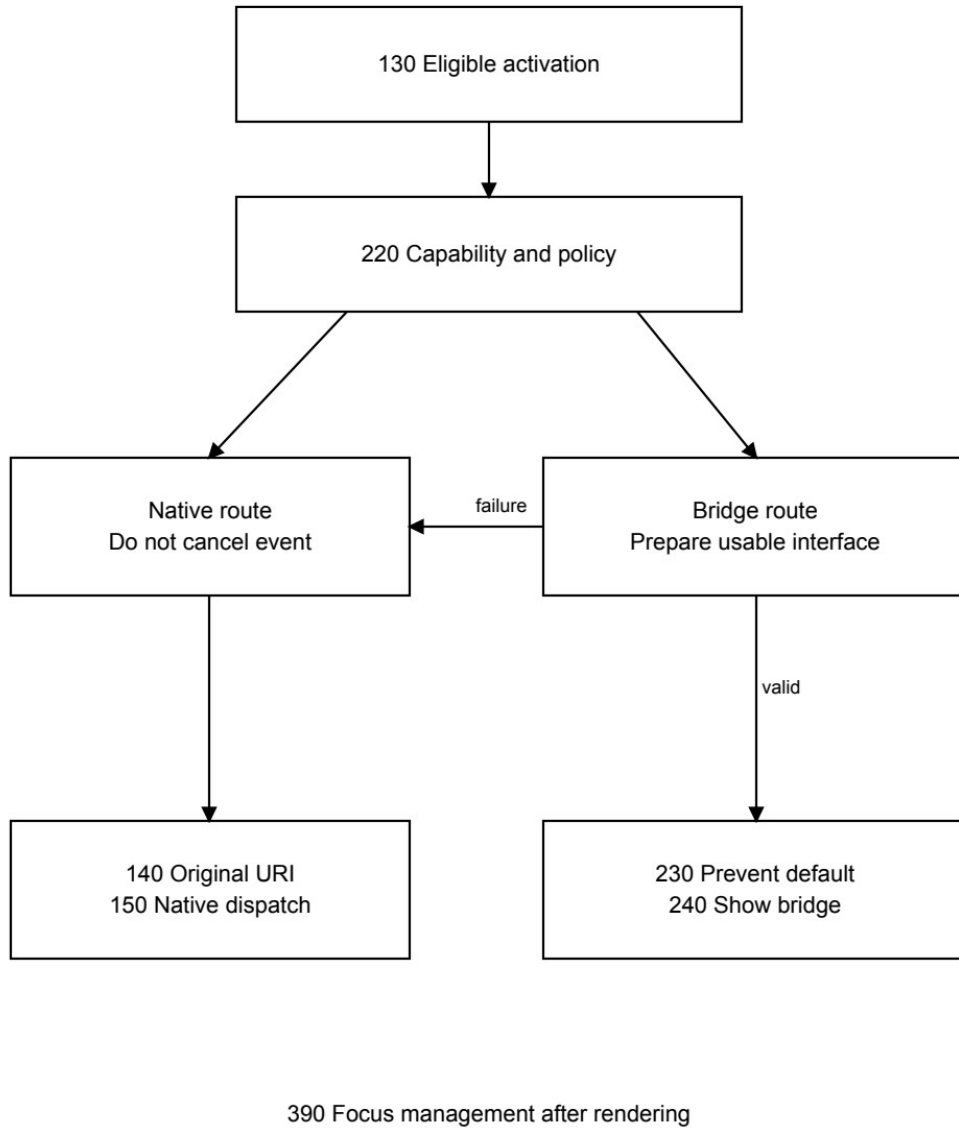
Hybrid input is not proof of telephone capability

Pointer query is implemented; combined policies and explicit overrides are proposed.

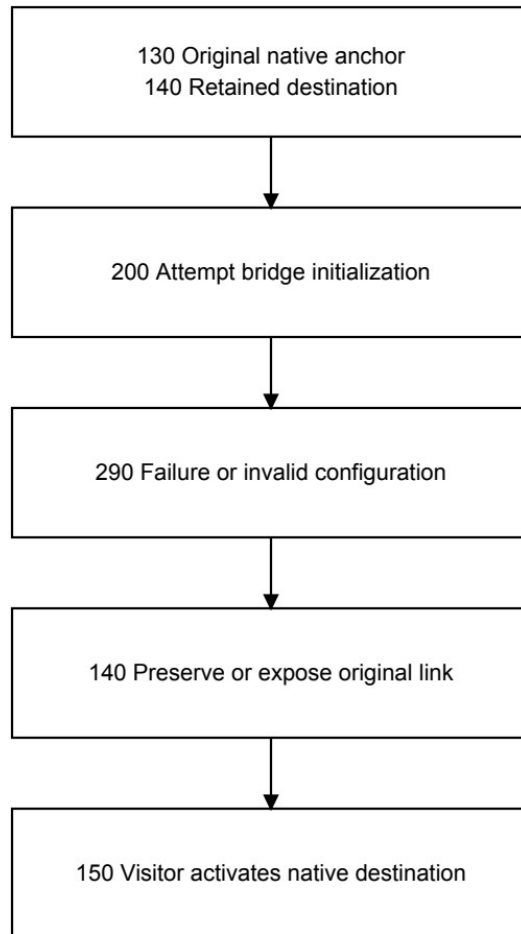
FIG. 4 Telephone trigger eligibility

Component or approved site wide adapter

The source intercepts its own component. General site adapters are proposed.

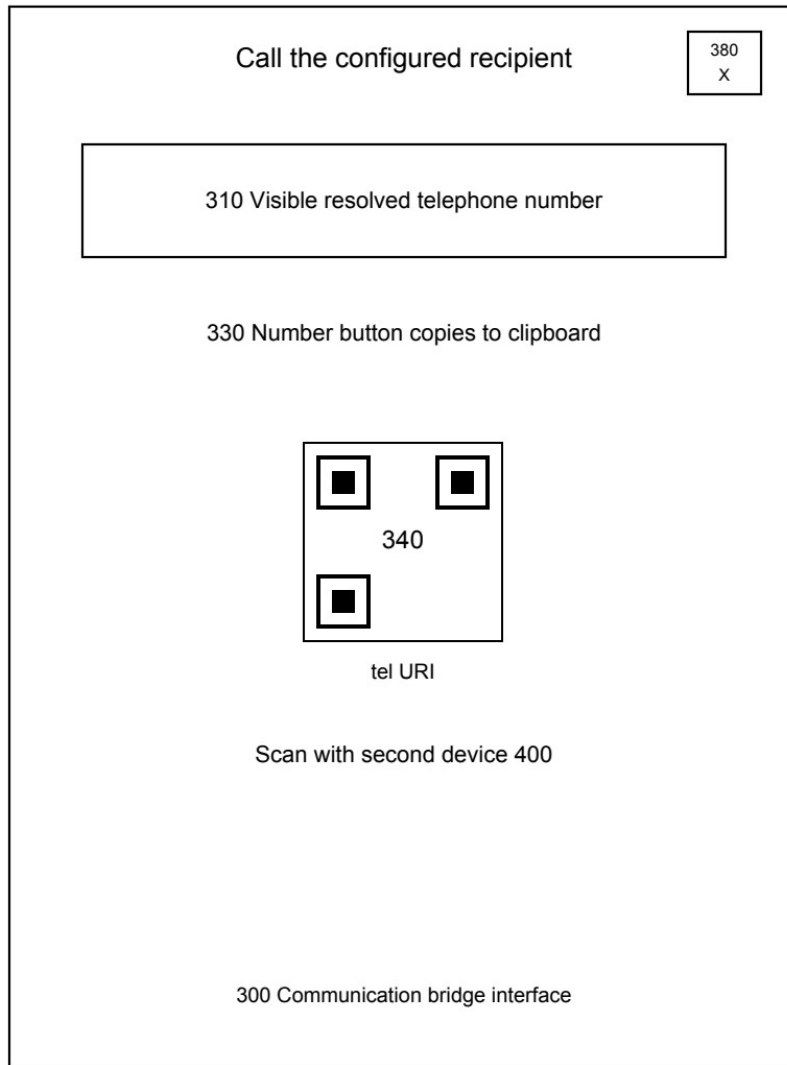
FIG. 5 Conditional interception decision

Preparation before cancellation and render recovery are proposed production details.

FIG. 6 Original destination and recovery

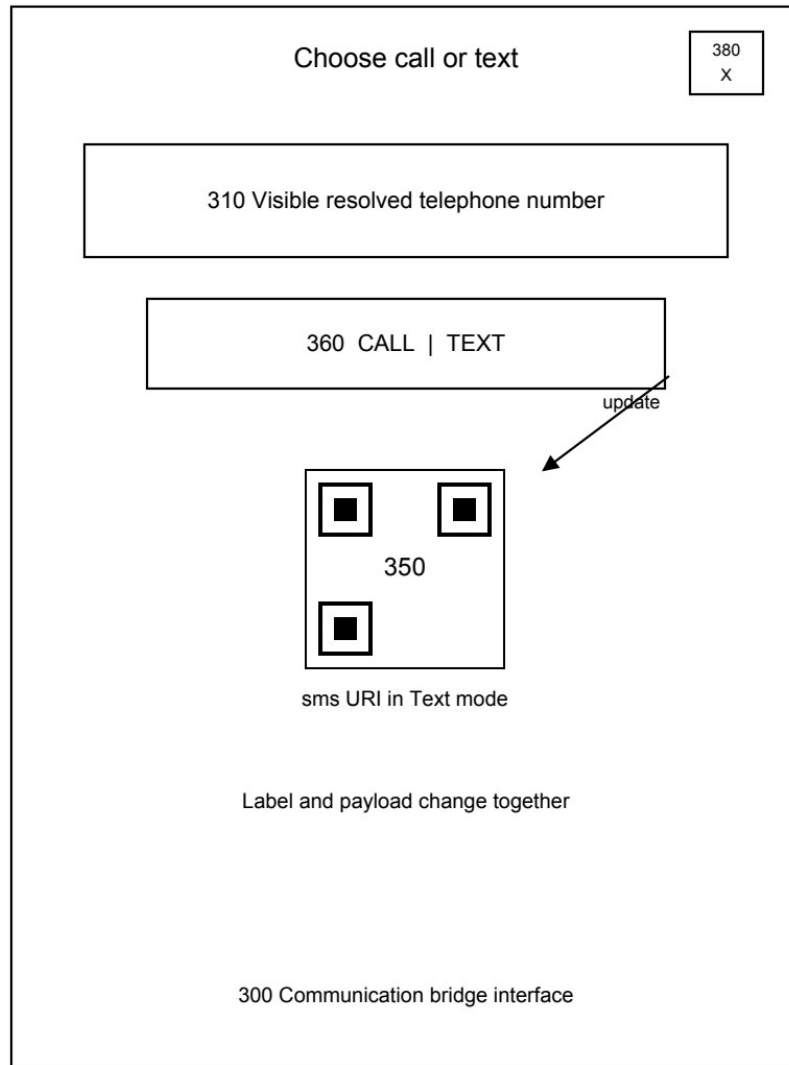
Retain native markup independently of enhancement where the platform permits it.

FIG. 7 Single call code bridge



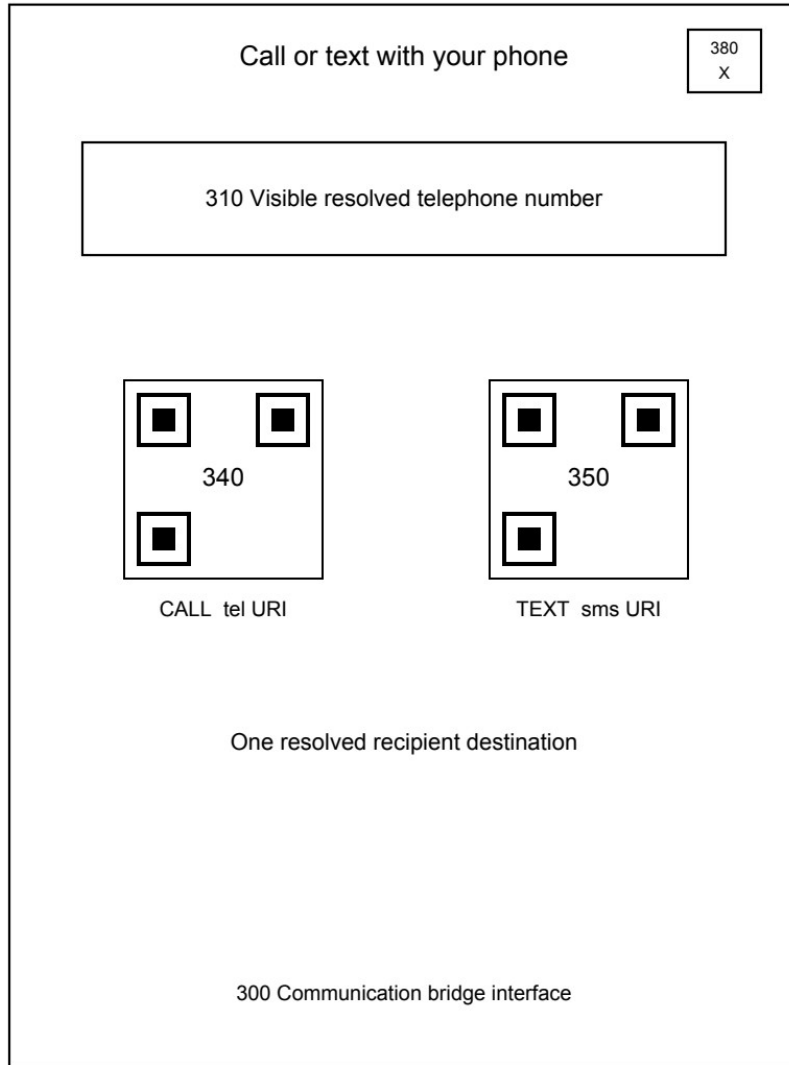
Core supplied presentation: visible copyable number and one tel code.

FIG. 8 Selectable call and text code



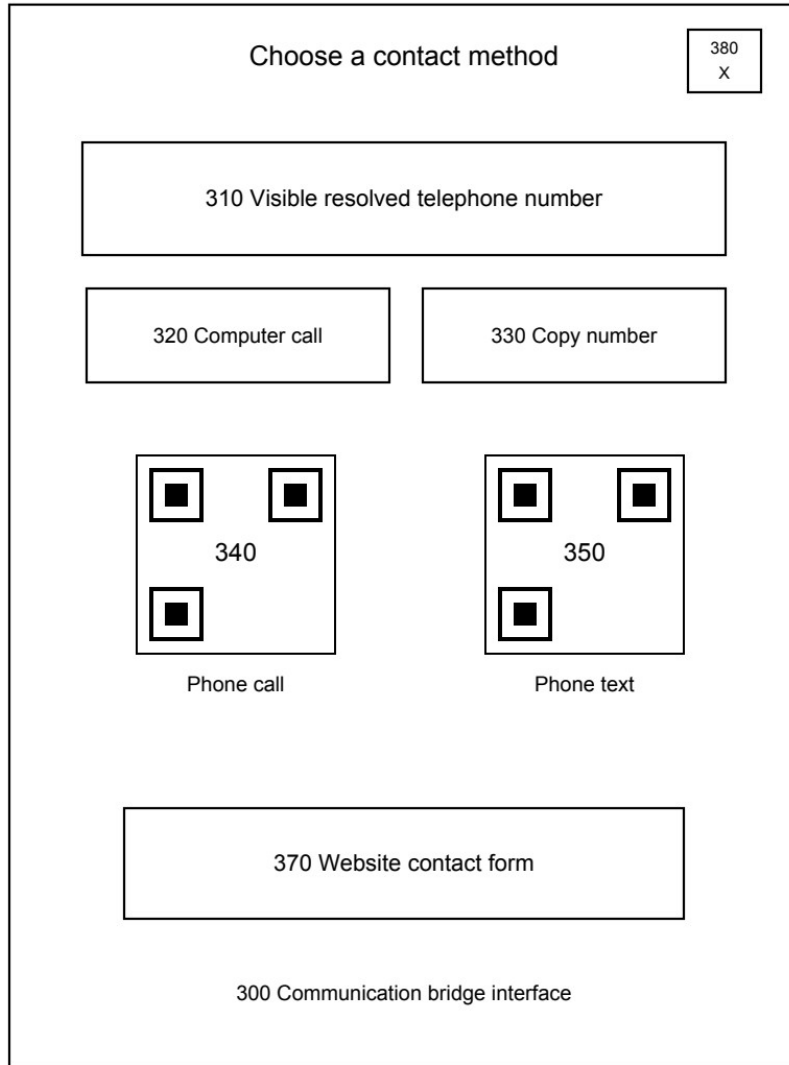
Proposed mode selector replaces scheme, label and symbol together.

FIG. 9 Simultaneous call and text codes



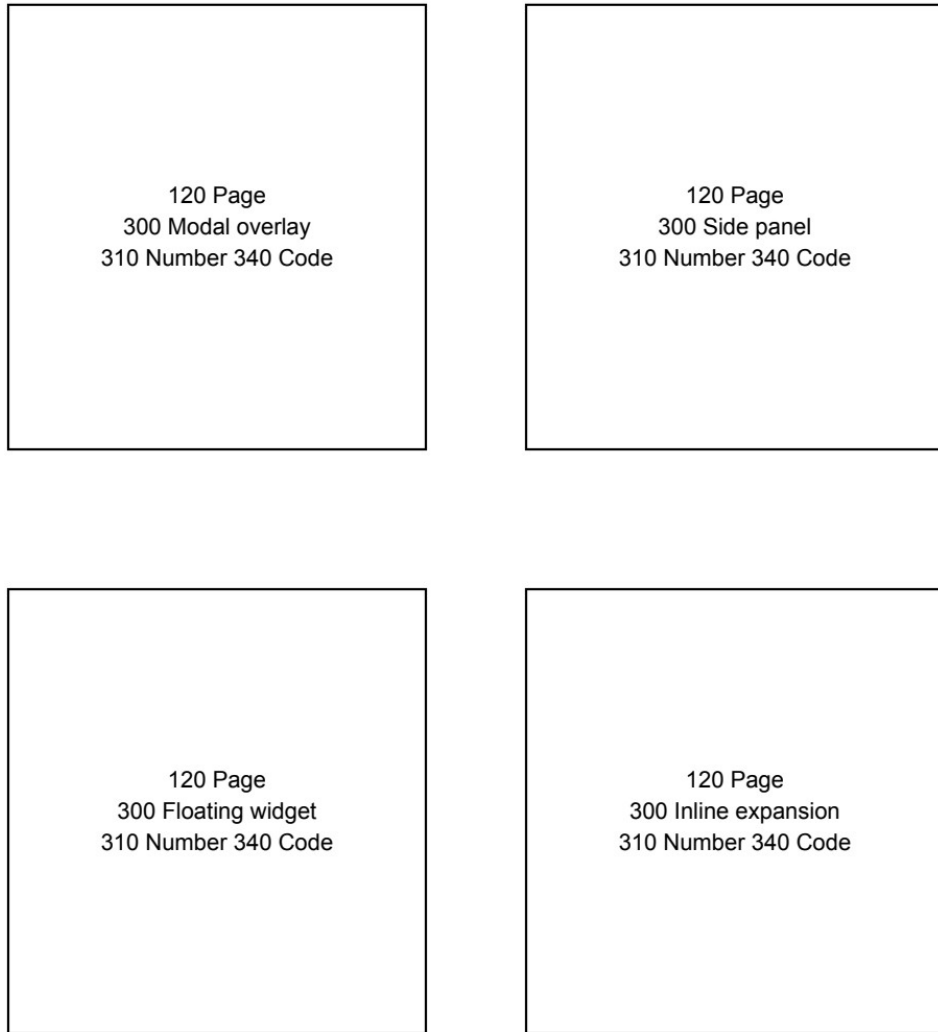
Proposed separately labeled symbols share a resolved recipient.

FIG. 10 Four communication paths



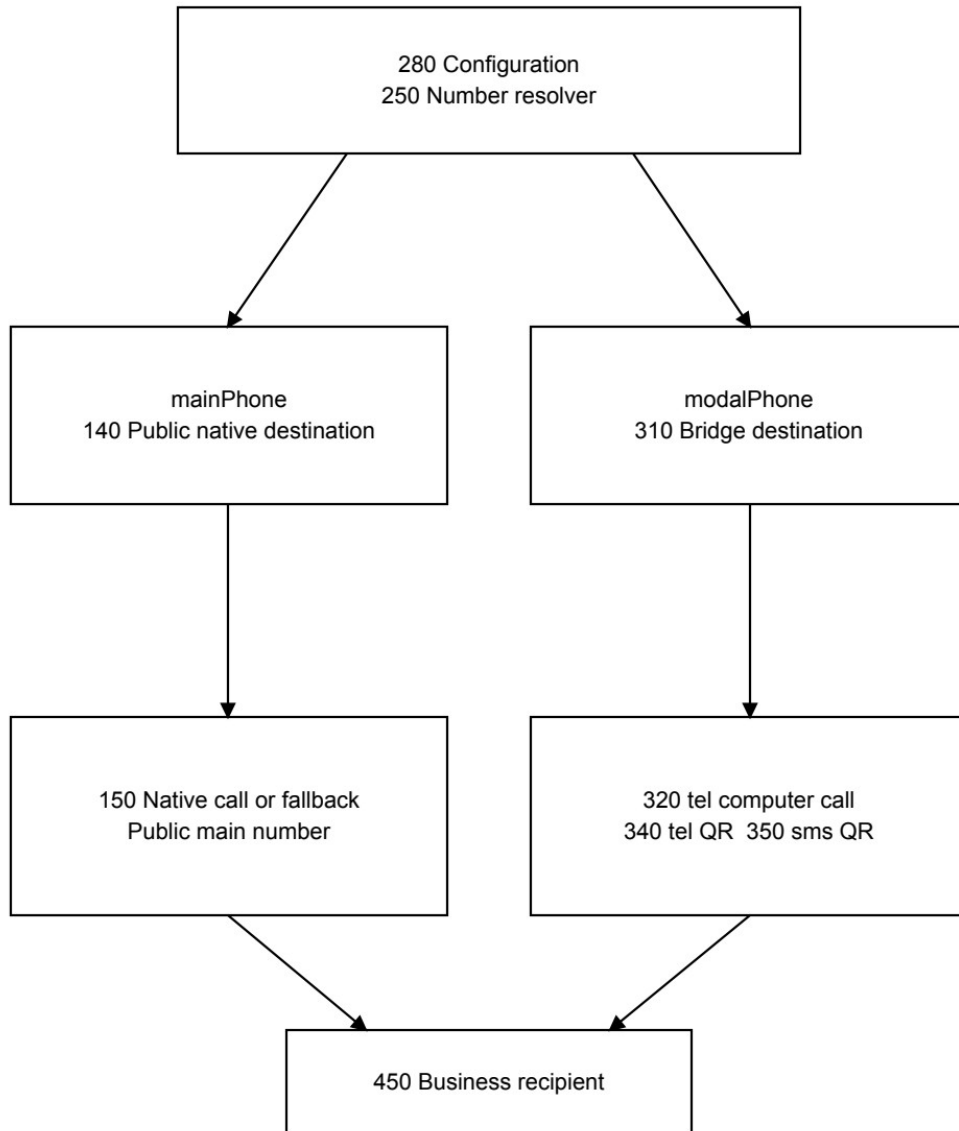
Proposed computer call, phone call, phone text and form paths; copy is separate.

FIG. 11 Alternative interface placements

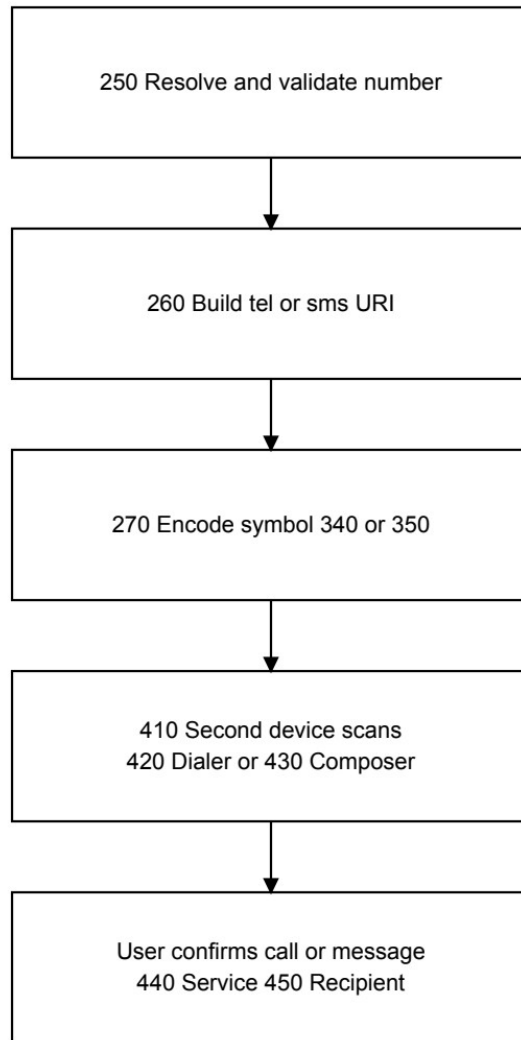


Same destination and routing contract

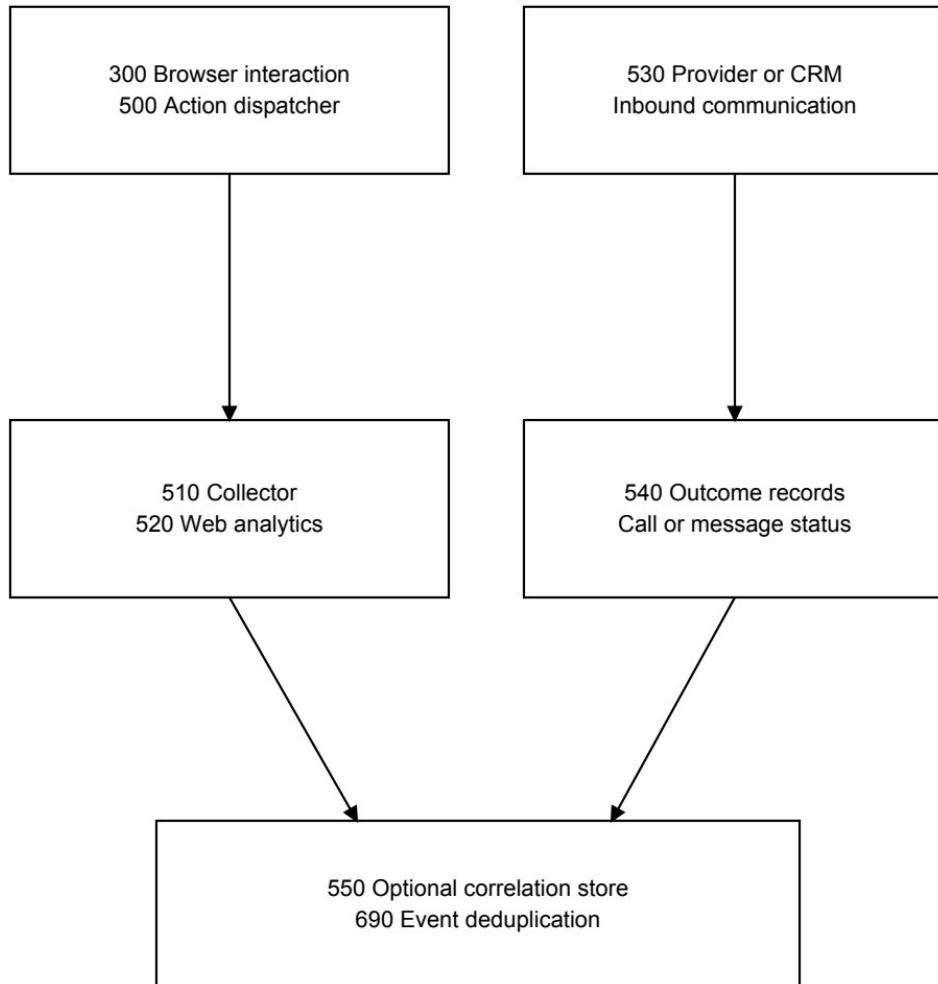
Modal exists in source; panel, floating and inline alternatives are proposed.

FIG. 12 Main and modal destination mapping

Separate number mapping is proposed; source uses one phone value.

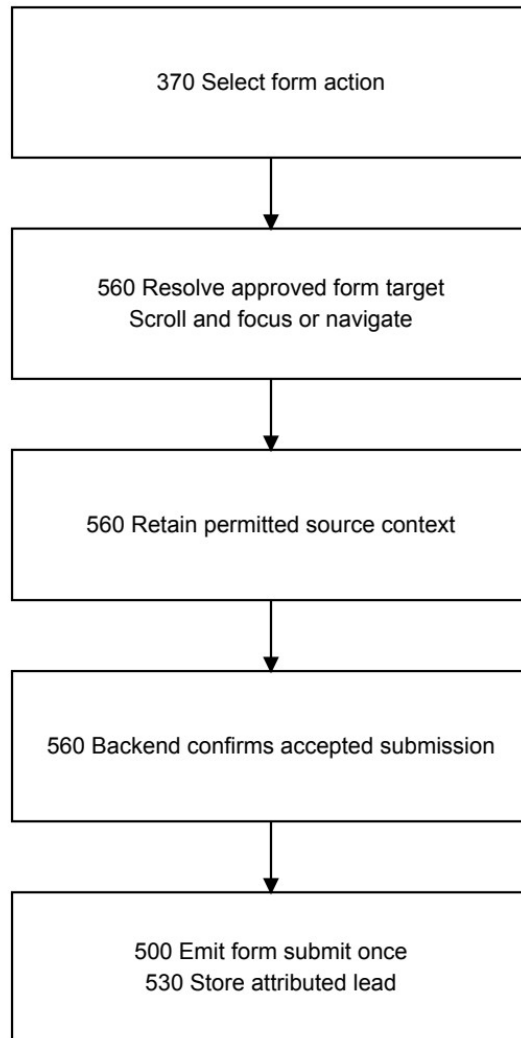
FIG. 13 URI encoding and second device use

The second device offers communication; the visitor confirms call or message.

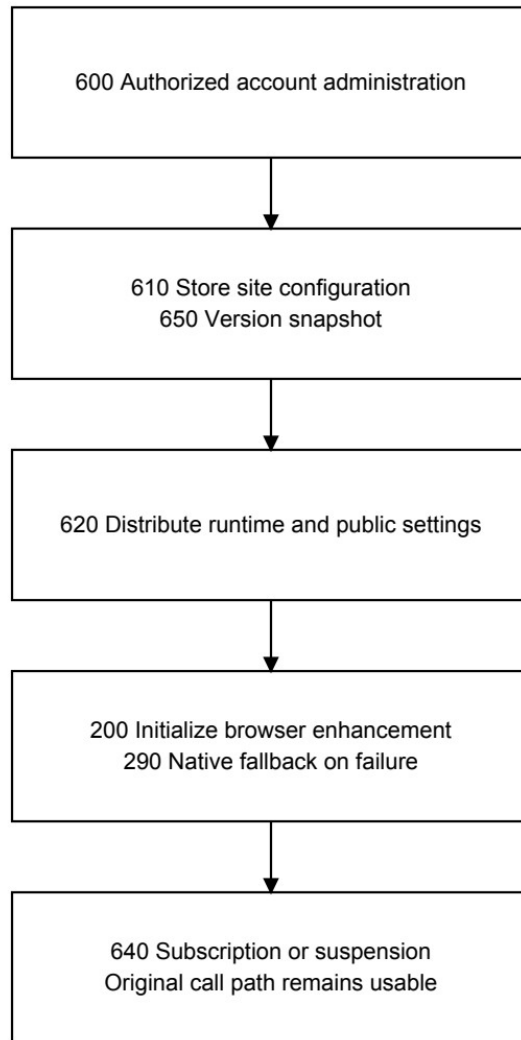
FIG. 14 Action evidence and outcome evidence

QR display is not scan or completed communication

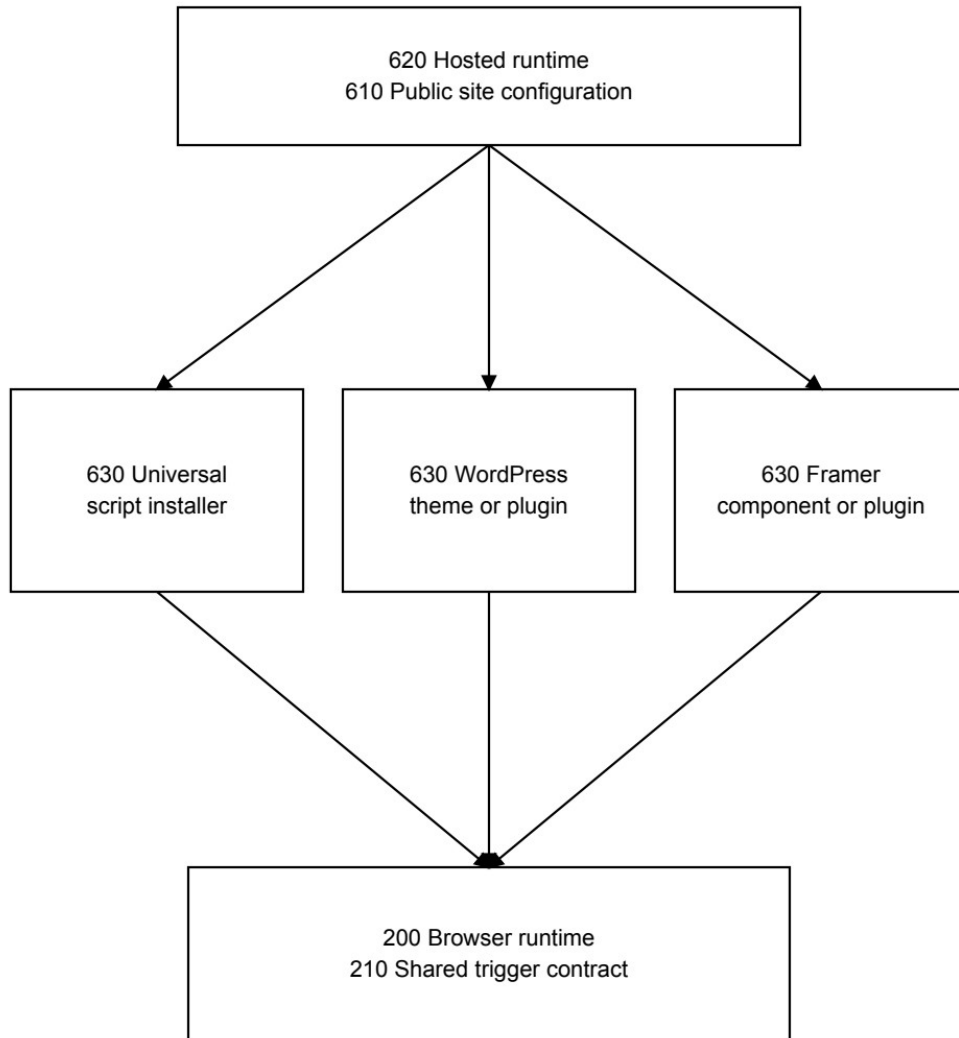
Proposed event layer. Direct tel or sms scanning sends no desktop event.

FIG. 15 Form navigation and confirmed submission

Proposed source retention and success event; opening a form is not submission.

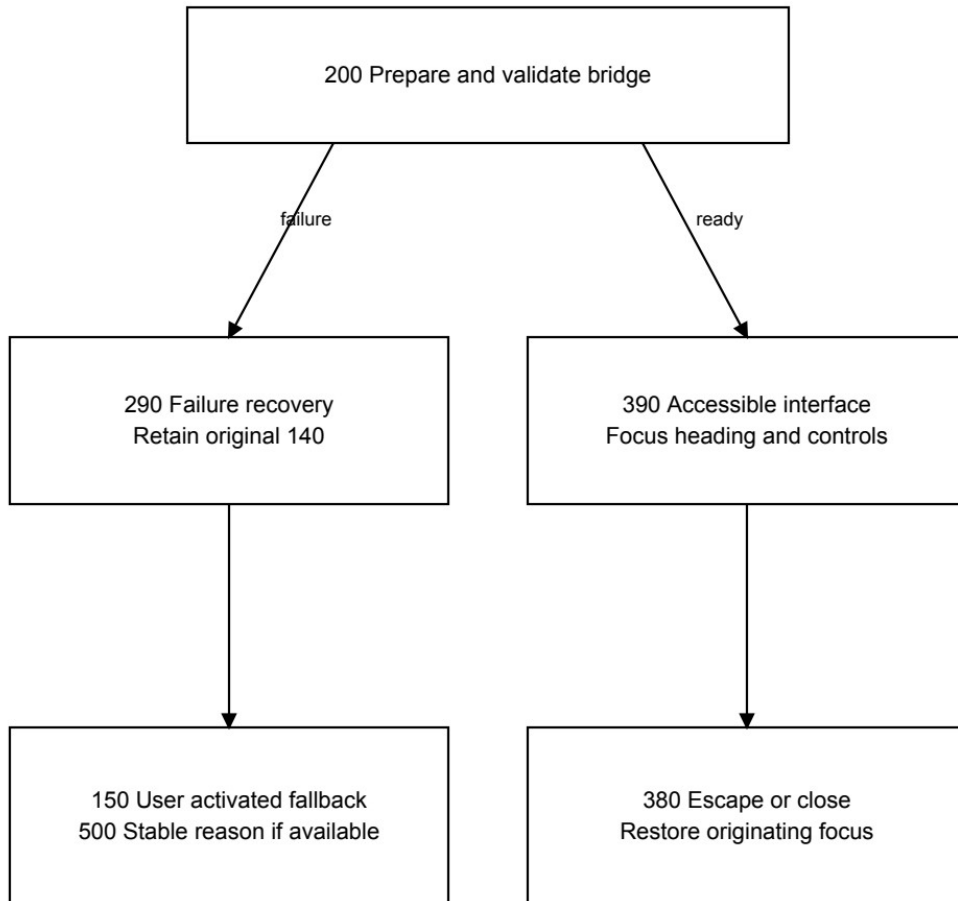
FIG. 16 Optional remote service lifecycle

Proposed account, public configuration, versioned runtime and subscription services.

FIG. 17 Platform installation alternatives

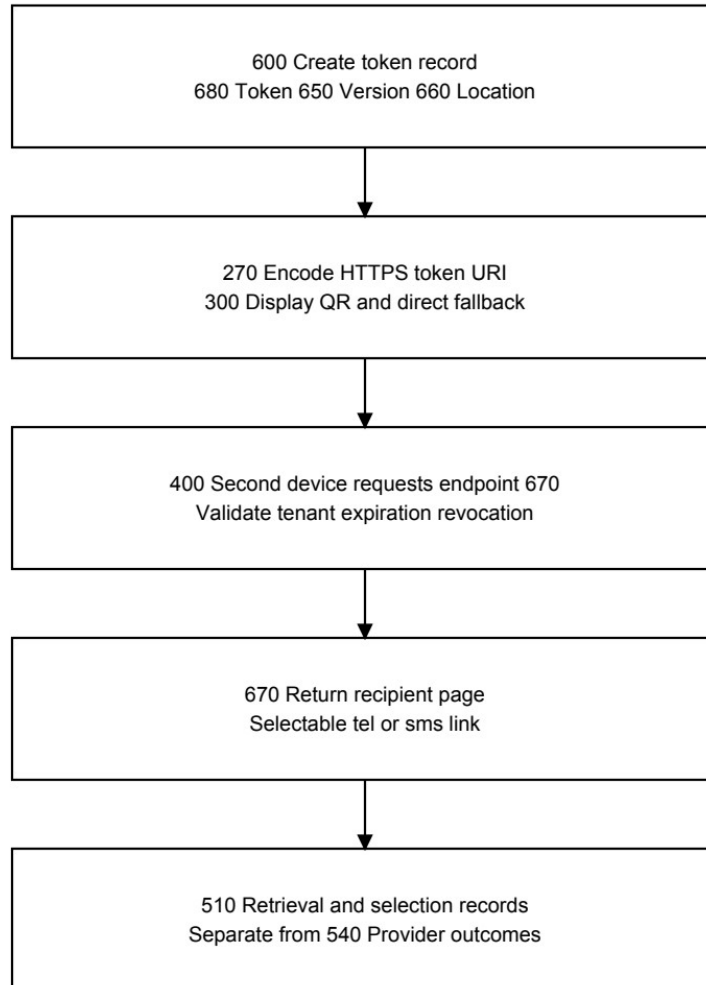
Distribution platform does not change core routing

Standalone Framer source exists; hosted script and installer paths are proposed.

FIG. 18 Recovery and accessible operation

No QR scan or outcome inferred from UI state

Dismissal exists in source; complete focus and robust failure handling are proposed.

FIG. 19 Tracked token retrieval sequence

Proposed server embodiment. Retrieval and selection do not prove a completed call.