RSW-0012 — JavaScript API

Revision 0.01 | Protocol rsw/1 | Format 1 | Experimental proposal

Developed by Pablo Murad — November 23, 2026 — pablo@pablomurad.com — https://rsw.works

Experimental Project Specification. This document is an experimental specification of the Really Small Web project. It is a work in progress and has not been published as an RFC or approved as an Internet Standard.

License: specifications CC0 1.0 (public domain); future reference implementation and official tools MIT.

RSW-0012 — JavaScript API

Revision 0.01 | Protocol rsw/1 | Format 1

Post-MVP — experimental draft; not enabled in basic rsw/1

Developed by Pablo Murad

Pablo Murad — November 23, 2026

pablo@pablomurad.com

MUST and MUST NOT indicate requirements of the described profile; SHOULD indicates a recommendation; MAY indicates an option. Experimental document, not yet ratified or implemented. The rules of RSW-0001 through RSW-0003 remain applicable.

01. Status

Post-MVP. Defines a candidate contract for JavaScript applications; it does not authorize execution in format=1, protocol=rsw/1. RSW-0002 requires an empty permissions and a text/static profile; a future generation formalized by RSW-0016 is a prerequisite for the interactive/connected profile. A v1 reader MUST NOT infer support from extensions or capabilities. No execution while the architecture and isolation are not audited.

02. Runtime and Identity

An application is a validated cartridge edition, with an isolated, ephemeral origin per execution. The host keeps a separate privileged API; content has no direct access to filesystem, shell, IPC, camera, microphone, clipboard, credentials, UI cookies, keys, or sockets. Scripts and local resources are only loaded from the validated inventory. Block direct network, fetch, WebSocket, and external subresources by isolation and an egress policy, not only by overriding JS objects. Prohibit remote code, eval/Function, plugins, WebAssembly, and workers until their own review. Separate processes and enforce limits when the engine allows; an iframe alone does not provide a CPU/memory budget.

03. Candidate Capabilities

storage.local: read/write of an own namespace, proposed quota of 256 KiB per (key, name); unsigned uses H_png and ephemeral storage. identity.read: access to the public key chosen by the user, without secrets. network.discover: query of local references; does not start a crawl. network.message: requests sending upon confirmation, using RSW-0008/0009/0014 when available. network.external: not supported in this first proposal. Permissions are denied by default, bound to identity, publication, and scope; an increase of permissions in a new edition requires new consent. Do not automatically inherit privileges through trust or a key change. Revocation stops pending operations.

04. Proposed API

Asynchronous interface with Promises; limited JSON responses, without privileged native objects. rsw.cartridge.info() returns {name,version,sha256,public_key}; for unsigned, public_key is absent. rsw.identity.get() returns {public_key} only with consent; rsw.storage.get(key), set(key,value), delete(key) under storage.local; key ASCII [a-z0-9._-]{1,64}, value JSON of up to 16 KiB, absent get returns null. rsw.network.discover() returns up to 32 local URIs, not global history. rsw.network.send(to,text) requests a

public message per RSW-0009, with privileged confirmation showing destination and text. A future private message must use a distinct method and RSW-0014. rsw.events.subscribe(type, callback) receives explicitly authorized events, without unlimited replay; returns a cancellation function. The host never offers a generic sign(bytes), exec(), readFile(path), or fetch(url).

05. Host/Application Channel

Dedicated channel per execution, preferably a MessagePort transferred at initialization by a trusted host, without accepting commands from generic window.postMessage. Message {id,method,args}, id an integer monotonic per channel, response {id,ok,result} or {id,ok,error}. Validate the initialization origin/context and the port's binding to the execution; do not trust an identity declared by the app. Limits: 16 KiB per call, 10 calls/s, 4 concurrent, 10 s timeout as initial targets. Methods and arguments by allowlist, without access to arbitrary host properties. Close the port on unload or revocation; prevent id reuse. Errors: DENIED, UNSUPPORTED, INVALID_ARGS, QUOTA, OFFLINE, TIMEOUT, CANCELLED.

06. Blockers for Activation

Ratify the manifest version and profiles; choose an engine that enforces real isolation and limits; specify module loading and network policy; test escape, XSS, exfiltration, prompt abuse, UI spoofing, races, persistence, and permission migration. Tests between apps of the same key and of different keys, revocation, update, storage quota, and a fake channel. Do not declare interactive conformance with only an iframe prototype. References: RSW-0002/0003/0008/0009/0011/0014/0016; HTML https://html.spec.whatwg.org/ ; CSP https://www.w3.org/TR/CSP3/ .

License: CC0 1.0 (public domain dedication; see CC0-1.0.txt in the downloads area). Developed by Pablo Murad | pablo@pablomurad.com END — RSW-0012

Author's Address

Pablo Murad
Independent
pablo@pablomurad.com
https://pablomurad.com
Official project website: https://rsw.works

Small files. Independent worlds.