Island

Solution Brief

Browser-Layer Security: Where It Fits in Your Architecture

A control point for the work that happens in the browser, alongside the stack you already run.

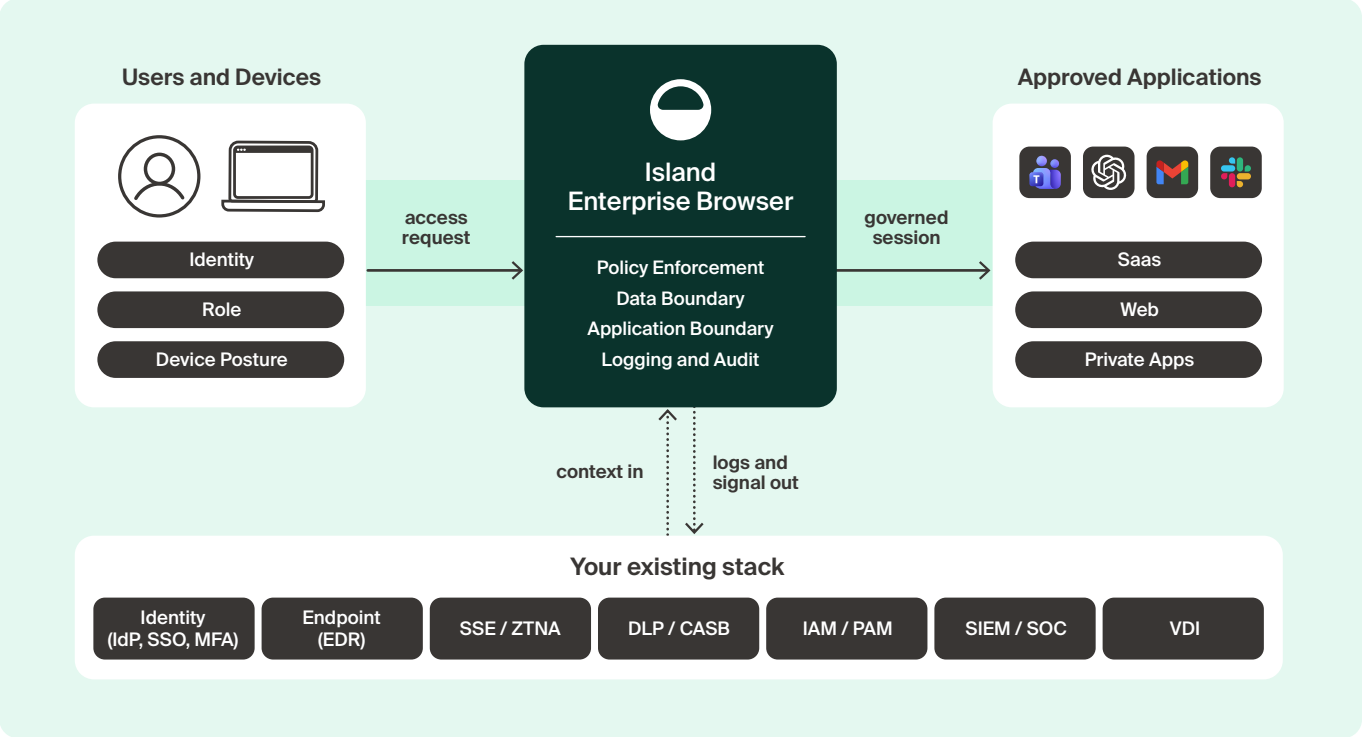
The browser is a control point, not just a client

Your identity, endpoint, and network controls decide who gets in, from which device, under which conditions. Once a session opens, in SaaS, web, and private apps, those controls have limited view of what happens next: what data moves, what actions a user takes, what leaves the session.

The browser is the one place that sees all of it. That makes it a control point, not just a client. This brief shows where that control point fits in your architecture, what it enforces, and where it isn't the right model.

The control model

A managed browser applies policy inside the session — using the identity and device context you already establish — and sends its signals back to the tools you already run.



Read left to right: the browser takes the user, role, and device context you already establish, enforces policy and data controls in the session, and opens a path only to approved applications. It draws context from your existing controls and returns its logs to them, so it strengthens the architecture rather than sitting outside it.

What the browser layer enforces

Identity-aware access

Access follows user, role, and device context, so one login can open different paths for different people.

Application and action policy

Rules apply by application and by action, not only by who authenticated.

Data-action controls

Govern copy, paste, upload, download, print, and local storage inside the session.

Access boundaries

Reach approved SaaS and web resources without extending broad network access to get there.

Logging and auditability

Capture session-level activity for investigation and audit.

Where it fits, and where it doesn't

The browser layer is the right model for work that already lives in the browser. It isn't a fit for everything, and it doesn't try to be.

Good candidates for browser-layer control:

- Secure workforce access
- Third-party and contractor access
- Sensitive SaaS and web work
- Browser-dominant workflows

Keep or validate separately:

- Desktop-dependent applications
- Specialized or peripheral-intensive tools
- Workflows that require broad network access
- Anything better served by a full virtual desktop

The goal isn't to move everything, it's to put control where the work is already browser-based, and leave the rest where it belongs.

It complements your stack

Island sits in the browser layer, alongside your investments, with no rip-and-replace.

Works with:

- Identity and access (IAM, PAM)
- Endpoint security
- SSE / ZTNA
- DLP / CASB
- SIEM / SOC
- VDI
- Zero Trust strategy

It uses their context to make decisions in the session, and feeds its logs back to them. The result is coverage where those tools stop: the live session, across SaaS, web, and private apps.

What changes

Control at the point of work

Policy and data handling apply where the action happens, not only at the door.

Consistent enforcement

The same policy holds across managed and unmanaged devices.

A narrower access path

Users reach approved apps without broad network exposure.

Evidence built in

Session activity is logged for audit and investigation.

Start with one workflow

You don't need to redesign your architecture to see whether this fits. Pick one browser-dominant workflow and check it against the model:

- The applications, users, and endpoints involved
- The identity and data sensitivity in play
- The controls already in place
- The criteria you'd hold it to

A short session is enough to map that workflow against your architecture and agree what a fit looks like.

Schedule a demo https://www.island.io/schedule-demo