

# Managed Components

Redefining third-party tools on the web,  
and beyond.



# Today's issues with third-parties

```
<script  
  src="//example.com/random/script.js">  
</script>
```



Performance



Security



Privacy

Zaraz, take 1

Do things  
that don't  
scale...

Zaraz, take 2

Do things  
that don't  
scale... *until*  
*they have to.*

# Managed Component

[ 'mæniɪdʒd kəm 'pəʊnənt ] noun

An open format for defining third-party tool behaviors on a website or an application. Managed Components can execute code outside and inside the user agent, can present interfaces in a parent application, can establish network connections and can provide custom APIs. Managed Components requires a runtime environment to work.

# Benefits of a Managed Component

- 👍 Events system
- 👍 Same domain
- 👍 Server logic
- 👍 SSR embeds
- 👍 Client events
- 👍 Pre-page-rendering actions
- 👍 Consent support

Pageview

```
export default function (manager) {
  manager.addEventListener("pageview", async (event) => {
    const { context, client } = event;
    fetch("https://example.com/collect", {
      method: "POST",
      data: {
        url: context.page.url.href,
        ip: client.device.ip,
      },
    });
  });
}
```

## Server Functionality

```
export default function (manager) {  
  manager.proxy("/api", "api.example.com");  
  manager.serve("/public", "public");  
  manager.route("/ping", (request) => {  
    return new Response(204);  
  });  
}
```



# Component Manager

## Implement the API

- Dispatch events
- Provide server-side capabilities
- Manage caching and storage
- Execute “server-side” code

# Component Manager

## Enforce user-configuration

- What module to load?
- When to load it?
- Which events should be passed to it?
- What permissions to grant a tool?

# Component Manager

## Potential implementations

- Proxy server
- HTTP server
- App SDK - Mobile / Desktop
- Service Worker

...?

# Open format, Open runtimes

- Specs defined by a committee
  - Tool vendors
  - Browsers
  - Infrastructure
  - Users
- Open source reference implementation
  - WebCM - [github.com/cloudflare/webcm](https://github.com/cloudflare/webcm)
- Open source the Zaraz library
  - Already > 30 releases components

# The future of third-parties

- Analytics tools that don't run in the browser
- Embed that are pre-rendered and cached
- Session recording tools that are light
- Complex widgets without security concerns
- Seamless Web <-> Mobile migration (& Desktop?)
- Controlled and audited execution environment
- **Standard way to deploy fast and safe third-party tools**