Holo Web Bridge v1.0 User Guide

A strategic gateway between web 2.0 and distributed applications

Overview

The Holo Web Bridge represents a transformative infrastructure component within the Holo Cloud ecosystem, enabling seamless integration between traditional web applications and distributed Holochain applications. This bridge technology fundamentally extends the reach of Holochain's distributed hash table (DHT) data by making it accessible through standard HTTP(S) in a read-only manner, creating powerful new possibilities for content delivery and data accessibility.

Structurally, the Web Bridge connects to Holo's `Holochain Cloud Nodes` and under-the-hood utilizes Holochain's HTTP Gateway to facilitate data access. This architectural approach successfully bridges the technological gap between Web 2.0 and Web 3.0 systems, creating new possibilities for distributed content delivery.

Technical Implementation Guide

Holochain HTTP Gateway and Holo Web Bridge Relationship

The Holo Web Bridge fetches data from Holo's `Holochain Cloud Nodes` by utilizing Holochain's HTTP Gateway. The Holochain HTTP Gateway is a tool designed for calling compatible zome functions over HTTP. When attaching a web bridge to a cloud node deployment, Holo automatically configures an HTTP gateway on each Cloud Node and exposes them to the public web at a URL unique to the deployment.

It's important to understand that the entire deployment is treated as a single entity with traffic routed to each Cloud Node, returning the response from whichever node is the first responder. This can affect data consistency since Holochain nodes work with eventual consistency—identical HTTP calls made simultaneously may return different responses if queried before the DHT reaches consistency for the queried data.

Configuration Process

In the current implementation, configuration requires specifying allowed zome calls through the HC_GW_ALLOWED_FNS_ environment variable. This list must be shared with Holo's deployment team.

The format of each allowed zome call is <zome_name>/<function_name>.

For example:

Unset

```
HC_GW_ALLOWED_FNS_=main/list_mews,main/count_likes
```

URL Format Specification

The Holo Web Bridge accepts GET requests in the following structured format:

Unset

```
http://<host>/<dna-hash>/<coordinator-identifier>/<zome-name>/<function-name>?payload=<payload>
```

Where:

- `<host>`: The URL for your deployment.
- `<dna-hash>`: A base64url-encoded identifier for the Holochain app's DNA.
- `<coordinator-identifier>`: A unique identifier for the app instance (e.g., a UUID).
- `<zome-name>`: The name of the zome (module) in the app containing the function.
- `<function-name>`: The specific function to invoke within the zome.
- `?payload=<payload>`: An optional base64url-encoded JSON string containing input data.

Examples of Implementation

1. **Retrieving a PDF without payload**:

Unset

```
https://web-bridge.holo.hosting/uhC0kwENLeSuselWQJtywbYB1QyFK1d-ujmFFtxsq6CYY7_Ohri2u/67d2ef2a67d4b619a54286c4/content/get_pdf
```

2. **Getting an attestation with payload**:

Unset

```
https://web-bridge.holotest.dev/uhC0kwENLeSuselWQJtywbYB1QyFK1d-u
jmFFtxsq6CYY7_Ohri2u/67d2ef2a67d4b619a54286c4/content/get_attesta
tion?payload=eyAiaGl2ZV9pZCI6Ik1UYzBNVEE4T0RnNU5EQTVOaTFpWm1WalpH
RXdaRFV4WVRNeE1qZ3oiLCAiY29udGVudF90eXBlIjogImh1bW1oaXZlLWV4dGVuc
2lvbi1zdG9yeS12MSIgfQo=
```

Payload Encoding Instructions

When your zome function requires input data, encode it as JSON, then convert it to base64url format:

1. Start with JSON data: `{"stream_id":"xx","content":"Hello"}`
2. Convert to base64url: `eyJzdHJlYW1faWQiOiJ4eCIsImNvbnRlbnQiOiJIZWxsbyJ9`

Various programming languages offer libraries to perform this encoding, or online base64 encoders can be used for testing purposes.

HTTP Response Codes

The Web Bridge returns standard HTTP status codes with JSON responses:

Code	Meaning	Example Response
200	Success	`{"data":"your_result_here"}`
400	Bad request	`{"error":"Invalid base64url payload"}`
403	Access denied	`{"error":"Function not exposed by gateway"}`
404	Resource not found	`{"error":"No app found for DNA hash"}`
405	Method not allowed	Response for using POST instead of GET
500	Internal error	`{"error":"Conductor error"}`

Note: Authenticated endpoints are not currently available and will be introduced in a future release.

Response Payload Handling

For developers transitioning from the Legacy Holo Network, there's an important difference in handling response payloads. In the Legacy Network, Chaperone decodes the msgpack-encoded zome call response from Holochain based on your defined types. In the new Allograph Network, Holo Web Bridge returns the result as a JSON. This can lead to subtle differences e.g. Uint8Arrays.

Technical Best Practices

1. **Allow Only Safe Functions**: Carefully select which functions to expose, limiting to read-only operations. The Web Bridge does not enforce read-only behavior, so misconfiguration could allow unintended data modifications.

2. **Manage Payload Size**: Keep payloads under the configured limit (default 10KB) to avoid 400 errors.

3. **Use HTTPS**: Always implement HTTPS (enabled by default) for production environments to ensure secure communication.

Implementation Path

To effectively implement the Holo Web Bridge:

1. **Identify Compatible Functions**: Determine which read-only zome functions should be exposed via the Web Bridge.

2. **Configure Allowed Functions**: Share the list of allowed functions with Holo's deployment team.

3. **Construct URLs**: Build HTTP GET requests following the specified URL format.

4. **Test Requests**: Use tools like `curl` or a browser to test the functionality:

Unset

```
curl -X GET
"https://web-bridge.holotest.dev/uhC0k...Y7_Ohri2u/67d2ef2a67d4b6
19a54286c4/content/get_data"
```

5. **Handle Responses**: Develop client-side logic to process the JSON responses appropriately.

Important Considerations for v1.0

- **Read-Only Not Enforced**: The Web Bridge will execute any allowed function, even those that modify data. Developers must carefully select which functions to expose.
- **Manual Configuration**: Allowed functions listing currently requires coordination with the Holo deployment team.

—--

Removed - good content below for a Blog, let's keep the User Guide more implementation focused for now and we can update it as the product evolves.

Strategic Application Benefits

Implementing the Holo Web Bridge delivers several transformative advantages for application developers:

1. Extended Content Reach

Applications built on Holochain can now expose select data and content to conventional web applications without requiring integration with specialized Holochain libraries. This strategic capability extends the potential audience of Holochain applications beyond those running Holochain conductors.

2. Read-Only Data Access

The Web Bridge is specifically designed to provide read-only access to DHT data, enabling safe exposure of Holochain content to web users while maintaining the integrity of the underlying data structures.

3. Data Sovereignty with Accessibility

Holochain applications inherently provide data sovereignty by design, with each node maintaining and validating its own data. The Web Bridge maintains this fundamental advantage while enhancing accessibility through standard web protocols.

Self-Service Evolution Roadmap

While the current implementation requires coordination with Holo's deployment team, a comprehensive self-service model is being developed to empower developers with greater autonomy:

Cloud Console Integration

The Web Bridge will soon be accessible as a self-service component through the Holo Cloud Console, enabling developers to:

- Configure and deploy Web Bridge instances independently
- Manage allowed function access through an intuitive interface
- Monitor traffic and performance metrics
- Scale resources based on usage patterns

API-Driven Management

A programmatic API layer will provide automation capabilities for Web Bridge deployments, allowing:

- Creation and configuration via API calls
- Integration with CI/CD pipelines
- Dynamic function allowlist management
- Operational monitoring integration

This evolution toward self-service represents a strategic initiative to empower developers while maintaining the security and performance guarantees of the Holo Cloud infrastructure.

Future Enhancements

The Web Bridge technology roadmap includes several planned improvements:

1. **Authentication Capabilities**: Support for authenticated endpoints to enable more sophisticated access control.
2. **Enhanced Security**: Improved function execution controls and security mechanisms.
3. **Self-Service Configuration**: Direct management of Web Bridge settings through the Cloud Console.

Conclusion

The Holo Web Bridge creates a powerful connection between traditional web applications and Holochain's distributed architecture. By following the technical implementation guidelines outlined above, developers can effectively expose DHT data to web clients, creating new possibilities for distributed application deployment while maintaining the fundamental advantages of Holochain's agent-centric design.