Smart Contracts serve the web

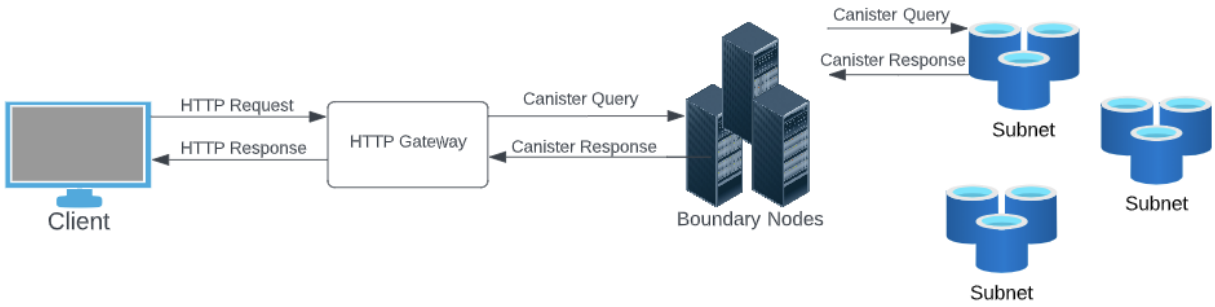
The Internet Computer is the only blockchain that can host a full dapp – frontend, backend and data. Any user can deploy their dapp as a canister (smart contract) on the Internet Computer. Canisters are computational units that bundle together code and state. Canisters can store data, deliver HTML, CSS and Javascript pages, and answer API requests. Canisters are incredibly fast and can deliver webpages within 200ms. Canisters can store up to 32 GB of data at an incredibly low cost ($5 per GB per annum). Browsing dapps hosted on the Internet Computer is as seamless as browsing Web2 apps hosted on cloud. All these factors enable developers to deploy even large-scale social media applications entirely on-chain without needing any cloud services. Try out a few dapps deployed on the Internet Computer.

Workflow

We now describe how a client can access a website deployed as a canister on the Internet Computer. The architecture involves 4 key components.

- Client - A device owned by the user. When the user browses a website, the client device sends a HTTP request.
- HTTP Gateway - A HTTP Gateway is a software that implements HTTP Gateway protocol. It converts HTTP requests into a format understandable by canisters. When the canister sends back a response, the HTTP Gateway converts the response into a HTTP response. The HTTP gateway can be run either on the client, on the boundary nodes or on independent servers.
- Boundary Node - Boundary nodes keep track of the architecture of the Internet Computer. In particular, boundary nodes keep track of the list of subnets, list of nodes on each subnet, the canisters run by each subnet, etc. On receiving a canister query, boundary nodes can route the request to one of the blockchain nodes running the canister.
- Canister - Developers can host their dapp as a canister. Canister consists of a bunch of methods. Anyone can send queries to the canister. A query consists of the canister method to

be executed and the inputs for the canister method. The Internet Computer receives the queries sent by the users, executes the corresponding canister method and returns the response to the user.



HTTP Gateway converts the format of HTTP Requests to canister queries, and canister responses to HTTP responses.
Boundary nodes route canister queries to appropriate subnet.

Deploying web apps on the Internet Computer

If a canister wishes to serve web content, it should implement a method that consumes a HTTP request (url, http method and headers) and outputs a HTTP response (status, headers and body). The canister method could return HTML, CSS and Javascript content as part of the HTTP response. Refer to Internet Computer Interface Spec for more details.

There's also an easy way to host existing static web apps (even those built using frameworks such as React and Angular) on the Internet Computer with minimal extra code by creating an "asset canister". An asset canister works similar to a regular canister, except that a lot of boilerplate code to host static websites is taken care of for us. To host a static website, we simply need to create a canister, specify its type as "asset" and specify the source folder of the web app. Once the asset canister is deployed to the Internet Computer, the website can be accessed at http://<canister id>.ic0.app and http://<canister id>.raw.ic0.app. See tutorial or watch the video.

HTTP gateway protocol

The browser only communicates with HTTP(s) protocol and doesn't know how to query a canister. To fill the gap between the browser and Internet Computer protocols, we utilize a HTTP Gateway, which is a software that sits in between the browser and the Internet Computer. The browser sends a http request to the http gateway. The gateway first interprets the URL in

the http request and extracts the corresponding canister id. It then converts the http request into a canister query and sends it to the boundary nodes. When the canister sends back a response, the http gateway interprets the response, verifies the signatures, converts into a http response and sends it to the browser.

There are many ways to implement the HTTP Gateway protocol. Currently, there are 2 implementations.

- The gateway protocol is implemented as a **service worker**. When the user enters a URL such as http://<canister id>.ic0.app, the browser calls the DNS service to resolve the query. Currently, the DNS maps the ic0.app domain to the **boundary nodes** of the Internet Computer. When the browser makes a request to a boundary node, it responds with a service worker that implements HTTP Gateway protocol. The browser then installs the service worker. From then on, whenever the user makes a request to http://<canister id>.ic0.app, the browser passes on the request to the service worker.
- Boundary nodes also implement the HTTP Gateway protocol. When the user enters a URL such as http://<canister id>.raw.ic0.app, the browser sends the http request to the boundary node, which acts as a http gateway. There are a few other ways of implementing the HTTP Gateway protocol. The gateway can be implemented as a browser extension. The chromium browser could also be modified to include HTTP Gateway as part of the browser.

SEO

The dapps running on the Internet Computer seamlessly integrate into the Web 2.0 world as crawlers are able to access them directly on-chain. This allows dapps to be indexed by search engines and for their metadata to be read in order to generate previews and cards on social platforms. A tutorial on using the Search Engine Optimization (SEO) features of the Internet Computer can be found in this **blog post**.

Serving web content

Building a front-end dapp on the IC

Hosting a static website on the IC

HTTP Gateway Protocol

[Web Serving Wiki Article]