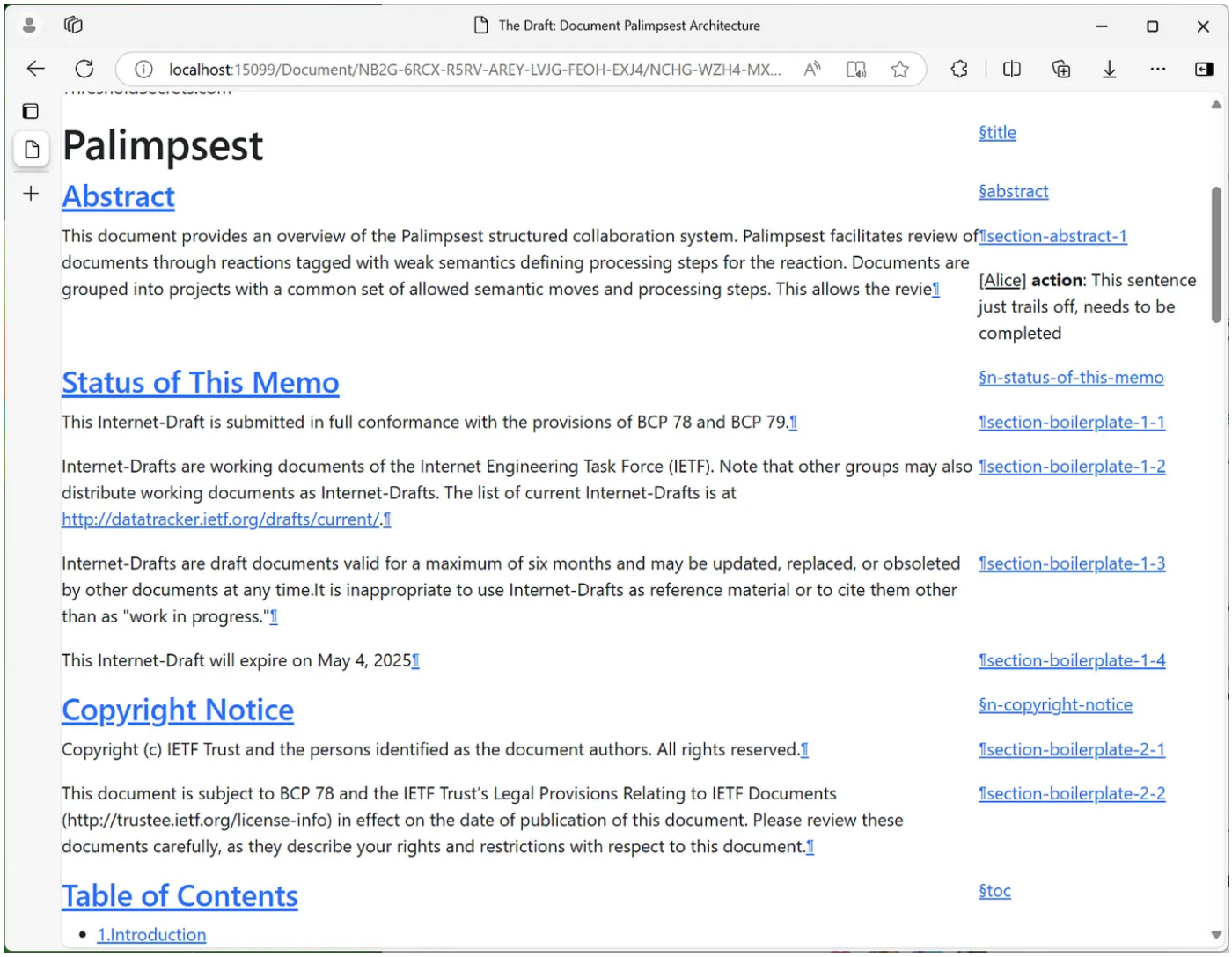
Connecting to a Blue Sky OAuth server Part 1

Phill Hallam-Baker Follow 5 min read · Dec 5, 2024

6

So you want to make use of BlueSky accounts in your non-BlueSky service? It is certainly possible, but finding out how to do it is hard. For some reason, anything connected with ‘identity’ causes specification writers to want to create a whole new rack of jargon and explain what it means with diagrams with arrows whizzing everywhere.

All I want to do is to enable people to use their BlueSky accounts to leave comments on my ‘Palimpsest’ document annotation and forum tool. So if someone reads one of my specifications and spots an issue, they can leave a note on that paragraph.



The Goal

The end goal is to allow people to enter their BlueSky handles into my forum server, do the OAUTH dance that redirects them to BlueSky to check it is them and create an account for their use on the Palimpsest server.

So instead of being ‘Alice’, Alice can be @alice.bsky.social and everyone using the forum will know that it is the same Alice they know from BlueSky.

Only here is the complicating factor, we are not actually using BlueSky handles, we are using ATprotocol handles and these can be created by anyone in any domain, not just BlueSky. And so, I quickly changed my

account from @hallam.bsky.social, to @phill.hallambaker.com.

The ability to use our own names and our own identity providers is one of the best features of the BlueSky ecosystem, it means that we can create accounts that belong to *us* and not to some corporation that might be bought by some random microdosing billionaire with a passion for fascist politics.

This is going to be a multi-part series of articles. In this article I am going to explain how to get from a handle '@phill.hallambaker.com' to the URI of the authentication and authorization server we are going to use.

Step 1: Convert the Handle to a DID

Handles are human readable names. Human readable is good for identifiers that are going to be put in front of a user. But they are not good as a machine identifier:

- If a handle is human readable, humans are likely to want to change it.
- DNS names only have meaning because they are resolved through the DNS. If the domain name registration they are bound to expires, the name stops working.

The type of identifiers BlueSky uses under the covers are identifiers that describe a public key that can be used to verify documents published to the ATmosphere.

These identifiers are encoded as W3C DIDs which are basically just a URI scheme that collects 'identity' identifiers together by slapping 'did:' followed

by a scheme specific label in front of the identifier.

The DID created for me by BlueSky is did:plc:k647x4n6h3jm347u3t5cm6ki.

To convert the handle to a DID, we do a DNS query for a TXT record at _atproto.<handle>.

TXT _atproto.phill.hallambaker.com ”
did=did:plc:k647x4n6h3jm347u3t5cm6ki”

Step 2: Resolve the DID

The next step is to resolve the DID to discover which service is currently servicing it. To do this, we make a HTTPS query to the resolution service

https://plc.directory/did:plc:k647x4n6h3jm347u3t5cm6ki

This step is not ideal in that PLC is a directory service that is run by BlueSky and thus represents a potential point of control for the whole system. But we don’t have to worry too much about that at this stage because we can define a new DID type later on.

You can see the full document returned by clicking on the link, for now we

```
"service": [
    {
        "id": "#atproto_pds",
        "type": "AtprotoPersonalDataServer",
        "serviceEndpoint": "https://shimeji.us-east.host.bsky.network"
    }
]
```


Step 3: Fetch the Resource Server metadata

This is another document and it is stored at <serviceEndpoint>.well-known/oauth-protected-resource. You can see the document describing my PDS here:

https://shimeji.us-east.host.bsky.network/.well-known/oauth-protected-resource

The Resource Server, known as a PDS in AT speak, is the service we would interact with if we were going to interact with the BlueSky world. This is something we might well be interested in at some point to allow a user to post a note to their BlueSky feed to say they have just done something in the private forum. But for now, we will ignore all of it apart from the part that tells us where to find the Authorization service:


```
{
    "resource": "https://shimeji.us-east.host.bsky.network",
    "authorization_servers": [
        "https://bsky.social"
    ],
    "scopes_supported": [],
    "bearer_methods_supported": [
        "header"
    ],
    "resource_documentation": "https://atproto.com"
}
```

Step 4: Get the Authorization Server description

At this point, we can grab the information telling us how to interact with the Authorization Server. This is another '.well-known' service, this time at /.well-known/oauth-authorization-server:

https://bsky.social/.well-known/oauth-authorization-server

Recap

At this point we have mined the BlueSky documents and extracted the information we need to interact with the OAUTH2 service itself:

- Resolve the handle to a DID via DNS
- Resolve the DID to a document describing the service binding
- Fetch the resource server metadata
- Fetch the authorization server metadata

The one thing we have not done in the example which we would definitely want to do in a production implementation is to cache any of this data. While there are 23 million Blue Sky accounts, these are hosted on a much smaller number of servers and so it is likely that most of them will be hosted on a small number of resource servers supported by a handful of authorization servers.

References

For more information see:

Handle specification: https://atproto.com/specs/handle

DID Specification: https://atproto.com/specs/did

PLC DID Documentation: https://github.com/did-method-plc/did-method-plc

The BlueSky description of the TXT record use diverges from the IETF standard which is RFC6763: https://www.rfc-editor.org/rfc/rfc6763

Rather than relying on the did entry being the first in the TXT record, I would strongly recommend not making that assumption as it is precisely the sort of change that standards organizations tend to insist on.

Oauth | Blue Sky | Open Services

Written by Phill Hallam-Baker

144 followers · 193 following

Follow