OAuth Identity and Authorization Chaining Across Domains

Arndt Schwenkschuster (Microsoft), Pieter Kasselman (Microsoft), Kelley Burgin (MITRE), Michael J. Jenkins (NSA-CCSS), Brian Campbell (Ping Identity)
Rough Agenda:
OAuth Identity and Authorization Chaining Across Domains

- Overview / refresher
- Changes since last time
- Open issues/questions
  - Next steps!

Abstract & ASCII

Defines describes a mechanism to preserve identity information and federate authorization across trust domains that use the OAuth 2.0 Framework.
A Life of WIMSE

- In Brisbane this work was presented to the newly formed Workload Identity in Multi System Environments (WIMSE) WG
What’s new in -02?

• remove recommendation to not use RFC8693's requested_token_type
• Corrected discrepancy between alphabetic numbering of the diagram and text in the resource acting as client example
# Open Issues & Requests to be Pulled

<table>
<thead>
<tr>
<th>Issue Description</th>
<th>Labels</th>
<th>Opened</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add sender constraining mechanisms</td>
<td></td>
<td>1</td>
<td>kburgin3</td>
</tr>
<tr>
<td>Recommended media type for JWT Authorization Grant</td>
<td></td>
<td>1</td>
<td>aaronpk</td>
</tr>
<tr>
<td>Should we allow identity chaining with DPoP tokens?</td>
<td></td>
<td>3</td>
<td>arndt-s</td>
</tr>
<tr>
<td>Describe or give examples of what kinds of tokens a client would exchange</td>
<td></td>
<td>4</td>
<td>aaronpk</td>
</tr>
<tr>
<td>Add use case to the appendix</td>
<td></td>
<td></td>
<td>PieterKas</td>
</tr>
</tbody>
</table>

__Closed Issues__

<table>
<thead>
<tr>
<th>Issue Description</th>
<th>Labels</th>
<th>Closed</th>
<th>Author</th>
</tr>
</thead>
</table>
| [Kelley] add sender constraints                                                   |        | 11     | arndt-s         | Changes requested
I've seen some things, man, and some stuff. I wouldn't recommend it.
“rough sketch of things that likely need to be accounted for or a least considered in this effort”

checking the binding on inbound token of **Token Exchange** request. No standard defined here. Might not even be feasible or make sense depending on deployment (i.e. the RS acting as the client doesn't have access to the bound key)
“… things that likely need to be accounted for or at least considered in this effort”

binding the token returned from Token Exchange. The request could be MTLS or have a DPoP header, which is kinda implied per standards. And the returned JWT could be bound with the respective cnf claim and method, which is maybe implied by standards but not explicitly written down anywhere (that I know of).
“… things that likely need to be accounted for or at least considered in this effort”

checking the binding on inbound token/assertion of the Access Token Request. The request could be MTLS or have a DPoP header, which is kinda implied per standards but kinda leaves much to the imagination, and checked against the cnf claim of the JWT authz grant. The presence of the cnf claim might be sufficient to say the thing is bound and suggest/require checking it (implementations I know about would just ignore it, however). But maybe a new grant type would make sense to clearly convey the intent (also using urn:ietf:params:oauth:grant-type:jwt-bearer for a non-bearer thing is awkward).
“… things that likely need to be accounted for or at least considered in this effort”

binding the resultant access token. The request could be MTLS or have a DPoP header and I think this is covered/implied by the existing specs to say that the issued access token should/could be bound a la DPoP Access Token Request and Mutual-TLS Client Certificate-Bound Access Tokens.
“… that likely need to be accounted for or at least considered in this effort”

<table>
<thead>
<tr>
<th>Resource</th>
<th>Authorization</th>
<th>Protected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server</td>
<td>Server</td>
</tr>
<tr>
<td>Domain A</td>
<td>Domain A</td>
<td>Domain B</td>
</tr>
</tbody>
</table>

The **Authorization server acting as client** flips around nearly all of the [prior] pieces to make them unworkable. I guess it'd need separate treatment with the definition of something like a proxied "trust me" this is the cnf I need in the final access token. IIRC Kelley had some prior work toward this end somewhere but I can't seem to find it at the moment. Or maybe sender constraining is just out of scope in the AS as client proxy case.
Next Steps ...

Never been to Dublin (IETF 121) so Bangkok (IETF 122)