Final Report of the OIDC Survey Working Group

Repository ID: TI.7.1

Authors: Steve Carmody
Eric Goodman
Patrick Radtke
Mark Scheible
David Walker
Albert Wu, Working Group Chair
Members of the OIDC Survey Working Group

Sponsor: InCommon Technical Advisory Committee
Superseded documents: (none)
Proposed future review date:
Subject tags: federation, OIDC, OAuth

Table of Contents

1. Executive Summary

2. Survey Methodology

3. OIDC/OAuth Use Cases

    3.1. Summary of Use Case Submissions

    3.2. Responses to Provided Use Cases

4. Highlights from the Survey Responses

    4.1. State of OIDC/OAuth Deployments

    4.2. Interest in Support for OIDC/OAuth

    4.3. Registration of RPs and OPs

    4.4. Attribute Release

    4.5. Software Development

5. Further Examination of the Issues

    5.1. Examination of Needs

    5.2. Additional Observations

6. Recommendations

    6.1. Support OIDC/OAuth in TIER Products

    6.2. Develop OIDC/OAuth Federation Standards and Practices

7. Appendix: Survey Questions

    7.1. Glossary of Terms Used in the Survey

8. Appendix: Full Text of Submitted Use Cases

9. Appendix: Feedback Received from the Community Consultation for This Report

10. Appendix: OIDC Survey Working Group Charter

    10.1. Problem Statement

    10.2. Charter

    10.3. Work Products

11. Appendix: Working Group Membership

1. Executive Summary

There has been a noticeable movement in higher education toward exploring/adopting alternate authentication/authorization protocols beyond SAML. Among them, OpenID Connect (OIDC) and OAuth (OAuth2) appear particularly popular. Today, InCommon has little involvement in this area of activity.

The InCommon Technical Advisory Committee (TAC) chartered the OIDC/OAuth Survey Working Group to survey the higher education community to identify OIDC/OAuth use cases, to gauge interest in adoption, and to recommend (if any) next step actions for InCommon and its related communities.

This report shows that the survey results confirm a strong community interest to adopt OIDC/OAuth. The move is driven primarily by three trends:

- The adoption of API-centric application architecture - organizations need to enable end-to-end authentication and authorization between SSO-enabled web applications and the APIs behind the applications.
- Native (mobile) application deployment - campuses are deploying native mobile applications seeking compatible authentication and authorization solutions for native applications accessing campus resources (usually in the form of API).
- Social and SaaS platform integration - organizations either need to support social identities in its applications, or need to integrate campus SSO with Social/SaaS applications that only support OIDC/OAuth.

84% of responses indicated that OIDC/OAuth support should be built into Shibboleth and/or Internet2 TIER products. Nearly half wants InCommon to develop OIDC/OAuth-based federation interoperability standards.

It is important to act now. OIDC/OAuth deployments have started and are ongoing, and each one moves us further from common, interoperable solutions for research and education. We recommend that Internet2 Trust and Identity initiate two parallel areas of activity related to OIDC/OAuth:

Support OIDC/OAuth in TIER products - Extending TIER products (Shibboleth and others) to fully support OIDC/OAuth protocol enable higher education community members to leverage their current investments in Shibboleth and other TIER products to enable end-to-end authentication and authorization between SSO-enabled web applications and the APIs behind the applications. It also provides a native mobile application friendly authentication and authorization solution. Further, having built-in support for OIDC/OAuth simplifies campus' effort to enable social platform integration.

Facilitate the development of OIDC/OAuth federation standards and practices - There is currently little to no federation support in OIDC/OAuth protocols. As the higher education community adopts OIDC/OAuth enabled services, we will need to address use cases, common to today's multilateral federation. While this is not an urgent need, the lead time to develop standards and practices in this area is long. InCommon should begin coordinating efforts with other federations to develop federation standards for OIDC/OAuth.

These activities are detailed in the Recommendations section of this report. They will require coordination and collaboration with multiple efforts both nationally and internationally.

2. Survey Methodology

The OIDC/OAuth Survey consisted of a mix of multiple choice and open ended questions. The survey attempts to determine:

- How interested is the community in adopting OIDC/OAuth protocols?
- How important is OIDC/OAuth support in Internet2 TIER offering?
- How important is federation support in OIDC/OAuth, and in what form?
- Among adopters, what are the tools/products in use today?
- If there is demand to adopt OIDC/OAuth, what are the implementation priorities?

Further, the survey is designed, as much as possible, to elicit actual OIDC/OAuth adoption use cases at organizational/campus level. The survey is constructed in 3 sections:

- The first section collects basic respondent demographic information such as location, job role, and contact information for follow up
- The second elicits use cases and gauges adoption interest in the community
- The third section probes deeper into OIDC/OAuth implementation specifics for those who have already deployed OIDC/OAuth solutions

Invitations to complete the survey were sent to the usual identity federation mailing lists, including InCommon Participants, InCommon Announce, and REFEDS. The survey group felt it was important to survey higher education application and technical communities, beyond the typical identity management circle. Additional invitations went to ITANA, EDUCAUSE, CLAC, BTAA/CIC, and several University of California and Cal State technical mailing lists. Further, the invitation asked everyone to forward the invitation to his/her respective IT communities.

A copy of the survey questions is attached in 9. Appendix: Survey Questions.

3. OIDC/OAuth Use Cases

3.1. Summary of Use Case Submissions

This section summarizes the open ended use case responses detailed in 10. Appendix: Full Text of Submitted Use Cases

Use Case Pattern	Number of Use Cases
Authorization of REST APIs, use API Gateway Service, or provision to cloud-based providers (e.g., BOX)[1]	29
Authentication from mobile devices[2]	12
Allow access with social credentials to campus resources (e.g., parents, applicants, alumni)[3]	8
SAML<->OIDC interoperability (gateway)[4]	4
Mid-tier(portal) can obtain and present delegated credentials to backend service[5]	3
Using a product that supports only OIDC[6]	3
Standard requires use of OIDC[7]	2
Use OIDC/OAuth for authentication to our on campus applications[8]	1
Impose some form of campus governance and policy over use of campus-associated accounts[9]	1
Looking to integrate two vendor products, both support OIDC/OAuth2[10]	1
Unable to categorize[11]	7

[1] Submissions: 6, 7, 8, 9, 10, 14, 16, 18, 20, 21, 22, 26, 27, 31, 32, 35, 36, 37, 38, 39, 42, 43, 44, 46, 48, 49, 50, U1, U4

[2] Submissions: 5, 9, 16, 25, 28, 34, 35, 37, 42, 48, U3, U4

[3] Submissions: 11, 12, 17, 19, 29, 41, 42, 48

[4] Submissions: 47, U2, U3, U4

[5] Submissions: 8, U1, U4

[6] Submissions: 13, 21, U2

[7] Submissions: 40 (FHIR), 39 (FHIR)

[8] Submission: 2

[9] Submission: 1

[10] Submission: 3

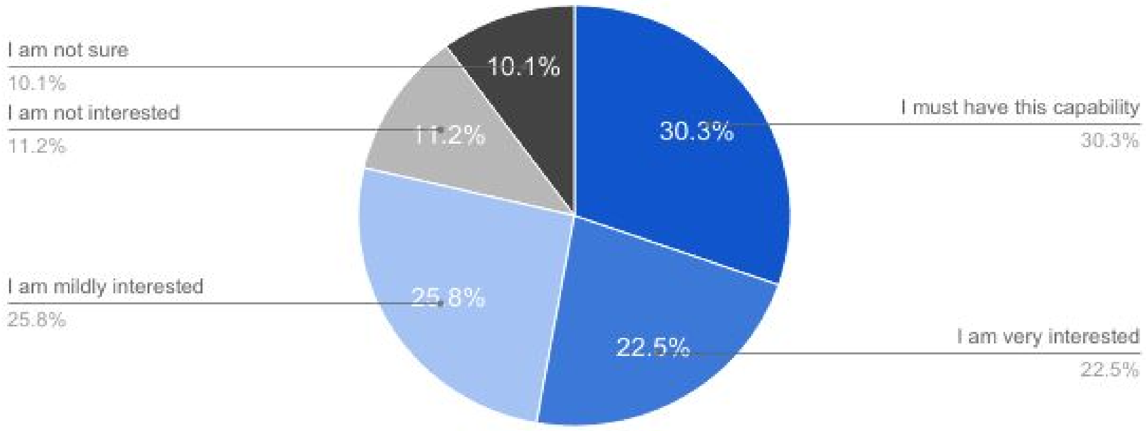
[11] Submissions: 15, 22 (para 1), 23, 24, 30, 33, 45

3.2. Responses to Provided Use Cases

In addition to soliciting open ended use cases, the survey asked the respondents to assess their interests in several archetype use cases previously submitted to the survey group during survey development:

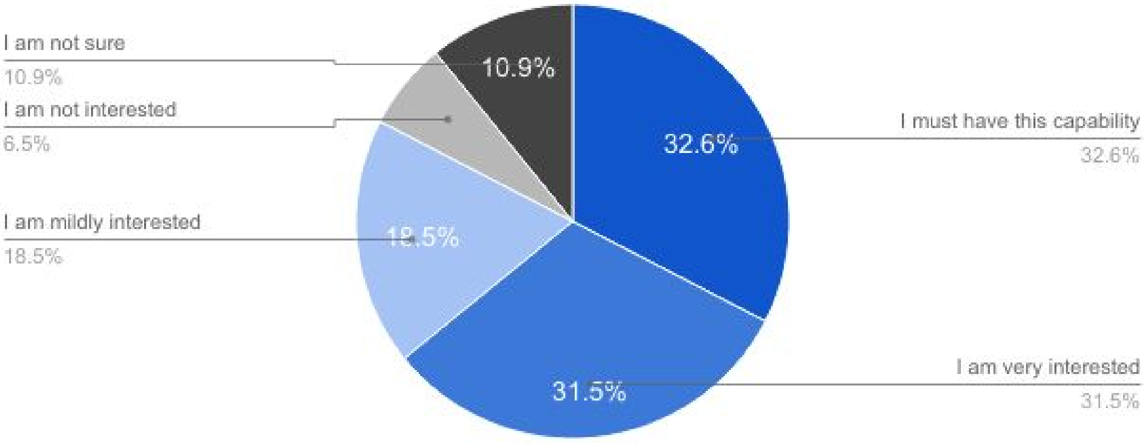
Use Case 1: As an API deployer, I need to authenticate the client applications accessing my APIs, but not the user of the client application (client credentials flow in OAuth)

Response	Count	Percent
I must have this capability	27	30.3%
I am very interested	20	22.5%
I am mildly interested	23	25.8%
I am not interested	10	11.2%
I am not sure	9	10.1%



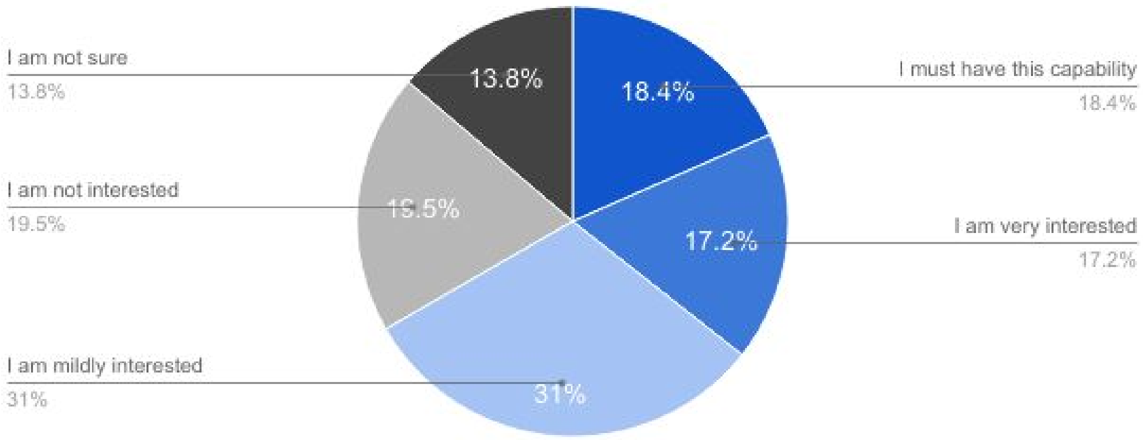
Use Case 2: As an API deployer, I need to authenticate the client application accessing my APIs. In addition, I need to know that the user has authorized the client application to act on their behalf (three-legged authentication in OAuth)

Response	Count	Percent
I must have this capability	30	32.6%
I am very interested	29	31.5%
I am mildly interested	17	18.5%
I am not interested	6	6.5%
I am not sure	10	10.9%



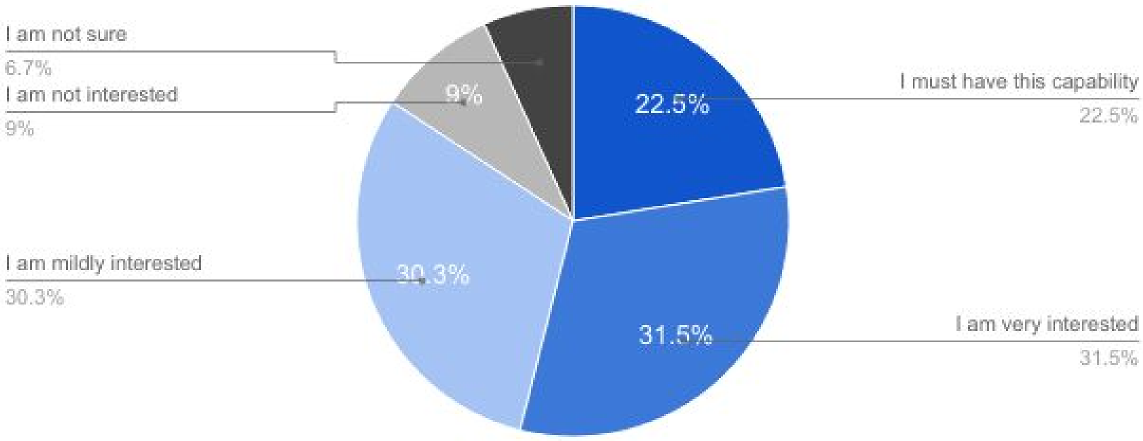
Use Case 3: As a learning management system operator, I need to integrate my LMS with learning resource providers

Response	Count	Percent
I must have this capability	16	18.4%
I am very interested	15	17.2%
I am mildly interested	27	31.0%
I am not interested	17	19.5%
I am not sure	12	13.8%



Use Case 4: For my OIDC/OAuth-enabled application, I would like to let users sign in from multiple universities using their respective campus credentials

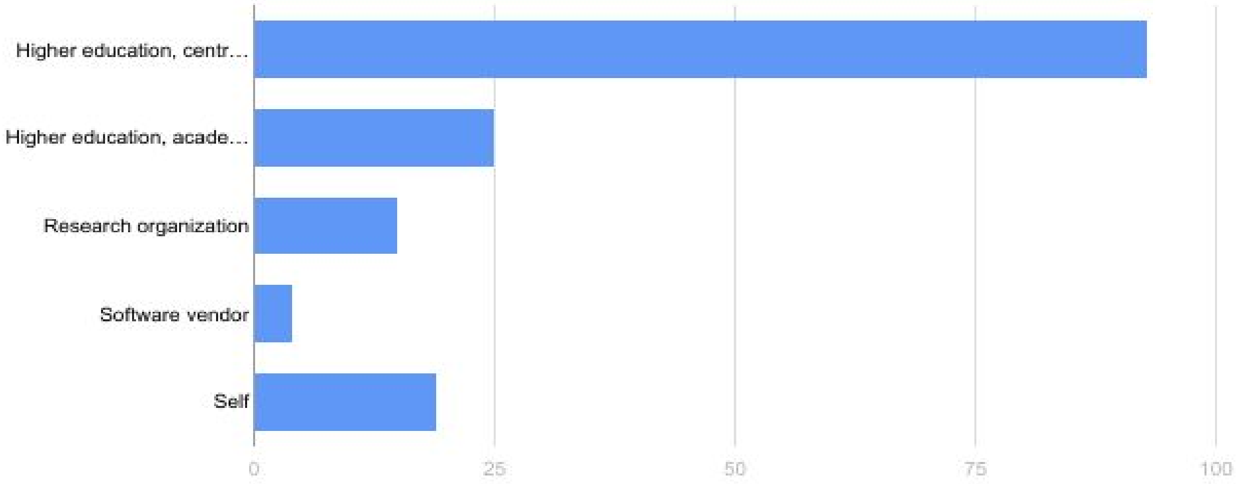
Response	Count	Percent
I must have this capability	20	22.5%
I am very interested	28	31.5%
I am mildly interested	27	30.3%
I am not interested	8	9.0%
I am not sure	6	6.7%


I am not sure
6.7%
I am not interested
9%
I am mildly interested
30.3%
9%
22.5%
30.3%
31.5%
I must have this capability
22.5%
I am very interested
31.5%

4. Highlights from the Survey Responses

There were 143 responses to the survey, 90% from the US. 13% represented only themselves, with the remaining 87% representing some form of organization with the following breakdown.[12]

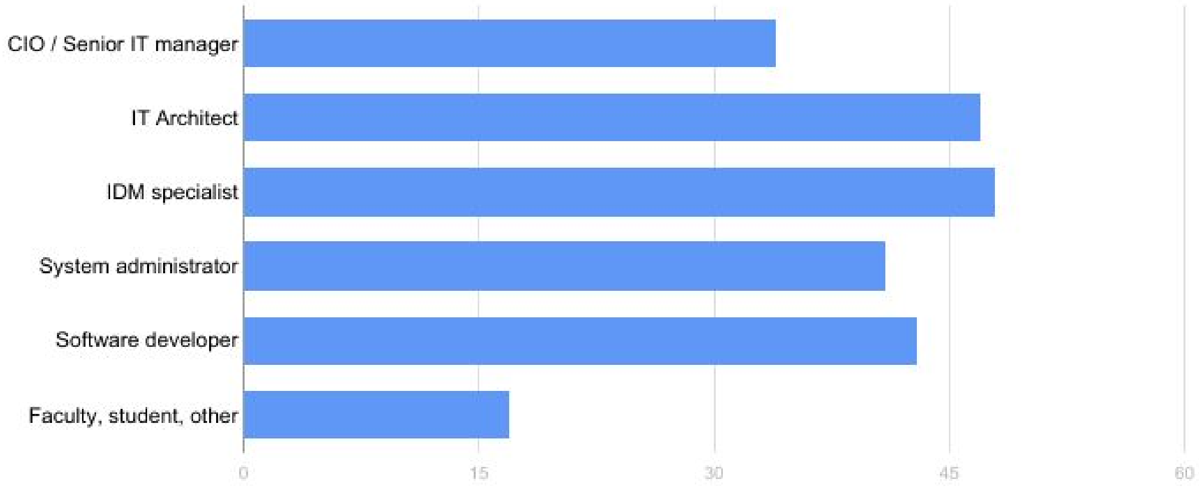
Organization	Count	Percent
Higher education, central IT	93	65.0%
Higher education, academic department	25	17.5%
Research organization	15	10.5%
Software vendor	4	2.8%
Self	19	13.3%



The respondents identified themselves in the following categories.[13]

[12] Questions 2 and 3
[13] Question 4

Role	**Count**	**Percent**
CIO / Senior IT manager	34	23.8%
IT Architect	47	32.9%
Identity management specialist	48	33.6%
System administrator	41	28.7%
Software developer	43	30.1%
Faculty, student, other	17	11.9%



4.1. State of OIDC/OAuth Deployments

15% of respondents said that they currently use OIDC/OAuth in production. An additional 12% are currently implementing. 24% have plans to implement but no funding, and 54% are monitoring without specific plans. 5% have no interest in OIDC/OAuth.[14]

The leading need for OIDC/OAuth is API authentication (72%), followed by mobile applications (55%), and off-the-shelf services (41%) and applications (16%) that require OIDC. 30% of

[14] Question 6

respondents feel that OIDC/OAuth provides an easier mechanism to register client applications than SAML, and 20% feel that OIDC/OAuth software is easier to implement than SAML.[15]

53% of respondents do not operate their own OIDC Provider (OP). There is no dominant OP that is used, but the two leaders are Google (13%) and Azure AD (9%).[16]

94% of institutions have a unique, never-reassigned identifier they could use for OIDC compliance.[17]

4.2. Interest in Support for OIDC/OAuth

84% of respondents would like to see support for OIDC/OAuth in Shibboleth, but only 4% have resources they could contribute to that effort.[18]

A large majority of respondents saw a need for OIDC/OAuth support for education and research's current federation models, but it was not universal. 68% said it was essential or very important, while 20% said it was nice to have, and 3% said it wasn't important. 12% were not sure.[19]

4.3. Registration of RPs and OPs

24% of respondents felt that any client/RP should be allowed to register with their OP without approval; others felt that some form of approval or affiliation with the OP's organization should be required. 43% would like the federation to have a registration process to help them decide which RPs to trust.[20]

94% of RPs felt they need a registration process to control which OPs they will trust.[21]

4.4. Attribute Release

The following table shows the percent of respondents that required different forms of control over attribute release.[22]

By default, any information may be released by your OP,	21%

[15] Question 7
[16] Question 9
[17] Question 17
[18] Question 11
[19] Question 12
[20] Question 14
[21] Question 16
[22] Question 15

if the user consents.	
By default, only certain information may be released by your OP, if the user consents.	56%
By default, no information may be released by your OP, even if the user consents.	15%
Certain information may be released by your OP to specific RPs without user consent.	53%
Users may delegate specific RPs the right to release certain information to other specific RPs.	35%
RPs may be delegated the right to release certain information to other specific RPs without user consent.	6%

4.5. Software Development

Developers using OIDC/OAuth are building all types of applications in our survey. Restful APIs are in the lead with 89%, followed by traditional web applications (80%) and mobile applications (71%), but server applications (60%) and single-page web applications (57% - the lowest category) are also high.[23]

Developers are choosing a wide variety of development frameworks and languages. Nothing is truly dominant.[24]

[23] Question 18

[24] Question 19

5. Further Examination of the Issues

5.1. Examination of Needs

The survey responses identify three primary OIDC/OAuth use cases:

> **API Access -** The majority of respondents are deploying or wish to deploy OIDC/OAuth as an authentication and authorization mechanism for API access. Whether social platform APIs (e.g., Google API) or custom-built APIs, OIDC/OAuth is emerging as the authentication/authorization protocol(s) of choice among application developers building today's API-centric applications.
>
> In particular, API operators need to authenticate and authorize the human as well as the device or application used by the human to interact with an API. One example is an n-tiered portal architecture, where a user portal interacts with back-end API services to perform business transactions. There is currently no widely adopted standard for APIs to obtain delegated assertions from the user for use by the portal.
>
> **Native Mobile Application Authentication** - Developers building native mobile applications wish to leverage campus SSO to authenticate the user while accessing backend APIs using OAuth. Absent native OAuth support within Shibboleth, developers are deploying their own SAML/OAuth gateway solutions. Today, there does not appear to be a natural consensus on the product used to perform this gateway function.
>
> **Social / SaaS Platform Integration -** Campuses wish to allow affiliates (parents, guests, applicants, alumni) access to campus services using their social credentials. Alternatively, campus community members may wish to use campus credential/SSO services to access SaaS services that only support OIDC/OAuth as an SSO solution. OIDC/OAuth support in core Internet2 Trust and Identity products can significantly ease integration and ongoing identity management efforts.

Further, 84% of responses requested that OIDC/OAuth support should be built into Shibboleth and/or Internet2 TIER products. 43% of respondents want InCommon to develop an OIDC/OAuth-based resource provider (RP) registration process to help them decide which RPs to trust.

The survey results suggest strong community demand for OIDC/OAuth support in InCommon services and TIER products. Although many submitted use cases do not require immediate services supporting federated interoperation, the responses indicated significant interest in developing and defining standard interoperation practices (e.g., standardization of claims, endpoints, etc). One particularly notable point is that there is currently significant fragmentation in OAuth product deployments among those who have deployed OAuth support. Absent built-in OIDC/OAuth support in TIER products, campus IDP and SP operators are likely to continue to adopt one-off solutions. None of these support federation use cases, because there are no widely adopted federation and interoperation standards in OIDC/OAuth protocols.

We appear to be in the same early adoption stage with OIDC/OAuth as we once were with SAML and Single Sign-On. In the early days of InCommon, Shibboleth deployed widely before federation support of SAML fully matured. The widespread deployment of the Shibboleth software significantly eased the deployment of federation practices. If each organization continues to adopt its own flavor of OAuth, it becomes much more difficult to deploy federation policy/practice changes by leveraging common software and standards.

OIDC/OAuth adoption is definitely on the rise. Campus IDM and application operators are increasingly pressed to support OAuth protocol in some way. OIDC/OAuth adoption is a disruptive force in the current SAML-only federation model. We believe it is important that Internet2 and InCommon begin a prioritized effort to incorporate support for OIDC/OAuth in its products and practices. Failing to do so in a timely manner will likely erode InCommon’s role in the community and its ability to continue to foster continued federated interoperation.

5.2. Additional Observations

The following are observations made during the working group’s discussions of the survey results.

- The survey was titled “OIDC/OAuth2 Use”; this likely biased the set of respondents, and biased how they framed their responses. It is easy to read their responses in an OIDC context when they may be describing protocol independent functionality and use cases.
- In the responses to Question 11, it was surprising to the group that only 6% of respondents said that “RPs may be delegated the right to release certain information to other specific RPs without user consent.” We believe it is common to release attributes to institutions’ enterprise software services without user consent.
- SAML is not going away, and there will be successors to OIDC/OAuth. We need to adopt a multi-protocol support model.
- A number of the submissions mentioned that it was "easier" for applications to use OIDC/OAuth2 than to use SAML. We believe this is true for simple use cases, certainly easier today for apps on mobile devices. More complex use cases that require, for example, a rich set of attributes, support for multilateral federation, or RP certification (e.g., R&S), however, are more readily implemented in SAML, where guidelines and standards are more available.

  OIDC may also require less coordination with others within the institution. SAML is more likely to require installation of additional libraries by system administrators, and compliance with (SAML) federation policy may require working with designated officials within the institution.

- There may be a significant number of use cases that do not require multilateral federation, in fact many require a specific OP. As mentioned in 4.1. State of OIDC/OAuth Deployments, 22% of the respondents use either Google or Azure AD for their OP.
- Historically, OIDC/OAuth implementations on mobile devices have utilized an “embedded web view,” which can make user-entered credentials available to the application, or have simply prompted for user credentials directly. This represents a security issue, as those applications must be trusted not to capture login secrets. More recently, however, Facebook and Google are addressing this by moving to a more secure strategy, similar to that used for SAML, where the local browser is invoked to request user credentials, where they are not visible to the application. In IOS, this strategy is called “Safari View Controller;” in Android, it is called “Chrome Custom Tabs.”
    - There’s a need for guidance on authenticating and/or federating mobile apps in general (not just OIDC/OAuth). Many see OIDC/OAuth as a solution targeted at mobile apps, but given the direction of mobile authentication, SAML may be just as reasonable to implement.
- Organizations are deploying API Gateway products to provide security and governance across all APIs. Internet2 Trust and Identity architects should reach out to the deployers and consider the role of API Gateways in the architecture.
- It is important to act now. OIDC/OAuth deployments are ongoing, and each one moves us further from common, interoperable solutions for research and education. There is even, in fact, the perception in some sectors (e.g., application developers) that SAML (and, by extension, InCommon) will soon no longer be relevant. Regardless of the reasoning, supporting the needs of applications is, nonetheless, crucial to InCommon’s success.

6. Recommendations

The survey responses indicate strong community demand to support OIDC/OAuth in the coming months. The workgroup recommends that Internet2 Trust and Identity initiate two parallel areas of activity related to OIDC/OAuth now.

First, we believe it is important that Internet2/InCommon take immediate action to build OIDC/OAuth support into TIER products (Shibboleth, TIER API, and others). Doing so allows higher education community members to leverage their current investments in Shibboleth and other TIER products to enable end-to-end authentication and authorization between SSO-enabled web applications and the APIs behind the applications. It provides a native mobile application friendly authentication and authorization solution. Further, having built-in OIDC/OAuth support in Shibboleth IDP simplifies campus' effort to enable social platform integration. Section 6.1 details specific recommended actions.

Second, InCommon should begin coordinating efforts with other federations and the standards bodies to develop federation standards for OIDC/OAuth. There is currently a significant lack of support for multilateral federation use cases in OIDC/OAuth. As the higher education community adopts OIDC/OAuth enabled services, we will likely need to tackle use cases common to multilateral federation. While this is not an urgent need, the lead time to develop standards and practices in this area is long. Several groups in European research and education community have begun working on federation support in OIDC/OAuth. Section 6.2 lists these federation work recommendations.

6.1. Support OIDC/OAuth in TIER Products

The community's demand for OIDC/OAuth appears to focus on bilateral use cases at the moment. While InCommon and other federations tackle the challenge of developing federation protocols and practices in OIDC/OAuth, Internet2 Trust & Identity should develop plans to build core OIDC/OAuth support into TIER products. The work includes extending Shibboleth to support OIDC and OAuth as well as implementing OAuth support in key developments such as Grouper and the TIER product APIs. In particular, we recommend the following actions:

Product Implementation

- Add OpenID Provider (OP; equivalent to IDP in Shibboleth) functionality to the TIER product to support end-to-end API and native mobile app authentication and authorization use cases. A likely approach is to extend Shibboleth IDP to function as an OIDC OP.
- Add OAuth Authorization Server functionality to the TIER product. Follow-on implementation teams should determine the best architectural approach to add OAuth Authorization Server into the TIER portfolio.

- Adopt OIDC/OAuth as the access protocol in TIER API that is currently under development.
- Examine strategy to support social platform integration where an application wishes to support access using social credentials.
- Consider the need for an extensible architecture capable of supporting multiple identity protocols. This could include use of proxies as well as direct inclusion of protocol support into products.
- Convene a follow-on technical working group to identify, research, and address architectural and implementation details for OIDC/OAuth within TIER products. This working group could be a possible work plan item for CACTI.
- Work with the organizations (such as Duke, Columbia, UCLA, and others) that are currently implementing these use cases to identify areas where organizations would be helped by increased standardization (e.g., use of OIDC claims, development of deployment profiles, etc.).
  - Track and document the lessons learned; develop recommended practices.
  - Create channels for sharing information within and from the existing Higher Ed OIDC/OAuth2 deployers and interested parties. This could include email lists, wiki pages, and regular webinars.

Standards Development

- Work within existing efforts or create new efforts to develop the required standards. New efforts might be created within existing standards bodies (e.g., I2 T&I, REFEDS, Kantara, http://openid.net/, etc.)
- Develop and share information about best practice with native mobile application authentication using SAML and OIDC/OAuth2.

Financial Impact

- Adding OIDC/OAuth support to core products, especially Shibboleth, will require substantial financial and technical investment. Internet2 Trust and Identity should consider an alternate funding model or implementation partnerships. Possible candidates might be campuses who are already deploying OAuth solutions, international organizations (REFEDS?) who are tackling similar challenges, and commercial companies who provide SAML/OAuth gateway solutions today.

It's worth reiterating that building OIDC/OAuth support into TIER products, with or without mature federation standards implementation, has significant immediate value: campuses using Shibboleth today gain the immediate benefit of being able to support OIDC/OAuth without having to manage additional products. More significantly, it maintains Shibboleth and related products' positions as the driving deployment vehicle for Higher Education federation practice changes.

6.2. Develop OIDC/OAuth Trust Federation Standards and Practices

While TIER product teams implement initial, core OIDC/OAuth support, a parallel group, likely a collaboration between InCommon and global federations, should work to develop a multilateral trust federation model, including policies and practices for OIDC/OAuth protocols. OIDC/OAuth has few standards around multilateral federation support. There is much work to do in this area, and the TIER products will require these new protocols and standards to emerge quickly in order to sustain its continued support for OIDC/OAuth. InCommon should either convene a follow-on InCommon working group or join existing efforts in European federations to develop federation protocol, policies, and standards in OIDC/OAuth. Work may include:

- Develop higher education deployment profile for OIDC/Oauth (e.g., profile similar in concept to the one for healthcare: http://openid.net/wg/heart/)
- Ensure the development of a Higher Ed attribute schema for OAuth2 claims (i.e., map eduPerson schema to OIDC/OAuth compatible format, likely in JSON Web Token forms), possibly by participating in the current European effort: https://wiki.refeds.org/display/GROUPS/Mapping+SAML+attributes+to+OIDC+Claims
- Build on Roland Hedberg's efforts to develop OAuth federation framework (which surfaces within both OTTO and an OIDC Federations WG).

7. Appendix: Survey Questions

1	What is the name of the institution you represent?
2	What type of institution do you represent?
3	Where is your institution located?
4	What is your role at your institution (check all applicable roles)?
5	This survey may lead to follow up work in InCommon or Internet2. If you would like to stay connected, please let us know how to contact you.
6	What is your involvement with OIDC/OAuth (please check all that apply)?
7	Why are you interested in deploying OIDC/OAuth? (please check all that apply)
8	We'd like to know more about how you use (or plan to use) OIDC/OAuth. Please describe your OIDC/OAuth use case(s). (Provide as much detail as you would like. The text box can be resized.)
9	Which OpenID Connect OP or OAuth 2.0 server product do you use? (Please check all that apply)
10	The following are some of the already contributed use cases. Please rank your interest in these.
11	Should OpenID Connect/OAuth 2.0 be built into future Shibboleth or other Internet2 TIER offering(s)?
12	If Internet2 TIER offers support for OpenID Connect/OAuth 2.0, how important is support for current models of federation in higher education (InCommon, eduGAIN, etc.) where a user accesses a resource using a credential issued by their home institution?
13	We have a few more questions for people who have deployed or are deploying OIDC/OAuth. Would you like to answer them?
14	Who should be allowed to register OIDC/OAuth clients? (please check all that apply)
15	Policies concerning information release, user consent, and delegation. Check all that apply.
16	Will your OIDC RP trust any OP, or will you need a registration process (perhaps offered by the federation) for OPs?
17	OpenID Connect requires each user to have a unique, never re-assigned identifier (e.g.,

	eduPersonUniqueId). Do your users have a unique, never re-assigned identifier that you could use for this purpose?
18	What are you developing that you will use OIDC or OAuth2? Check all that apply
19	What development frameworks and languages do you use with OIDC or OAuth2?
20	Is there anything else you would like us to know?

7.1. Glossary of Terms Used in the Survey

- **OIDC/OAuth.** The family of protocols known as OpenID Connect, OAuth 1.0, or OAuth 2.0.
- **Identity provider.** The generic concept of an entity that can provide [authentication and] identity information, i.e., a SAML Identity Provider (IdP) or an OpenID Provider (OP).
- **OpenID Connect (OIDC).** See http://openid.net/connect/faq/.
- **OAuth.** See https://en.wikipedia.org/wiki/OAuth. Unless otherwise noted, this survey uses OAuth to refer to either version of OAuth (OAuth 1.0 and OAuth 2.0).
- **OpenID Provider (OP).** In OIDC, a service that provides identity information.
- **Relying Party (RP).** In OIDC, a service that requires identity information from an OP.
- **InCommon Federation.** See https://www.incommon.org/federation/.
- **Internet 2 TIER.** See http://www.internet2.edu/vision-initiatives/initiatives/trust-identity-education-research/.

8. Appendix: Full Text of Submitted Use Cases

1	Would be good to replace the existing use cases where people have set up connections to oAuth at Microsoft or Google using our university accounts, and they have managed to do so without governance. This may be why we are not actually receiving a large influx of requests to support this technology, rather, just a few requests.
2	We currently use OIDC/OAuth for authentication to our primary IdM applications (administrative account management, HR enrollment of new user accounts, new user enrollments, password changes, etc.) We also use OIDC and OAuth (and -ish) via Cirrus Identity for social-to-SAML use cases. We have had a couple of instances where vendor applications could support OIDC more readily than SAML, but we haven't currently been prepared to open our existing server up broadly. (Note that in these cases, the vendors felt that it would be _easier_ to implement; they didn't already have a compatible SSO protocol. SAML, and in particular, the Shibboleth SP, was simply a difficult integration model.)
3	- Google Apps for Education - Atlassian integration
4	No specific use case at this time.
5	Our largest interest is the use of OIDC / OAuth to authorize mobile applications and APIs. We do not currently have an official OIDC / OAuth service offering, though I believe that some on our campus may be piggy-backing on our Google Apps for Education instance as an OIDC provider.
6	We would like support for end users and end user apps/applications to be able to access campus apis through an api gateway service, which allows the end user to authenticate using their familiar campus SSO (shibboleth).
7	We're a Google Apps school; as such, OIDC is important given interest around using social channels as alternative identity (i.e., Google+). In contrast, OAuth is necessary for delegated access via client software and APIs currently being deployed on campus.
8	Brown is in the process of developing and deploying a new "front door" for our alumnae. This site will be a portal (yea, that word; we're breathing new life into it). The site will obtain content from multiple backend services. Some content will come

	from a service providing information to the broader Brown community; other content will come from the Alumni system (essentially a CRM) and will be specific to each individual (eg giving history); I expect additional content to come from a wide variety of other systems. In the second case (CRM), it is critically important that the CRM provides the correct information -- ie the information associated with this alum, and not some other alum. Over time, the functionality in the portal will likely expand to provide Advancement staff with the ability to look at individual alumns, and groups of alumns. The security approaches in this multi-tier system should NOT offer simpler ways to impersonate an individual (especially Advancement staff) when accessing some of these backend systems. The current thinking is that the front door will be responsive, and will be programmed in javascript using node.js. This code will also run on a server (people who have disabled javascript in their browser will rely on the server based version). The code on the server will handle authentication (using passport, and strategies such as saml, LinkedIn, etc). The server will share with the javascript running in the browser a token containing information about the user; the javascript running in the browser could present this token when making requests to the various services. The front end will actually send its requests to an "API gateway" (which will be a product from MuleSoft). This approach provides a layer of abstraction, since the implementations of some of the backend systems may change over time (eg the CRM, the IDM system, etc). The API gateway will export a set of enterprise APIs to the alumnae portal, and to other systems and frontends (eg the usual apps that students want -- "today's menu"). The initial implementation will be what is described above. However, we all know that an app will have to be provided for mobile devices, and that app should support the authentication models in use on those devices.
9	The project started as a "data hub" - a central place to publish APIs that users in the community could consume to access public, and eventually, private data. To facilitate this, an API portal was built that provided OAuth support. Although we have yet to get permission from stakeholders to expose private APIs to the community, we are now using OAuth to secure access to private APIs for mobile apps that we develop in central IT. We hope to expand the use of OAuth out to the community once we get permission.
10	We are currently using OAuth for securing REST APIs. Currently, these apis are being accessed by integration ESB or by departmental IT. It doesn't use user consent at this point but it will be used in future
11	Current concern is as a consumer, rather than a provider. Have a requirement for a system that multiple parties can log into. Most of the higher education partners are likely to be eduGAIN / InCommon members, and will leverage that. However, plan to

	use OIDC/OAuth for authentication for institutions that aren't eduGAIN and for industry members as well.
12	Allow students' parents to "login with google" to our student information system.
13	We use cPanel for web hosting. In cPanel/WHM 11.56, functionality to support an external IdP (OIDC) was introduced, but we have no OIDC/OAuth IdP to leverage. Integrating cPanel with OIDC would eliminate redundant passwords and improve the security of the platform.
14	Some apps in the medical center already use this. Also seems to be increasingly common in securing APIs, which is our most likely use case.
15	Web apps that support it can easily integrate with our existing Google Apps domain to allow account provisioning and access through a familiar single sign-on interface.
16	We have Gluu depoyed and have bought and locally developed mobile apps using GLuu authN/authZ for login and API protection
17	Many of our members do not have SAML IDPs. For low security apps, OIDC/OAuth may be an easy way to authenticate them off of other accounts like Google Apps, Social Identities, etc
18	The main driver is Application-to-API authorization. The most common implementation of OAuth 2.0 is the so-called 3-legged flow, which expects the user to authorize access to his/her data, but student data does not belong to students, so students using a University application should not be asked whether they approve access (can you imagine asking a student if they delegate authorization in order to be invoiced for tuition?). As a result, it seems that the two-legged flow is more appropriate. But then why not use some kind of a shared-secret method? Such methods are not standard - I don't need every division creating their own implementations...and how do these get revoked when compromised? An industry-standard solution supported by central IT is a preferred option. An API, when accessed by a request with a valid token, may need the userid on whose behalf the request is being made so that authorization can be validated. For example, if a Commerce registrar is accessing the Service that provides a student's academic history, the student ID must be in the Commerce division (faculty). The API must be able to identify who the Registrar is and whether they are authorized to access the student's in the request. An OIDC JWT seems the best way to ensure that the Registrar's ID has not been modified. This means that both OAuth and OIDC need to be implemented, even for a simple two-legged flow. We are in the process of getting quotes to implement an OAuth Authorization Server, Token Service, and Policy Enforcement Point using IBM's DataPower Service

	Gateway (we own five of these in various environments). I still have a gap regarding OIDC. I'm investigating options.
19	We would be interested in using this to allow for our applicants, who have not yet been issued an institutional credential, to login with a credential they are more used to, as opposed to having to create another one.
20	We use the BOX API to provision box accounts for our site - this requires the use of Oath2 for the provisioning account, hence we use it for that.
21	Many vendor products purchased by campus sponsors support OAuth authentication but not SAML or OpenAM WPA authentication. It's unclear to us what additional authentication security measures are supported by these vendor products, (such as OpenID Connect), as we are still in the process of making sense of OAuth 2 vs OAuth 2 with OIDC ourselves. The more concrete use-case for us at this time is an ESB called MuleSoft, which has been purchased and is being integrated on campus. The team integrating MuleSoft on campus has requested that we enable their to authorize client API calls via OAuth token validation to our authentication service. We run OpenAM 11, which does not support OIDC natively, but does have OAuth 2 support. We are in the process of upgrading to OpenAM 13, which supports OpenID Connect out of the box.
22	We currently use OAuth/OIDC with Yahoo! And Google accounts as part of our account recovery (self-service password reset) service instead of secret questions and in combination with other information provided by account holders. Due primarily to our large scale 2-factor implementation, our account recovery service is now be analyzed for it to fit better with 2-factor. OAuth2 in combination with biometric capabilities on mobile devices is one of the potential methods that might improve account recovery functionality/usability. We are also looking at OAuth2 for web service authentication/authorization for both service accounts and end user accounts. As we continue to add vendor provided services, many private sector vendors are migrating their AD/LDAP-based offerings to either SAML2 or OAuth/OIDC (or both), but we see OAuth/OIDC providing easier to implement functionality that allows for securing service to service and service to user communication beyond just a browser-based path.
23	As faculty and staff departments look for best of class solutions for various business & teaching needs there's a growing need to integrate disparate systems in a secure way that also simplifies access to multiple resources using SSO.
24	Too early, use case development is underway and being defined

25	Mobile Forms portal in planning stages that would aggregate diverse / distributed data sources via web services which would benefit from OIDC/OAuth authentication.
26	One key use case is with the ESB that our campus is deploying. OAuth will provide improved authentication capabilities for applications making use of the ESB.
27	We need a way and instruction set to connect with our Netscaler for authentication and identity access to specific parts of our infrastructure we supply to campus users (all faculty, staff, and students). Right now I cannot find a simple way to do this, unless we use Google.edu
28	Primarily mobile usage. API use cases are interesting, but are currently system to system and are handled without end user oauth.
29	I particularly think OIDC/OAuth holds promise for providing services to alumni. It may also be promising to help current students integrate services into the single "single stream" of processing that they seem to like now.
30	InCommon is potentially interested in pursuing a pilot implementation of OpenID Connect Federation and seeing OIDC OP functionality built into popular federating software.
31	authenticating campus researchers to OIDC-enabled research services like https://docs.globus.org/api/auth/
32	External client application authentication/authorization (API consumption). Single Sign On coverage to decouple UI(s) from backend(s) via API(s) for reusability such as above.
33	Campus development efforts and third party cloud solutions supporting mobile applications, makes the need for an SSO solution critical. Consider these campus pain points that illustrate the challenges facing mobile users and organizations: SHADOW IT: The average enterprise has over 500 cloud applications in use, however fewer than 15% are enterprise ready MOBILE ACCESS: Nearly half of all cloud app activities occur on mobile devices. Yet, most mobile apps don’t support SAML for SSO. For those mobile apps that do support SAML, the authentication user experience is poor and security is weakened as user sessions are not frequently revalidated. The industry is moving to solve this problem with the introduction of NAPPS or Native Applications, a standard protocol to provide SSO for users on mobile devices through

	a “token agent,” which enables native mobile applications to authenticate users more easily. As is the case with SAML and SCIM for web applications, the promotion of NAPPS to mobile application developers is imperative to provide a more secure and integrated user experience. The NAPPS specification is part of the OpenID Foundation and is defined by the Native Applications Working Group. It is based on the OpenID Connect and OAuth 2.0 standards. It provides a seamless sign-on experience where an identity provider can federate access across numerous applications, and sessions can be validated repeatedly without degrading the user experience.
34	We needed to authenticate users through the campus Shibboleth and receive attributes via a mobile app. We used Keycloak as an identity broker to login in users through Shibboleth, and then have Keycloak manage user OAuth tokens for a mobile app. The OAuth token is then used by the mobile app to talk to an API.
35	We've had multiple people on campus requesting for delegated access to APIs by applications on behalf of, and authorized by, users. This would include things like Box API or home grown API's We've also had users wanting support for native mobile apps.
36	We need a solution for Epic and the future of API Connectivity using FHIR. If we don't have an OIDC/OAuth solution, we may be stuck with Epic being the main authorization service, which could lead to integration changes in the future.
37	We have had a few requests for OIDC/OAuth for mobile applications and for API work where a full SAML2 protocol is too much work.
38	We use OIDC in front of a number of internal applications (or student record system being the largest). We use it with a smaller number of cloud applications. We use it in front of our Moodle (LMS) instance. We use it in front of our Shibboleth server with a reverse proxy back to our IdP for all SAML applications. When we went Google 9 years ago, we made the decision to ship them our passwords - today, this makes it easier to do the above. (This has allowed faster/broader adoption of 2-step auth - since we didn't have to pay for and spend time integrating Duo.) This is our version of Identity as a Service, as was brought up in one of the I2TechEx sessions.
39	We run EPIC and various APIs and a Enterprise Service Bus. For Epic and the future of API connectivity using FHIR, if we do not have an OIDC/OAuth solution we’ll likely

	end up with Epic being the main authorization service which could lead to integration challenges in the future.
40	As part of our healthcare division, it is required for use with the HL7 standard known as FHIR. If we do not implement an enterprise solution, our EMR system will become the defacto enterprise system which will lead to future integration challenges. We currently have an API management solution which handles the application registration process, but it is lacking a true OAuth/OIDC implementation. They do support working with plenty of 3rd party vendors overall I feel like the market is lacking with true enterprise solutions.
41	I have both use cases for native applications as well as web client applications which are using social logins (Facebook/Google/etc). Both cases would be supported much better through OIDC. I've also developed an OIDC/OAuth proxy for Shibboleth requests which I use for this purpose. I'm currently working on widening the system to work with InCommon across a variety of IdPs.
42	Three main reasons. We have implemented it for the first: 1. API securing 2. Compatibility with OIDC/OAuth IdPs 3. Native mobile support
43	There is some demand for single-page apps, SOA-based architectures, etc. plus an API manager (with low adoption rate). No demand yet for cloud apps. As I see it the real need is for JWT and OIDC/OAuth is just one way to accomplish it.
44	We already use many OAuth-protected endpoints in our custom applications. However, we are not using OIDC at this point - we have users initially authenticate via SAML and then assign OAuth tokens.
45	We plan to officially take OIDC into production for our federation (next to SAML) in the beginning of 2017. SP's can choose how they would like to offer their service through SURFconext; using SAML or OIDC. A blogpost with some more info: https://blog.surf.nl/en/connection-to-surfconext-becomes-easier-for-service-providers-with-openid-connect/
46	We need delegated access where a portal acts on behalf of the user and OIDC is easier than SAML. For details on the use case, see: https://docs.google.com/presentation/d/1BLO1_5v7Zl2CxPezICF9a71lI-fMbwCiLUDpJT3yx88/edit N.B. There are also US Relying Parties in the use case.

47	We are envisioning the creation of a portal to let (SAML) IdP administrators the possibility of linking new applications to their already existing SAML Identity Providers. For that reason, our main interest is to determine SAML<->OIDC interoperability.
48	We user the social/saml gateway now for parent access. We have three use cases we want to serve, one is to use OAuth on our mobile app. Right now we have to kludge something to make SAML work. Two, we want use google login for our applicant and alumni login. We feel this will lessen support issues. Three, we want to establish some open api's for hackathons and feel OAuth would better support that.
49	1) As an easier option for SPs to implement 2) CAS = Simple & Local, Shib = Complicated & Federated, ... 2-4 years pass ... Maybe OIDC can replace both CAS & Shib 3) Some apps only offer user-generated API keys via OIDC (some have local token mechanisms) 4) Integrate with our BuzzAPI (soa-like) API infrastructure.
50	We mainly use it with APIs for cloud hosted services.

In addition, the following UCLA use cases were submitted to the working group outside of the survey:

U1	**Securing UCLA API** UCLA is in the process of building out a complete “UCLA API”. These API’s will encompass all major UCLA business functions, including student, financial, identity management, etc. We will be using OAuth as the API’s authorization protocol. In addition to authenticating the device and application consuming the API’s, the API owners wish to authenticate the human using the device/application to interact with the API. We would very much like to bridge SAML and OAuth in order to provide seamless SSO experience for the users. Further, we wish to do so in a standard, federation-friendly manner. In particular, UCLA provides business systems support for other UC campuses. Some of these APIs will immediately have cross campus (federation) access needs. Today, significant gaps exist in OIDC/OAuth’s support for federated access. We look to

	Internet2/InCommon to lead the development of federated OIDC/OAuth standards. We also wish to leverage, and contribute to the development of, I2 and/or federated support software to avoid maintaining local, one-off solutions.
U2	UCLA uses TicketMaster as its student events ticket sales agency. TicketMaster, after years of lobbying, has finally announced SSO support, except they are doing it via OAuth.
U3	UCLA Office of Information Technology (Academic/Research IT) is deploying a series of native mobile apps. SSO is very highly desirable there. SAML doesn't work well in native mobile app scenarios. The OIT developers have already deployed an SAML/OAuth gateway to work around the problem. They are not the first. We have others making similar request (or are just doing it without our knowledge). Effectively, teams are standing up shadow IDP proxies because native mobile apps don't play well with SAML. The users who are asking for native iOS/Android apps don't care about SAML or any of the technical complexity of integrating apps with campus SSO. They just want a working app.
U4	Box Sync, Box's desktop and native mobile apps, uses OAuth. It has its own SAML/OAuth bridge to enable SSO. It works. Several other notable SaaS providers do the same. I believe Sales Force is one of them. Similar to the OIT use case, we have the start of "shadow IDP proxy" problem. Once this pattern sets in, we, as the campus IDM office, potentially begin to lose the ability to enforce IDM security and privacy policies. We would like to OAuth support built into Shibboleth. In addition, we'd like InCommon to spearhead the development of OAuth federation interop policies and practices in the same way it did for SAML. Further, we'd like InCommon to lead the effort to work with SaaS providers to develop better integration patterns/solutions.

9. Appendix: Feedback Received from the Community Consultation for This Report

A community consultation for this report was conducted between March 6, 2017 through March 24, 2017. The following was the result of that consultation.

Current Text	Proposed Text / Query / Suggestion	Proposer	Action by the Working Group
Throughout section 6.1	Planning to add OIDC/OAuth protocol to existing products is warranted but more attention should probably be given at outset to importance of an Authorization Server capability. There seems to be a large gap between, on the one hand - Shib's current responsibility for authentication; Grouper's responsibility for group management - and, on the other, the need for an OIDC/OAuth-responsive infrastructure that has a fairly deep awareness of both the use case variants outlined and authorization claims at the level of business applications.	B. Savage	The OIDC *survey* group doesn't want to be overly prescriptive of what the TIER project should do. We have, however, added Authorization Server as an example of other services that should be considered for the TIER strategy.

10. Appendix: OIDC Survey Working Group Charter

10.1. Problem Statement

There is already a noticeable amount of campus-based activity exploring the use of the OIDC-related protocols, and particularly OAuth2. Today, very little of this activity is related to InCommon, and none of it is occurring within InCommon. Over the longer term, if a consensus emerges about the value of these protocols to campus communities, there may be actions that InCommon can take to help campuses use these protocols effectively. Today, it is more important to learn about what people are doing, and trying to do, without necessarily worrying about how InCommon might help them. Timing is import since this could be one of those situations where there is a risk in not taking action now.

10.2. Charter

1. Survey the campus community, and ask them to share information on the problems they are trying to solve that they think OIDC/OAuth2 can help to solve. Include their current and planned use of the OIDC/OAuth2 protocols, and encourage them to submit ideas that have not yet reached the planning stage. Encourage them to submit these descriptions as Use Cases.
2. Review the collected Use Cases, particularly the concrete ones that seem to generalize. Verify that these Use Cases seem like an appropriate use of OIDC, etc
3. Write a report summarizing the findings and conclusions. Include recommendations on next steps, if any, for
    a. TIER, and developing IDP/SP federating software
    b. InCommon, and Federation level support for these use cases
    c. and thoughts on the urgency, if any, of the recommended actions.
4. Include in the report the group's concerns, if any, about implementing scalable trust in conjunction with using the OIDC/OAuth2 protocols. Identify issues related to scaling trust, and issues InCommon would face in providing a trust fabric for these protocols.

10.3. Work Products

1. The group should submit its report to the TAC by Feb, 2017

11. Appendix: Working Group Membership

- Jim Basney, University of Illinois
- Russell Beall, University of Southern California
- Steve Carmody, Brown University
- Blair Christensen, University of Chicago
- Alan Crosswell, Columbia University
- Nathan Dors, University of Washington
- Eric Goodman, University of California, Office of the President
- Greg Haverkamp, Lawrence Berkeley National Lab
- Roland Hedberg, Catalogix
- Anil Kadiyala, University of Miami
- John Kazmerzak, University of Iowa
- Maarten Kremers, SURFnet
- Ethan Kromhout, University of North Carolina
- David Langenberg, University of Chicago
- Thomas Leggenhager, SWITCH
- Judah McAuley, Treetop Commons
- Chris Phillips, CANARIE
- Patrick Radtke, Cirrus Identity
- Nick Roy, InCommon / Internet2
- Tom Scavo, InCommon / Internet2
- Mark Scheible, MCNC
- Satwinder Singh, Columbia University
- Janusz Ulanowski, HEAnet
- David Walker <https://orcid.org/0000-0003-2540-0644>, InCommon / Internet2, Flywheel
- Albert Wu, University of California, Los Angeles, Chair
- Jule Ziegler, IRZ