Data Sharing and Traceability: Improving User Trust in Data Management within Open Banking and Beyond

by

Quinn Magendanz

S.B. Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2019

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by:	Quinn Magendanz Department of Electrical Engineering and Computer Science January 19, 2024
Certified by:	Daniel Weitzner Senior Research Scientist, MIT CSAIL Thesis Advisor
Accepted by:	Katrina LaCurts Chair, Master of Engineering Thesis Committee

Data Sharing and Traceability: Improving User Trust in Data Management within Open Banking and Beyond

by

Quinn Magendanz

Submitted to the Department of Electrical Engineering and Computer Science
on January 19, 2024, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This paper identifies the declining trust in proper data handling throughout the past decades, reviews studies into User Trust, and explores existing frameworks that have been developed to secure, streamline, and make accessible the processes of receiving authenticated User consent, sharing User data, and expressing data usage and collection preferences. Together, these realizations illustrate the customer need, market understanding, and optimum mode of integration which will demand and enable the development of the OTrace Traceability and Accountability Protocol. This protocol allows a User to track the sharing and usage of their personal data after it has been provided to, or collected by, an initial Data Provider that has explicitly received User consent. For the purpose of monitoring and auditing, the Data Provider and Data Recipient submit records to a Traceability Server to record initial User consent for data sharing as well as ensuing sharing and usage of the User's data. This specification introduces new standards for recording data sharing and usage as Traceability Records into a consent framework which builds off elements of the OAuth 2.0, PAR, PKCE, JWT, JWS, and TB protocols as well as the FAPI and FDX standards for financial data sharing.

Acknowledgments

First and foremost, I would like to thank my thesis advisor, Dr. Daniel Weitzner, and Ph.D. candidate Kevin Liao as well of the other members of the Future of Data Initiative, for their invaluable insights, advice, and support. Without Danny and Kevin's guidance, I would have undoubtedly been unable to finish this thesis remotely while simultaneously serving on active duty with the U.S. Navy.

A further thanks are extended to my roommates, Kaz and Daf, who distracted me all the time and made this take longer than it needed to.

Tough times build tough men. Challenges and puzzles are necessary to strengthen both mind and body. The best sailors and best software engineers are both **Forged By The C.**

Contents

List of Figures

Chapter 1

Introduction

"Trust but verify." -Russian Proverb[1]

1.1 Check Here to Acknowledge the Terms and Conditions

Apple's iOS 16 Software License Agreement is a 582 page document which begins with "BY USING YOUR iPHONE, iPAD OR iPOD TOUCH ("DEVICE"), YOU ARE AGREEING TO BE BOUND BY THE FOLLOWING TERMS" and goes on to state that "you agree and consent to Apple's and its subsidiaries' and agents' transmission, collection, maintenance, processing, and use of all of the foregoing information, to provide these Services" [4].

JPMorgan Chase Bank states within their U.S. Consumer Privacy Notice the following "reasons that we can share your personal information" [5]:

- "For our everyday business purposes – such as to process your transactions, maintain your account(s), respond to court orders and legal investigations, or report to credit bureaus. Does Chase share? Yes. Can you limit this sharing? No. "

[1]Famously used by President Ronald Reagan during nuclear disarmament discussions with the Soviet Union.

- "For our marketing purposes - to offer our products and services to you. Does Chase share? Yes. Can you limit this sharing? No."

- "For joint marketing with other financial companies. Does Chase share? Yes. Can you limit this sharing? No."

- "For our affiliates' everyday business purposes - information about your transactions and experiences. Does Chase share? Yes. Can you limit this sharing? No."

Apple and Chase are not alone in the cryptic nature of its privacy collections and sharing policies – they are merely a few examples. Users should not need to have the knowledge and time to parse the intertwined web of software license agreements, privacy policies, terms and conditions, legal documents, and government regulations in order to get an understanding of exactly who has their data and how it is being used. Even when a User does manage to read through all of these documents, many factors remain vague and generalized. What information is required to "provide these Services?" What individuals or collaborating corporations could fall under the label of Apple "agent?" How much data is shared because it is relevant to Chase "affiliates' everyday business purposes?"

Furthermore, parsing all of these documents to get answers is a barrier that many Users are either incapable of or do not have the time for. According to a set of 2019 Pew Research Center studies [6], 40% of Americans never read any of the privacy policies that they acknowledge when using new computer applications, systems, and accounts, while only 9% always do so. Of the 60% that may read them, only 22% read the policies all the way through.

This current system of User agreement and consent is flawed. Users need a new way to granularly classify their data and express explicit consent for how each category of data may be used and shared. A framework for tracking the movement of these data to specific organizations is then necessary for Users to verify these exchanges.

1.2 Policy Compliance

Modern privacy laws place appropriately high standards on companies handling personal data. Yet, consumers are still reporting declining trust in how companies handle their data, companies are finding it impossible to comply with the ever-shifting global patchwork of modern privacy legislation, and regulators around the world are struggling with the scale of the enforcement challenge [7].

1.2.1 Regional Regulatory Evolution

Ecosystem development has varied markedly by region, due in no small part to regulatory divergence as a result of differences in cultural philosophies and local economic priorities. A 2017 whitepaper by a leading US consulting firm describes the high-level evolution of data sharing regulation through various regions [8].

The most programmatic approach has been taken in the European Union, whose Payment Services Directive (PSD2) is administered by the European Commission in an effort to harmonize payments regulation and consumer protections across the European Union. Of particular interest, is an emphasis on online protections and attempts to foster marketplace innovation through open banking principles. A key principle of the PSD2 stipulates that upon the account holder's consent, a third-party provider (TPP) must be granted access to execute instructions on the account holder's behalf. TPPs can take several forms. Account Information Service Providers (AISPs), which include offerings such as Mint in the United States, are already empowered by the United Kingdom's open banking standard to deliver less sensitive information, such as branch and ATM locations. It is widely believed that enabling Payment Initiation Service Providers (PISPs) will drive significantly more innovation – and disruption – as it opens the field to actual money movement (Klarna and Alipay are examples of thriving PISPs).

India experienced remarkable fintech growth in late 2016 in the wake of the government's controversial decision to reissue fully 86 percent of its legal tender. The resulting cash shortage gave a jolt to an already growing mobile wallet segment, which

is now beginning to enter a consolidation phase.

Singapore has developed a large fintech market built largely around APIs, for instance, for risk-decisioning in the absence of formal credit-scoring agencies. The Monetary Authority of Singapore has now established a fintech division in order to provide structure and oversight to the process.

Open banking is also gaining traction in Iran (through the newly established Finnotech portal).

Australia is considering steps mirroring those being taken by the United Kingdom and European Union.

By contrast, the absence of a centralized US approach to data governance has given rise to a series of fintech innovators as well as a patchwork of one-off bank agreements (such as partnerships struck in the United States by Chase and Wells Fargo with Xero and Finicity) – a model that is not scalable in a market with roughly 12,000 financial institutions. Recently, the U.S. Office of the Comptroller of the Currency solicited public comments regarding potential issuance of a new special purpose charter enabling fintechs to engage in limited banking functions. While the charter's intent focuses more on lending and cost of capital, it also represents a step toward making it easier for nonbanks to compete in financial services and conceivably paves the road for data-sharing protocols similar to PSD2.

1.2.2 Common Requirements for Handling Personal Data

PSD2 [9] and General Data Protection Regulation (GDPR) [10] in the European Union, open banking standard in the United Kingdom, and Dodd-Frank [11] and the California Consumer Privacy Act (CCPA) [12] in the United States are some of the more significant data handling regulations (notably all within democratic regions of government) which are aimed at unifying personal data protections across member countries/states.

Despite the different priorities, regulations, and jurisdiction which contributed to the ratification of these regulations for data governance, some common elements are shared:

- *Privacy Notices.* Organizations are required to inform users annually of privacy policies and also upon changes to existing privacy policies.

- *Consent to Data Sharing.* Depending on the regulatory policy, users will either be required to provide explicit, granular consent to any data sharing or will reserve the right to opt-out of any data sharing (which they are notified of in privacy policies).

- *Third-Party Data Processing Agreements.* Third-parties that receive shared data must comply with the same set of regulations as the organization sending the data.

- *Data Portability.* Organizations must provide a human readable and machine readable mechanism for a user to extract their data for the purpose of viewing said data or migrating to competing services.

- *Right to Erasure.* Users reserve the right to request that their data be deleted from an organization's systems.

1.2.3 Discussed Requirements

These regulations are continuing to develop to meet the new challenges and pitfalls which are revealed as a result of advancing technology and novel edge/use cases of data sharing. Below are some of the examples of more recently justifiable and/or proactive regulations and the active discussions around the policies:

- *Right to Be Forgotten.* An organization's responsibility to take reasonable steps to inform all third-parties of a user's request for data deletion.

  The discussion around the right to be forgotten is how to define and enforce "reasonable steps." Furthermore, how a user can confirm the difference between if "reasonable steps" have been taken vs. if all third-parties have confirmed data deletion in addition to the original organization.?

- *Prohibition on "Dark Patterns".* Ensuring that there is no discrimination from organizations against customers based on their data sharing preferences.

  The discussion around the prohibition on dark patterns lies in how to prove that a organization's decisions were based solely on discrimination while proprietary algorithms (ex: algorithm for loan selection, source code for services) are not required to be disclosed.

- *Layered Notices.* Describes a mechanism for providing information on consent to a user in an injestible manner.

  The discussion around layered notices is on how to best provide granular and specific information to a user on consent without overwhelming the user. Should the mechanism used to achieve this be defined in policy?

- *Two Party Consent.* Getting consent for data sharing from all parties involved in a transaction (payer and payee).

  The discussion around two party consent lies in the difficulty of implementation. If a user wants to share their transaction history with a third-party, how does the bank receive consent from all of the other parties involved in all of the user's transactions?

- *Purpose Limitation.* Personal data must be erased if they are no longer necessary for the purpose(s) they were initially collected for.

  The discussion around purpose limitation is that determining when a purpose no longer exists is not a straightforward matter, as it varies from case to case. Fixed deadlines cannot be set to address this issue.

- *Freedom of Expression and Information.* This provides an exception to the right to erasure and to be forgotten if the processing is necessary for exercising the right to freedom of expression and information.

  The discussion around freedom of expression and information lies in defining freedom of expression. Some definitions state that any opinion is freedom of

expression, and thus, one could justify denying the right of erasure by recording an opinion which references the data as justification for the opinion.

- *Traceability.* See section 1.4.2 for definition. See section 1.2.4 for a new, proposed U.S. regulation to introduce standards for data sharing and a requirement that organizations make available to users all records of such data and financial transactions.

1.2.4 Current State of Affairs

This thesis will propose technical specifications aimed at simplifying regulation of data governance policies and improving transparency to the end user on how their data is being used. Such protocols will provide common implementation and interface solutions to many of the different policies discussed above. The wording of these policies is often intentionally left abstract in order to allow for implementations which both align with policies from jurisdictions and can grow with future needs and policy additions/modifications.

Towards the finalization of this paper, a specific measure was proposed within the U.S. which will provide a well timed opportunity for our technical analysis of the issues at hand to be received by policymakers and for public reception of our new protocols.

The Notice of Proposed Rulemaking - Required Rulemaking on Personal Financial Data Rights, deriving authority from Section 1033 of the Dodd-Frank Act, states that "the proposed rule would require depository and nondepository entities to make available to consumers and authorized third parties certain data relating to consumers' transactions and accounts; establish obligations for third parties accessing a consumer's data, including important privacy protections for that data; provide basic standards for data access; and promote fair, open, and inclusive industry standards" [13].

As the comment period for the proposed rule is currently open, and the Consumer Financial Protection Bureau (CFPB) is accepting comments until Dec. 29, 2023, it is

prime time to explicitly define data sharing protocols and standardized mechanisms to make data documenting consumers' transactions and accounts available to consumers (i.e. traceability protocols). As the U.S. is the largest national economy and the U.S. Dollar is currently the currency of global trade, building the United States a more unified data sharing and open banking interface set and regulatory mechanism will provide the best chance of proliferating such standards to the global data sharing and open banking space.

1.3 Existing Technologies

Existing cryptographic protocols systems cannot provide the guarantees required to control and enforce compliance of data usage. Protocols such as OAuth 2.0 exist to manage authorization and the user-initiated sharing of data. However, it provides no assurances of the sharing of data that is not user-initiated or of the proper use of data once it has been acquired. Cryptographic hashes, signatures, and encryption may serve to protect data in transit, but once an authorized agent holds the raw data, there is no way to enforce continued use of those cryptographic schemes. Once an agent holds the raw data, they can use it at will or share it to additional organizations without user knowledge. New protocols must be developed to manage these behaviors and account for correct data use and handling.

1.4 Future of Data

To address these global privacy challenges, the MIT Internet Policy Research Initiative (IPRI) and the Computer Science and Artificial Intelligence Laboratory (CSAIL) founded the MIT Future of Data [14]. This research group believes that any solution will need to focus on three properties in order to start building solutions within this space: informed consent, data traceability, and accountability [15].

1.4.1 Consent

At the time of this document's publishing, organizations provide inadequate modes for a User to view all the data sharing and data usage that they have consented to via both implicit and explicit User Agreements. Both users and governing bodies require a more detailed breakdown of consents granted. Any solution to informed consent must share the following characteristics to serve as a viable solution:

- *Consistency across service providers.* Issuing consent preferences must be standardized so that data sharing across organizations can carry with it the consent metadata.
- *Granularity in consent choices.* A user should be able to express specific ways data can be used while still disallowing other types of use. This must be granular enough to capture the different types of use while not so fragmented as to confuse users.
- *Equal access to services regardless of consent choices.* Mechanisms should be in place to monitor user consents and services rendered in order to provide transparency and assure consistency which will prevent the use of "dark patterns" to coerce, wheedle, and manipulate users who do not wish to agree to some/all consents.
- *Flexible update plan.* As new laws are ratified, the protocol may need to incorporate additional consent metadata.
- *Secure, scalable communication protocols.*

1.4.2 Traceability

Organizations need a medium to prove their compliance to both Users and governing bodies, especially as data is shared to third parties. Any solution to traceability must provide a method of both detecting misuse of data and of handling changes in consent preferences. Traceability monitoring must also be scalable for large amounts

of both automated and manual data processing and for sharing across many different, untrusted organizations. As discussed in section 1.3, existing cryptographic protocols cannot make the required guarantees, so the solution will likely depend on accumulated attestations of data use and sharing from multiple different organizations to verify compliance. To monitor these traceability records, a simple, unified platform should be available which is capable of processing incoming information from many organizations and displaying summaries and reports to users.

1.4.3 Accountability

"Uses of personal data are controlled in a manner to enable monitoring appropriate use as well as detection and consequences for misuse of data (Lampson 2009). Accountable systems will indicate when data uses are tied to a necessary, clear, and legitimate interest such as fraud prevention, legal compliance, and other consumer consented uses, and when, on the other hand, there is misuse. As the complexity of personal data services grows and new privacy laws are enacted, enterprises and regulators face corresponding complexity in navigating their legal obligations regarding processing and transferring personal data. Building on traceability solutions, we can make it possible for organizations to analyze and maintain indicators of data use to monitor compliance with internal policies, consumer consent, contractual obligations and legal requirements." [15]

1.5 Open Banking

Open banking is designed to allow consumers and businesses to enable third-party apps to access financial data instantly and securely for the purpose of providing additional insights or services [16]. Some examples include using an app to pay back friends after a night out, transferring money between accounts at different institutions to buy stocks, using a budgeting app to track spending, or receiving a breakdown of how elements of your financial state and history will effect your loan application.

Research shows that 87 percent of U.S. consumers are using open banking to link

their financial accounts to third parties, however only 43 percent of U.S. consumers are aware that they are using open banking [17].

Open banking ecosystems being deployed around the world will facilitate innovation in consumer banking services, but also raise novel questions regarding user trust and the need for personal data governance across organizational boundaries. The growing open banking environment will depend on systems of accountability and traceability for enforcing adherence to user consent of personal data use while enabling more open flow and analysis of personal financial information. Both users and regulators are demanding that personal data governance capabilities be deployed alongside open banking APIs, but scalable and secure systems have yet to be designed and deployed.

1.6 Traceability Protocol Proposal

This paper will ultimately lay out a new protocol to try and address these issues arising with the proliferation of user data sharing. A traceability service will provide a mechanism within the open banking environment for users to view a list of all the policies they have consented to in a granular, ingestible format as well as a log of sharing a usage of their data in order to verify the claims made by the businesses that handle their data. The goal of making such information available to the user is to improve user trust in the open banking environment and to ensure continued accountability of all parties involved.

This protocol defines a new entity known as a Traceability Server and details its API. The Traceability Server will allow users to track the sharing of their data to third parties as well as the ensuing use and sharing of that data. This protocol focuses on providing a consistent, secure, scalable solution for specifying granular consent in a unified standard.

1.7 Roles

This paper will refer to the following roles when describing frameworks, protocols, and APIs:

- User

  The end user whose personal information is held within the data. An example of the definition of personal information can be found in California Cival Code §1798.140(o)(1) - CCPA [12]. This can include, but it not limited to Personally Identifiable Information, bank account numbers, location, activity history, etc.

- Authentication Server (AS)

  The service that performs User authentication and receives initial, explicit consent to collect and store User data.

- Resource Server (RS)

  The server which contains the data/resources requested by the Data Recipient on behalf of the user. The Resource Server must confirm that the Data Recipient has sufficient consent to retrieve the requested data.

- Data Provider (DP)

  The entity that owns both the Authentication Server and Resource Server. Where differentiating between the AS and RS roles is not necessary, this role may be used to refer generally to both the AS and RS servers or a server that implements both roles.

  Users usually have an existing relationship, account, and direct interactions with Data Providers. However, if a Data Recipient is given consent to further share User data, it assumes the role of Data Provider with the secondary sharing, potentially without direct User interaction.

- Data Recipient (DR)

The third party which seeks to act on behalf of the User and/or access their data.

Chapter 2

User Trust

2.1 Studies on User Trust

Fellow MIT Future of Data Initiative graduate researcher, Nicola Lawford, reviewed a series of studies seeking to understand what role User trust plays within the adoption of new online banking tools and towards providing consent to data sharing [18]. The following are some of the common conclusions between the collection of studies:

- Regarding the building of trust between Users and financial/financial technology (fintech) organizations:
  - Trust builds through consistency, integrity, shared values, and perceived security and service quality [19] [20] [21] [22] [23].
  - Trust is influenced by pro-cyclical, economy-linked factors, including inflation, recession, and employment [24].
  - Perceived usefulness and personalization increases privacy concern as users witness the technology in action [25].
  - Financial literacy reduces initial trust [26] [27].
  - U.S. households trust traditional financial institutions with their personal data more than government agencies or fintech companies. Racial minori-

ties trust financial institutions less. Younger generations trust financial technology companies more [28].

  - The primary concerns of a data leak include, identity theft, personal safety, and abuse of personal data for secondary purposes (such as news media, political agendas, or targeted advertising) [28] [28].
  - There is insufficient trust in banks regarding the safeguard of personal data, and conflating trust in banks to safeguard money with trust in banks to safeguard personal data will not be the remedy [29].

- Regarding the willingness to use new open banking products,

  - Service quality, information quality, system quality, frequent communications, trust in the app, and social concerns (such as COVID-19 anxiety) increases rates of adoption. Specifically, it is a contest between perceived usefulness and trust [30] [31] [32] [33] [34] [28].
  - Brand trust mediates the influences of perceived risk, and initial trust reduces the effects of perceived risk [35] [27].
  - Banks are more trusted to utilize personal data than any other payment service provider; nonetheless, most respondents would not give their consent for payments data usage to facilitate a financial overview and personal offers from those banks [36].
  - Those with more to gain from sharing data are more likely to consent to more data sharing [37] [37].
  - It is commonly believed that there is a need for greater independent oversight over privacy, data protection, and customized privacy sensitivity [29].

2.2 Analysis of User Trust Studies

Analysis upon the above collection of studies yielded the following observations and lingering questions concerning how effective a traceability protocol would be in

influencing User trust and how to get Users to adopt said traceability protocol.

- There appears to be a need for both the User and regulatory agents to audit data sharing and usage [29]. A traceability protocol can fill that need.
- Privacy concerns which result from extensive personalization can be addressed through a traceability service providing visualization of proper personalization data use [25].
- Though this paper is limited to laying out the technical specifications of a traceability protocol, some of the most significant factors for adoption by Users are usability and performance. This will depend on the implementation and integration of the Traceability Server into the User's current banking experience.
- Will a traceability protocol influence "fair treatment" within the open banking space? Is this something that must be taken into account?
- Users are influenced to share more data based on how much they have to gain (i.e. fiscal stability). Though some low-income users who opt out of data sharing may face higher prices due to adverse selection inferece effects [38]. A traceability protocol can provide a mode of accountability to prevent "dark patterns", thereby protecting Users.
- The traceability protocol must strike the correct balance between privacy, fairness, and utility.
- Some qualifying observations on the studies themselves:
  - These studies were mostly survey-style, and thus, there may be a discrepancy between what Users report and their actual behaviors.
  - The studies may have identified predictive factors, but not necessarily causational.
  - Many studies were conducted in different parts of the world. What factors are universal and what is dependent on local culture and politics?

Chapter 3

Related Technical Specifications

3.1 Financial Data Exchange (FDX) API

In the U.S. and Canada, Financial Data Exchange (FDX), a non-profit technical standards body, provides standard tools for secure and reliable consumer data access, which could increase the adoption of open-banking API frameworks worldwide. FDX is working to align the industry around one common, interoperable open banking API called FDX. Regulatory bodies in the U.S. and other parts of the world are also considering how best to ensure that financial service providers support competition and innovation while protecting consumer data [16].

The Financial Data Exchange (FDX) API [1] lays out a standardized method for a user to provide consent for sharing financial data from an original Data Provider to a third party Data Receiver. v.5.1 specifies the creation of Consent Grant objects which manage the permissioning of Data Recipient access to user data. This protocol tightly couples consent creation with an OAuth 2.0 Authorization Code [39] grant flow, Rich Authorization Requests (RAR) [40] and Pushed Authorization Requests (PAR) [41] specifications [1]. In addition to the consent grant control flow shown in figure 3-1, FDX defines a few specific structures which will be utilized either directly or indirectly in the Traceability Protocol.

First, the FDX defines a specific type of authorization_details field from the Rich Authorization Requests protocol to carry fine-grained authorization data. FDX

defines this field to be a JSON-formatted object with two members: "type" with the value "fdx_v1.0", and "consentRequest" containing a valid JSON ConsentRequest entity as defined in [1] which describes the scope of the data sharing requested by the Data Recipient of the Data Provider. Below is an example of authorization_details:

```
"authorization_details": {
    "type": "fdx_v1.0",
    "consentRequest": {
        "durationType": "ONE_TIME",
        "lookbackPeriod": 60,
        "resources": [
            {
            "resourceType": "ACCOUNT",
            "dataClusters": [ ###TODO: Change to match Fides
                "ACCOUNT_DETAILED",
                "TRANSACTIONS",
                "STATEMENTS"
            ]
            }
        ]
    }
}
```

Secondly, FDX defines the ConsentGrant structure for internal Data Provider tracking of the User's consent. Though the ConsentGrant provides a sufficient example of this internal tracking, this internal behavior is out of scope of the Traceability Protocol and will not be required by Data Providers.

Thirdly, FDX defines its own system of categorizing consent that they call, Data Clusters. However, the Traceability Protocol opts instead to leave the format of consents open to allow for use of consent frameworks to evolve with the leading-edge

research. At the time of the writing of this paper, Fides Language (see 3.7) is one of the more granular, specific, and widely-accepted framework, and it can modularly replace the DataClusters structure within the FDX specification.

Finally, FDX states a few conditions on the relationships between the User, Data Provider, and Data Recipient which are held within the Traceability Protocol:

- For the Data Recipient to initiate data sharing with a Data Provider, the User MUST have independent relationships with both the Data Recipient and the Data Provider.
- The User MUST initiate the consent request from within the Data Recipient's experience/application.
- Before beginning the consent request, the Data Recipient determines the types of data access it intends to access from Data Provider and MUST disclose its intent to the User.
- During User Authentication to the Data Provider, the User MUST actively authorize the Data Provider to enable the Data Recipient's access to End User's data.

3.2 Financial-grade API (FAPI) 1.0

OpenID's Financial-grade API (FAPI) is a REST/JSON model based in OAuth 2.0 which has been designed for the high-risk scenarios which occur when third parties are requesting access to a user's financial data and making transactions on behalf of a user [42]. FAPI aims to be secure against very strong attackers by employing a range of mechanisms that have been developed to harden OAuth 2.0.

Under the Traceability Protocol, these mechanisms are integrated into the FDX control flow in order to harden FDX as FAPI hardens OAuth 2.0. Below are summaries of each of the introduced mechanisms along with their documentation. Figure 3-2 illustrates the combination of these mechanisms into different FAPI control flows:

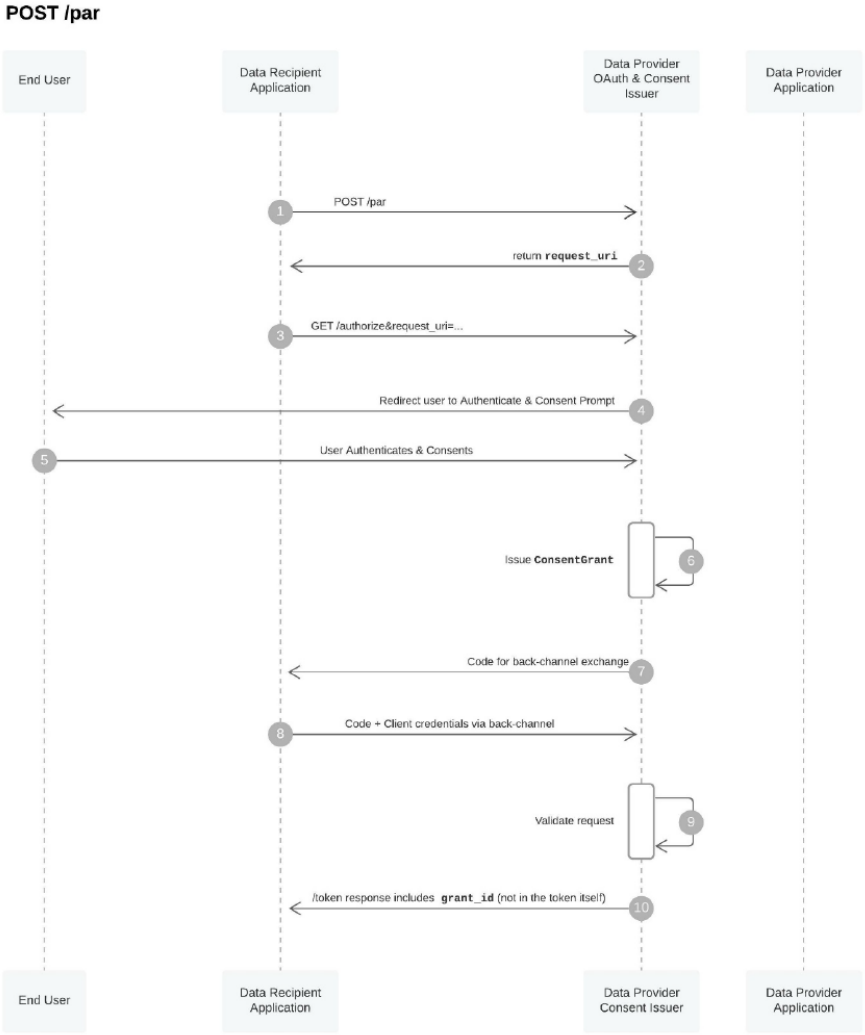

Figure 3-1: FDX API Consent Granting [1]

- OpenID Connect

  OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It enables Data Recipients to verify the identity of the User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the User in an interoperable and REST-like manner. The OpenID Connect specification [43] defines the core functionality to be authentication built on top of OAuth 2.0 and the use of Claims to communicate information about the User.

- Proof Key for Code Exchange (PKCE)

  OAuth 2.0 public Data Recipients utilizing the Authorization Code Grant are susceptible to the authorization code interception attack. RFC 7636 [44] defined that to mitigate this attack, this extension utilizes a dynamically created cryptographically random key called "code verifier". A unique code verifier is created for every authorization request, and its transformed value, called "code challenge", is sent to the Authorization Server to obtain the authorization code. The authorization code obtained is then sent to the token endpoint with the "code verifier", and the Authorization Server compares it with the previously received request code so that it can perform the proof of possession of the "code verifier" by the Data Recipient. This works as the mitigation since the attacker would not know this one-time key, since it is sent over TLS and cannot be intercepted.

- JSON Web Signatures

  A JSON Web Token (JWT), as defined in RFC 7519 [45], is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object that is used as the payload of a JSON Web Signature (JWS) structure (as defined in RFC 7515 [46]) or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed or integrity protected with a Message Authentication Code (MAC) and/or encrypted.

- Token Binding

  The Token Binding protocol, as defined in RFC 8471 [47], allows client/server applications to create long-lived, uniquely identifiable TLS bindings spanning multiple TLS sessions and connections. Applications are then enabled to cryptographically bind security tokens to the TLS layer, preventing token export and replay attacks. To protect privacy, the Token Binding identifiers are only conveyed over TLS and can be reset by the user at any time.

  The draft-ietf-oauth-token-binding-08 RFC [48] further defines an implementation of Token Binding for OAuth 2.0 to apply Token Binding to Access Tokens, Authorization Codes, Refresh Tokens, JWT Authorization Grants, and JWT Client Authentication. This cryptographically binds these tokens to a Data Recipient's Token Binding key pair, possession of which is proven on the TLS connections over which the tokens are intended to be used. This use of Token Binding protects these tokens from man-in-the-middle and token export and replay attacks.

As of the writing of this paper, FAPI 2.0 is currently under development and in draft form. As this draft is currently undergoing development and is not stable, this paper focused attention on FAPI 1.0. However, it is believed that FAPI 2.0 is also working towards the integration of the security remediations discussed in section 3.3 as well as some of the PAR features similar to the FDX Consent Grant workflow. When FAPI 2.0 is finalized, the consent grant workflow of the traceability protocol should be made compatible with that framework.

3.3 FAPI Security Analysis

In 2019, Daniel Fett, Pedram Hosseyni, and Ralf Küsters performed a formal security analysis of FAPI under a very strong attacker model. Using the Web Infrastructure Model to simulate the different network components involved in FAPI exchanges, the researchers were able to identify vulnerabilities in the protocol, intro-

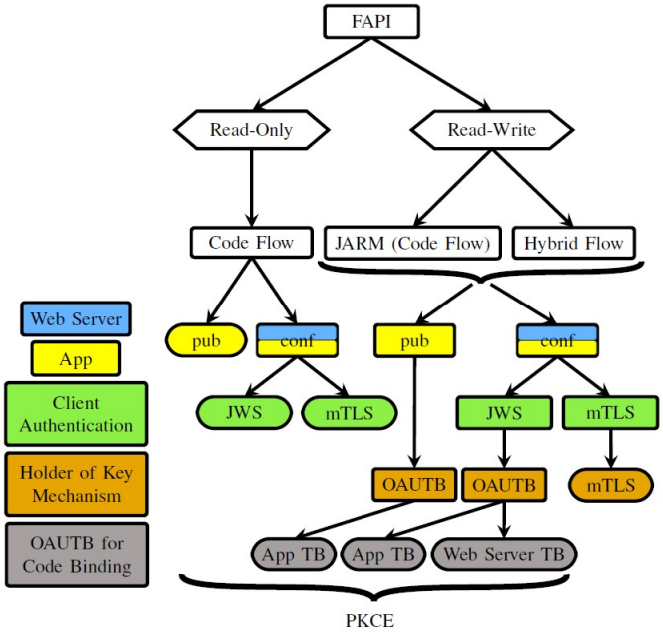

Figure 3-2: Overview of the FAPI. One path (terminated by a box with rounded corners) describes one possible configuration of the FAPI. The paths marked with PKCE use PKCE. JARM and Hybrid flows both allow for the configurations shown. [2]

duce remediations, and formally verify the central security properties of FAPI.

The proven theorem states, "Let *FAPI* be a FAPI web system with a network attacker. Then, *FAPI* is secure w.r.t. authorization and authentication. Furthermore, *FAPI* is secure w.r.t. session integrity for web server clients with OAUTB." [2]

The following are the vulnerabilities identified during formal verification along with the applied remediations:

- Cuckoo's Token Attack

  A malicious Authentication Server passes a stolen access token back to the Data Recipient which will then retrieve data from the Resource Server on behalf of a malicious User. This can be prevented if the Data Recipient includes the identity of the Authentication Server that provided the access token while requesting data from the Resource Server.

- Access Token Injection with ID Token Replay

A malicious user redirects the Data Recipient to a malicious Authentication Server after providing credentials to a valid Authentication Server. This malicious Authentication Server then provides the stolen access token to the Data Recipient as in a Cuckoo's Token Attack. This can be prevented if the signed ID token contains the access token proving that the access token came from the relevant Authentication Server.

- PKCE Chosen Challenge Attack

  A malicious Data Recipient may break session integrity, forcing a User to use the Attacker's resources if they can leak the Proof Key for Code Exchange (PKCE) challenge from the memory of the User. This can be prevented by including a JWS signed JWT with the authorization request to ensure that the stated Data Recipient is the one that actually initiated the request.

- Authorization Request Leak Attacks

  A malicious Data Recipient may defeat Cross Site Request Forgery (CSRF) protection if the authorization request state value is leaked from the memory of the User. This can be prevented in FAPI web servers by implementing OAuth Token Binding.

These remediations must be applied to any implementation of FAPI within a traceability service in order to maintain secure data exchange.

3.4 Consumer Reports Data Rights Protocol (DRP)

The Data Rights Protocol (DRP) is a set of request/response data flows which aim to standardize data exchanges across organizations and regulatory regions. By providing a shared protocol and vocabulary for expressing data rights and sharing preferences, DRP reduces the administrative burdens on consumers and businesses while providing a basis of trust for verifiable identity attestation which can be used by (individual) consumers (or by an agent intermediating the relationship on behalf

of consumers) and businesses [49]. As of the writing of this paper, Consumer Reports is currently going through integration testing of DRP v0.8 on the live data of multiple real-world companies.

As DRP is currently undergoing development and is not stable, this paper has not tried to align vocabulary of the traceability protocol with DRP. However, the long-term goal is for both protocols to leverage the same vocabulary and protocol/endpoint definitions so that there is a universal standard and services can be easily integrated. Specifically, early DRP models indicate it could be paired with a traceability protocol by defining the exchanges and requests that take place directly between the Data Providers and Data Recipient while the traceability protocol defines the format and frequency of reporting these interactions to a traceability service.

3.5 Global Privacy Control (GPC)

In response to the requirement imposed by the CCPA (along with other legal frameworks) that gives users the right to request that their data not be sold or shared beyond the business with which they intend to interact, Global Privacy Control [50] was pioneered to standardize and simplify such requests. GPC introduces a new field within HTTP headers which specifies a user's data sharing preferences. In addition, browser plugins were developed which add simple UI options to each website where a user can select their preferences and automate the insertion of preferences into outgoing HTTP request headers.

With regards to traceability, GPC provides a precedent on how a User may communicate initial consent and data sharing preferences via HTTP headers, and how to streamline the User experience involved in expressing those preferences.

3.6 Solid

Solid is a specification focused on giving users increased control over their data while remaining decentralized [51]. Users can stand up Pods to store any type of

data, and both broad and specific access can be given (and revoked) to external organizations via a REST API on the Pod. Users can pay to have Pods hosted within existing cloud environments or can clone and run the open-source Pod web server templates on their own infrastructure.

Solid varies from a traceability service in that Solid does not seek to monitor data exchange beyond managing direct accesses to the data hosted within the Pod. A traceability service would not store data directly or manage access to data, but will receive reports of data being shared from all organizations involved in the sharing. Instead, Solid may be a framework that Resource Servers build their storage off of. Thus, we can pre-build traceability service reporting into the Pod APIs.

3.7 Fides Language

Fideslang (fee-dez-læŋg, from the Latin term "Fidēs" + "language") is a proposed model for a human-readable "taxonomy" of privacy-related data types, behaviors, and usages. Fideslang hopes to develop an interoperable community standard for building privacy regulation compliance into the typical software development process [3].

Specifically, a traceability protocol should leverage Fideslang's Taxonomy Classification Groups. The classifications provide a structured community standard to granularly and specifically describe types of User data, known by the Data Categories data type (see figure 3-3), and the permitted actions over such data, known by the Data Uses data type (see figure 3-4).

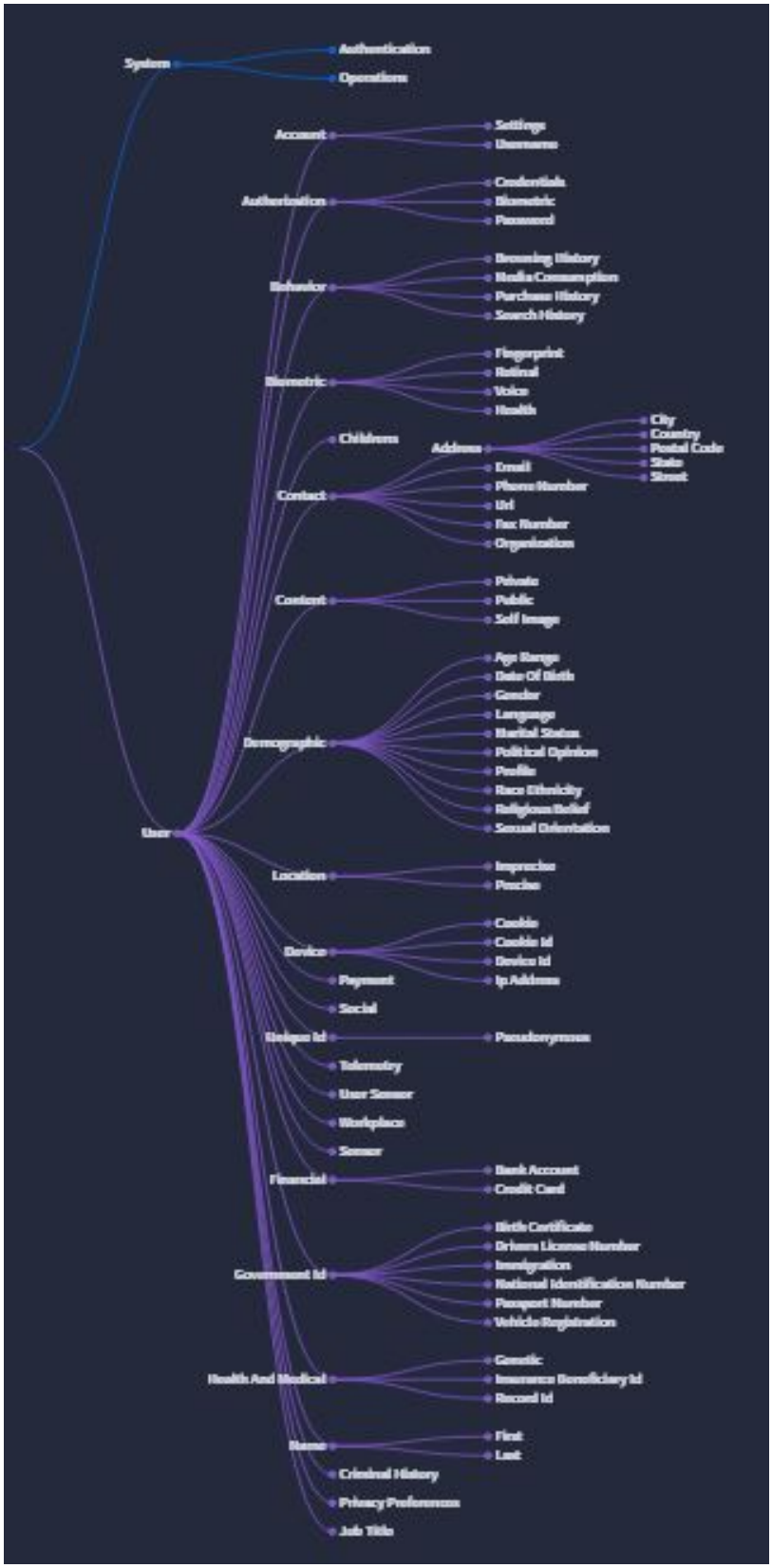

Figure 3-3: Fideslang's Taxonomy Classification Groups Data Categories [3]

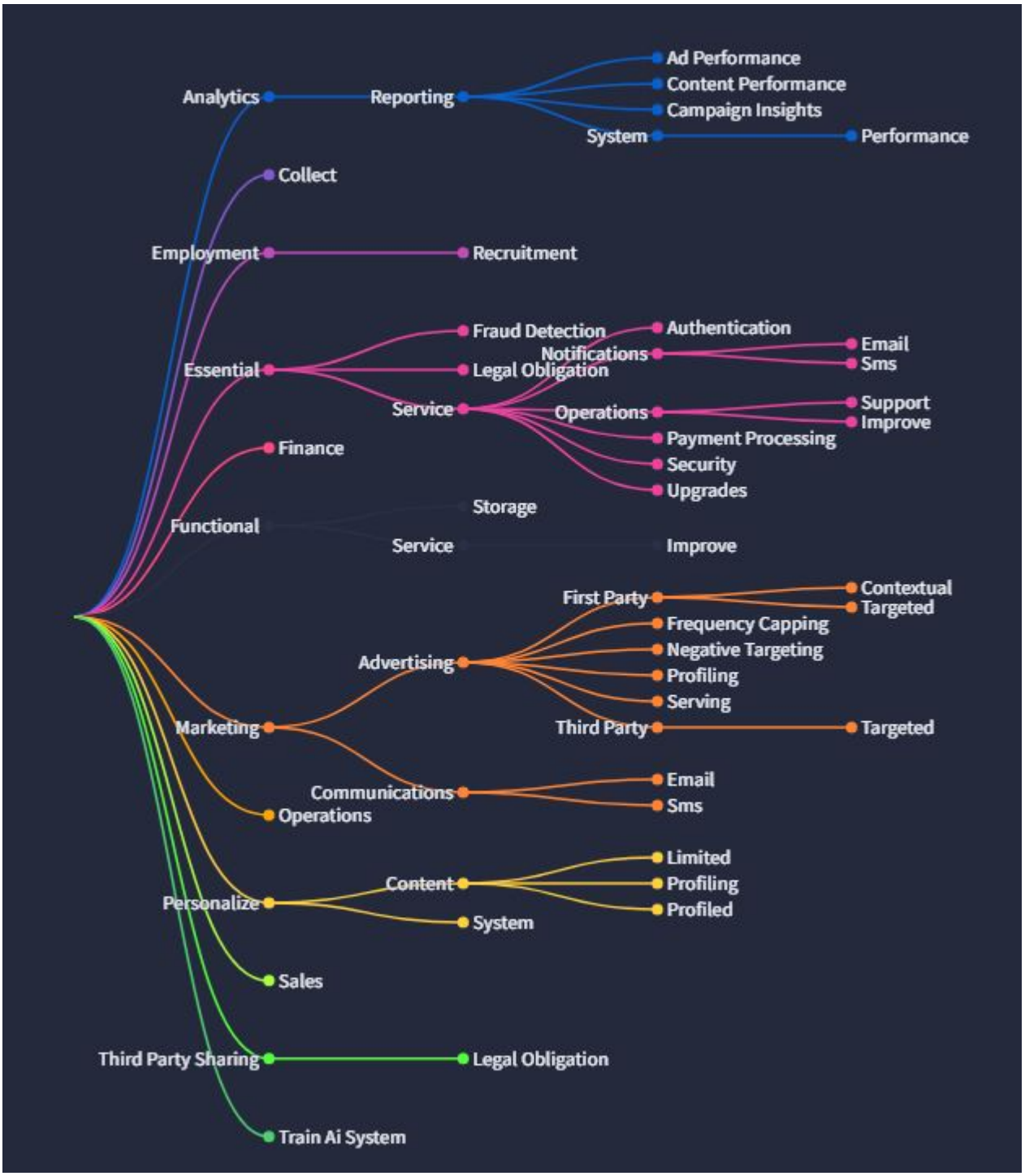

Figure 3-4: Fideslang's Taxonomy Classification Groups Data Uses [3]

Chapter 4

Traceability Protocol

The initial research conducted by MIT's Future of Data Initiative has identified the declining trust in proper data handling throughout the past decades. Nations and states craft new legislation mandating semi-specific standards for security and data handling as well as the rights of a customer over their data. Existing organizations invested in open banking have reached out directly to the Future of Data (as well as to other research organizations) to seek a solution.

The studies into User trust start to paint us a picture of the landscape of User-financial organization relationships. What are ways to improve or hurt a User's trust in an organization. What factors positively impact the adoption of new services that hinge on the sharing of data within an open banking market.

Existing protocols and frameworks have been developed to secure, streamline, and make accessible the processes of receiving authenticated User consent, sharing User data, and expressing data usage and collection preferences. These protocols are continuously evolving to provide the more efficient, more secure, globally accepted technical solutions.

Together, these realizations illustrate the customer need, our understanding of the market, and the optimum point of integration within existing frameworks. The OTrace Traceability Protocol will fill the need for monitoring and ensuring accountability; it will focus on factors shown to increase user trust to improve adoption as a new service ; it will leverage and compatibly integrate within the current open

banking infrastructure.

4.1 The OTrace Traceability Framework

Appendix A defines the OTrace Traceability Server and the accepted Traceability Records that can be exchanged within an authorization framework that extends and combines OAuth 2.0 along with other existing frameworks. OTrace enables a User to monitor records of the initial consent for data sharing alongside instances of subsequent sharing and usage in a manner which emphasizes the properties of informed consent, data traceability, and accountable use [15].

The OTrace Traceability Framework enables a User to track the sharing and usage of their personal data after it has been provided to, or collected by, an initial Data Provider that has explicitly received User consent. For the purpose of monitoring and auditing, the Data Provider and Data Recipient submit records to a Traceability Server to record initial User consent for data sharing as well as ensuing sharing and usage of the User's data. This specification introduces new standards for recording data sharing and usage as Traceability Records into a advanced consent grant framework.

4.1.1 Consent Grant Framework

OTrace consent framework builds off of elements of the OAuth 2.0, PAR, PKCE, JWT, JWS, and TB protocols from RFCs 6749 [39], 9126 [41], 7636 [44], 7519 [45], 7515 [46], and 8471 [47] respectively, as well as the FAPI [42] and FDX [1] standards for financial data sharing. See Chapter 3 for a breakdown on what each component protocol brings to the overall OTrace consent grant framework. See Appendix A for a breakdown of the individual steps and parameters of the process.

4.1.2 Roles

OTrace defines the same roles used throughout this paper – User, Authentication Server, Resource Server, Data Provider, and Data Recipient – but also defines the new role of Traceability Server:

- Traceability Server (TS)

  The server which receives and stores records of data sharing and usage for the purpose of User monitoring and legal auditing.

4.1.3 Record Definitions

OTrace defines the following terms to describe the types and categories of records within the traceability framework:

- Traceability Key-pair

  A Traceability Key-pair is a key-pair used to sign all JWS's sent to the Traceability Server. Each of the Data Provider and Data Recipient MUST have a Traceability Key-pair that is unique to the actor. The public key of the key-pair will be used to identify each actor as they interact with the Traceability Server.

- Traceability Record

  Traceability Records are received by the Traceability Server from both the Data Provider and Data Recipient and logged for verification by the User.

  Each Traceability Record MUST be represented as a JWT which is then signed within a JWS by the Data Provider or Data Recipient Traceability Key-pair.

  The Traceability Server MUST verify each JWS and check that the public key used for signing matches the Data Recipient or Data Provider.

  The following are the types of Traceability Records:

  - **Traceability Policy Record**

Represents an explicit consent provided by a User. Includes identifiers for the Data Subject (the User), Data Provider, and Data Recipient.

Specifies the types of User data pertaining to the consent as well as the explicitly permitted types of usage for the Data Recipient. OTrace leaves the definition of the consents field intentionally vague (specifying that it must only be expressed as a JSON array of structures) as there is not currently a universally accepted format or language to granularly and concisely express the data type and actions consented to.

As of the writing of this paper, the Fides Language Taxonomy Classification Groups [3] appears to be the most effective format of conveying consent as it provides a more structured and standardized language for specifying multiple levels of consent. Fides is also abstract enough to allow organizations to add in custom consent parameters without overlapping with less vague descriptors. However, other frameworks, such as FDX Data Clusters and Visa Consent Items, exist. Below are a few examples of how these frameworks would be expressed:

```
Fides Language Taxonomy Classification Groups {
    "consents": [
        {
            "category": "user.contact",
            "uses": "marketing.communications",
            "subject": "anonymous_user"
        },
        {
            "category": "user.demographic",
            "uses": "marketing.advertising",
            "subject": "anonymous_user"
        },
        {
```

```
            "category": "user.location.impercise",
            "uses": "personalize.content",
            "subject": "anonymous_user"
        }
    ]
}

FDX Data Clusters {
    "consents": [
        "ACCOUNT_DETAILED",
        "TRANSACTIONS",
        "STATEMENTS"
    ]
}

Visa Consent Item {
    "consents": [
        {
            "Consent Item Type": {
                "Description": "Bank Transactions"
            },
            "Duration": 31536000,
            "Valid From-To": {
                "Start": "2023-12-28T19:41:40Z",
                "End": "2024-12-28T19:41:40Z"
            },
            "Frequency": "STREAMING",
            "Time Span": "PERIOD",
            "Consent State": "GRANTED",
            "Consent Status": "ACTIVE",
```

```
            "Primary Purpose(s)": "connecting a bank account
                to a Personal Financial Management tool to
                manage finances",
            "Secondary Purpose(s)": "sharing bank account
                information with lenders, aggregators or
                intermediaries who want to use data for
                something in addition to fulfilling the
                Primary Purpose(s)"
        }
    ]
}
```

When the parameters of the consent are changed (usually by the User), this must be communicated to all parties, and a new Traceability Policy Record will be issued by both the Data Provider and Data Recipient.

A Traceability Policy Record with an empty or absent consent parameter is equivalent to deletion of all shared data.

– **Traceability Share Record**

Represents the movement of User data from a Data Provider to Data Recipient. This record will specify the type of data being moved. The type of data must be within the consent laid out by the Traceability Policy Record.

– **Traceability Usage Record**

Represents the use of User data for any purpose beyond possession. When User data is used, this record specifies exactly what type of data was used and in what manner.

– **Traceability Migration Record**

Represents the request to migrate all previous Traceability Records in the current Traceability Record Set to a different Traceability Server. This

type of record keeps the Traceability Server within compliance of laws such as GDPR [10] and CCPA [12] which require the ability for a User to export their data at any point to a different service (in this case, a new Traceability Server).

- Traceability Record Set

  A Traceability Record Set is a collection of Traceability Records which all derive their consent from the same User consent for the sharing and/or use of data. They are initiated by a Traceability Policy Record from the Data Provider which documents the details of the User consent and is attested by a matching Traceability Policy Record from the Data Recipient. Further records will be considered to be part of this Traceability Record Set if they contain the same trace_id (which is returned by the Traceability Server) and are signed by either the Data Provider or Data Recipient.

  If a Data Recipient receives consent to further share User data, they MUST create and record records to a new Traceability Record Set. This new Traceability Record Set MUST use the same Traceability Server as the original and MUST use the parent_ids parameter of the new Traceability Policy Record to refer to the original Traceability Record Set.

It is important to note that OTrace is not based in attestable cryptography to guarantee complete and accurate reporting of data sharing and data use.

Where possible, OTrace leverages verification between the Data Recipient and Data Provider. This occurs when the Data Provider, a party already trusted by the User with their data, creates a Traceability Record when actions such as consent and data sharing occur, and then the Data Recipient submits a Traceability Record with matching information.

However, when data usage occurs, it is often impossible for the Data Provider to independently provide a verifying Traceability Record. Instead, it is believed that the market will pressure Data Recipients to honestly report data usage, as those organizations who honestly report will improve User trust, increasing their likelihood

of successfully introducing new products and maintaining User business per Chapter 2.

Chapter 5

Conclusion

5.1 Open Banking Research Sandbox

The MIT Future of Data Initiative's Research Sandbox offers resources, such as data sets, technical specifications, code implementations, and relevant policies to aid in the exploration of accountability and traceability in the open banking ecosystem. Initial versions of the Sandbox from earlier research efforts implemented a proof of concept of the FDX API v.5.1, written in Golang [52]. This environment is meant to be open-sourced and maintained in order to provide APIs (and associated example proof-of-concept code) which simplify the implementation of data sharing/use protocols which are compatible with traceability and consent frameworks.

In addition to producing the technical specifications detailing the OTrace Traceability Protocol, this thesis advanced the state of the Open Banking Research Sandbox by incorporating the enhanced security features of the consent grant process as well as a sample implementation of a rudimentary Traceability Server along with some Traceability Record reporting according to the defined OTrace protocol. In this OTrace proof-of-concept, a Data Recipient requests access to a simulated User's data hosted on a Data Provider. The Data Provider performs User authentication (unimplemented) before generating an internal consent grant structure and a Traceability Policy Record to share back with the Data Recipient and a Traceability Server. The Data Recipient signs an attesting Traceability Policy Record for the Traceability

Server and then subsequent sharing and use records are sent when the data is shared and read.

Furthermore, rudimentary performance evaluations were conducted over the new implementations within the Open Banking Research Sandbox in order to get a basic sense of feasibility within existing open banking markets and organizations. These performance evaluations were conducted by measuring the time spent on each step of the OAuth2.0/OTrace protocol during Consent for User Initiated Data Sharing (see appendix A.4.1). To reduce the impact of network latencies on measurements, response time is not included in measurements. These evaluations found that the new protocols increased total latency of consent grant by 9.3% for the Data Provider (from 166.0ms to 181.5ms) and 68.8% for the Data Recipient (from 213.5ms to 360.4ms) [1]. This latency increase was expected due to the increased cryptologic complexity of the improved consent grant process and the total latency remains within predicted acceptable limits as not to inconvenience Users. These performance evaluations were rudimentary, and further evaluation is recommended at scale to more accurately represent maximum loads and normal use patterns expected in a fully saturated open banking market.

5.2 Continuing Work

5.2.1 Traceability Protocol Libraries

In order to facilitate implementation of the traceability protocol within an expanding open banking market and beyond into other realms of data sharing, it will be necessary to build out libraries in multiple programming languages to present a simplified API for leveraging the traceability protocol. These APIs should build upon the Open Banking Research Sandbox, producing implementation papers and documentation as well as a series of open-source examples that more thoroughly illustrate the use, benefits, and performance characteristics of the traceability protocol.

[1] These evaluations were conducted on a Intel(R) 8 Core(TM) i5-8250U CPU @ 1.60GHz. Only one instance of the Data Recipient was measured at a time interacting with the Data Provider.

These APIs, documentation, and libraries should be made available within multiple programming languages.

5.2.2 Building Towards Current Understanding of User Trust

Chapter 2 describes our current understanding of how User trust is established and grown as well as how to influence the adoption of new data sharing products. The studies revealed that an important aspect of User trust and consent is the user experience and the convenience/performance of the system. Thus, it is will be crucial to conduct more extensive performance analysis of not only the traceability protocol, but also of the User's interactions with the Traceability Server in order to view/audit Traceability Records. In addition, it will be important to identify the most effective manner of presenting the User with the Traceability Record information in a manner that conveys the honest data use practices of Data Recipients without overwhelming the User. One recommendation in this venture is to look into current uses of Layered Notices as introduced by GDPR [53].

5.2.3 Identifying Causes of User Trust

As mentioned at the end of Chapter 2, the studies that we reviewed to gain a better understanding of User trust are flawed. These studies identify predictive factors rather than factor of causation, They hold discrepancies between surveyed and actual practices. There is no study that reveals differences in how proclivities surrounding User trust vary throughout country, culture, and demographic. Better studies must be conducted to give us a better understanding of User trust so we can more effectively structure a traceability protocol to capture the information which will most positively influence that trust.

5.2.4 Refining Data Sharing Model

Though this thesis greatly benefited from the relationships with industry partners and discourse on the concurrent development of related protocols, interactions and

exchange of real-world practices remained limited. Open-source research can only get a team so far, and it will be critical to further leverage those partnerships so that future traceability researchers can get a better understanding of how to evolve the traceability protocol to better account for the real-world complexities, corner cases, and system integration which may serve as obstacles in the roll-out of the traceability protocol should these factors be overlooked or misunderstood.

Furthermore, it is crucial that protocol developers remain involved with these partners during and following protocol implementation such that any issues can be documented and addressed in later versions.

5.2.5 Traceability Server Migration

In its current form, the preconditions for migrating the Traceability Server are overly strict (all Traceability Record Sets connected by the parent_ids field of the Traceability Policy Record must contain Traceability Migrate Records from the Data Provider and the Data Recipient). This decision was made to prevent related Traceability Records Sets from becoming fragmented between different Traceability Servers. Furthermore, the migration is fragile and does not incorporate modern concepts of distributed systems, as when a Traceability Migrate Record is submitted, the actor must wait for a response to the provided migrated_uri before continuing to submit Traceability Records. This decision was made to prevent an Traceability Server from getting out of sync with a new Traceability Server in the middle of migration. However, both these design decisions were made as they were the simple solutions from a protocol design perspective on the part of the protocol developers (me). This migration process should be improved, applying principles of distributed systems, to make migration more practical, efficient, flexible, and most importantly, convenient for Users.

Appendix A

RFC

Internet Engineering Task Force (IETF)
Request for Comments: XXXX
Obsoletes: XXXX
Category: Standards Track
ISSN: XXXX-XXXX

Magendanz, Quinn
MIT Future of Data Initiative

February 2024

The OTrace Traceability Framework

A.0.0.1 Abstract

The OTrace Traceability Framework enables a User to track the sharing and usage of their personal data after it has been provided to, or collected by, an initial Data Provider that has explicitly received User consent. For the purpose of monitoring and auditing, the Data Provider and Data Recipient submit records to a Traceability Server to record initial User consent for data sharing as well as subsequent sharing and usage of the User's data. This specification introduces new standards for recording data sharing and usage as Traceability Records into a consent framework which builds off elements of the OAuth 2.0, PAR, PKCE, JWT, JWS, and TB protocols from RFCs 6749 [39], 9126 [41], 7636 [44], 7519 [45], 7515 [46], and 8471 [47] respectively, as well as the FAPI [42] and FDX [1] standards for financial data sharing.

A.0.0.2 Status of This Memo

This is a Massachusetts Institute of Technology (MIT) Computer Science and Artificial Intelligence Laboratory (CSAIL) document and is a product of the Future of Data Initiative (FDI).

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at `https://datatracker.ietf.org/drafts/current/`.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February XX, XXXX.

A.0.0.3 Copyright Notice

POSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A.0.0.4 Table of Contents

A.1 Introduction

This document defines the OTrace Traceability Server and the accepted Traceability Records within an authorization framework that extends upon OAuth 2.0 and other existing frameworks. OTrace enables a User to monitor records of initial consent to data sharing alongside subsequent sharing and usage in a manner which

emphasizes the properties of informed consent, data traceability, and accountable use [15].

A.1.1 Consent

At the time of this document's publishing, organizations provide inadequate modes for a User to view all the data sharing and data usage that they have consented to via both implicit and explicit User Agreements. Both users and governing bodies require a more detailed breakdown of consents granted. Any solution to informed consent must share the following characteristics to serve as a viable solution:

- *Consistency across service providers.* Issuing consent preferences must be standardized so that data sharing across organizations can carry with it the consent metadata.
- *Granularity in consent choices.* A user should be able to express specific ways data can be used while still disallowing other types of use. This must be granular enough to capture the different types of use while not so fragmented as to confuse users.
- *Equal access to services regardless of consent choices.* Use of "dark patterns" to coerce, wheedle, and manipulate users to grant consent shall not be permitted.
- *Flexible update plan.* As new laws are ratified, the protocol may need to incorporate additional consent metadata.
- *Secure, scalable communication protocols.*

A.1.2 Traceability

Organizations need a medium to prove their compliance to both Users and governing bodies, especially as data is shared to third parties. Any solution to traceability must provide a method of both detecting misuse of data and of handling changes in consent preferences. Traceability monitoring must also be scalable for large amounts

of both automated and manual data processing and for sharing across many different, untrusted organizations. Existing cryptographic systems cannot be relied on to prove compliance as they can neither sufficiently scale nor be properly managed across different environments. Instead, the solution will likely depend on accumulated attestations of data use and sharing from multiple different organizations to verify compliance. To monitor these traceability records, a simple, unified platform should be available which is capable of processing incoming information from many organizations and displaying summaries and reports to users.

By defining a framework for reporting and viewing consent, OTrace provides a consistent, secure, scalable solution for specifying granular consent in a unified standard where updates can be applied as they arise. The mode of authorization extends the OAuth 2.0 [39] and OpenID Connect (OIDC) [43] frameworks in a manner similar to FAPI [42], aiming to provide enhanced security features tailored to the needs of high-stakes exchange of personal data.

OTrace relies on doubly attested Traceability Records to document and verify the history of data sharing as both parties publish records to a Traceability Server for each step of a data sharing transaction. However, OTrace provide cannot doubly attested verification of data use as the data user may be the only entity aware of data use. Instead, OTrace relyies on market pressures and enforcement of federal/state regulation to pressure organizations to accurately report data usage after sharing has occured.

A.1.3 Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119].

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234]. Additionally, the rule URI-reference is included from "Uniform Resource Identifier (URI): Generic Syntax" [RFC3986].

Certain security-related terms are to be understood in the sense defined in

[RFC4949]. These terms include, but are not limited to, "attack", "authentication", "authorization", "certificate", "confidentiality", "credential", "encryption", "identity", "sign", "signature", "trust", "validate", and "verify".

Unless otherwise noted, all the protocol parameter names and values are case sensitive.

A.2 Definitions

A.2.1 Roles

- User

  The end user whose personal information is held within the data. An example of the definition of personal information can be found in California Civil Code §1798.140(o)(1) - CCPA [12]. This can include, but it not limited to Personally Identifiable Information, bank account numbers, location, activity history, etc.

- Authentication Server (AS)

  The service that performs User authentication and receives initial, explicit consent to collect and store User data.

- Resource Server (RS)

  The server which contains the data/resources requested by the Data Recipient on behalf of the user. The Resource Server must confirm that the Data Recipient has sufficient consent to retrieve the requested data.

- Data Provider (DP)

  The entity that owns both the Authentication Server and Resource Server. Where differentiating between the AS and RS roles is not necessary, this role may be used to refer generally to both the AS and RS servers or a server that implements both rolls.

Users usually have an existing relationship, account, and direct interactions with Data Providers. However, if a Data Recipient is given consent to further share User data, it assumes the role of Data Provider with the secondary sharing, potentially without direct User interaction.

- Data Recipient (DR)

  The third party which seeks to act on behalf of the User and/or access their data.

- Traceability Server (TS)

  The server which receives and stores records of data sharing and usage for the purpose of User monitoring and legal auditing.

A.2.2 Record Definitions

- Traceability Key-pair

  A Traceability Key-pair is a key-pair used to sign all JWS's sent to the Traceability Server. Each of the Data Provider and Data Recipient MUST have a Traceability Key-pair that is unique to the actor. The public key of the key-pair will be used to identify each actor as they interact with the Traceability Server.

- Traceability Record

  Traceability Records are received by the Traceability Server from both the Data Provider and Data Recipient and logged for verification by the User.

  Each Traceability Record MUST be represented as a JSON Web Token (JWT) [45]. Parameter names and string values MUST be included as JSON strings. Since Traceability Records are handled across domains and potentially outside of a closed ecosystem, per Section 8.1 of RFC 8259 [54], these JSON strings MUST be encoded using UTF-8 RFC 3629 [55]. Numerical values MUST be included as JSON numbers. Traceability Records MAY include any extension parameters. This JSON object of parameters constitutes the JWT Claims Set defined in

JWT RFC 7519 [45]. The JWT Claims Set is then signed within a JSON Web Signature (JWS) RFC 7515 [46] by the Data Provider or Data Recipient Traceability Key-pair. The result is a JWS-signed JWT of the following format:

```
base64url-encoded(UTF8(JWS Protected Header)) || '.' ||
base64url-encoded(JWS Payload) || '.' ||
base64url-encoded(JWS Signature)
```

The Traceability Server MUST verify each JWS and check that the public key used for signing matches one of the Proof Key of Code Exchange (PKCE) challenges [44] provided in the Traceability Record Set's most recent attested Traceability Policy Record.

Each Traceability Record contains the trace_id field for identifying the Traceability Record Set and the time field to prevent replay of Traceability Records. The Traceability Server MUST not include duplicate copies of records generated at the same time.

The following are the types of Traceability Records:

  - Traceability Policy Record
  - Traceability Share Record
  - Traceability Usage Record
  - Traceability Migration Record

- Traceability Record Set

  A Traceability Record Set is a collection of Traceability Records which all derive their consent from the same User consent for the sharing and/or use of data. They are initiated by a Traceability Policy Record from the Data Provider which documents the details of the User consent and is attested by a matching Traceability Policy Record from the Data Recipient. Further records will be

considered to be part of this Traceability Record Set if they contain the same trace_id (which is returned by the Traceability Server) and are signed by either the Data Provider or Data Recipient.

If a Data Recipient receives consent to further share User data, they MUST create and record records to a new Traceability Record Set. This new Traceability Record Set MUST use the same Traceability Server as the original and MUST use the parent_ids parameter of the new Traceability Policy Record to refer to the original Traceability Record Set.

A.3 Token Binding Format

Within OTrace, Token Bindings will be applied to Access Tokens, Authorization Codes, Refresh Tokens, JWT Authorization Grants, and JWT Data Recipient Authentication [48]. This cryptographically binds these tokens to a Data Recipient's Token Binding key pair, possession of which is proven on the TLS connections over which the tokens are intended to be used. This use of Token Binding protects these tokens from man-in-the-middle and token export and replay attacks. The Token Binding message format is defined using the TLS presentation language of RFC 8446 [56]:

```
enum {
   rsa2048_pkcs1.5(0), rsa2048_pss(1), ecdsap256(2), (255)
} TokenBindingKeyParameters;

struct {
   opaque modulus<1..2^16-1>;
   opaque publicexponent<1..2^8-1>;
} RSAPublicKey;

struct {
```

```
   opaque point <1..2^8-1>;
} TB_ECPoint;

struct {
   TokenBindingKeyParameters key_parameters;
   uint16 key_length;  /* Length (in bytes) of the following
                          TokenBindingID.TokenBindingPublicKey */
   select (key_parameters) {
       case rsa2048_pkcs1.5:
       case rsa2048_pss:
           RSAPublicKey rsapubkey;
       case ecdsap256:
           TB_ECPoint point;
   } TokenBindingPublicKey;
} TokenBindingID;

enum {
   (255)        /* No initial TB_ExtensionType registrations */
} TB_ExtensionType;

struct {
   TB_ExtensionType extension_type;
   opaque extension_data<0..2^16-1>;
} TB_Extension;

enum {
   provided_token_binding(0), referred_token_binding(1), (255)
} TokenBindingType;

struct {
```

```
   TokenBindingType tokenbinding_type;
   TokenBindingID tokenbindingid;
   opaque signature<64..2^16-1>; /* Signature over the concatenation
                                    of tokenbinding_type,
                                    key_parameters, and EKM */
   TB_Extension extensions<0..2^16-1>;
} TokenBinding;

struct {
   TokenBinding tokenbindings<132..2^16-1>;
} TokenBindingMessage;
```

A.4 Protocol Flow

A.4.1 Consent for Data Sharing

A.4.1.1 User Initiated

For the User to initiate data sharing with a Data Provider on behalf of a Data Recipient, the User MUST have independent relationships with both the Data Recipient and the Data Provider. The User MUST initiate the consent request from within the Data Recipient's experience/application. Before beginning the consent request, the Data Recipient determines the types of data access it intends to access from Data Provider and MUST disclose its intent to the User. During User Authentication to the Data Provider, the User MUST actively authorize the Data Provider to enable the Data Recipient's access to End User's data [1].

1. (1) PAR

   The Pushed Authorization Request (PAR) is an OAuth 2.0 JWT-Secured Authorization Request (JAR) Request Object as defined in RFC 9101 [57]. A Request Object is used to provide authorization request parameters for an OAuth

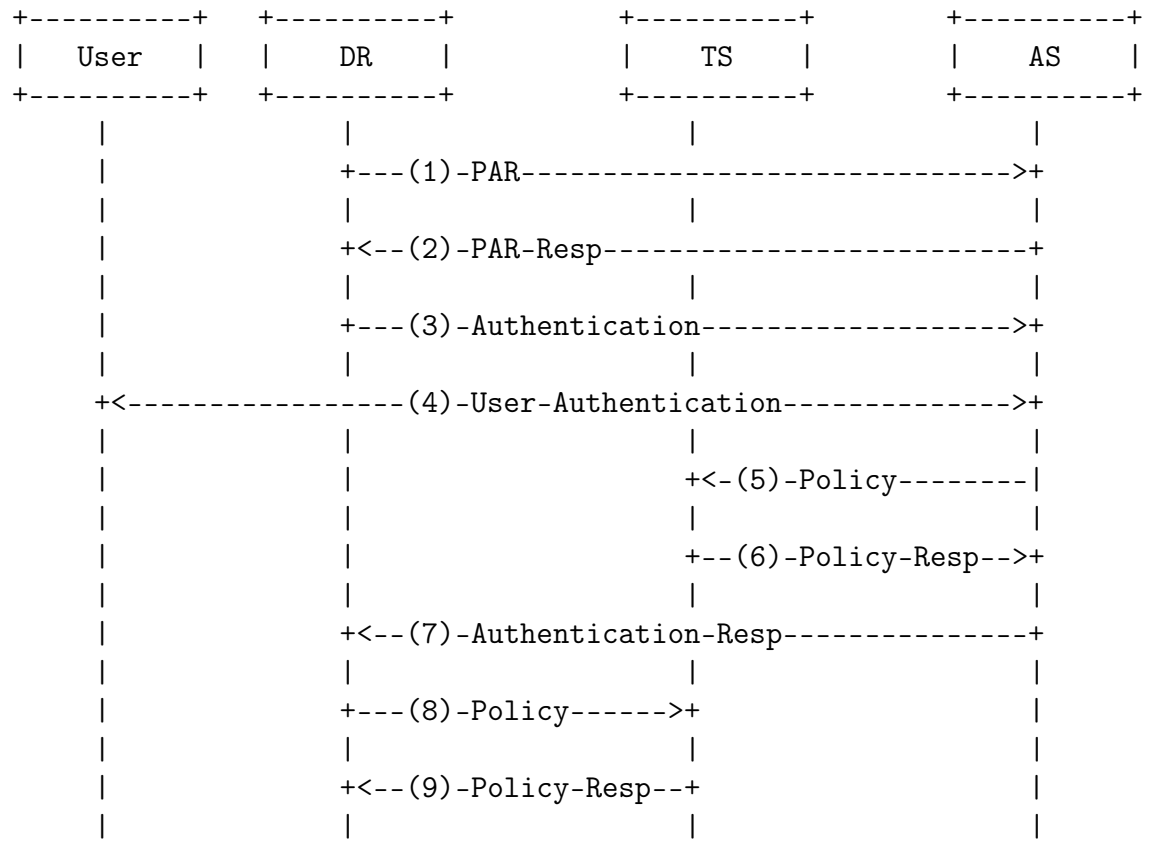

Figure A-1: User Initiated Consent

2.0 authorization request. The parameters are represented as the JWT Claims of the object. Parameter names and string values MUST be included as JSON strings. Since Request Objects are handled across domains and potentially outside of a closed ecosystem, per Section 8.1 of RFC 8259 [54], these JSON strings MUST be encoded using UTF-8 RFC 3629 [55]. Numerical values MUST be included as JSON numbers. The Request Object MAY include any extension parameters.

This JSON object constitutes the JWT Claims Set defined in JWT [45]. The JWT Claims Set is then signed within a JSON Web Signature (JWS) [46]. The result is a JWS-signed JWT.

The following parameters are included as top-level members in the JSON message of the Request Object with any additional parameters necessary for Data Recipient authentication:

- client_id

  REQUIRED. The Data Recipient identifier issued to the Data Recipient during the registration process described by [39].

- response_type

  REQUIRED. Value MUST be set to "code" per [39].

- state

  REQUIRED. An opaque value used by the Data Recipient to maintain state between the request and callback. The Authorization Server includes this value when redirecting the User back to the Data Recipient. The parameter MUST be used for preventing cross-site request forgery as described in [39]

- code_challenge

  REQUIRED. The PKCE code challenge is derived from [44] and [48]. The value is the base64url encoding (per Section 5 of [RFC4648] with all trailing padding ('=') characters omitted and without the inclusion of any line

breaks or whitespace) of the SHA-256 hash of the Provided Token Binding ID that the Data Recipient will use when calling the authorization server's token endpoint. The Provided Token Binding ID MUST be the same that will be used for Token Binding during the Authentication and Token Request steps. See section A.3 for the Provided Token Binding ID format.

- code_challenge_method

  REQUIRED. Value MUST be set to "TB-S256" per [48]

- authorization_details

  REQUIRED. Value MUST contain a JSON-formatted object with two members in compliance with the RAR format specified by [40]. These two members are "type" with the value "fdx_v1.0" and "consentRequest" containing a valid JSON ConsentRequest entity as defined in [1]. Below is an example of authorization_details:

```
"authorization_details": {
    "type": "fdx_v1.0",
    "consentRequest": {
        "durationType": "ONE_TIME",
        "lookbackPeriod": 60,
        "resources": [
            {
                "resourceType": "ACCOUNT",
                "consents": [
                    {
                        "category": "user.contact",
                        "uses": "marketing.communications",
                    },
                    {
                        "category": "user.demographic",
                        "uses": "marketing.advertising",
```

```
                    },
                    {
                        "category": "user.location.impercise",
                        "uses": "personalize.content",
                    }
                ]
            }
        ]
    }
}
```

The Request Object may contain other parameters as recommended in [41] and [39] such as redirect_uri, client_secret, iss, aud, scope, etc.

2. PAR Response

   If the verification is successful, the server MUST generate a request URI and provide it in the response with a 201 HTTP status code. The following parameters are included as top-level members in the message body of the HTTP response using the "application/json" media type as defined by RFC 8259 [54]:

   (a) request_uri

   REQUIRED. The request URI corresponding to the authorization request posted. This URI is a single-use reference to the respective request data in the subsequent authorization request. The way the authorization process obtains the authorization request data is at the discretion of the authorization server and is out of scope of this specification. There is no need to make the authorization request data available to other parties via this URI.

   (b) expires_in

RECOMMENDED. A JSON number that represents the lifetime of the request URI in seconds as a positive integer. The request URI lifetime is at the discretion of the authorization server but will typically be relatively short (e.g., between 5 and 600 seconds).

3. Authorization Request

The Data Recipient requests an authorization code after successful User authentication by adding the following parameters to the query component of the authorization endpoint URI using the "application/x-www-form-urlencoded" format, per [39]:

(a) client_id

REQUIRED. The Data Recipient identifier issued to the Data Recipient during the registration process described by [39].

(b) request_uri

REQUIRED. The request_uri returned from the PAR response.

(c) code_challenge_TS

REQUIRED. The PKCE code challenge is derived from [44] and [48]. The value is the base64url encoding (per Section 5 of [RFC4648] with all trailing padding ('=') characters omitted and without the inclusion of any line breaks or whitespace) of the SHA-256 hash of the public key of the Data Recipient's Traceability Key-pair.

(d) code_challenge_TS_method

REQUIRED. Value MUST be set to "TB-S256" per [48]

The Authorization MUST also contain the "Sec-Token-Binding" header for the purpose of performing OAuth Token Binding [48]. See section A.3 for details.

The provided token binding public key MUST be the same public key used in the PKCE challenge of the PAR Request Object. The token binding will use the Exported Keying Material (EKM) of the current TLS connection through

which the Authentication Request is made per [48]. The value of "Sec-Token-Binding" MUST be the base64url-encoded concatenation of the provided TokenBindingMessage preceded by two bytes denoting the length of the bytes that follow.

```
base64url-encoded(uint16 following_bytes ||
    provided_TokenBindingMessage)
```

4. User Authentication

   Data Provider performs User authentication. User consents to specific types and granularity of data sharing and authorized actions to the Data Recipient.

   While the User is providing consent, Data Provider MUST provide an option to the User to either supply a custom Traceability Server URL or agree to utilize a User-accessible, default Traceability Server. The Data Provider SHOULD provide a mode for Users to upload the public key of a custom traceability server.

5. Traceability Policy Record

   The Data Provider submits a Traceability Policy Record of the User's consent to the Traceability Server. This record contains a specific description of the type of data that the consent concerns as well as granular permissions limiting the types of actions that the Data Recipient can take concerning the shared data. Per section A.2.2, all Traceability Records are signed JWTs. Traceability Policy Records contain the following parameters:

   - trace_id

     REQUIRED. This value is required to be "0" when the Data Provider initiates a new consent trace. The Traceability Server will reply with the assigned trace_id for use in all subsequent records submitted to the Traceability Server.

- time

  REQUIRED. Time at which the Traceability Record was generated by the sender. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

- data_subject

  REQUIRED. The WebID HTTP URI representing the User who owns the data that this policy concerns [58]. This WebID Profile MUST be unique to the User relative to the Data Provider and SHOULD NOT provide any compromising information to those who view the WebID Profile. If the User does not host and provide their own unique WebID, it is expected for the Data Provider to host such a WebID that provides at minimum a unique identifier for the User. This follows the same standards as the W3C Solid Community Project [59].

- description

  REQUIRED. A human-readable summary for the User describing the relevant data categories and the purpose of consent, as phrased to the User during acknowledgment.

- consents

  REQUIRED. A JSON structure which describes the data categories that the User consent covers and what data uses are permitted upon each data category. The consents available SHOULD offer granularity in consent choices.

  The structure of this field is intentionally left open so that Data Providers may define the structure with proper granularity and specificity to match their use cases and to aid the User's understanding of how data will be used.

  In order to ensure consistency across service providers, it is recommended that Data Providers use an widely recognized structure such as the Fides Language Taxonomy Classification Groups [3]. An example would be a

JSON array of structures with the following fields:

  - data_categories

    Data Categories are labels to describe the type of data processed by your software. See section [3] for further details.

  - data_uses

    Data Uses are labels that describe how, or for what purpose(s) a component of your system is using data. See section [3] for further details.

- parent_ids

  OPTIONAL. A JSON array of trace_ids of other Traceability Policy Records that this Traceability Policy Record derives consent from.

- provider_challenge

  REQUIRED. This PKCE code challenge is derived from [44] and [48]. The value is the base64url encoding (per Section 5 of [RFC4648] with all trailing padding ('=') characters omitted and without the inclusion of any line breaks or whitespace) of the SHA-256 hash of the public key from the Data Provider's Traceability Key-pair.

- provider_challenge_method

  REQUIRED. Value MUST be set to "TB-S256" per [48]

- recipient_challenge

  REQUIRED. This PKCE code challenge is derived from [44] and [48]. The value is the base64url encoding (per Section 5 of [RFC4648] with all trailing padding ('=') characters omitted and without the inclusion of any line breaks or whitespace) of the SHA-256 hash of the public key from the Data Recipient's Traceability Key-pair.

- recipient_challenge_method

  REQUIRED. Value MUST be set to "TB-S256" per [48]

- trace_uri

  REQUIRED. A URI identifying the Traceability Server for this consent.

- trace_cert (optional)

  RECOMMENDED. The Data Provider may submit a certificate by which the Traceability Server can identify itself. This certificate should be the base64url-encoding of the certificate format described in either section 4.4.2. of RFC 8446 [56] or RFC 8879 [60]

6. Traceability Policy Record Response

   If the Traceability Policy Record is successfully received and processed, the server MUST generate a response with a 201 HTTP status code. The following parameters are included as top-level members in the message body of the HTTP response using the "application/json" media type as defined by RFC 8259 [54]:

   - trace_id

     REQUIRED. The trace_id generated by the Traceability Server to identify the Traceability Policy Record and all subsequent records under the policy. This MUST be unique to this set of Traceability Record Set.

7. Authentication Response

   If the User grants the access request and the Authentication message Token Binding is verified, the Authorization Server issues an authorization code and delivers it to the Data Recipient by adding the following parameters to the query component of the redirection URI using the "application/x-www-form-urlencoded" format, per RFC 6749 [39].

   - code

     REQUIRED. Per RFC 6749 [39], the authorization code generated by the Authorization Server. The authorization code MUST expire shortly after it is issued to mitigate the risk of leaks. A maximum authorization code lifetime of 10 minutes is RECOMMENDED. The Data Recipient MUST NOT use the authorization code more than once. If an authorization code is used more than once, the authorization server MUST deny the request

and SHOULD revoke (when possible) all tokens previously issued based on that authorization code. The authorization code is bound to the Data Recipient identifier and redirection URI.

- state

  REQUIRED if the "state" parameter was present in the Data Recipient Authorization Request per RFC 6749 [39]. The exact value received from the Data Recipient.

- trace_policy

  The same Traceability Policy Record sent by the Data Provider to the Traceability Server, re-signed after the returned trace_id value has been applied.

- id_token

  As defined in OpenID Connect [43], the ID Token data structure enables Users to be authenticated. The ID Token is a security token that contains Claims about the Authentication of an User by an Authorization Server when using a Data Recipient, and potentially other requested Claims. The ID Token is represented as a JWT [45]. The JWT MUST be signed as a JWS [46].

  The following Claims are used within the ID Token within the Authentication Response of OTrace. ID Tokens MAY contain other Claims. Any Claims used that are not understood MUST be ignored.

  - iss

    REQUIRED. Issuer Identifier for the Issuer of the response. See OpenID Connect for further details [43]

  - sub

    REQUIRED. Subject Identifier. See OpenID Connect for further details [43]

  - aud

REQUIRED. Audience(s) that this ID Token is intended for. See OpenID Connect for further details [43]

- exp

  REQUIRED. Expiration time on or after which the ID Token MUST NOT be accepted for processing. See OpenID Connect for further details [43]

- iat

  REQUIRED. Time at which the JWT was issued. See OpenID Connect for further details [43]

- code_hash

  REQUIRED. The SHA256 hash of the code parameter.

- state_hash

  REQUIRED. The SHA256 hash of the state parameter.

- trace_policy_hash

  REQUIRED. The SHA256 hash of the trace_policy parameter.

8. Traceability Policy Record

   The Data Recipient submits a record of the User's consent to the Traceability Server. This message follows the same format as the Traceability Policy Record step from the Data Provider above and MUST contain the same parameter values as provided by the Data Provider in the Authorization Response. See section 5 for required fields.

9. Traceability Policy Record Response

   If the Traceability Policy Record is successfully received and processed, the server MUST generate a response with a 201 HTTP status code.

A.4.1.2 Data Provider Initiated

The Data Provider may initiate Consent for Data Sharing at any point within their relationship with the User while they are already authenticated. This often occurs

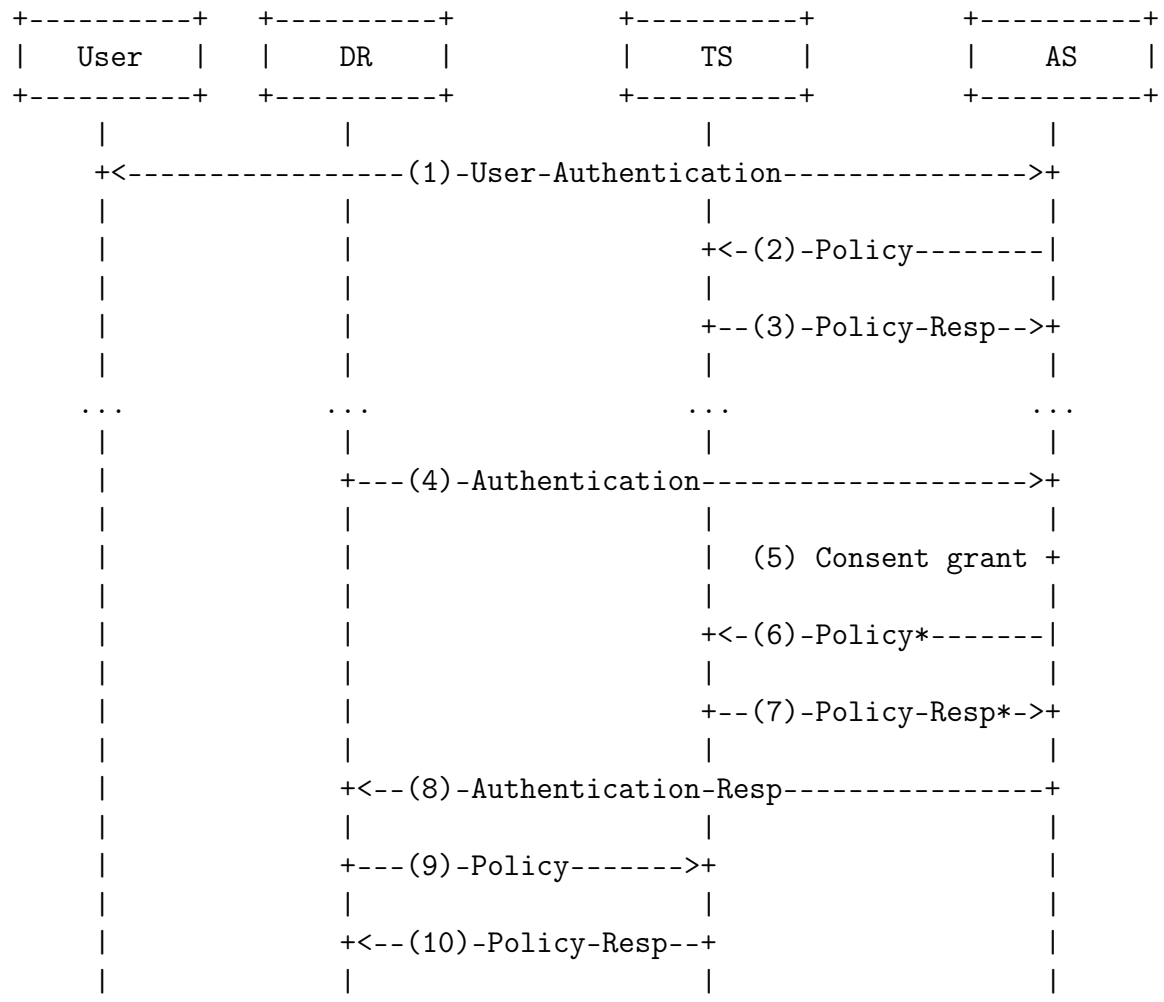

Figure A-2: Data Provider Initiated Consent

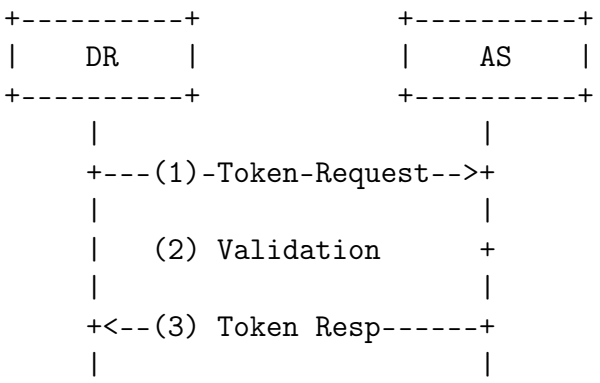

Figure A-3: Token Acquisition

when Data Providers regularly utilize the services of a third-party that requires User data to perform these services. Users often agree to this type of bulk, categorical data sharing in initial User agreements.

Consent for data sharing in these scenarios occurs with the same steps as when Data Recipients initiate data sharing but in a slightly different order. See Figure X for details.

However, only when the Traceability Policy Record from step (2) specifies a broad sharing consent do steps (6) and (7) occur. This Traceability Policy Record will reference the trace_id returned from step (3) in the list of parent_ids and must contain the same provider_challenge. Additionally in this case, steps (4) through (10) must occur every time a new Data Recipient requests access to bulk consented data from step (2).

A.4.1.3 Token Acquisition

1. Token Request

   The Data Recipient makes a request to the token endpoint by sending the following parameters using the "application/x-www-form-urlencoded" format per Appendix B with a character encoding of UTF-8 in the HTTP request entity-body per RFC 6749 [39]:

   - grant_type

REQUIRED. Value MUST be set to "authorization_code".

- code

  REQUIRED. The authorization code received from the authorization server.

- redirect_uri

  REQUIRED, if the "redirect_uri" parameter was included in the authorization request, and their values MUST be identical.

- client_id

  REQUIRED, if the client is not authenticating with the authorization server.

The Token Request MUST also contain the "Sec-Token-Binding" header for the purpose of performing OAuth Token Binding [48]. This header will contain both a provided and referred token according to the structure laid out in [47]. The provided token binding public key MUST be the same public key used in the PKCE challenge of the Request Object. The referred token binding public key will be used to sign a provided token binding when later communicating to the Resource Server. Both token bindings will use the EKM of the current TLS connection through which the Authentication Request is made per [48]. The value of "Sec-Token-Binding" MUST be the base64url-encoded concatenation of the provided and the referred TokenBindingMessage preceded by two bytes denoting the length of the bytes that follow.

```
base64url-encoded(uint16 following_bytes ||
    provided_TokenBindingMessage || referred_TokenBindingMessage)
```

2. Validation

   Follow validation steps required by RFC 6749 [39] authentication code grant workflow, specifically, ensuring that the authorization code was issued to the client, the authorization code is valid, and the "redirect_uri" parameter is

present if it was included in the initial authorization request. Additionally, ensure that the token binding is valid and matches the original PKCE per [48].

3. Token Response

   Per RFC 6749 [39], if the access token request is valid and authorized, the authorization server issues an access token and optional refresh token. The following parameters are included as top-level members in the message body of the HTTP response using the "application/json" media type as defined by [RFC8259] with a 200 (OK) status code:

   - access_token

     REQUIRED. The access token issued by the authorization server as described in RFC 6749 [39].

   - token_type

     REQUIRED. The type of the token issued as described in RFC 6749 [39].

   - grant_id

     REQUIRED. Contains the ConsentID per [1].

   - id_token

     As defined in OpenID Connect [43], the ID Token data structure enables Users to be authenticated. The ID Token is a security token that contains Claims about the Authentication of an User by an Authorization Server when using a Data Recipient, and potentially other requested Claims. The ID Token is represented as a JWT [45]. The JWT MUST be signed as a JWS [46].

     The following Claims are used within the ID Token within the Authentication Response of OTrace. ID Tokens MAY contain other Claims. Any Claims used that are not understood MUST be ignored.

     – iss

       REQUIRED. Issuer Identifier for the Issuer of the response. See OpenID Connect for further details [43]

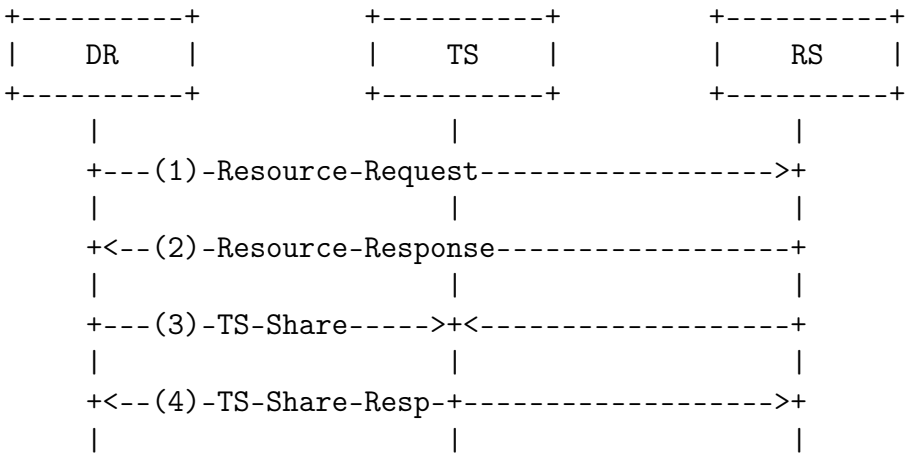

Figure A-4: Data Sharing

- sub

  REQUIRED. Subject Identifier. See OpenID Connect for further details [43]

- aud

  REQUIRED. Audience(s) that this ID Token is intended for. See OpenID Connect for further details [43]

- exp

  REQUIRED. Expiration time on or after which the ID Token MUST NOT be accepted for processing. See OpenID Connect for further details [43]

- iat

  REQUIRED. Time at which the JWT was issued. See OpenID Connect for further details [43]

- access_token_hash

  REQUIRED. The SHA256 hash of the access_token parameter per [2].

A.4.2 Data Sharing

1. Resource Request

The client accesses protected resources by presenting the access token to the resource server. The resource server MUST validate the access token and ensure that it has not expired and that its scope covers the requested resource. The methods used by the resource server to validate the access token (as well as any error responses) are beyond the scope of this specification but generally involve an interaction or coordination between the resource server and the authorization server.

The method in which the client utilizes the access token to authenticate with the resource server depends on the type of access token issued by the authorization server. Typically, it involves using the HTTP "Authorization" request header field [RFC2617] with an authentication scheme defined by the specification of the access token type used, such as [RFC6750]. See RFC 6749 [39] for further details on access token types.

In addition to access token, the Resource Request should include the following parameters as top-level members in the message body of the HTTP response using the "application/json" media type as defined by [RFC8259]:

- id_token

  The id_token parameter received during the Token Request step.

The Token Request MUST also contain the "Sec-Token-Binding" header for the purpose of performing OAuth Token Binding [48]. This header will contain a provided token according to the structure laid out in [47]. The provided token binding public key MUST match the referred token binding previously issued to the Authentication Server. The token binding will use the EKM of the current TLS connection through which the Resource Request is made per [48]. The value of "Sec-Token-Binding" MUST be the base64url-encoded concatenation of the provided TokenBindingMessage preceded by two bytes denoting the length of the bytes that follow.

2. Resource Response

The Data Provider provides the requested resource to the Data Recipient.

3. Traceability Share Record

   The Data Provider and Data Recipient submit a Traceability Share Record of the sharing of User data with the Data Recipient within the bounds of the consent detailed in the Traceability Policy Record. Per section A.2.2, all Traceability Records are signed JWTs. Traceability Share Records contain the following parameters:

   - trace_id

     REQUIRED. The trace_id for this Traceability Record Set.

   - time

     REQUIRED. Time at which the Traceability Record was generated by the sender. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

   - data_shared

     REQUIRED. A JSON structure which describes the data categories that the shared data falls within and what data uses are permitted upon each data category. This should follow the format used for the consents parameter of the Traceability Policy Record.

   - description

     REQUIRED. A human-readable summary for the User describing the relevant data being shared and the purpose of share.

4. Traceability Share Record Response

   If the Traceability Share Record is successfully received and processed, the server MUST generate a response with a 201 HTTP status code.

A.4.3 Data Usage

1. Data Usage

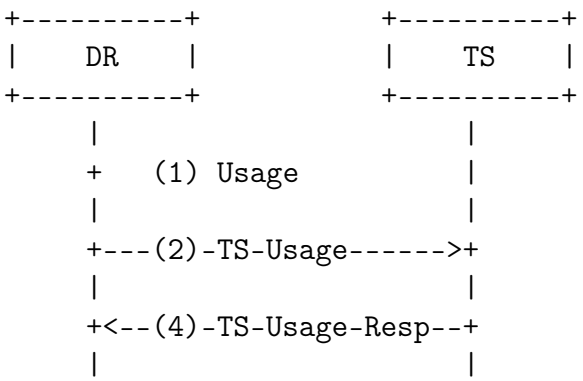

Figure A-5: Data Usage

The Data Recipient uses the shared User data within the bounds of the consent detailed in the Traceability Policy Record.

2. Traceability Usage Record

   The Data Recipient submits a Traceability Usage Record when the shared User data is used. The use MUST be within the bounds of the consent detailed in the Traceability Policy Record. Per section A.2.2, all Traceability Records are signed JWTs. Traceability Usage Records contain the following parameters:

   - trace_id

     REQUIRED. The trace_id for this Traceability Record Set.

   - time

     REQUIRED. Time at which the Traceability Record was generated by the sender. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

   - data_used

     REQUIRED. A JSON structure which describes the data categories that the used data falls within and what data uses are taking place upon each data category. This should follow the format used for the consents parameter of the Traceability Policy Record.

   - description

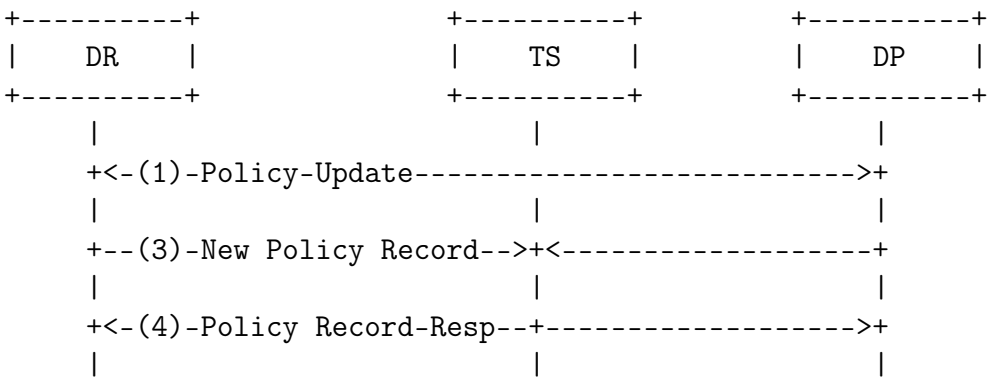

Figure A-6: Policy Update

REQUIRED. A human-readable summary for the User describing the relevant data being used and the purpose for use.

3. Traceability Usage Record Response

   If the Traceability Usage Record is successfully received and processed, the server MUST generate a response with a 201 HTTP status code.

A.4.4 Policy Update

1. Policy Update

   The Data Provider and Data Recipient exchange a new Traceability Policy Record if there is a change in consent or another field of the Traceability Policy Set's most recent Traceability Policy Record. This may result from changes in, for example, User data sharing preferences, Data Provider policies, or Data Recipient data use.

   Any new permissions beyond the original Traceability Policy Record MUST receive User consent as in the initial Consent process.

   A removal of all permissions is equivalent to a request for data deletion.

   The new Traceability Policy Record MUST NOT change the User, Data Provider, Data Recipient, and Traceability Server. As such, the new Traceability Policy Record SHOULD NOT contain changes to any field other than

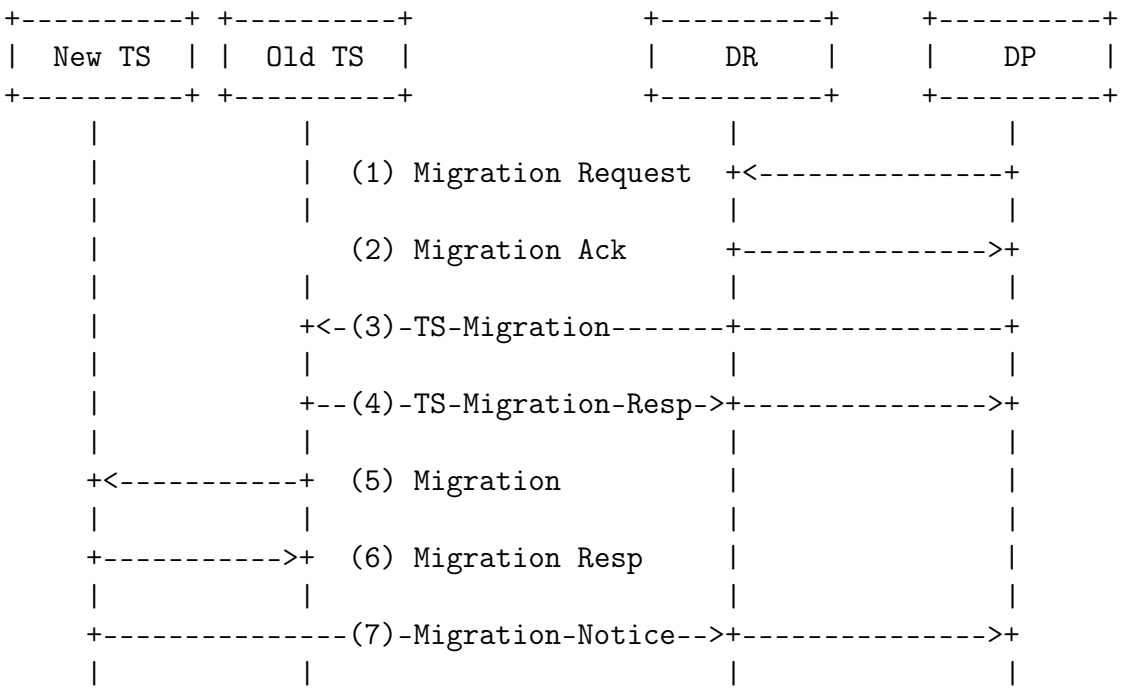

Figure A-7: Data Migration

description, consents, and parent_ids unless required to update keys or URIs.

Per section A.2.2, all Traceability Records are signed JWTs. See section 5 for required fields.

If the new Traceability Policy Record is successfully received, it MUST be acknowledged with a 200 HTTP status code.

2. Traceability Policy Record

   Both the Data Provider and Recipient submit their individually signed new Traceability Policy Records to the Traceability Server.

3. Traceability Policy Record Response

   If the new Traceability Policy Record is successfully received and processed, the server MUST generate a response with a 201 HTTP status code.

A.4.5 Migrate Traceability Server

1. Migration Request

The Data Provider submits a Traceability Migration Record to the Data Recipient if there is a change in the Traceability Server. This may result from changes in, for example, User requested change in Traceability Server or Data Provider change in default Traceability Server. Per section A.2.2, all Traceability Records are signed JWTs. Traceability Migration Records contain the following parameters:

- trace_id

  REQUIRED. The trace_id for this Traceability Record Set.

- time

  REQUIRED. Time at which the Traceability Record was generated by the sender. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

- force

  OPTIONAL. "true" if the Data Provider and Data Recipient should start using the new Traceability Server immediately, submitting this Traceability Migration Record to the new Traceability Server.

- migrated_uri

  REQUIRED. A URI identifying the Data Provider's callback for when Traceability Server migration is complete.

- trace_uri

  REQUIRED. A URI identifying the new Traceability Server for this Traceability Record Set.

- trace_cert

  RECOMMENDED. The Data Provider may submit a certificate by which the Traceability Server can identify itself. This certificate should be the base64url-encoding of the certificate format described in either section 4.4.2. of RFC 8446 [56] or RFC 8879 [60].

2. Migration Acknowledgement

   If the Traceability Migration Record is successfully received and processed, the Data Recipient MUST generate a response with a 202 HTTP status code.

3. Traceability Migration Record

   Both the Data Provider and Recipient submit their individually signed Traceability Migration Records to the Traceability Server. The Data Recipient should replace the Data Provider migrated_uri with its own before sending its Traceability Migration Record.

4. Migration

   The migration of a Traceability Record Set from one Traceability Server to another uses the "application/json" media type as defined by RFC 8259 [54]. All Traceability Records within the Traceability Record Set MUST be wrapped within the following structure and sent as a JSON array:

   - type

     REQUIRED. Indicates the type of the Traceability Record. Value is either "policy", "share", "use", "change", or "migration".

   - time

     REQUIRED. Time at which the Traceability Record was received by the Traceability Server. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

   - trace

     REQUIRED. The Traceability Record represented as a JWS signed JWT, as originally received by the Traceability Server.

5. Migration Response

   If the Traceability Migration Record is successfully received and processed, the new Traceability Server MUST generate a response with a 201 HTTP status

code. The new Traceability Server MUST verify the signatures from each Traceability Record in the migrated Traceability Record Set. The new Traceability Server MUST assign a new trace_id to the Traceability Record Set for future Traceability Record submission if the old trace_id is not unique within the new Traceability Server.

6. Migration Notice x2

   If the migration to the new Traceability Server is successfully received and processed, the new Traceability Server MUST send a PUT request to both the Data Provider and Data Recipient's migrate_uri conveyed within the last Traceability Migration Records. The following parameters are included as top-level members in the message body of the HTTP response using the "application/json" media type as defined by RFC 8259 [54]:

   - old_trace_id

     REQUIRED. The trace_id on the Traceability Migration Request.

   - new_trace_id

     REQUIRED. A new trace_id to uniquely represent the migrated Traceability Record Set on the new Traceability Server. This MAY be the same as old_trace_id if it is unique within the new Traceability Server.

     If trace_cert from the Traceability Migration Request was provided, these fields should instead be provided as a JWS that can be verified with trace_cert.

For a Traceability Server to commence migration, all Traceability Record Sets connected by the parent_ids field of the Traceability Policy Record must contain Traceability Migrate Records from the Data Provider and the Data Recipient (unless not present) detailing the same migration information [1]. While waiting on a response

[1]This prerequisite for migration is significant and potentially impractical and inconvenient for Users. This decision was made to reduce the complexity of implementation for v1.0 of this protocol. It is recommend that subsequent versions create a more flexible migration framework that builds on principles of distributed computing.

to indicating completion of migration which will contain a new trace_id, no additional Traceability Records should be submitted to the Traceability Server.

Bibliography

[1] Financial-Data-Exchange, "Financial data exchange api specification," May 2022. v5.1.

[2] D. Fett, P. Hosseyni, and R. Küsters, "An extensive formal security analysis of the openid financial-grade api." https://arxiv.org/pdf/1901.11520.pdf, Jan. 2019. arXivLabs.

[3] ethyca, "Fides language." https://ethyca.github.io/fideslang/.

[4] Apple, "ios and ipados software license agreement." https://www.apple.com/legal/sla/docs/iOS16_iPadOS16.pdf, 2022. v16.

[5] J. M. C. Bank, "Chase u.s. consumer privacy notice." https://www.chase.com/digital/resources/privacy-security/privacy/consumer-privacy-notice, apr 2022.

[6] B. Auxier, L. Raine, M. Anderson, A. Perrin, M. Kumar, and E. Turner, "Americans' attitudes and experiences with privacy policies and laws." https://www.pewresearch.org/internet/2019/11/15/americans-attitudes-and-experiences-with-privacy-policies-and-laws/, Nov. 2019. Pew Research Center.

[7] K. Liao, E. Radler, and D. Weitzner, *Building Accountable Systems Through Data Traceability (Extended Abstract)*. MIT Internet Policy Research Initiative.

[8] L. Brodsky and L. Oakes, "Data sharing and open banking," *McKinsey & Company*, vol. 1105, 2017.

[9] "Payment services directive (psd2)." https://finance.ec.europa.eu/regulation-and-supervision/financial-services-legislation/implementing-and-delegated-acts/payment-services-directive_en, 2015.

[10] "General data protection regulation (gdpr)." https://gdpr-info.eu, 2016.

[11] "Section 1033 of the dodd-frank wall street reform and consumer protection act." https://www.congress.gov/bill/111th-congress/house-bill/4173, 2010.

[12] "California consumer privacy act (ccpa)." https://oag.ca.gov/privacy/ccpa, 2018.

[13] "Notice of proposed rulemaking - required rulemaking on personal financial data rights." https://files.consumerfinance.gov/f/documents/cfpb-1033-nprm-fr-notice_2023-10.pdf, 2023.

[14] "Mit future of data." https://futureofdata.mit.edu/.

[15] "Accountability and traceability white paper and research roadmap." https://futureofdata.mit.edu/tr/2023/fod-account-trace-20230418.pdf, Apr. 2023. Future of Data Initiative, Massachusetts Institute of Technology.

[16] B. Glock, "Unlocking the opportunities of open banking." https://navigate.visa.com/na/money-movement/unlocking-the-opportunities-of-open-banking/, July 2022. Visa.

[17] "Visa open banking consumer survey," Apr. 2022. n=1,500, representative sample of U.S. population based on Census Bureau. Administered digitally.

[18] N. Lawford, "Literature review of user trust in online and open banking technologies (draft)." MIT Future of Data Initiative, 2023.

[19] B. Vatanasombut, M. Igbaria, A. C. Stylianou, and W. Rodgers, "Information systems continuance intention of web-based applications customers: The case of online banking," *Information & management*, vol. 45, no. 7, pp. 419–428, 2008.

[20] P. L. Yu, M. Balaji, and K. W. Khong, "Building trust in internet banking: a trustworthiness perspective," *Industrial Management & Data Systems*, vol. 115, no. 2, pp. 235–252, 2015.

[21] R. F. Hasandoust and M. M. Saravi, "Identifying the effect of successful e-banking on customers' satisfaction, trust, commitment and loyalty," *QUID: Investigación, Ciencia y Tecnología*, no. 1, pp. 1716–1726, 2017.

[22] A. Mukherjee and P. Nath, "A model of trust in online relationship banking," *International journal of bank marketing*, vol. 21, no. 1, pp. 5–15, 2003.

[23] R. Kumra, R. Mittal, and L. Gunupudi, "Trust and its determinants in internet banking: A study of private sector banks in india 1," in *Information Systems*, pp. 141–158, Routledge India, 2019.

[24] C. Cruijsen, J. de Haan, and R. Roerink, "Trust in financial institutions: A survey," tech. rep., Netherlands Central Bank, Research Department, 2020.

[25] P. Palos-Sanchez, J. R. Saura, and F. Martin-Velicia, "A study of the effects of programmatic advertising on users' concerns about privacy overtime," *Journal of Business Research*, vol. 96, pp. 61–72, 2019.

[26] A. Sanayei and A. Noroozi, "Security of internet banking services and its linkage with users' trust: A case study of parsian bank of iran and cimb bank of malaysia," in *2009 International Conference on Information Management and Engineering*, pp. 3–7, IEEE, 2009.

[27] R. Chan, I. Troshani, S. Rao Hill, and A. Hoffmann, "Towards an understanding of consumers' fintech adoption: The case of open banking," *International Journal of Bank Marketing*, vol. 40, no. 4, pp. 886–917, 2022.

[28] O. Armantier, S. Doerr, J. Frost, A. Fuster, and K. Shue, "Whom do consumers trust with their data? us survey evidence," tech. rep., Bank for International Settlements, 2021.

[29] I. Van Zeeland and J. Pierson, "In banks we trust: Banks as custodians of personal data in open banking ecosystems," in *TPRC49: The 49th Research Conference on Communication, Information and Internet Policy*, 2021.

[30] S. Rotchanakitumnuai and M. Speece, "Corporate customer perspectives on business value of thai internet banking services," *Journal of electronic commerce research*, vol. 5, no. 4, pp. 270–286, 2004.

[31] A. Ali, A. Hameed, M. F. Moin, and N. A. Khan, "Exploring factors affecting mobile-banking app adoption: a perspective from adaptive structuration theory," *Aslib Journal of Information Management*, vol. 75, no. 4, pp. 773–795, 2023.

[32] E. Masoud and H. AbuTaqa, "Factors affecting customers' adoption of e-banking services in jordan," *Information Resources Management Journal (IRMJ)*, vol. 30, no. 2, pp. 44–60, 2017.

[33] R. Al-Dmour, M. Alnafouri, and A. Al-Alwan, "The mediating role of e-satisfaction in the relationship between e-service quality and customer e-loyalty in internet banking," *Jordan Journal of Business Administration*, vol. 15, no. 2, 2019.

[34] G. Briones de Araluze and N. Cassinello Plaza, "The relevance of initial trust and social influence in the intention to use open banking-based services: An empirical study," *SAGE Open*, vol. 13, no. 3, p. 21582440231187607, 2023.

[35] H. A. Zadha and G. Suparna, "The role of brand trust mediates the effect of perceived risk and brand image on intention to use digital banking service,"

[36] M. Bijlsma, C. van der Cruijsen, and N. Jonker, "Consumer propensity to adopt psd2 services: trust for sale?," 2020.

[37] R. J. Nam, "Open banking and customer data sharing: Implications for fintech borrowers," 2022.

[38] Z. He, J. Huang, and J. Zhou, "Open banking: Credit market competition when borrowers own the data," *Journal of financial economics*, vol. 147, no. 2, pp. 449–474, 2023.

[39] "The oauth 2.0 authorization framework." https://datatracker.ietf.org/doc/html/rfc6749, Oct. 2012. Internet Engineering Task Force (IETF).

[40] "Oauth 2.0 rich authorization requests." https://datatracker.ietf.org/doc/html/rfc9396, May 2023. Internet Engineering Task Force (IETF).

[41] "Oauth 2.0 pushed authorization requests." https://datatracker.ietf.org/doc/html/rfc9126, Sept. 2021. Internet Engineering Task Force (IETF).

[42] C. Michael, F. Gyara, J. Heenan, T. Lodderstedt, D. Postnikov, and D. Tonge, "Financial-grade api (fapi) profiles." https://openid.net/wg/fapi/, July 2022. OpenID.

[43] "Openid connect core 1.0." https://openid.net/specs/openid-connect-core-1_0.html, Nov. 2014. OpenID.

[44] "Proof key for code exchange by oauth public clients." https://datatracker.ietf.org/doc/html/rfc7636, Sept. 2023. Internet Engineering Task Force (IETF).

[45] "Json web token." https://datatracker.ietf.org/doc/html/rfc7519, May 2015. Internet Engineering Task Force (IETF).

[46] "Json web signatures." https://datatracker.ietf.org/doc/html/rfc7515, May 2015.

[47] "The token binding protocol version 1.0." https://datatracker.ietf.org/doc/html/rfc8471, Oct. 2018. Internet Engineering Task Force (IETF).

[48] "Oauth 2.0 token binding." https://datatracker.ietf.org/doc/html/draft-ietf-oauth-token-binding-08, Oct. 2018. Internet Engineering Task Force (IETF).

[49] "Data rights protocol." https://github.com/consumer-reports-innovation-lab/data-rights-protocol/. Consumer Reports Innovation Lab.

[50] S. Zimmeck, P. Snyder, J. Brookman, and A. Zucker-Scharff, "Global privacy control." https://privacycg.github.io/gpc-spec/, July 2023.

[51] "Solid specification." https://solidproject.org/TR/protocol, Dec. 2022. W3C Solid Community Group.

[52] K. Liao, "Traceability protocol for open banking (draft)." MIT Future of Data Initiative, 2023.

[53] "Guidelines 01/2022 on data subject rights - right of access." https://edpb.europa.eu/system/files/2022-01/edpb_guidelines_012022_right-of-access_0.pdf, Jan. 2022. European Data Protection Board.

[54] "The javascript object notation (json) data interchange format." https://datatracker.ietf.org/doc/html/rfc8259, Dec. 2017. Internet Engineering Task Force (IETF).

[55] "Utf-8, a transformation format of iso 10646." https://datatracker.ietf.org/doc/html/rfc3629, Nov. 2003. Internet Engineering Task Force (IETF).

[56] "The transport layer security (tls) protocol version 1.3." https://datatracker.ietf.org/doc/html/rfc8446, Aug. 2018.

[57] "The oauth 2.0 authorization framework: Jwt-secured authorization request (jar)." https://datatracker.ietf.org/doc/html/rfc9101, Aug. 2021. Internet Engineering Task Force (IETF).

[58] A. Sambra, H. Story, and T. Berners-Lee, "Webid 1.0." https://www.w3.org/2005/Incubator/webid/spec/identity/, 2014. W3C.

[59] S. Capadisli and T. Berners-Lee, "Solid webid profile." https://solid.github.io/webid-profile/#solid-profile, 2023. W3C Solid Community Group.

[60] "Tls certificate compression." https://datatracker.ietf.org/doc/html/rfc8879, Dec. 2020. Internet Engineering Task Force (IETF).