STRONGER TOGETHER: CROSS SERVICE MEDIA RECOMMENDATIONS

H. Ricklefs[1], M. Leonard[2], J. Loveridge[1], J. Carter [1], K. Mackay[1], J. Allnutt[1], T. Preece[1], T. Nooney[1], K. Bennett[1], J. Cox[1], A. Greenham[1], T. Broom[1], A. Balantyne[1], T. Al-Ali Ahmed[1], B. Thompson[1]

[1]BBC, UK and [2]BBC now Inrupt, UK

ABSTRACT

This paper presents a way to use an individual's entire media streaming history from all their providers to inform recommendations without any of the contributing services needing to process or hold the data. To enable this, the user's personal data is held in a Solid pod, a secure personal web server designed to give the user greater control over their personal data supporting a more flexible approach to how it is used.

We present an application and supporting architecture based around Solid pods that can process media consumption data in a way that is controlled by the user, generating a general purpose 'Media Profile' which media organisations can use to inform their recommendation services. This profile could form the basis of an open standard and would allow broadcasters to offer audiences more relevant content in a way that limits their data liability risk and safeguards their privacy.

We will demonstrate the system in operation along with a platform and supporting application.

INTRODUCTION

Media organisations expend enormous energy collecting and analysing data, which is primarily used to enhance products or to support targeted advertising, and so user data has traditionally been seen as a competitive advantage and as an asset owned by the company rather than something a user has any rights to. One consequence of this is that user data has historically been tightly coupled to its originating service and rendered non-interoperable and inaccessible. Organisations like Spotify, Netflix and the BBC operate separate infrastructures for data acquisition, analysis and use driving recommendations.

At the same time, new data protection laws, such as the EU GDPR (1) and California Consumer Privacy Act (2), have been introduced around the world establishing clear frameworks within which users have the right to access and even remove personal data held by organisations.

One reason for data gathering is the use of user data to drive personalisation of services. From a user perspective all this data is spread over many locations, users have limited or no control over what data is captured, processed, and analysed about them, and often processes set up to conform with data protection laws mean that data cannot easily be

collected and shared. There are no standardised data formats to support sharing across organisations or services, even if there was a desire to make such data reusable.

In line with the BBC's role as a public service organisation[1] seeking to deliver value to audiences, society, and the wider economy, we have been exploring alternative models for data stewardship that could enhance trust, support standardisation of data schemas, and offer support for the creation of cross-service media profiles that can drive enhanced recommendations.

We have developed a working system based on the open-source Solid technology (3), as implemented by the commercial operator Inrupt (4), that demonstrates a full end-to-end solution that can ingest data from multiple media providers and create an exportable cross-service media profile capable of driving recommendation services, all based on a permissions-based model that retains user agency.

BACKGROUND TO THE PROJECT

BBC Research and Development (R&D), Technology Strategy and Architecture (TS&A) and User Experience and Design (UX&D) departments have been actively exploring the nature of human data interaction (8, 9, 10) for several years and were collaborators in a research programme funded by the UK Engineering and Physical Sciences Research Council (EPSRC) called Databox (5), a personal datastore system that allowed data-driven applications to run on any arbitrary personal data, locally on a user-owned secured device. This work culminated in the development of the BBC Box (11), a prototype consumer device based on an internet-connected Raspberry Pi, running Databox that offered users local storage and control of their personal data.

The BBC Box created sufficient interest to justify further exploration, and as the Databox project had concluded we decided to use an alternative data storage service, Solid (the name is derived from SOcial LInked Data), which is a set of open specifications developed by a dedicated W3C Community Group (3).

TECHNICAL APPROACH

Data Ecosystem Context

Throughout the build of our project we have always considered how our system would fit within the wider context of a *public service data ecosystem*, understood as a bounded collection of infrastructure, analytics, and applications used to acquire, store, analyse and operationalise data that acts in the public interest (6)

We consider such a data ecosystem to at least have three main components:

1. A place to securely store data: a secure repository for personal data and other data that individuals want to keep control over that afford the user greater oversight, access and control over data about them.

[1] as noted in (ref: https://www.bbc.co.uk/rd/projects/new-forms-value-bbc-data-economy), the BBC seeks to play a different role: "As a non-commercial research department with an obligation to explore emerging technologies for the good of licence fee payers, the UK and the world, BBC R&D is well-placed to apply public service considerations to the online environment for the betterment of society and to develop a new range of tools and services that provide real value to audiences."

2. A data exchange: a service that offers access to aggregate, open, and licensed non-personal data by enabling data transfer from third party services by exposing a unified interface to data.
3. A set of services for individuals, groups and entities that use the stored data to deliver public value.

The dominant set of technologies for providing repositories are called personal data stores or personal data services (PDS). There are many players in the PDS landscape, the business models are still emerging, and they vary in how user data is stored and processed, but they each offer data storage and a range of tools to facilitate the integration of third-party apps.

The Solid Platform

To build our data repositories we chose Solid, a set of specifications that allows for an ecosystem of interoperable applications and data (3). Solid uses W3C standards and offers open solutions to deal with decentralised identity, global identifiers, authentication and authorization mechanisms, data interoperability and query interfaces. The PDSs created by Solid are called 'pods'. Pods are a decentralised data store like a secure personal web server for people's data. Once data is stored in a user's pod, the user can decide who and what has access to it, and revoke access at any time. A pod can store any kind of data, such as image, video or text files. However, the Solid ecosystem works primarily with linked data resources (7) in RDF format[2], which gives different applications the ability to work with the same data. The use of linked data and other interoperable data formats allows for a consistent description of the data and a way for the relationship between resources to be described and understood by applications. We have used Solid pods to build a user-centred data storage system and a prototype media discovery tool to test our ideas in practice, and deployed this system using the Enterprise Solid Server (ESS) developed by Inrupt (13).

There are several excellent technologies and people working on personal data store projects over the world, so why did we choose to use Solid? Firstly, it is open-source and is a set of proposed W3C standards ie - we can build our own and dig deep into any aspect for this initial trial. Secondly it is Web native - it embodies the principles of the web, especially that of universal access, which is one of the essential principles in the way we deliver our services. Thirdly, there is a large and active developer community, and finally, commercial support is available through companies like Inrupt, who want to provide an enterprise scale software solution of the Solid specification.

We are open to developing on other platforms too should another technology prove more suitable in the future, and we are continuously monitoring the market for alternative solutions. The ability for multiple organisations to be part of this data ecosystem is essential, some of the proposed legislations such as the EU Digital Services Act and Digital Markets Act (14) call out that organisations might have to make a choice on the role they want to perform within the data ecosystem such as data store provider, identity provider, verifying attributes (such as address, or age) or providing services. Therefore, we expect there to be several organisations and service providers within the data ecosystem.

[2] https://www.w3.org/RDF/

THE TECHNICAL SOLUTION

System Setup

We have built a working system (Figure 1) which provides each user with a Solid pod and allows them to import their media consumption data from a variety of sources. Using this data, they are then able to see their combined media viewing history, as well as receive recommendations for BBC content based on analysis of their data from other services. For hosting the Solid pods we initially used the open source Node solid server (12) implementation but decided to evaluate Inrupt's production-grade Solid server, ESS (13), due to the advanced security features and operational tooling.

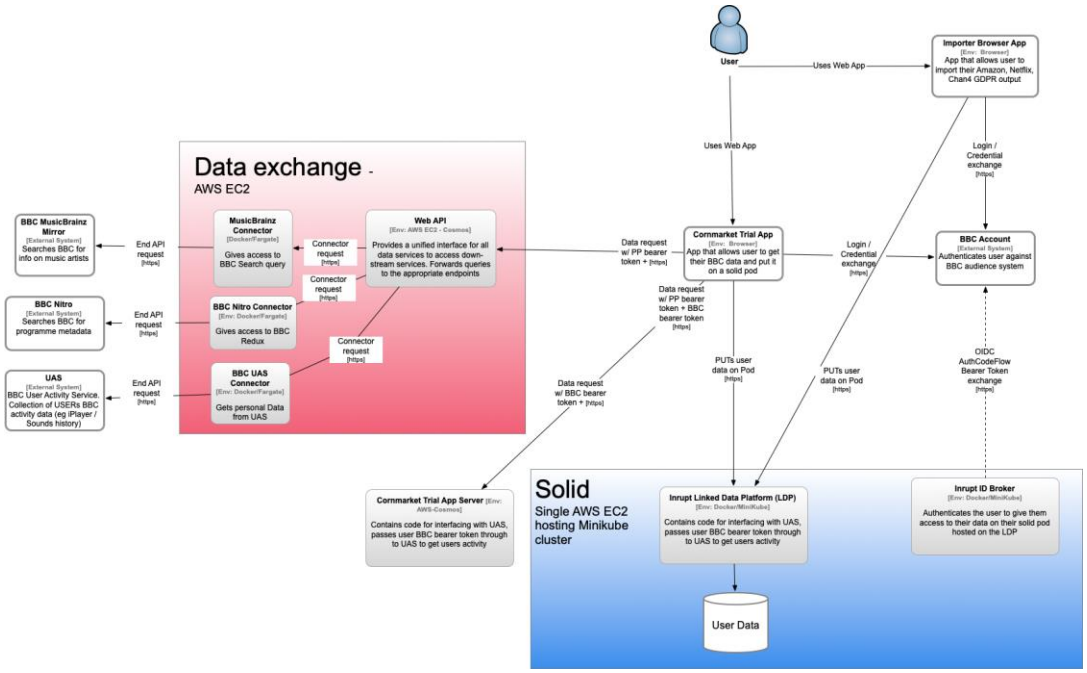

Figure 1 – System Architecture Diagram

A fully-specified mobile app called *My PDS* was designed by the UX team, and a slice of this larger app was prototyped as a web application, referred to as the demonstrator. The demonstrator was built using React[3] and TypeScript[4] with the look and feel of a mobile app. This application was used to support our user research into attitudes to personal data (to be published separately).

Demonstrator

The demonstrator allows a user to sign into their Solid pod and then choose which media services they want to import data from. Importing Spotify and BBC data is supported through existing APIs, while for Netflix a user has to obtain their viewing history through a data access request outside of the demonstrator. For all three services, viewing/listening history is imported and saved to the pod as **things** and **actions**, where a *thing* represents

[3] https://reactjs.org/
[4] https://www.typescriptlang.org/

an item of media (tv episode, film, song etc.) and an *action* represents a particular item of media being played at a particular time by the user.

Reading and writing data from and to a pod is done using Inrupt's JavaScript Client Libraries (15). Things are stored in a dataset per type: one dataset for tv episodes, one dataset for films etc. All actions were stored in a single, large dataset. This data storage arrangement was a slight compromise as the preferred arrangement was to have each thing stored as an individual dataset, which offers more control over individual resource access. However, at the time of development, this approach led to exceedingly long waiting times while each dataset was fetched individually.

After a user has connected data to their Solid pod, it then needs to be converted to a media profile and displayed. In the demonstrator, the conversion involves creating a local "MediaItem" object for each thing found on the user's pod. The MediaItem type is generic and is used to represent all Things, regardless of type. Corresponding actions are then identified for each thing, and the timestamp from the action is added to the MediaItem object, resulting in an array of MediaItems, each with a title, description, thumbnail, and timestamp. The use of a generic type meant that all items could be easily displayed on a common timeline, allowing the user to see their combined media consumption history. A calendar view is included which allows the user to choose a day and see which BBC/Netflix/Spotify items they watched or listened to on that day.

Although storing resources in a small number of large datasets was suitable for the demonstrator, storing them individually would improve interoperability of the data. As well as storing them as individual datasets, these datasets should be stored in a flat hierarchy, with one container for each type of resource, and no further nesting. This makes discovery simple as there is only one location for each type of resource. We could then explore ways to provide virtual grouping and hierarchy of these resources, while allowing the actual storage hierarchy to remain flat. This data is then processed to generate a media profile, which can be viewed and configured by the user and made available to selected services. UI controls exist within the demonstrator to allow users to include or exclude music artists from their media profile, impacting the recommendations received.

Media Recommendations

In addition to seeing a combined media timeline, the demonstrator also offers a page that replicates the BBC Sounds audio player and uses the media profile to provide alternative recommendations. It does this by searching against BBC News, music, podcasts and programme archive to allow users to find content related to their favourite artists.

This is done by taking the names of the user's favourite artists (as determined by Spotify) and searching for BBC content tagged with that artist's name. If matching BBC content is found, this is then presented back to the user as a recommendation. The recommendation can be viewed or listened to within the demonstrator; "app switching" is mocked to represent the user being taken to their phone's BBC Sounds app. Furthermore, we used the resulting media profile to recommend events or museum exhibits that relate to the user's favourite musicians or programmes, showing how the use of a standardised media profile plus integration of open third-party data can further enhance the user experience.

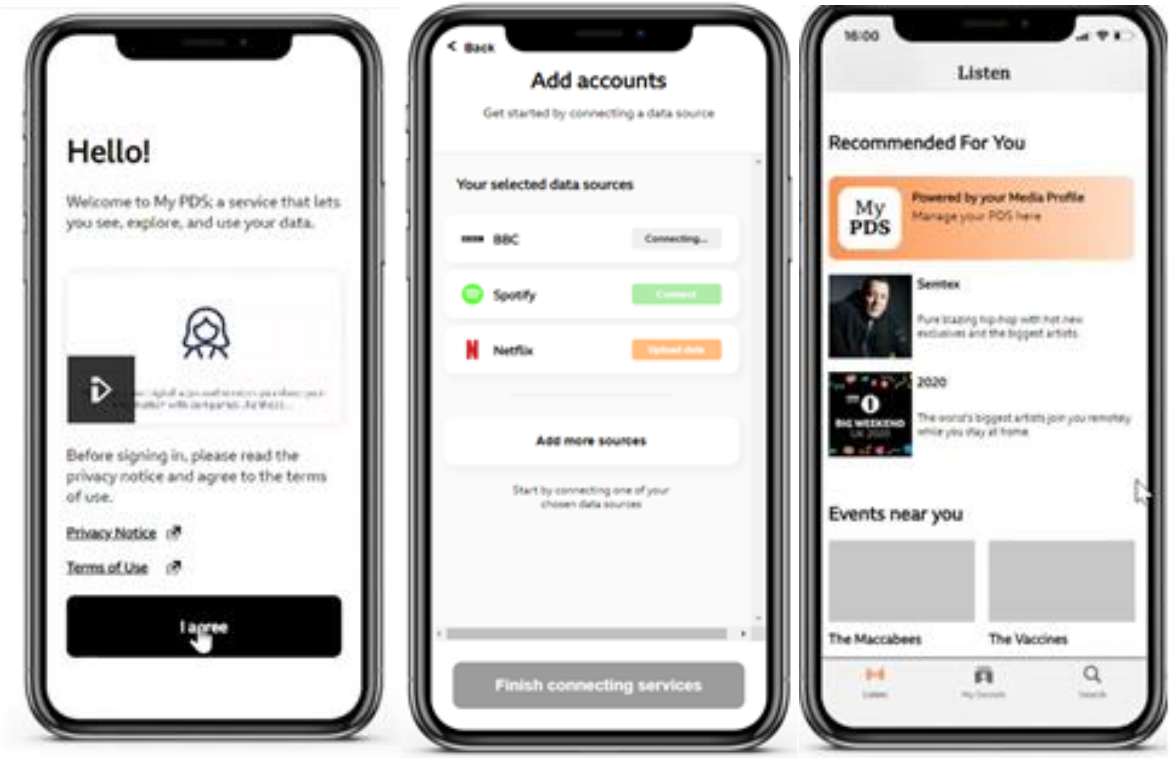

Figure 2 – Demonstrator Screenshots

This is a simple but effective demonstration of a secure cross-service application. At no time does the BBC get to see the user's Spotify data, and Spotify does not receive a copy of the user's BBC data, as all data processing is done in the demonstrator app and all data is stored in the pod. As well as keeping data safe, it supports data portability as user data is always under the user's control.

Although "recommendations" is part of the title for this paper and presents the core use-case of the demonstrator, we do not make any claims about the quality of the recommendations. Our 'recommender' was a very simple one, but we have evidence that media assets that would not have been discovered by analysis of the viewing/listening history of only one of the three services were proposed through use of our cross-service profile. This supports our hypothesis that given more knowledge about a user's listening and viewing habits from other services enables media providers to extract more relevant content from their catalogue and archive.

Feedback from internal testing with BBC staff was positive and formative research probing user's perception and appetite for the prototype of a PDS enabled Cross Media Profiler found it to be well received (16). The service was seen as innovative, engaging and useful - largely in part due to the PDS as a means to improve transparency and control. Users preferred the media service when supported by a PDS. A live closed trial has just concluded, and results are forthcoming.

KEY COMPONENTS

Identity

In the Solid ecosystem, a user is identified by their WebID, a unique identifier which is generated for the user upon signing up to the service and follows an existing

specification[5], providing universal identification across many services. Solid has extended the OpenID Connect (OIDC) authentication protocol[6] which allow users to log into their pod with an existing OIDC service, matching their WebID with the session. As such we have integrated our BBC Accounts OIDC system into our demonstrator and used an OpenID broker provided by Inrupt's ESS as the middle layer between a Pod and the authenticated OIDC session. This leverages existing live BBC infrastructure, allowing users to use their current BBC credentials for our demonstrator, facilitating the onboarding process.

Authorisation

Our recommender service is one of theoretically many applications that may interact with a user's pod. As each application will request access to different data, we need a consent screen. Permissions for an application are controlled via Access Control Policies[7]. We make it clear that our service will need specific consent to proceed with the option of limiting the duration of that consent. Figure 3 is an example of our application requesting user authorisation to access their pod on the ESS, improving the way we communicate this to our users is an area for future improvement for both our application and ESS itself. Authorisation in the Solid ecosystem covers authorisation for users to edit data on their pod, and permissions for applications to use data on a user's pod. The first part is covered by the Solid OIDC protocol, built on top vanilla OIDC and OAuth2[8].

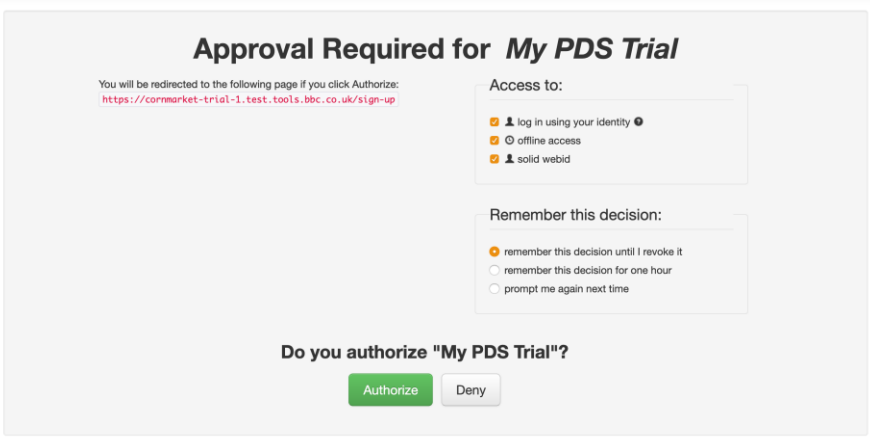

Figure 3 – Solid Authorisation Screen

Authorisation and general consent flow raised challenges around designing for informed consent and more dynamic permission models, key for this project was working closely with UX and our operational privacy team to ensure an easy-to-understand language was used clearly articulating what is happening to a user's data and how this is different to existing approaches. For further detail see E. Sharp et al. (6)

Semantic Data Harmonisation

A key aspect of the data ecosystem is to enable interoperability and collaboration in a secure way across various data sources and data domains (healthcare, finance, media).

There are several technologies and standards that enable the extraction, transformation and loading of different data sources into a consistent representation.

---

[5] https://www.w3.org/2005/Incubator/webid/spec/identity/
[6] https://solid.github.io/authentication-panel/solid-oidc/
[7] https://github.com/solid/authorization-panel/blob/main/proposals/acp/index.md
[8] https://tools.ietf.org/html/rfc6749

In our demonstrator, we use Linked data, and Shapes to describe a user's media data. A *shape* defines the **fields and structure** that client and apps can expect to find in a view over a piece of data (17).

We wanted to use shapes to express media items and consumption patterns in a generic way. This would enable us to pull in media data from a variety of sources and store it on a user's Pod in a common, interoperable format. We therefore defined several shapes named: **MediaThing**, **MediaAction** and **MediaProfile**. Our shapes are written in SHEX format. We have defined them using vocabularies from places such as schema.org. Our shapes can be seen at https://shapes.bboxservices.net/.

Our basic building block is a **MediaThing**. This allows us to describe an 'entity' in its most basic form. This could be a TV show, a music group, or a podcast. Data from all three services was sufficient to create a MediaThing for each entity, with a type assigned to it. BBC content was marked as either 'tv episode' or 'podcast episode' depending on whether the BBC API identified it as TV or radio content. Spotify tracks and artists were marked as 'music recordings' and 'music groups' respectively. Netflix data, after enrichment, was marked as 'movie' or 'tv episode' according to the metadata received from TMDb[9].

We also created shapes that store a user's consumption of these MediaThings as **MediaActions**. In our demonstrator, an action is an interaction with a MediaThing – usually a watch or a listen. An identifier property allows us to point to the associated MediaThing. With this arrangement, it is possible to have multiple actions associated with the same thing, if a user watches a film on multiple occasions for example. All three services provide accurate timestamps allowing us to easily create actions.

This enabled us to build a **MediaProfile** for each of our users including the things they consumed and when. This profile can be edited by the user in the interface of the application. In our demonstrator, lists of MediaThings are initially populated directly from the data imported from Spotify, BBC and Netflix. The user is then able to "exclude" entities from any list, thus removing them from their media profile. Through this mechanism, it is hoped that a richer recommendation experience could be developed, based only on the media items a user wishes to influence their profile.

Populating Our Shapes

By collecting user data from the BBC, Netflix and Spotify, we were able to populate our shapes. The process is illustrated in Figure 4.

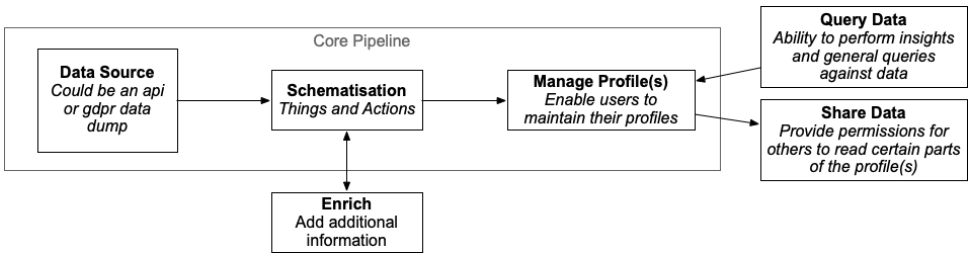

Figure 4 – Basic data pipeline

To gather a user's BBC viewing data, we used the BBC's User Activity Service (UAS) API, which provides details of a user's activity across BBC products. We requested a user's played items from BBC iPlayer and BBC

[9] https://www.themoviedb.org/

Sounds. Each item returned contained a timestamp of when it was played and a “pid” (programme id), which could be used to fetch additional metadata (title and description) from a separate BBC API.

```
[{resourceId: 't1234567', resourceDomain: 'tv',  programme: { name: 'Test Card F', description: 'The standard BBC Test Card', image: 'http://www.bbc.co.uk/staticarchive/e5392e3bfca450291543eb1e45a4009db5bfc6a9.jpg’}]
```

We were also able to use the Spotify API to request a user’s top Spotify artists, tracks and most recently played tracks. In our recommender application, we used the artist's name returned by Spotify to search against the BBC’s backend systems (combining MusicBrainz and the BBC’s own artist identifier). If an artist was identified in the BBC system, BBC content relating to that artist would be surfaced to the user.

```
[{"endTime" : "2019-03-20 09:25","artistName" : "The Prodigy","trackName" : "Their Law","msPlayed" : 66873}]';
```

Netflix do not provide a web API that would enable us to support the app directly. Hence, we had to ask users to perform a subject access request, which under GDPR allows users to request a copy of their personal information from Netflix, which they then uploaded to their pod separately. This provides a copy of the account viewing data, which contains the profile name, start time, title, and supplemental video type for each play action on the platform. We added a data enrichment process which matched the title string to a corresponding film, series, or episode on TMDb. Using their public API, we were also able to retrieve associated synopses and images for the media item.

```
Profile Name,Start Time,Duration,Attributes,Title,Supplemental Video Type,Device Type,Bookmark,Latest Bookmark,Country\r\nCarl,2015-02-09 19:37:07,00:20:15,,Archer: Season 1: The Rock (Episode 8),,Chrome PC (Cadmium),00:20:27,00:20:27,GB (United Kingdom)';
```

Although our work has shown that it is possible to populate a media related profile with data from multiple sources, we believe that without standardisation on common data formats scaling this approach would become un-maintainable for one organisation alone as each new data source will require its own importer and associated maintenance cost.

Security

Use of the system moves all the user’s data from disparate services into a single place, which significantly increases our security risks as we now have a specific location containing personal user data which needs to be protected. Throughout the project we employed the “Secure by Design” principle (18); making sure that our system is designed with security in mind from the start. We had regular meetings with InfoSec Architects which allowed us to use their knowledge to make sure we did not miss any security issues. We also created and maintained accurate architecture diagrams using the C4 methodology[10] to get an accurate overview of our system which we could then perform STRIDE threat modelling on. These together resulted in many security issues being fixed with the 3 major changes to the system detailed below.

Firstly, we protected all frontend UIs using our internal staff login system. This resulted in only approved trial members being able to access the system reducing our potential attack

[10] https://c4model.com/

surface. Secondly, we implemented and monitored audit logging on the system for user and developer actions, allowing us to monitor who is accessing user's data. Finally, we leveraged the BBC's Cosmos system and AWS's ECR[11] scanning system to automatically scan our deployments for vulnerable dependencies allowing us to quickly fix critical issues In our codebase, we also used custom ESLint[12] rulesets with a focus on security.

By applying the "Secure by Design" principle we were able to fix most security issues during development and reduced our security risk categorisation significantly ensuring a smooth information security approval stage.

DEPLOYMENT AND INFRASTRUCTURE

Infrastructure

For the project we have been primarily using AWS for our infrastructure. Our recommender application is hosted on an EC2 instance[13] and microservices such as our connectors (which facilitate the fetching and transformation of the data between third party services and our applications) are hosted on AWS Elastic Container Service (ECS)[14]

Our Solid server, the ESS is hosted on top of a Kubernetes cluster. Initially on an EC2 instance in a development environment, we are in the process of migrating it to a more robust production environment, on our onsite hardware using OpenStack[15].

Deployment

Service releases and deployments were handled using our internal BBC Cosmos, a managed service for creating repeatable deployments using 'baked' server images. A server image is a snapshot of our application alongside an up-to-date version of CentOS and any software dependencies we require. These images can then be deployed to Test or Live environments. Cosmos also notifies the team if the latest image has a security vulnerability and requires a redeployment for security updates.

Furthermore, the images also include automated management of a Public Key Infrastructure[16], assigning a BBC-CA-signed certificate to each service it deploys and the appropriate BBC Certificate Authority chains to authenticate other BBC services and staff - this means that services can easily be limited to BBC staff and can be further tailored to a subset BBC staff, such as BBC developers or other internal services.

We also created a monorepo for our project using Lerna[17]. This allowed us to easily create a common base set of packages and configurations that would be available to all microservices, such as applying the aforementioned linting ruleset to the entire project codebase and provides a central location for storing our Architectural Decision Records[18].

---

[11] https://aws.amazon.com/ecr/
[12] https://eslint.org/
[13] https://aws.amazon.com/ec2/
[14] https://aws.amazon.com/ecs/
[15] https://www.openstack.org/
[16] https://en.wikipedia.org/wiki/Public_key_infrastructure
[17] https://lerna.js.org/
[18] https://adr.github.io/

FURTHER RESEARCH

Interoperability

A central element of the Solid ecosystem is decoupling the data from the applications that use it and making it fully interoperable across any application. The My PDS and recommender proposition demonstrate this first part by showcasing an application that uses media data stored on a user's pod to drive a recommender service. The next step would be to demonstrate how that data can be used across several different applications. To do that, we must ensure that our data is in a format which allows machine-to-machine interoperability. By storing data in RDF format and shapes, we are part-way there. The last step is using Shape Trees. Shape Trees express how different resources are inter-connected, where each resource is associated with a shape. It allows the semantics of the data to be expressed at all levels of complexity, which enables applications to fully understand a user's data and build on that. Adopting this model internally within the BBC would enable us to create a common schema for all the different content that the BBC produces. Although we currently have systems in place to provide representations of our content and user data, this would allow us to have common standards for how applications exchange and interact with data.

This shared semantic of the data also presents the opportunity for a granular authorization process, where applications must make explicit requests for the data they require from a pod. Intuitive UIs can be designed and presented to users, who in turn can make informed decisions on the access they grant to the application, at the granularity that they want.

Verifiable Credentials

Storing user information on pods also opens the door for new types of sensitive digital data to be used, such as credentials. Credentials in the physical world are well established but it is difficult to express machine-readable and verifiable information on the web. As more traditional services and transactions become digital, using credentials on the web to verify the authenticity of a piece of data become crucial. A recent W3C standard has been put forward to address this issue, called Verifiable Credentials (VC). It provides a standard way to express credentials on the web that is cryptographically secure, respects privacy, and is machine verifiable. The data model of VC sees a data authority, issue a credential about a subject in the form of a claim, and give it to a holder. The holder is responsible for storing and managing the credential. This model marries very well with the Solid ecosystem where a user's holder could be their Pod. As a simple example for our recommender, we could require a user's verifiable date of birth ensure the content surfaced to that user is within the appropriate age rating.

There are further aspects, such as verifiable compute which addresses the issue of not being in control how data will be used once it has left the user's pod. Projects such as Google Oak (19) have already started investigating the use of Trusted Execution Environments, which ensures the data is processed in a secure and verifiable manner.

OPPORTUNITIES FOR PUBLIC SERVICE MEDIA

Audience data for media organisations tends to be stored centrally in large databases, which necessitates strict security measures, and mechanisms to comply with subject access requests. Storing audience data in a per user personal data store enables a new paradigm for storing, processing, and managing audience data. Particularly interesting is

the opportunity to provide a GDPR compliant mechanism as a view and access to a user's pod is the equivalent to a subject access request.

This enabled us to rethink what a BBC privacy promise would look like in this new environment. The privacy promise deployed as part of the trial covered a renewed commitment to transparency, choice and control, as enabled by using Solid pods. For transparency, we can be even more declarative about the data we collect and how it is being used. Applications need to fully state the data they need to access, and this access can be revoked at any time. Choice is achieved by giving users granular controls over what application has access to what data. Users could also choose to move their data to different pod providers at any point. We believe this will increase the trust our audiences have in our approach to personal data handling.

CONCLUSION

In an effort to inform the BBC's position within the new data landscape, we investigated PDS technology as a novel alternative approach to data stewardship, exploring use cases which combine user data from multiple sources, and the balance between data protection and technical implementation. We wanted to understand this technology landscape and investigate how it might enable us deliver to our public purpose in the future.

Our demonstrator allowed us to explore the impact, and readiness of this technology for our users, the BBC, and other media services. The work provided a vehicle to understand the benefits, challenges, risks of this approach to both the user and organisation when providing greater control to the user and providing richer recommendations.

We need to define a set of common standards and protocols to enable this open ecosystem to work and scale to support transparency, interoperability, and choice for all participants. The hope is that organisations can build together common data formats, such as the Media Profile, which can be used by all services (e.g., BBC and other Public Service Broadcasters) as these all move towards a PDS model.

The Solid specification is still in active development and consists of several sub-specifications (such as the Solid protocol on how to read and write resource in a pod, Solid OIDC to authenticate within a pod, or interoperability in the Solid Ecosystem). Most of these are still in draft version and are waiting to be ratified by the W3C. This means that there could still be a lot of change before a common set of standards emerge.

Adopting this approach will be a shift for organisations whose business models thrive on personal data gathering. But the potential offered by the Solid ecosystem for users and organisations alike justify further investment in our research and will drive adoption. Experiments with this technology in government (20) and healthcare (21) sectors also suggest this approach could become a regulatory requirement, and the PDS could play an important part in the future technology landscape.

REFERENCES

1. General Data Protection and Regulation. https://gdpr-info.eu
2. California Legislative Information. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375
3. Solid Technical Reports. https://solid.github.io/specification/
4. Inrupt. https://inrupt.com/

5. Engineering and Physical Sciences Research Council. https://gow.epsrc.ukri.org/NGBOViewGrant.aspx?GrantRef=EP/N028260/1
6. Thompson et al. Enhancing media through the development of a public service data ecosystem. IBC 2021 (forthcoming)
7. W3C, Linked Data. https://www.w3.org/standards/semanticweb/data
8. Mortier, R, Haddadi, H, Henderson, T, McAuley, D & Crowcroft, J 2014 'Human-Data Interaction: The Human Face of the Data-Driven Society'. https://doi.org/10.2139/ssrn.2508051
9. BBC R&D Blog. https://www.bbc.co.uk/rd/projects/human-data-interaction
10. N Sailaja. CHI EA '21: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems May 2021 Article No.: 271 Pages 1–7 https://doi.org/10.1145/3411763.3451808
11. Thompson and Jones, BBC R&D Blog Post Introducing the BBC Box. https://www.bbc.co.uk/rd/blog/2019-06-bbc-box-personal-data-privacy
12. Solid, Node Solid Server. https://solidproject.org/self-hosting/nss
13. Inrupt, Enterprise Solid Server. https://inrupt.com/products/enterprise-solid-server/
14. European Comission. https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/digital-services-act-ensuring-safe-and-accountable-online-environment_en
15. Inrupt, JS Libraries. https://docs.inrupt.com/developer-tools/javascript/client-libraries/
16. N. Sailja et al. Human Data Interaction in Data driven media experiences: An exploration of data sensitive responses to the socio-technical challenges of personal data
17. R. Verborgh. https://ruben.verborgh.org/blog/2019/06/17/shaping-linked-data-apps/
18. Secure by Design. https://en.wikipedia.org/wiki/Secure_by_design
19. Google Oak. https://github.com/project-oak/oak
20. T. Berners-Lee Inrupt/Flanders. https://inrupt.com/flanders-solid
21. Greater Manchester Authority/NHS/JaneiroDigitial. https://www.srft.nhs.uk/media-centre/latest-news/news-archive/news-2020/gm-digital-platform/