Secure Web Objects: A Data-Oriented Paradigm

Tianyuan Yu
UCLA
USA

Xinyu Ma
UCLA
USA

Varun Patil
UCLA
USA

Yekta Kocaogullar
UCLA
USA

Yulong Zhang
HKUST(GZ)
China

Jeff Burke
UCLA REMAP
USA

Dirk Kutscher
HKUST(GZ)
China

Lixia Zhang
UCLA
USA

ABSTRACT

This paper explores how to support the Web's evolution through an underlying data-centric approach that better matches the data-orientedness of modern and emerging applications. We revisit the original vision of the Web as a hypermedia system that supports document composability and application interoperability via name-based data access. We propose the use of *secure web objects* (SWO), a data-oriented communication approach that can reduce complexity, centrality, and inefficiency, particularly for collaborative and local-first applications such as the Metaverse and collaborative editors. SWO are named, signed, application-defined objects that are secured independently of their communications channels, an approach that leverages the results from over a decade-long data--centric networking research. The approach does not require intermediation by aggregators of identity, storage, and other services that is common today. We present a brief design overview, illustrated through prototypes for two editors of shared hypermedia documents: one for 3D and one for LaTeX. We discuss our findings and suggest a roadmap for future research.

CCS CONCEPTS

• **Information systems → Web applications**; **Internet communications tools**; • **Networks → Programming interfaces**; **Network design principles**.

KEYWORDS

Web, Metaverse, Data-Orientation, Information-Centric Networking, Local-First Software, Decentralization

1 INTRODUCTION

The Web was conceived as a hypermedia information system enabling the publishing of Web objects and access to them by name (URI). Many Web applications today remain intrinsically data-oriented. For example, multimedia document editors work with what can be viewed as catalogs of media objects of different types. Video streaming can be seen as a system of continuous production and consumption of media objects. Contemporary AI applications revolve around a variety of data - training, models, and vector storage, for example. Emerging Metaverse systems work with interwoven collections of 3D scenes that organize media objects, which can be created, consumed, modified, and re-organized in new contexts.

What these systems have in common are concepts of data objects accessed via namespaces often expressed as URIs. However, the current Web stack effectively supports only a limited approximation of the original hypermedia concepts. Data-centric behavior is supported at the application layer and within applications, but not at lower layers or as a means for interoperability between applications. There, a client-server communication approach is dominant; replacing single servers with centrally secured cloud resources has actually helped cement silos around data object collections. The dependence on channel-oriented security, built on TLS, results in a lack of simple mechanisms to efficiently access shared data from peers or across domains, especially in local-first communication scenarios. Client-server connection-based communication patterns do not correspond well to the data-oriented nature of applications they support: hence the popularity of a variety of messaging and middleware solutions that implement sit atop Web protocols and within cloud silos. These solutions, while data-centric for applications, typically rely on centralized and online services for authentication, transport, and other key services.

While such contemporary approaches have been successful in turning the Web into an application platform, they have moved from the original vision of an interoperable hypermedia information system, even as domains themselves become more data-centric. They also struggled to support intermittent connectivity scenarios and local-first communication. For example, today, most collaborative editing occurs entirely within siloed, online Web applications rather than through multiple editors that can operate on the same hypermedia data in either internet- or locally-connected scenarios. Radoff describes **interoperability** on different layers (connectivity, persistence, presentation, meaning, and behavior)[1] as a major challenge for future Metaverse systems, and discusses the importance of **composability**[2], alluding to composable computing and distributed computing as an example. ARENA [6] is an example of a **modular system design** for multi-user and multi-application setups, enabling the development and hosting of collaborative XR experiences on WebXR-capable browsers with transparency, allowing data to migrate seamlessly across computing resources.

This paper begins to re-imagine the Web (or, really, revisit its original promise) as a data-oriented hypermedia system that enables *data interoperability* and service composability within and among applications without requiring intermediation by central server platforms for user authentication and message relaying. [3] In Section 2, we further discuss the opportunities in more detail, which we illustrate by a description of two prototype designs in Section 3. We summarize insights gained from prototype development in Section 4 and articulate a suggested research agenda in Section 5.

[1] https://medium.com/building-the-metaverse/metaverse-interoperability-part-1-challenges-716455ca439e

[2] https://medium.com/building-the-metaverse/composability-is-the-most-powerful-creative-force-in-the-universe-e82e3dd83ccd

[3] This is distinct from our understanding of *Solid* [7], for example, which decentralizes control over data but does not change the underlying communication paradigm.)

2 A NEW PATH

> *Information systems start small and grow. They also start isolated and then merge. A new system must allow existing systems to be linked together without requiring any central control or coordination.* – Tim Berners-Lee[4]

Our primary aim is simple: we seek to enable Web applications to create and exchange Web objects, accessed by URI-like names, without relying on the communication channel for security. By decoupling secure data exchange from channels, we can (re-)enable an array of different interaction styles, from individual Web object access across applications to scalable multi-destination distribution.

By *Web objects*, we mean named data objects that have semantic meaning within an application domain, composed from lower-level application data units (ADUs) [1] that can be individually accessed by their names. Examples include file-like objects as well as objects such as groups-of-pictures (GOPs) in streaming video, and more granular data. We propose that a data-oriented security model, in which each ADU is cryptographically secured (e.g., signed and/or encrypted), yields **secure Web objects** (SWO). SWO can support important emergent features at the Web layer as cryptographic operations on the objects themselves, rather than tied to the channel they are carried over, including authentication, authorization, binding to real-world identities, and data encryption. Access to secured Web objects becomes location-agnostic: data can be stored, replicated, and accessed just by name. By securing data directly, we reduce security dependency on client-server connections, enabling Web applications to communicate securely in local, decentralized contexts. SWO can be carried over any transport: HTTP, QUIC, UDP, Bluetooth, etc., with or without transport security.

Exchanging SWO *by names* requires name-based rendezvous (e.g., leveraging DNS to find rendezvous servers) or transport connectivity between all parties that has already been established. Then, this communication approach unifies single-destination and multi-destination delivery and could dramatically simplify both server scalability and distribution infrastructure. E.g., server offloading through caching becomes an intrinsic feature, as SWO can be easily replicated in managed ways or opportunistically cached by network elements and/or other endpoints. Because their security properties do not emerge from the channel that carries them, SWO can be used directly in decentralized applications over potentially insecure transport as well as over more “traditional” means. This approach enables expanding the practical possibilities of the Web:

- application interoperability can emerge directly from common data and naming formats of SWO, with cryptographic key exchange used to provide access to data.
- moving from browser interfaces that replicate server silos (e.g., the tab) to layered, possibly 3D-structured UIs more suitable to concepts like the Metaverse;
- enabling such cross-origin access to elements from multiple “applications”, while tracking identity and provenance as described in [3]; and
- on-demand data transformation in the network, such as rendering an object containing a 3D scene description into 2D tile objects and data compression/de-compression functions.

3 PROTOTYPING

Using SWO as the basic building block, we have applied the data-oriented approach articulated above to develop two prototype web apps. We leverage *Named Data Networking* (NDN [11]) to support name-based rendezvous for exchanging named SWO. In an NDN setting, communications are carried out by exchanging named, secured data objects agnostically to object locations. We have developed a set of “local-first” web editors using NDN libraries in TypeScript [2]. These editors operate on shared hypermedia documents expressed in SWO, which are used to build serverless apps.

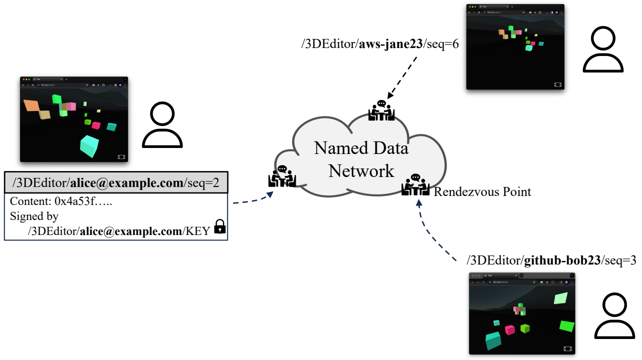
/3DEditor/**aws-jane23**/seq=6
Named Data Network
Rendezvous Point
/3DEditor/**alice@example.com**/seq=2
Content: 0x4a53f…..
Signed by
/3DEditor/**alice@example.com**/KEY
/3DEditor/**github-bob23**/seq=3

Figure 1: A collaborative 3D editing scenario among Alice, Bob and Jane, who have obtained their identifiers by adding the application prefix “/3DEditor” in front of their own identifiers.

The first prototype we developed, the *Microverse Editor*, enables collaborative editing of 3D scenes expressed in SWO.[5] To jointly edit 3D scenes, *Microverse Editor* enables users to directly exchange SWO with each other in the same web of trust. To join an existing web of trust, a user uses his/her existing Internet identifier, such as an email address, cloud username, or a DNS name if one has one, as his/her identifier and produces a self-signed certificate. Users can then mutually authenticate each other through direct or indirect trust relations, such as QR code exchanges or transitive trust relations. Our prototype assumed that, within a collaboration group, every user can find direct or indirect trust relations to some other users, therefore even in the worst case where only out-of-band peer authentications are possible, one can have users in the same group authenticated through the group’s web of trust [10].

As Figure 1 shows, scene updates are carried in SWO with URI-like names, and each user secures both asset and update objects by signing the object using his/her certificate, with the certificate itself a SWO. With object security decoupled from channel security, users only need a rendezvous place to exchange semantically named and secured objects. Given broadcast communication and discovery are prohibited for security concerns in today’s browsers, we run NDN over a simple, schema-agnostic Websocket server as an NDN relay among users.

Although our prototype *Microverse Editor* has limited editing functions, its use of SWO model enables several attractive features:

[4] https://www.w3.org/History/1989/proposal.html

[5] see https://named-data.net/microverse/ for a more detailed description of the current project.

the scene and object descriptions are expressed as individual SWO and collections of SWO. A collection can be seen as a manifest that refers to individually names SWO, including other collections. Aggregation and re-use can be easily achieved by creating such collections. Since the SWO are uniquely named, copying an existing complex 3D scene can be efficient. Mapping SWO to NDN network delivery would enable additional benefits: requests for SWO can be broadcast in a local network and obtained from the closest peer without name translation and re-encoding.

A second prototype we developed is a shared LaTeX editor called *Workspace*[6]. In developing *Workspace*, we found that it shares the same set of needs as *Microverse Editor*: assigning each user an identifier based on his/her unique existing Internet identifier, facilitating user mutual authentication, securing named data directly, and enabling exchanging named SWO through a rendezvous place. When a group of people jointly work on the same paper, *Workspace* can keep all the traffic local if a rendezvous point is nearby. One feature of Workspace that the *Microverse Editor* does not have (yet) is the capability to merge changes made by different users. As multiple users may edit at the same time, Workspace must keep consistent ordering of all the changed seen by all the users. To address this problem, Workspace represents the files in a shared LaTeX editor by a collection of Conflict-Free Replicated Data Types (CRDT) [8] data structures, which ensures the consistent user views by achieving eventual consistency on shared data structures. Workspace also allows users to work offline. Once offline users come online, they exchange SWO with the others to have the edits merged with the rest of the group.

4 NEW INSIGHTS

The vision of a data-oriented Web is based on data immutability as a fundamental, universal concept in facilitating the Web as a distributed hypermedia information system. We argue that aligning hypermedia object models and access methods as well as the underlying network transport better with the inherent data-oriented application structure has significant advantages with respect to ease of software development (both within and outside of the "app" paradigm), scalable sharing and multi-destination distribution, decentralized and asynchronous communication, and data re-use.

While Web servers will continue to play important roles in applications, we have shown that secure Web objects can support other communication models with additional advantages, for example, decentralized editing of shared documents without requiring a central server. Our approach makes data objects first-class citizens of the Web again, and has security attached to data instead of data containers and communication channels. The use of semantic identifiers helps develop decentralized security solutions based on trust relations between users, instead of being determined by 3rd party (the current WebPKI security model).

5 RESEARCH AGENDA

Our prototype development has demonstrated the potential of the SWO approach. Based on related ongoing research in named data networking [11], we suggest the following research agenda for continuing this work and creating a general SWO Web platform.

5.1 Naming

Enabling distributed data-oriented applications requires **namespace design strategies** for application domains, i.e., design strategies for developing name hierarchies that best express an application domain's semantics and support its security needs. When considering interoperability between applications enabled by shared data access, the namespace design challenge becomes even more relevant: for example, it will be important to **map common file and interchange formats intro named data** in ways that promote cross-application compatibility and increase granularity of access and security. In general, more work is required to elaborate the **analogies and technical mappings that are possible between Web semantics and named data**.

5.2 Security

Using SWO also requires easy-to-use data-centric security in which trust relationships can be described using names [5] and bootstrapped in a variety of ways. In the prototypes described in section 3, we have developed a pragmatic approach for **security bootstrapping** in local networks or closed user groups, i.e., obtaining cloud-independent identifiers and security credentials. This is certainly an area that needs more attention, including initial deployment approaches for starting the development and deployment of SWO apps in today's Internet and Web environment, as described in [9]. Another interesting area will be the development of **group security solutions** for data confidentiality, authenticity and access control under an SWO framework. In some performance-critical applications, universal public-key-based security (for signing and encrypting objects) is not the first choice, and technologies such as broadcast security could be leveraged. Additional requirements for supporting intra- and inter application security (e.g., securely identifying creators of artifacts across application boundaries and securely controlling access to certain objects) should be considered.

5.3 Interactions

With the foundation of channel-agnostic exchange of named, secured objects, we can **rethink application layer communication semantics and interaction styles**. In today's Web, the client-server model has led to representational state transfer (REST) as a universal vehicle for any form of data exchange, leading to unnecessary complications for applications that do not require common state evolution on a client and a server. In a fundamentally data-oriented Web, simple publishing and access semantics can be realized consistently across a variety of situations. Knowing the data naming and security approach is sufficient for publishing data or synchronizing data sets among inter-operating applications. APIs for publishing and access also could be agnostic to the number of peers because data-oriented communication has no concept of individual connections or peers. In NDN, there are implementations of so-called distributed dataset synchronization schemes that can be perceived as a multi-party transport layer for SWO communication. Client-server applications with robust state evolution can still be achieved through data-oriented REST approaches such as the one described in [4].

With semantically meaningful identifiers, interoperability can be facilitated and security properties and access control rules can

[6] see https://github.com/UCLA-IRL/ndn-workspace-solid

be linked to application-defined names and name prefixes. Further work is required to **explore different options for naming and data re-use**. For example, data re-use should be enabled across application boundaries which suggests a level of indirection in naming. Such approaches would benefit from collection concepts for file-like *and* dynamic objects.

5.4 Application Frameworks

Based on such abstractions, we can also explore how **serialization to secure Web objects can be integrated into file systems, databases, and other storage mechanisms**, and how the interaction between server-side logic and in-browser based counterparts can benefit from such features. For example, reactive programming concepts (as used by libraries such as React[7] have led to the development of application platforms such as Next.js[8] and Remix[9] which enable end-to-end application development based on a reactive (partly functional programming) paradigm. Newer systems such as Electric Clojure[10] go a step further and fully abstract over client/server state synchronization at the programming language layer, to achieve strong composition across the frontend/backend boundary in dynamic Web apps. We believe that such systems can benefit from an SWO-based abstraction and native communication layer. This could be especially useful for mobile application development platforms such as Firebase[11] which can benefit from the above-mentioned efficiency gains by mapping SWO directly to named data on the network.

Additionally, computation on data ("in-network computing") becomes easier to integrate, for example by a named-function design where functions create new (location-independent) secure Web objects following the same principles. Decentralized communication and processing are easily available to any Web application.

5.5 Communication

We want to **enable direct user-to-user SWO exchange on top of today's available communication underlays**, i.e., Internet protocols. While we can obviously tunnel SWO communication over existing QUIC or TCP and rely on relay servers, better performance and scalability properties can be achieved by direct local SWO exchanges as in our prototypes. More work is needed on bootstrapping such communication. Moreover, distributing SWO as network layer data objects would enable the use of wireless broadcast/multicast, which is currently not achievable for Web communication but could provide significant scalability benefits. More work is needed to enable this in real-world networks.

6 CONCLUSION

Variations of this concept of named, secured data objects already exist within individual applications, especially on those that cannot rely solely on channel security. We propose that adopting consistent naming, security, and access mechanisms to create an interoperable, channel security-independent approach for secure exchange of data opens up important new possibilities.

The SWO approach should not be confused with approaches that are sometimes referred to as *web3*[12], such as IPFS. These systems, while claiming to provide a platform for the next Web, are merely providing a (conceptually) decentralized storage platform for accessing file-like objects, which 1) is not a useful service model for any future Web, and 2) does not enable the data-oriented benefits as described here. While the SWO approach can in principle leverage existing transports as underlays, it is important to note that a direct mapping to named-data oriented communication would provide significant benefits, such as avoiding combinatorial interoperability problems and enabling efficient, local, and decentralized communication.

SWO-based systems require name-based rendezvous services. In traditional systems, this is typically a central place (server) that can recognize SWO names. In an NDN system, names are used in the network, where rendezvous functions can be realized directly.

Hence, in this paper, we present an early concept for secure Web objects that leverages research in data-centric networking to help evolve the Web into a more effective environment for local-first, decentralized, and interoperable applications that resurface the early promise and excitement of the Web.

REFERENCES

[1] David D Clark and David L Tennenhouse. 1990. Architectural considerations for a new generation of protocols. *ACM SIGCOMM Computer Communication Review* 20, 4 (1990), 200–208.

[2] NDNts contributors. 2024. https://github.com/yoursunny/NDNts. Accessed: 2024-2-3.

[3] Dirk Kutscher, Jeff Burke, Giuseppe Fioccola, and Paulo Mendes. 2023. Statement: The Metaverse as an Information-Centric Network. In *Proceedings of the 10th ACM Conference on Information-Centric Networking* (Reykjavik, Iceland) *(ACM ICN '23)*. Association for Computing Machinery, New York, NY, USA, 112–114. https://doi.org/10.1145/3623565.3623761

[4] Dirk Kutscher and David Oran. 2022. RESTful information-centric networking: statement. In *Proceedings of the 9th ACM Conference on Information-Centric Networking* (Osaka, Japan) *(ICN '22)*. Association for Computing Machinery, New York, NY, USA, 150–152. https://doi.org/10.1145/3517212.3558089

[5] Boubakr Nour, Hakima Khelifi, Rasheed Hussain, Spyridon Mastorakis, and Hassine Moungla. 2021. Access control mechanisms in named data networks: A comprehensive survey. *Acm computing Surveys (cSuR)* 54, 3 (2021), 1–35.

[6] Nuno Pereira, Anthony Rowe, Michael W Farb, Ivan Liang, Edward Lu, and Eric Riebling. 2021. Arena: The augmented reality edge networking architecture. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 479–488.

[7] Andrei Vlad Sambra, Essam Mansour, Sandro Hawke, Maged Zereba, Nicola Greco, Abdurrahman Ghanem, Dmitri Zagidulin, Ashraf Aboulnaga, and Tim Berners-Lee. 2016. Solid: a platform for decentralized social applications based on linked data. *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.* (2016).

[8] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. 2011. Conflict-free replicated data types. In *Stabilization, Safety, and Security of Distributed Systems: 13th International Symposium, SSS 2011, Grenoble, France, October 10-12, 2011. Proceedings 13*. Springer, 386–400.

[9] Tianyuan Yu, Xinyu Ma, Hongcheng Xie, Dirk Kutscher, and Lixia Zhang. 2023. Cornerstone: Automating Remote NDN Entity Bootstrapping. In *Proceedings of the 18th Asian Internet Engineering Conference* (Hanoi, Vietnam) *(AINTEC '23)*. Association for Computing Machinery, New York, NY, USA, 62–68. https://doi.org/10.1145/3630590.3630598

[10] Yingdi Yu, Alexander Afanasyev, Zhenkai Zhu, and Lixia Zhang. 2014. *An Endorsement-based Key Management System for Decentralized NDN Chat Application.* Technical Report NDN-0023, Revision 1. Named Data Networking.

[11] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named Data Networking. *ACM SIGCOMM Computer Communication Review (CCR)* 44, 3 (July 2014), 66–73.

[7] https://react.dev/

[8] https://nextjs.org/

[9] https://remix.run/

[10] https://github.com/hyperfiddle/electric

[11] https://firebase.google.com/

[12] https://www.cnbc.com/2022/11/04/web-inventor-tim-berners-lee-wants-us-to-ignore-web3.html