Browsing an Augmented Reality with Named Data Networking (Invited Paper)

Jeff Burke
Center for Research in Engineering, Media and Performance
UCLA School of Theater, Film and Television
Los Angeles, CA 90095
jburke@remap.ucla.edu

***Abstract*—Augmented and virtual reality (AR and VR) are entering the consumer market, and attracting substantial attention from content creators. However, delivering AR and VR experiences using existing IP networks presents significant challenges that tend to force creators into "stovepipe" solutions, in contrast with the openness of the early World Wide Web, which led to its widespread content revolution. Taking advantage of multiple network interfaces and providing resilience to intermittent connectivity in mobile scenarios are difficult, as is handling trust for experiences built up from the content of heterogeneous providers. Additionally, streaming AR and VR content, including video components, must be user-navigable across multiple dimensions. This paper explores opportunities for an *augmented reality web* using Named Data Networking (NDN), a proposed future Internet architecture in which the network forwards intrinsically secure data packets directly based on application-defined names. By providing web semantics at packet granularity, NDN enables the success of the web to be pursued for low-latency, high-granularity, and context-dependent media in AR. The paper outlines emerging media types that could be part of a new AR browsing experience, briefly introduces NDN, describes benefits the architecture should provide via an example browser design, and enumerates related open research challenges.**

I. INTRODUCTION

The Internet has become fundamental to delivering modern media; it is as defining to content today as celluloid was to film in the past. As physical and engineering possibilities and constraints of film, television, theater, sound recording and reproduction shaped media in the 20th century, limitations and choices made in all layers of the network stack will shape media in this decade and beyond.

This paper explores how to enable a second wave of content innovation, building on that of the World Wide Web, by creating new possibilities for media authorship and distribution through changes to the "thin waist" of the Internet currently occupied by IP. Specifically, it considers how NDN [1], part of the larger field of information-centric networking (ICN), can contribute to the creative landscape of storytelling using augmented reality (AR) as an integrative environment.

NDN's forwarding based on data names and data-centric security enable extremely granular content and flexible, potentially decentralized approaches to trust and access control. NDN supports content publishing both by major providers in now-traditional, cloud-based ways, as well as local and on-demand services that follow a multi-tier, edge/fog/cloud model. [2] But AR is inherently context-dependent, as it overlays content on the "real world" as seen by a particular viewer. NDN also enables viewer context to be named as first-class data on the network, appropriately for AR. The architecture's intrinsic multicast and caching support enable the context from a single viewer to be efficiently accessed by any number of potential content providers on the network, allowing viewing context to become as vital to networked media as content.

As a fundamental architecture proposed for the network layer, NDN also offers consistency with networking of other types of devices and data, potentially simplifying the integration of Internet of Things (IoT) data with AR content. Furthermore, it enables on-demand edge computing and acceleration resources to be used consistently with other sources of context—via names.

Our goal in this paper is to present how these potential benefits can be used together to enable next-generation streaming AR applications. We suggest they offer the potential to enable a "web" of AR content that can provoke a new cycle of experimentation and innovation.

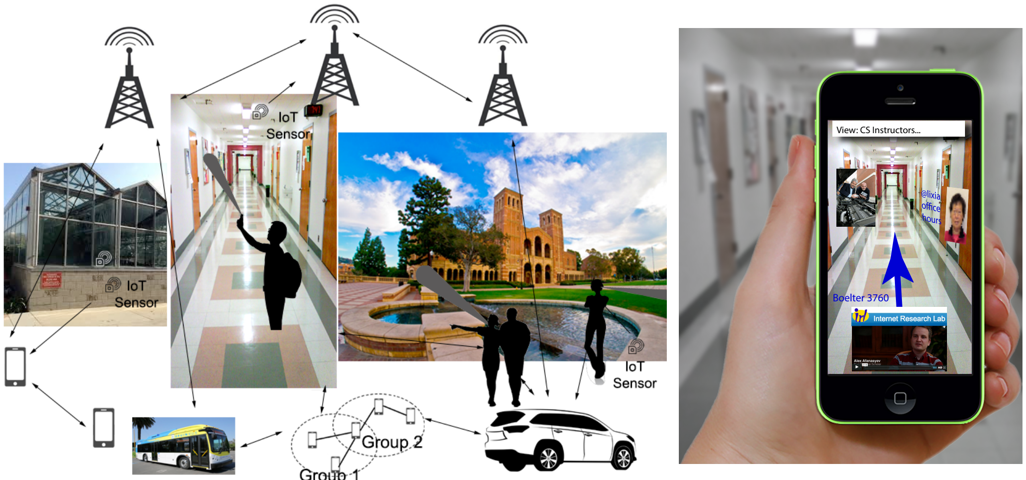
II. APPLICATION REQUIREMENTS

We define an AR application as one that delivers content that is 1) selected and navigated based on viewer context, and 2) intended to be overlaid on a live view of the real world.[1] The network thus must enable low-latency, granular exchange of context and content among many sources. While most current concepts of AR involve a client-server relationship between viewing devices and content sources, we do not make this assumption, because NDN does not require this; our requirement is instead support for mobile terminals to be potential content publishers as well as consumers. One can envision applications in which mobile devices in the same physical environment share content locally—e.g., sharing different points of view on the same event or collaboratively annotating the physical world. So, our view of AR applications includes both peer-to-peer and client-server relationships.

Rather than focusing on a set of specific application scenarios, we envision a "meta-application"—an augmented reality

---

[1] More or less, we follow the general definition given in the seminal survey by Azuma in 1997, which reminds us that this is an area that has been waiting for enabling technologies for some time. [3].



(a) System concept.

(b) Example overlay "tab."

Fig. 1: Illustration of a smartphone-based AR browser on a university campus.

browser—that requires a broad set of capabilities representative of AR applications. The AR browser expands the concept of a web browser to incorporate viewer spatiotemporal context (at a minimum) and suggests that named data following certain naming conventions can form an *augmented reality web*, overlaying a "global information universe" like that of the web [4] on the real world.

To visualize an AR browser, imagine an instance of Chrome, Firefox, Safari, or another popular browser running on a set of AR glasses. Instead of a fixed page background, there is a live view as seen from the device camera. The input URL provides an initial rendezvous point from which to bootstrap what content will be shown—from a child's make-believe game to a social media application, or a tool for a construction superintendent to annotate a physical job site (or a familiar university campus setting, shown in Figure 1). Instead of pages, these could be thought of as overlays on the environment. Organized like descendants of the browser "tab," more than one overlay could be shown at once. In this AR browser, each overlay has 1) a sandboxed code execution space, 2) at least one trust root, potentially based on the rendezvous point (URL) provided, and 3) may in turn pull granular content from a variety of local and remote sources as needed.

Each overlay (formerly, "tab") in our AR browser shares viewing context (such as location, orientation, field of view, etc.) and receives content, including executable code, from one or more sources. It also offers the possibility to follow new paths navigated in space, time, and content-specific dimensions as the viewer moves and interacts. For example, imagine a "spatial search engine" overlay at an amusement park that provides directions to upcoming events based on preferences, times of day, and distance from visitors' locations. Additionally, the overlay may provide real-time interaction with displayed content, as in many popular AR games today.

This browser might be an application or part of a future mobile operating system in which individual applications generate their own overlays. In either case, by emphasizing a wide world of granular content and many simultaneous overlays involving the exchange of viewer context for content, it enables us to illustrate the benefits of NDN over TCP/IP more clearly than a single, optimized application, which can be delivered reasonably well today.

Before discussing the requirements for this AR browser, we first briefly evaluate the types of media assets it could be browsing. After that, we detail specific requirements for the meta-application. Then, after introducing the NDN architecture, we describe opportunities that NDN offers to meet those requirements effectively.

A. *Navigable Media Assets*

> Up until now... [motion] picture editors have thought almost exclusively in the horizontal direction. The question to be answered was simply, "What's next?" ...that's complicated enoughthere are a tremendous number of options in the construction of a film. In the future, that number is going to become even more cosmic because film editors will have to start thinking vertically as well, which is to say: "What can I edit within the frame?"
>
> - Walter Murch, *In the Blink of an Eye*, 2011

Discussions of AR are often dominated by the emergence of new hardware for experiencing it—headsets, graphics engines, and sensors, for example. Theoretical and implemented advances in computer vision also receive significant attention from potential application developers. Less often discussed are two critical areas: 1) the evolving nature of media content that will be delivered in and influence AR, and 2) the approach to networking that will make seamless, streaming AR possible. In this section, we briefly explore the former, so that we may continue on to the latter in subsequent sections.

AR delivers content that is intended to be overlaid on a live view of the real world based on the user's location and point of view. We generalize this *context-dependency* of AR to include a variety of user context such as content preferences, semantic information about the environment in which the content is viewed, story-related choices made by the viewer, and other information.

Increasingly multidimensional content can take advantage of this viewer context for navigation. To select a few examples of various types of content, we focus on AR experiences built on real-time 3D graphics engines that incorporate media assets into a 3D virtual space, the "background" for which is live video. We expect AR to thus incorporate new media types needed for 1) capturing and rendering the physical world as a navigable 3D environment, 2) next-generation 2D video, 3) 3D assets that are a mixture of computer-generated and acquired content, and 4) integration with the real world environment. We discuss examples briefly below and summarize them in Table I along with potential dimensions of navigation (and thus, of names, in an NDN approach).

a) Immersive backgrounds: In today's mobile augmented reality applications, a live stream from a smartphone's 2D camera is sufficient to create a "window" into the environment. For more immersive display types and features such as prerecorded backgrounds, immersive acquisition may also be used. The most familiar mechanism in consumer markets is 360-degree capture from one or more cameras. The resulting media is intended to be navigated by shifting the rotation of the virtual observer at the centerpoint of a sphere. (One could imagine this as a base layer for certain types of remotely viewed augmented reality experiences—crossing over with VR.)

However, 360-degree video from a central viewpoint is only one type of immersive media. Other examples include free viewpoint television [5], which records a scene from multiple angles and enables interpolation between views. Such techniques could be used to provide prerecorded full or partial overlays for AR exploration, or offer semi-immersive assets within a larger viewing field. Another relevant technology is light field photography, which enables computational photography techniques for manipulating focus, depth of field, orientation, and other parameters after the fact. It may enable full or partial environments and assets to have realistic focus dynamics and other novel benefits. [6]

b) 2D and 3D assets: AR environments will include 2D assets that are navigated in an integrated way with the environment. Even two-dimensional video can support such dynamic selectivity—e.g., through the use of MPEG-DASH Spatial Representation Descriptions (SRD) to select areas on-the-fly from within a high resolution master shot of a specific area corresponding to, for example, a close up. In this way, the 2D master becomes a world of content from which the point of view is selected. This is what renowned editor Walter Murch (*Apocalypse Now*, *The Godfather I & II*) above calls "vertical editing" in film.[2]

In addition to navigable video, new types of 3D assets will emerge as well, generated through volumetric capture of performers, objects, and other items.[3]

c) Content-driven editing: While the types of navigation above revolve around point of view, content navigation may also involve story choices. Interactive video that supports discrete choices by the viewer provides a good example. Examples include real-time or just-in-time assembly of video frames for live streaming based on story choices [9] and simple cases of being able to change content within a video as it is playing. While the underlying technical approaches are related to those described above, they are controlled by application developers in different ways, and may suggest different naming schemes are necessary to match application semantics.

d) Real-world IoT Integration: Finally, AR experiences can leverage control over the real world as well. While not the focus of this paper, we note that integration with IoT technologies providing control over environmental lighting, sound, and actuation is likely to become commonplace, given the rise in smart home technologies integrated with consumer entertainment and lifestyle brands. See the Illumiroom [10] for one example.

In sum, browsing an augmented reality will integrate not only the content that makes up the bulk of the traffic on the internet today—2D video and images—but also a variety of other types of media (and real-world elements) with various dimensions of navigation, which can be broken up into different types of grains for transmission. Experiences of this content must be able to leverage the network to adapt format, quality, story and other choices on the fly, while being delivered into a display environment with a dynamic viewpoint.

B. Requirements

From the concept of an AR browser and the new media types introduce above, we can outline an initial (and incomplete) set of requirements placed on the underlying approach to networking.

Context-dependent multi-media retrieval at high granularity. As a basic requirement, the browser must efficiently retrieve *potential* content based on viewer context and choices, in order to be able to respond to changing location, field of view, gaze, and other sensor inputs with low latency, from both nearby and cloud sources. Conceptually, as the user's context shifts, the application must be able to respond by changing the "path" through many media options rapidly.

Transparent multi-interface, disruption tolerant, local/global network access. To deliver appropriate performance for this type of content retrieval, the network stack must transparently support simultaneous use of multiple interfaces, disruption tolerant behavior, and local communication either with or without global connectivity. Traditional channel

---

[2] See also [7], an implementation of automatic vertical editing using computer vision.

[3] See [8] as well as commercial endeavors like those of 8i, http://8i.com/.

TABLE I: Example navigation dimensions of new media content types.

Media type	*Navigation / Naming Dimensions*	*AR Application*
Immersive 360^o video	Content identifier, timecode, 3D surface patch of interest, resolution, quality.	Live backgrounds for immersive displays and headsets.
Free viewpoint television	Channel identifier, timecode, point of view, audio simulcast and captioning, quality.	Integration of broadcast content into navigable AR environments.
Light-field photography	Content identifier, timecode, point of view, focus / depth of field, stereo, resolution, quality (if rendered remotely).	Immersive video with realistic focus, parallax, reflections, etc.
Spatially selectable video	Content identifier, timecode, 2D region of interest, resolution or zoom level, quality.	Select only non-occluded or of-interest portions from 2D video feeds.
Volumetric capture	Content identifier, timecode, point of view or 3D volume, quality.	Avatars / characters, telepresence, super-imposed virtual objects, scene reproduction and manipulation.
Dynamic CG	Point of view, any other user context, texture quality.	Virtual elements and, eventually, scene reproductions that are fully navigable.
Content-driven editing	Content identifier, story choices, timecode, user context.	Story- and interaction-based navigation.
Real-world IoT	Location and region of responsibility, various parameters of physical impact.	Integration of lighting, actuation, environmental sound, sensing to supplement viewer context, etc.

and session-based application designs, based on existing web technologies in particular, are likely to fail to provide the necessary granularity or flexibility, especially outside of stovepipe designs, due to the combined overhead of managing control and/or data connections across multiple content sources accessed over multiple interfaces.

Multi-tier content and processing sources. The architecture should support diverse communication and processing models possible in edge/fog/cloud networks. As discussed above, mobile users themselves might be publishers to a local audience, supported by edge nodes for storage and computing capacity. For example, they may share their point of view with others nearby, or enterprise-level or geographically distributed repositories might provide locally specific information that is impractical, unnecessary, or undesirable for cloud storage. Telemetry, inter-participant communication, user-generated content acquisition, and archives of their own AR experiences all require the ability to have data sources close to the edge.

Diversity of content publishers and trust models. The browser concept returns to one of the most appealing qualities of the World Wide Web—the ability for content to incorporate assets from a variety of sources dynamically. The "browsing of the physical environment" suggests that rich and complex notions of trust—such as those we have with various entities in the physical environment—is required. We envision applications that move beyond the binary, connection-based notions of trust in TLS and the centralized management of social network-based trust in commercial social networks. Examples include: a) content that falls into a hierarchical trust model with campus authorities as the root; b) web-of-trust established independently of campus authority by student groups, friends, events, etc.; and c) evidentiary trust where infrastructure / other nodes vouch that a person (their device, at least) was present at a given location at a certain time.

Integration with IoT environments. Along with the examples of actuating in the real world via IoT given above, IoT integration can also be used to gather data that acts as viewer context (as well as content itself). To obtain user context and support multi-user and multi-media experiences, we anticipate that AR applications will coordinate with loT infrastructure, including indoor positioning, sensors, displays, and environmental controls.

III. BRIEF BACKGROUND ON NDN

We briefly introduce important NDN concepts relevant to this application, following recent papers such as [11] on real-time streaming. In the next section, we use the basic concepts of NDN to describe an approach to implementing an AR browser based on data-centric principles.

NDN shifts the "thin waist" of a network from the host-centric communication model of IP to a data-centric, information dissemination model. It is a prominent example of information-centric networking [12].

Communication over NDN employs two types of packets: *Interest* and *Data*. An application wishing to consume data sends Interests for named Data packets to retrieve from the network. Each forwarding node that receives these Interests forwards them to the next hop based on a longest-prefix match against its name-based Forwarding Information Base (FIB). It also stores the incoming interface, from which it received the Interest, in a Pending Interest Table (PIT), where it also aggregates duplicate Interests. When an Interest reaches a node with matching Data, whether an original producer, in-path cache, or any other source, the hop-by-hop state in the PIT is used to return the matching Data, along the original path back to the requesting node (or nodes, providing intrinsic multicast distribution). Each Data packet is signed, enabling any node to answer any request for data (in-network caching) and providing an important building block for data-

TABLE II: Summary of AR browser requirements related to architectural capabilities.

AR Browser Requirement	*IP Limitation*	*NDN Benefit*	*NDN Design Questions*
Context-dependent, high-granularity retrieval	Navigation of multidimensional content must happen at the application level; chunk sizes too large in playout-oriented protocols.	Navigation of granular, multidimensional content via name construction and network forwarding.	Namespace design for navigable content, access paths, metadata, and related certificates/keys.
Robust multi-interface communication	Difficult to leverage multiple radios that are becoming increasingly common, as well as adapting transparently to disruption/mobility.	Intrinsic support for multiple interfaces, asynchronous fetching patterns straightforward to implement.	Best approach to dynamic content, e.g., leveraging Named Function Networking; confidentiality designs; support for publisher mobility.
Ecosystem of multi-tier content and processing	Different content (and processing) rendezvous approaches needed for different network tiers among the edge/fog/cloud.	Consistent access to local and global content and services; network-assisted, host-independent scalability for distributed data.	Best method to synchronize subsets of content of interest; forwarding strategy for application support in E/F/C deployment.
Diverse content publishers	Scaling and other requirements above drive stovepipe designs that increase latency of both coordination and delivery. Host and channel-centric security, along with content delivery cost, steer applications towards stovepipe designs.	Caching and intrinsic multicast support low-capacity publishers and, along with granular naming, lower latency. Consistent, expressive, granular name-based trust mechanisms separable from confidentiality; extensible to edge processing via techniques such as NFN.	Characterization of local network resources needed for typical apps. Economic models for granular content from heterogeneous sources and CDN evolution. Naming and trust designs for non-hierarchical trust; distributed processing models and related security approaches.
IoT integration	Different protocols for low capacity devices, requiring application or middleware-layer integration.	Same architecture can be used with IoT devices; any device can cache any content, supporting simple devices.[4]	Compatibility with low-capability devices; low-latency protocols for sensors impacting interactive navigation.

centric security.[5] Below, we briefly introduce a few higher-level concepts built on the Interest/Data exchange of NDN.

A. Integrative view of storage and processing

NDN's name-based forwarding unifies requests for data and requests for computation or other action; a name, /video_root/track=0/h264-1024k/_v1/_s0, might require a transcoding action to be performed or just retrieve pre-existing data. It is important to note that, in the description of packet processing above, an NDN node can choose to *generate* the Data upon receipt of an Interest containing a particular name prefix. This enables services and in-network processing to be integrated at the same layer. Named Function Networking (NFN) [20] articulates a generalization of this idea.

B. Authenticating everything

Whether created by a service or retrieved from storage, each Interest returns a Data packet. Fundamental to dealing with highly granular content delivered as part of an augmented reality experience is NDN's approach to *securing each of those named data packets* at the network layer directly, through its signature,[6] rather than securing connections or channels between endpoints. In this way, content consumers are able to validate, and publishers can control access to granular content on a packet-by-packet or object-by-object basis. Note that each signed data packet in NDN is immutable, furthering the notion of a growing "universe" of data to be browsed.[7]

Per-packet signatures must be evaluated by applications in terms of some type of trust policy, perhaps selected as part of the content rendezvous process or within browser preferences. In NDN, trust decisions can leverage the structure of names to schematize decision-making on a packet-by-packet basis that does not require channel or session-based semantics; this is commonly called *schematized trust.* [22] This is an important building block that enables consumers to evaluate their trust in the highly granular content, with many entry points, proposed in our approach, without relying on anything about the channel over which they received the data. As each Data packet has its own name and also carries the signer's name in its *KeyLocator* field, NDN enables applications to express trust relationships through rules regulating allowable relationships between the Data packet name and signing key name.

C. Granular access control

Names can also be used to organize fine-grained access control, an example of which is found in recent work on Name-based Access Control (NAC) motivated by applications to granular mobile health data [23]. NAC enables a data owner to enforce access control policies based on data names. It aims to enable the principle of least privilege security to be applied to NDN data access. In parallel with the data namespace, NAC makes use of additional access control namespace elements to facilitate the distribution of encryption and decryption keys to authorized users. Other techniques have been explored that

---

[5] For a more complete description of the architecture, please see [1], [13]. Particularly relevant applications explored over NDN are video conferencing [14], live and streaming video [15], lighting control [16], person tracking [17], vehicular networking [18], and early integrations with the Unity game engine [19].

[6] Note that the term signature here is used loosely and may include other types of authentication, such as HMACs in cases where computational efficiency is important.

[7] The implications of data immutability are discussed in [21].

also have the potential to schematize access control relationships in terms of names. The use of attribute-based encryption is one such approach, where attributes and names/namespaces may be able to be related. [24], [25]

D. Multiparty communication

Building on the Interest-Data exchange, NDN can provide high-level data dissemination functionality. Synchronization (sync) refers to a multi-party communication paradigm that aims to reconcile collections of named data efficiently. Example protocols are given in [26], [27], [28]. They allow members of a group to exchange knowledge about data published under a namespace. When new data is generated, nodes advertise their updated knowledge about the collection, using tools such as digest trees, to represent them efficiently and synchronize with other nodes. The synchronization model of communication is distributed, multi-party, and sessionless. This makes it particularly useful in assisting information dissemination in disruptive environments, where the network exhibits intermittent connectivity or dynamic topology, or can communicate over multiple media.

IV. APPROACH TO AN AR BROWSER

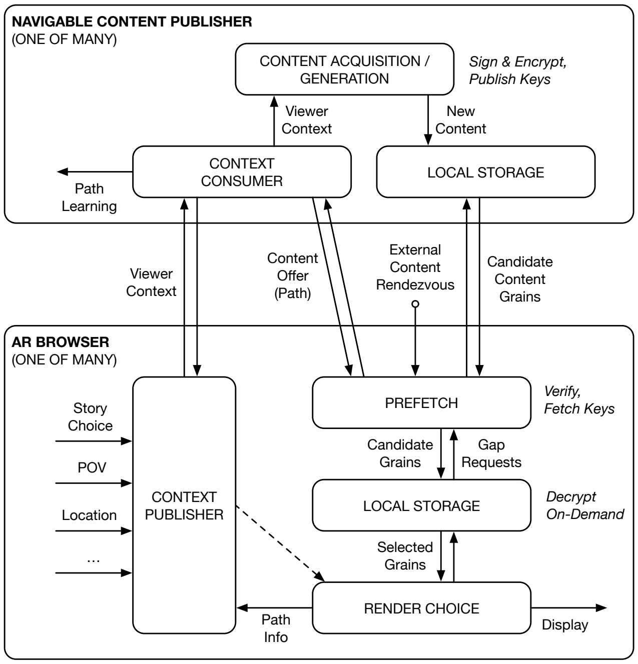


Fig. 2: Conceptual architecture for an NDN-based AR browser.

Figure 2 shows a conceptual application architecture for an NDN-based augmented reality browser, as proposed in Section II, with a corresponding navigable content publisher. In this section, we discuss these components in more detail, and highlight some specific design challenges. We use schematized trust for managing trust relationships between entities, and name-based access control, or other data-centric techniques, for content encryption. Due to space constraints, they are not discussed in detail here, though some high-level security functionality is shown in italics in the above-referenced figure.

A. Major Components

a) Navigable Content Publisher: We assume a content ecosystem consisting of many independent publishers of navigable media, including the types discussed in Section II-A. They use distributed storage and processing that leverages the location independence enabled by NDN. These publishers provide static or dynamically created content as uniquely named *grains*. Grains are NDN Data packets containing chunks of media with name components containing important dimensions of navigation, such as story reference markers, scalable coding quality level, linear timecode, associated geographic location, etc. Grains may include manifests that point to collections of other grains, referenced by name—potentially including ordering or other access information.

Properties of the content grains relevant to navigation are contained in the names or in metadata accessed through a well-known naming relationship with the main media prefix. (This might be known in advance for given content types or described in a schema fetched during rendezvous with a content entry point.)

b) External Content Rendezvous: Users rendezvous with starting points amongst these content grains through similar means to those in today's network. A location-based search engine may provide starting points for content relevant to a user's location and preferences. A descendant of today's streaming services might offer storytelling experiences that can be applied over the current environment of the user. A URL might simply be provided directly by scanning a QR code or reading an NFC tag in the environment. NDN may also offer new opportunities to discover nearby relevant content, by enabling a browser to issue discovery Interests with standardized names over location-limited channels (e.g., a name like /localhop), to which local rendezvous services[8] could respond. In each case, the result of the rendezvous process will be a root name that acts as a high-level entry point into more navigable media elements available to the user, much like a web page today. These are shown in the Figure as "External Content Rendezvous" input into the AR Browser.

B. AR Browser operation

Once it obtains the starting point (formerly, a "page" URL) for a content overlay, the AR Browser renders it in a sandboxed code and virtual display (formerly, a "tab") using the process described below.

a) Prefetch: The goal of an AR Browser is to be prepared to render content for the viewer based on their context with as low latency as possible. Therefore, it must prefetch the most likely set of content to be needed based on context from

---

[8] In each of these cases, we assume that trust in received data will be evaluated using application-specific designs, depending on the type of services involved. For non-free or otherwise confidential content, payment interactions could occur during this step, using an additional exchange between viewing application and content service, to provide the necessary keys to decrypt confidential content.

the user—for example, where they are looking or likely to look. The content *publisher* will be the most likely to have knowledge of the navigation options available from a given content rendezvous point (and indeed, may have to generate content on demand based on user context). Thus, it supports the prefetch operation by providing *content offers* based on user context provided by the browser, retrieved asynchronously through Interest/Data exchange by the browser and placed in local storage.[9] Publishers' content offers represent a set of prioritized content grains most likely to be needed for rendering based on users' context. These may take into account media-specific coding. For example, grain prioritization might apply recent research in how to code efficiently and transmit such 360-degree video, such as using layered coding [29] and "perceptual pruning" [30], with content options expressed in an NDN namespace.

b) Context Publisher: To enable many publishers to provide content offers, the browser must provide the user's context to a potentially large number of content publishers efficiently. The browser—providing a single *context publisher* component—collects, names, encrypts and signs[10] user context. This encrypted user context is offered for publishers to use in deciding what content grains to offer for prefetching by the browser.

Context-content exchange exchange can leverage multi-party synchronization protocols described previously, as well as in-network caching, to limit redundant communication, especially between edge services all within the same radio range. (Sync would notify content publishers of new context items from consumers when available, and vice versa.) Browser-controlled context exchange subsumes the notion of cookies for most non-service-specific context, and enables it to be placed in more direct control of the user.

c) Context Consumer: The publisher's *context consumer* retrieves viewer context that it needs from the set available in the synchronized namespace. This may include the original rendezvous point, common context (POV, location), overlay-specific state, and other information, and formulates content offers associated with content rendezvous points.[11] This may include the acquisition or generation of content on the publisher side, or other operations to prepare the media for the viewer's context, such as transcoding, encryption, etc.

Though context-content exchange appears to require a stateful relationship between publishing and consumer components, this may not be necessary for every type of content. For example, after initial rendezvous with a 360-degree video stream origin point, consumer-side content fetching can be done according to simple rules based on view location and field-of-view, which can be fetched immediately after rendezvous and executed in the AR browser sandbox.

d) Path Learning: The browser's local logic, including application-specific sandboxed code for the overlay, will ultimately decide what content to render for the user based on their current context and interactive choices. The result will be a *path* through the content that is more specific than what has been prefetched. This path becomes part of the user's context, and thus made available to the content publisher, subject to agreed privacy constraints. It can be used for offering new candidate grains, and for learning the probability distribution of paths through content, given certain context. The latter can enable more effective prioritization of content offered to viewers in the future.[12]

C. Edge acceleration

The above descriptions and Figure 2 do not explicitly discuss the use of mobile edge computing (MEC) [31] to support the application. In fact, as suggested, the browser model for our application is designed such that code intended to run in sandboxed execution space on the mobile device itself (like Javascript in a web browser) can be used to do many lightweight local tasks, notably the construction of names for content prefetching based on user context.[13]

However, many AR functions are compute-intensive and/or require operations to be performed on privileged devices. The availability of service resources close to mobile terminals is very important to providing a robust AR experience. NDN repositories in the edges can perform CDN-like functions, holding content that is likely to be accessed based on the statistical profiles of associated consumers' content paths and other context. Using techniques such as Named Function Networking [20], they can provide distributed processing resources for generalized transcoding, re-encryption or security support functions, as well as renaming and encapsulation of data within new namespaces to support alternative access patterns. In future ultra-low-latency wireless networks, they may be able to support some types of processing at rates and latencies close to that of the render loop, making it possible to offload support for complicated signal processing tasks such as simultaneous localization and mapping (SLAM) [32], feature extraction via SIFT and SURF [33], [34], and neural-network / deep learning-based processing for image classification.

V. CHALLENGES

Based on the approach described above, we outline several research challenges for NDN, and information-centric networking more generally, to realize the AR browser and content ecosystem.

---

[9] In practice, to lower latency and/or keep the publisher's proprietary navigation strategies secure, the functional components shown in the Figure may be implemented in a mixture of publisher-side functionality with client-side code in the sandboxed "overlay" space, which is handed off to browser APIs for executing actual fetch and rendering.

[10] We expect that keys have limited temporal validity and may vary for different types of context, enabling granular control over what publishers can access what context.

[11] It might also provide code for use locally in the browser that translates context to names to fetch, improving bandwidth by taking virtual distance and orientation of the object into account (e.g., [30]).

[12] See, for example, the recent paper [29], in which video quality for 360-degree streaming video is adjusted based on user viewport prediction.

[13] Even in this locally managed case, however, we would point out that multiple "tabs" or overlays are likely to consume the same context, suggesting the use NDN-style data access methods locally.

a) Content naming: Namespace designs must be developed for content grains, associated metadata and manifests, as well as certificates and content keys. These designs must take into account not only the dimensions of navigation but also the desire to standardize prefetching behavior, so that it can be executed by library code (or the core code of the AR browser), rather than being reimplemented by each application. (Of course, each content type presents its own specific challenges for naming, suggested by the table of navigation dimensions.)

b) Context naming: Similarly, user context naming designs are also needed. They must also be general enough that publishing can be generally supported by the AR browser rather than application code; the namespaces should also support the envisioned synchronization patterns described above. The *context-content exchange* pattern is critical to this approach and must be developed carefully.

c) Name engineering: In both content and context naming, one of the open research challenges is to balance the many properties of a given grain of media (or context) that could be expressed in a name with engineering restrictions on name size and structure. Potential solutions include moving some descriptive information into metadata objects, using standard naming conventions or linked explicitly in manifests. Guiding this tradeoff are considerations of whether network forwarding or application demuxing is required based on a certain property or not.

d) Trust models: The above namespace designs must be developed in parallel with trust models for different types of content and context. Potential trust models include hierarchical trust with a well-known root—which has been well-explored in NDN research to date—as well as less researched ways to express web-of-trust relationships among socially shared content, and various definitions of "anonymous" data models used for both context and content. These trust models will yield namespace and schematized trust relationships for certificates and keys.

e) Access control: Providing comprehensive encryption-based access control mechanisms for content and context that enable the highly granular retrieval patterns described, while protecting confidentiality, is vital to the success of an NDN-based approach to AR browsing. From our experience in working with NAC [23], we believe several parallel paths of research are necessary: 1) usable security – simplification of the concepts and APIs needed for developers to implement common design patterns; 2) name confidentiality – approaches to protecting information leakage from names themselves, while preserving the benefits of network forwarding based on structured names; 3) more expressive and direct techniques for schematizing access control based on names (approaches such as attribute-based encryption [35] are promising, but unlikely to provide sufficient performance in their current form).

f) Attribute-based naming: Hierarchical names are very powerful but present challenges for expressing the types of attribute-based classification schemes often used with content. General solutions for handling set and attribute-based content descriptions with a mix of hierarchical names and metadata objects would be very valuable to application developers—whether they are implemented in higher-level protocols or directly in the network.

g) Subset sync: Efficient techniques to synchronize subsets of namespaces would open up significant possibilities to support the type of continuous nearest-neighbor retrieval patterns [36] that appear to be required by AR content prefetching.

h) Distributed designs: The ability to mix browser and publisher-side code in the current web has yielded a powerful and flexible (if often, quite complicated) environment for authoring distributed applications with sophisticated user interfaces. We propose here that this pattern should be continued in AR browsers, where client-side code can also be used to map context information to content grain naming for efficient retrieval. Distributed application design patterns are needed that mix browser-side code and appropriate naming of content grains to minimize demand on active server-side components.

i) Edge acceleration: Approaches to designing namespaces and data exchange patterns are needed for distributing computation for common compute-intensive tasks, especially where intermediate products may be of use to many users. In our experience with NFN, a significant challenge is the appropriate security model. NAC-type techniques appear as if they will be quite powerful for distributed "read" access to content, but determining the appropriate name and key space to use for publishing the resulting distributed computations is challenging—especially in chained computations, as is managing data provenance.

VI. CONCLUSION

In his classic children's book *Where the Wild Things Are*, Maurice Sendak populated a jungle island with wondrous, loud, dangerous beasts inspired by the adults of his childhood. A young boy, Max, whose bedroom has mysteriously transformed into this island, becomes their king and conspirator before returning home in time for supper. For Max (and the reader), what is reality? What is augmented?

In the context of technology research, it's often easiest to view the future as a set of capabilities that are somehow both inevitable and yet to be figured out. We could look at AR this way to generate a concluding example scenario. Instead, we invite the reader to think of all the ways that the human race has "augmented" reality until now, and extend that thinking into an open ecosystem of AR content, browsed as easily as the web. For our lab, the next research step for NDN-based AR is to follow Max's cue and let fictional experiences drive our experimentation.

This paper has considered how Named Data Networking's shift of web semantics to the "thin waist" of the Internet can enable the creation of such experiences that mix the real and the fictional, the everyday and the fantastic. The concepts and application architecture presented are preliminary, but suggest significant promise to 1) enable content creators to realize navigable, networked media; 2) make viewer context (and thus agency) a first class part of media experiences; 3) create an ecosystem, or web, of that media that benefits from the cloud without creating stovepipe solutions.

ACKNOWLEDGMENT

The author would like to thank Lixia Zhang and Peter Gusev for helping develop ideas related to AR over NDN.

REFERENCES

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[3] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and virtual environments*, vol. 6, no. 4, pp. 355–385, 1997.

[4] T. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann, "World-wide web: The information universe," *Internet Research*, vol. 20, no. 4, pp. 461–471, 2010.

[5] M. Tanimoto, "Overview of free viewpoint television," *Signal Processing: Image Communication*, vol. 21, no. 6, pp. 454–461, 2006.

[6] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005.

[7] V. Gandhi and R. Ronfard, "A computational framework for vertical video editing," in *4th Workshop on Intelligent Camera Control, Cinematography and Editing*, 2015.

[8] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor *et al.*, "Fusion4d: Real-time performance capture of challenging scenes," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 114, 2016.

[9] I. Literat, "The work of art in the age of mediated participation: Crowdsourced art and collective creativity," *International Journal of Communication*, vol. 6, p. 23, 2012.

[10] B. R. Jones, H. Benko, E. Ofek, and A. D. Wilson, "Illumiroom: peripheral projected illusions for interactive experiences," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 869–878.

[11] P. Gusev, W. Zhehao, J. Burke, L. Zhang, T. Yoneda, R. Ohnishi, and E. Muramoto, "Real-time streaming data delivery over named data networking," *IEICE Transactions on Communications*, vol. 99, no. 5, pp. 974–991, 2016.

[12] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

[13] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[14] P. Gusev and J. Burke, "NDN-RTC: Real-time videoconferencing over Named Data Networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 117–126.

[15] D. Kulinski, J. Burke, and L. Zhang, "Video streaming over Named Data Networking," *IEEE COMSOC MMTC E-LETTER*, Jul 2013.

[16] J. Burke, P. Gasti, N. Nathan, and G. Tsudik, "Securing instrumented environments over Content-Centric Networking: the case of lighting control," in *Proc. of IEEE INFOCOM 2013 NOMEN Workshop*, Apr. 2013.

[17] P. Gusev, Z. Wang, J. Burke, L. Zhang, T. Yoneda, R. Ohnishi, and E. Muramoto, "Real-time streaming data delivery over named data networking," *IEICE Transactions*, 2016.

[18] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "VANET via Named Data Networking," in *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April/May 2014, pp. 410–415.

[19] Z. Qu and J. Burke, "Egal car: A peer-to-peer car racing game synchronized over Named Data Networking," NDN Project, Tech. Rep. NDN-0010, October 2012.

[20] C. Tschudin and M. Sifalakis, "Named functions and cached computations," in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. IEEE, 2014, pp. 851–857.

[21] P. Helland, "Immutability Changes Everything," *Commun. ACM*, vol. 59, no. 1, pp. 64–70, Dec. 2015. [Online]. Available: http://doi.acm.org/10.1145/2844112

[22] Y. Yu, A. Afanasyev, D. Clark, V. Jacobson, L. Zhang *et al.*, "Schematizing trust in named data networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 177–186.

[23] Y. Yu, A. Afanasyev, and L. Zhang, "Name-Based Access Control," NDN Project, Tech. Rep. NDN-0034, Revision 2, jan 2016.

[24] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the iot," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 725–730.

[25] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in icn: Attribute-based encryption and routing," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. ACM, 2013, pp. 39–40.

[26] Z. Zhu, A. Afanasyev, and L. Zhang, "Let's ChronoSync: Decentralized dataset state synchronization in Named Data Networking," 2013, under submission.

[27] W. Fu, H. Ben Abraham, and P. Crowley, "Synchronizing namespaces with invertible bloom filters," in *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*, May 2015, pp. 123–134.

[28] M. Zhang, V. Lehman, and L. Wang, "Partialsync: Efficient synchronization of a partial namespace in ndn," Technical Report NDN-0039, NDN, Tech. Rep., 2016.

[29] A. TaghaviNasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using layered video coding," in *Virtual Reality (VR), 2017 IEEE*. IEEE, 2017, pp. 347–348.

[30] P. Pourashraf and F. Safaei, "Perceptual pruning: A context-aware transcoder for immersive video conferencing systems," *IEEE Transactions on Multimedia*, 2017.

[31] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5g," *ETSI White Paper*, vol. 11, 2015.

[32] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 593–598. [Online]. Available: http://dl.acm.org/citation.cfm?id=777092.777184

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[34] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014

[35] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in icn: Attribute-based encryption and routing," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13. New York, NY, USA: ACM, 2013, pp. 39–40. [Online]. Available: http://doi.acm.org/10.1145/2491224.2491237

[36] Y. Tao, D. Papadias, and Q. Shen, "Continuous nearest neighbor search," in *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 287–298.