

# **“Second Generation” NDN Applications:**

Design Patterns, Libraries, and Architectural Support

ACM ICN 2018 Tutorial

Boston, MA, September 21, 2018

## **NDN Thinking**

Alex Afanasyev, Jeff Burke

“It is widely accepted that creative design is not a matter of first fixing the problem and then searching for a satisfactory solution concept; instead it seems more to be a matter **of developing and refining together both the formulation of the problem and ideas for its solution.**”

Cross & Dorst (1999), quoted by Brooks (2010)

That is, what happens if we...

Name ALL THE  
THINGS!



## Section Objectives

- Discuss how using NDN rather than TCP/IP can impact application design and development.
- Outline affordances of NDN and design principles that we believe are good starting points for application research.
- Introduce an example application – AR Browser.
- Motivate sections of the tutorial that follow.

## What NDN *should offer* to Augmented Reality by naming all the things:

- **Low latency interactivity** via packet granularity requests with app semantics and enabling edge acceleration;
- **Context-dependent** retrieval of media, ranging from location to content preferences and demographics;
- **High throughput** for scene video and content overlays;
- **Progressive retrieval** for responsive/scalable display, variable level-of-detail, predictive fetching;
- **Reverse CDN to scale consumers:** Content *and context* publishing, incl. users' mobile terminals;
- **Heterogeneous wireless:** Mobile terminals should be able to transparently use a variety of comm technologies;
- **Real-world trust:** Diverse, non-binary trust models;
- **IoT Integration:** Ability to integrate with data/devices that may not / should not be Internet-accessible.



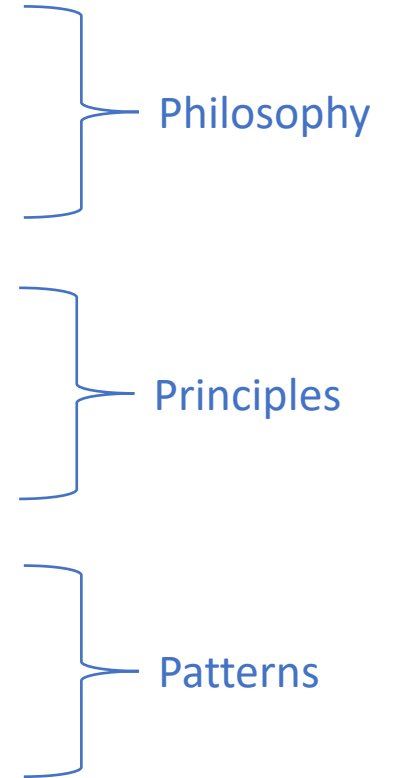
"Pokémon Go Will Make You Crave Augmented Reality",  
*New Yorker*, July 12, 2016.

# Challenges with IP left for your offline review

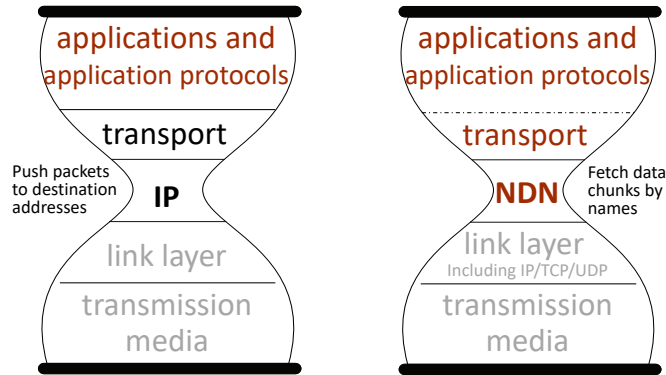
<i>AR Browser Requirement</i>	<i>IP Limitation</i>	<i>NDN Benefit</i>
<b>Context-dependent, high-granularity retrieval</b>	Navigation of multidimensional content must happen at the application level; chunk sizes too large in playout-oriented protocols.	Navigation of granular, multidimensional content via name construction and network forwarding.
<b>Robust multi-interface communication</b>	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.
<b>Ecosystem of multi-tier content and processing</b>	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.
<b>Diverse content publishers</b>	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.
<b>IoT integration</b>	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. <sup>4</sup>

# (Our) NDN Thinking: *From Architectural Basis to App Realization*

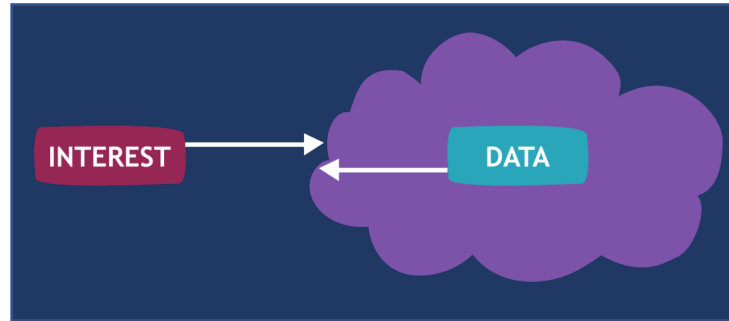
- 1. Conceptualize the application in terms of named data**  
Emphasize real-world principals, data ecosystems, rather than devices
- 2. Borrow what works from the existing Internet**  
End-to-end principle, request-response model, SPKI-style security
- 3. Implement w/data-centric design and deployment patterns**  
Avoid what comes with point-to-point connection assumptions



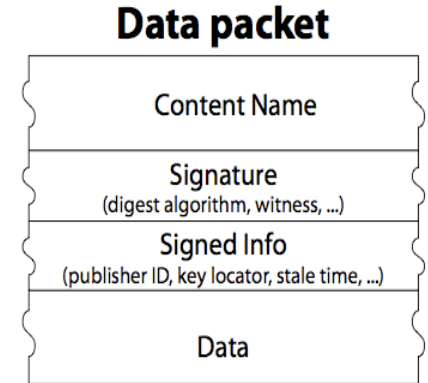
# 1. Conceptualize the application in terms of named data



**Data is the Thin Waist**



**Interests get Data from anywhere**



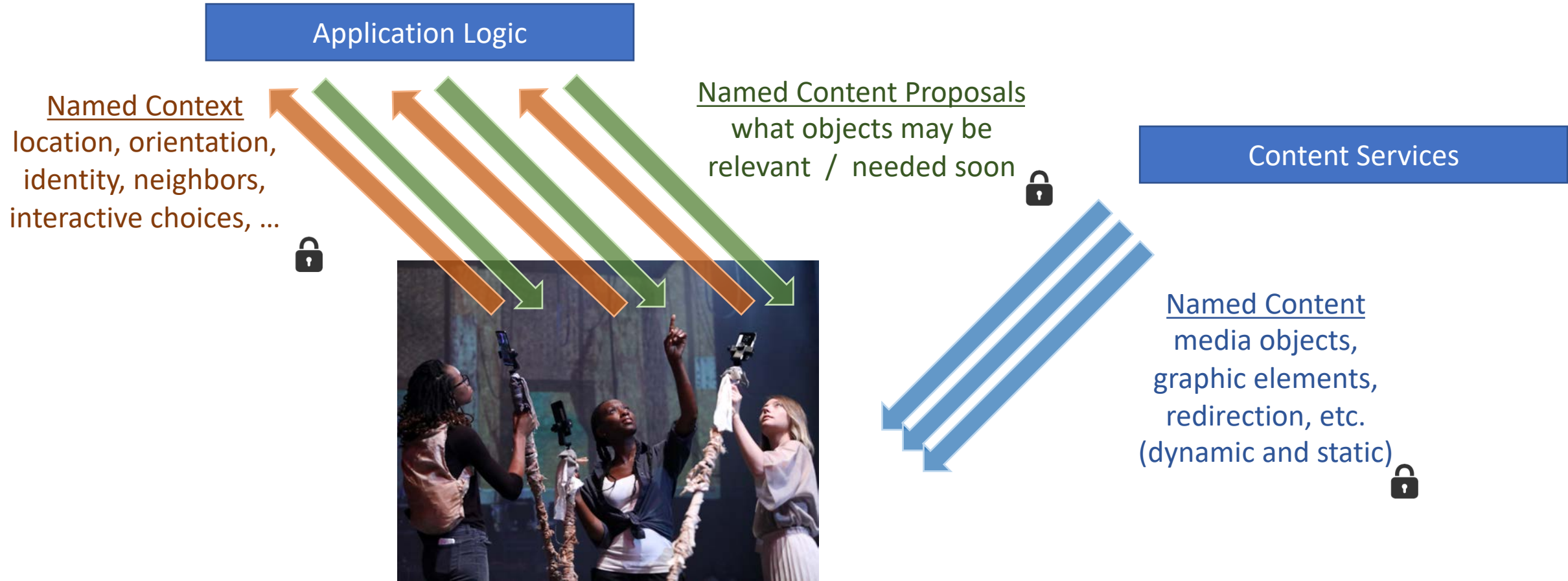
**All Data is signed**

*Names facilitate management and trust derivation.  
Data is signed by producer and verified by consumers.  
Repos enable persistent storage of named data objects.  
Sync enables multi-party synchronized state.*



# Generalizing AR (and the web)

Observation  
Context and content require processing and storage not achievable on mobile handsets or well-served by cloud.



## “In terms of named data”

- What are the various kinds of data?
  - What process creates them?
  - Who owns them?
  - Who (and what) processes can access them?
  - When are they needed?
- How will the data evolve over time?



# AR Example #1: Even cinematic content will not be linear

Watch game like an actual spectator

Free Viewpoint Video Synthesis and Delivery

**C-17** Personalized Video Playback from Arbitrary Viewpoint

NIT R&D EXHIBIT 2015

RGB/depth sensor

Capturing the scene using multiple sensors

360° range sensor

360° spherical camera

Features

Application Scenarios

Personalized video playback in s

Multi-viewpoint television



9/21/18

Light field photography



360 video

FAST COMPANY

LEADERSHIP MAGAZINE MOST INNOVATIVE COMPANIES MOST CREATIV

Why Volumetric vr Is The Real Future Of Virtual Reality

With 8i's technology, it's possible to walk around a human subject in a VR experience. That makes it more immersive, more real, say experts.



Volumetric imaging

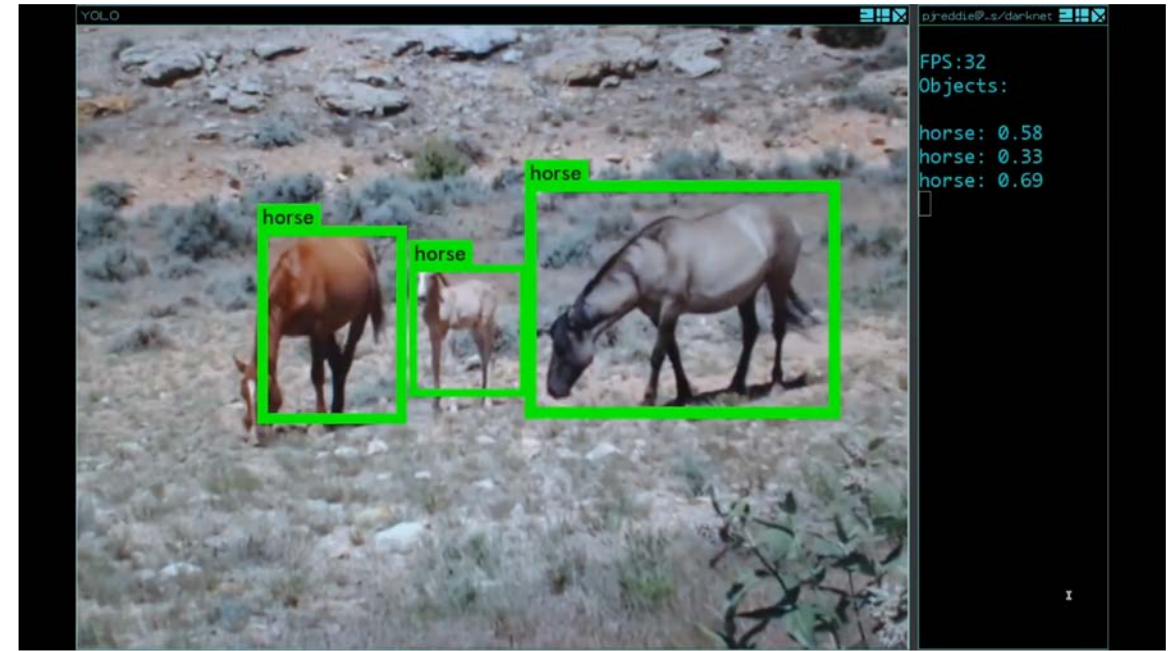


Subframe selection

# AR Example #2: Context will be determined in real-time, with edge assistance



Google Tango @ GTC 2015



Darknet YOLO (You Only Look Once)



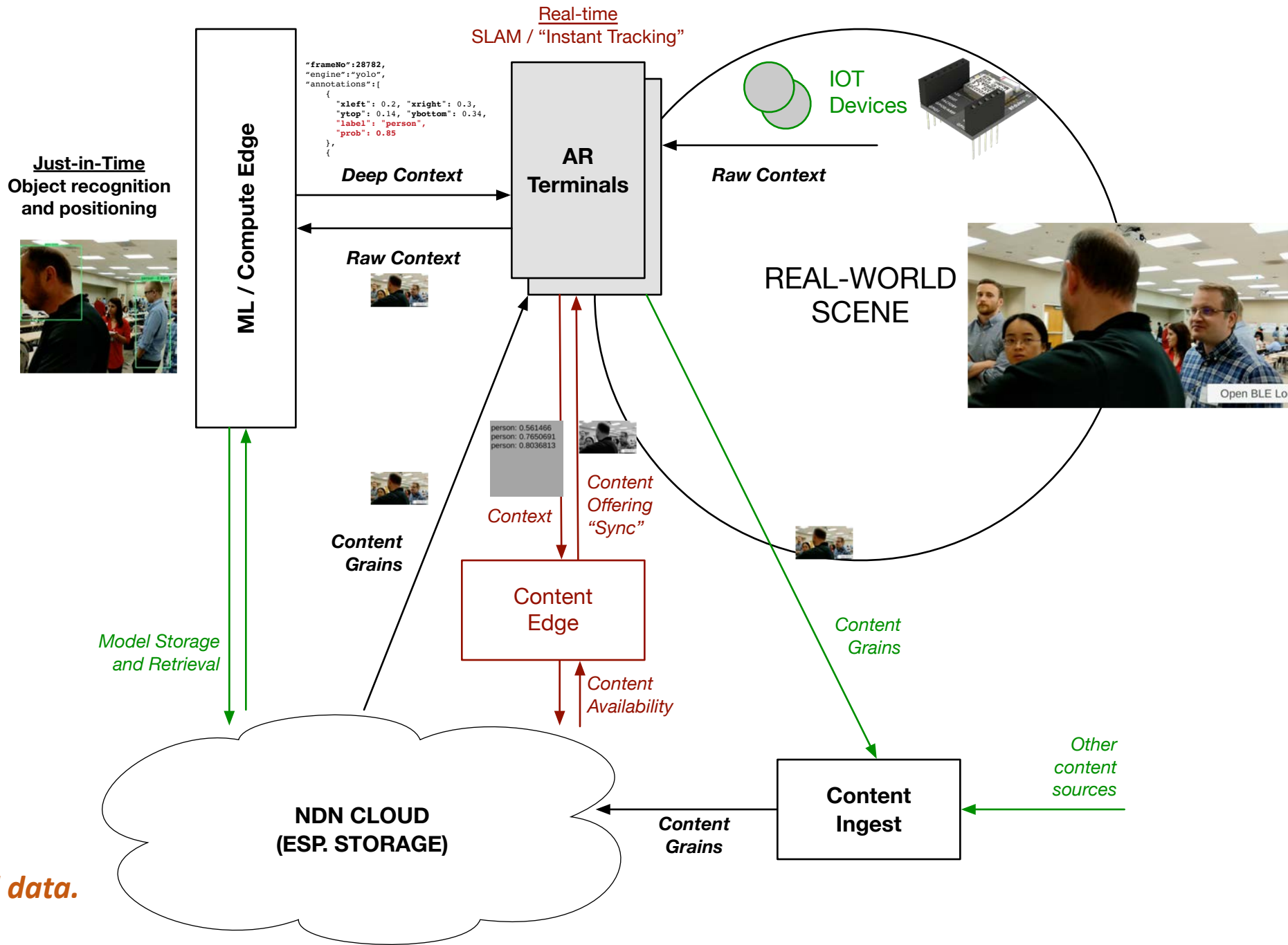
## 2. Borrow what works from the existing\* Internet

### Augmented Reality as web...

- **A (decentralized, interoperable) data web integrated with physical world.** Layers instead of tabs.
- Apps as per-layer unikernels / containers.
- Consumer responsible for requesting what it needs.
- Sessions replaced by **multi-party context-content exchange.**
- URLs as user-facing names that start in-network discovery. Many entry points into content navigation – brand, location, etc.
- Common service interactions expressed as data-centric protocols.
- *Edge supported context analysis (and content generation).*



\* - Not always how it is done in the walled-garden, cloud-centric models of today.



➤ **Context, Content, Metadata, Certs, Service, Storage, Identity – all named data.**

### 3. Implement w/data-centric design and deployment patterns\*

Some examples:

- Create host-independent behavior
- Embrace Multicast
- Enable (and Use) Storage Everywhere
- Communicate Assertively & Opportunistically
- Share Namespace Updates, Not Connections



Aspects of these show up in cloud-based designs but don't work well in highly dynamic / mobile environments.

A prerequisite for how NDN achieves the above:

- **Secure data at its creation.**

\* - Here, closer to Christopher Alexander's use of the word in *A Pattern Language: Towns, Buildings, Construction* (1977), than in the more specific software sense that seems to have followed it.

# Secure the Data at its creation.

- Simple concept.  
Required for all the rest to work.
- Key granularity (in data type, time, identity dimensions) is closely tied to granularity of control.
- Foregrounds identity and ownership.
- Applications must consider retention, privacy, etc. explicitly – what were repository / storage considerations are now important general app strategies.
- Topic of this afternoon

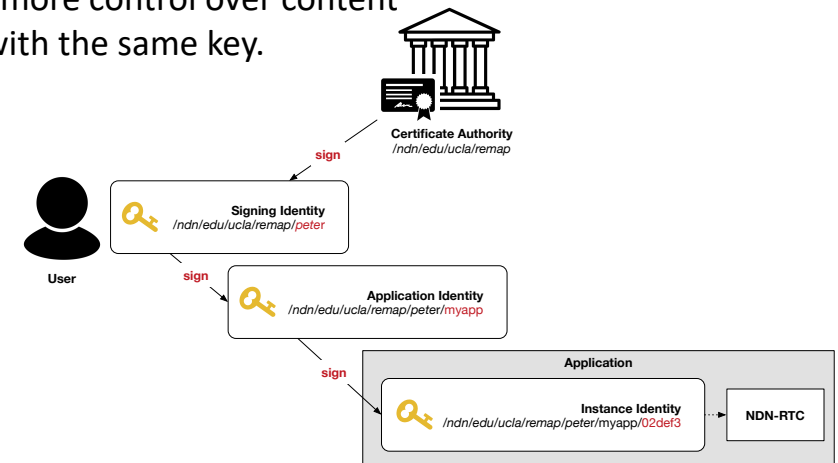
## AR Browser

Identities: **Access, Application, Lens Sandbox, Persona**

Key granularity

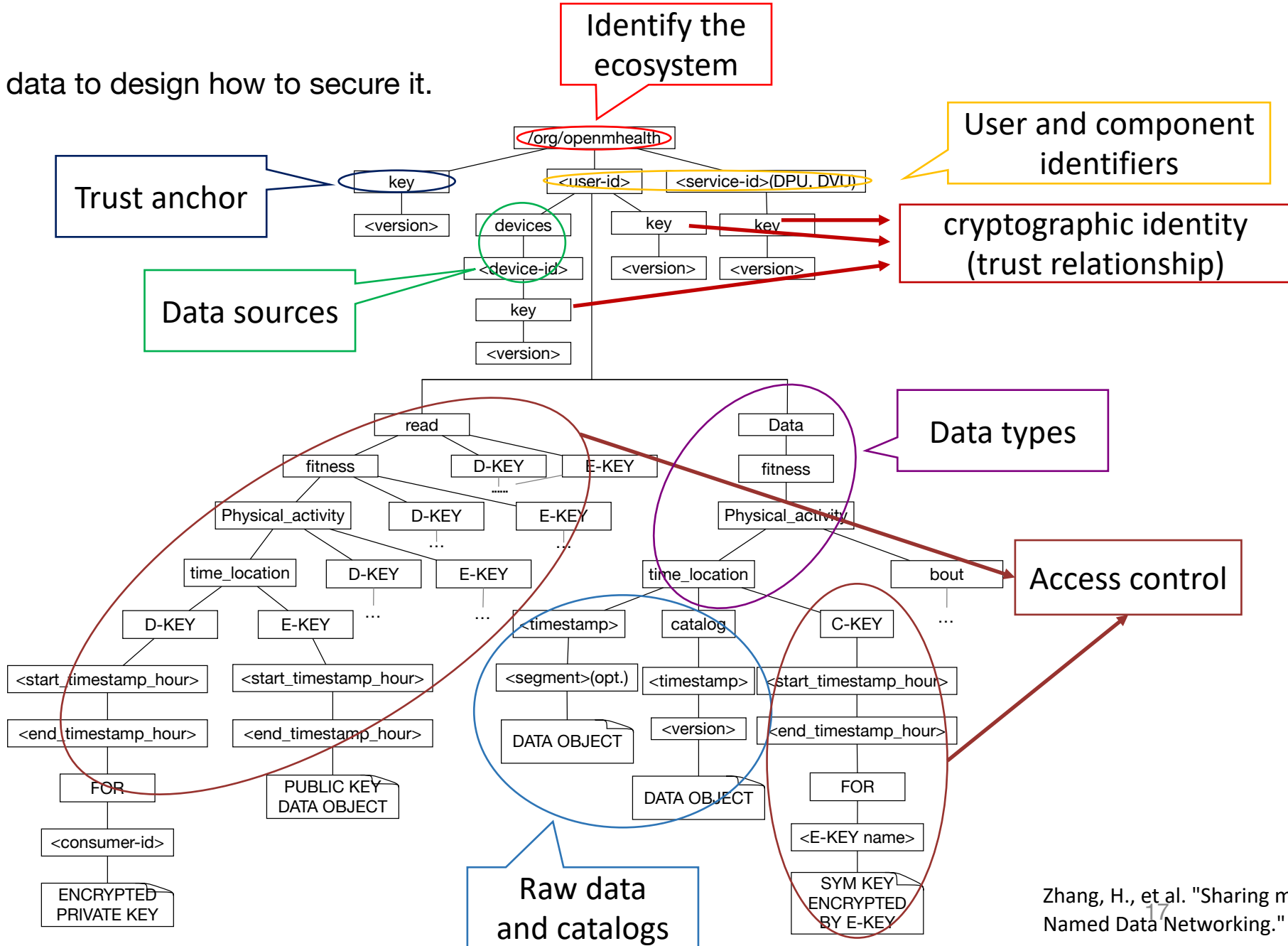
- *Context*: How much linkability allowable for different types of user context?
- *Content*: How much visibility of what others are consuming is acceptable?

New techniques such as ABE enable wider use of multicast, caching by allowing more control over content encrypted with the same key.

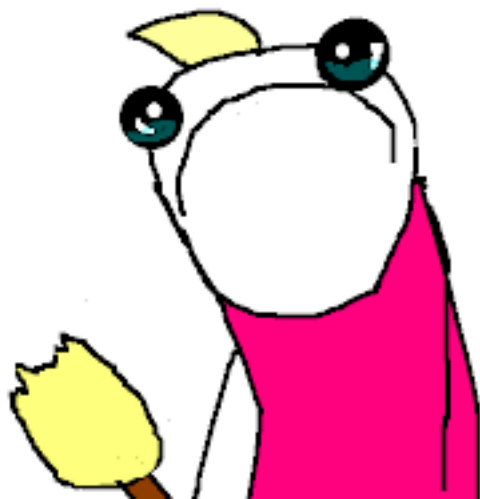




Have to **name** *all* the data to design how to secure it.



Name and sign all the things?



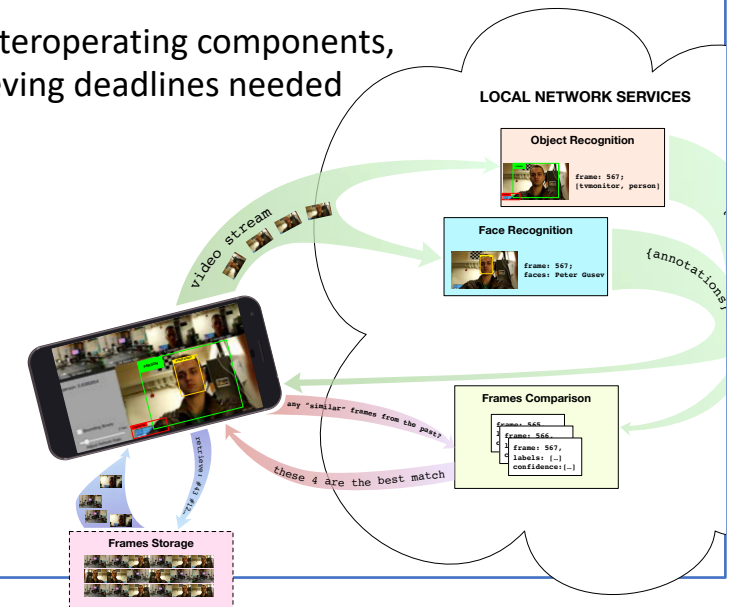
# Create Host-Independent Behavior

- A reason for success of “the cloud”
- Now possibly in a wider variety of networking situations by using named, signed data.
- In IP, requires mapping services, which add brittleness / complexity and increase attack surface.
- Instead, name (and sign) all the things.
- Requires designing (and coding) with host-independent thinking, even for data *about hosts*.
- More work to do: In-network name discovery.

## AR Browser

Design choices:

- On the mobile, explicitly separate device, browser instance, user identity in design discussion and abstraction
- Similar on the edge, don't even prototype on a host anymore – container.
- Minimal information about the device itself.
- Asynchronous, interoperating components, coupling by achieving deadlines needed by others.



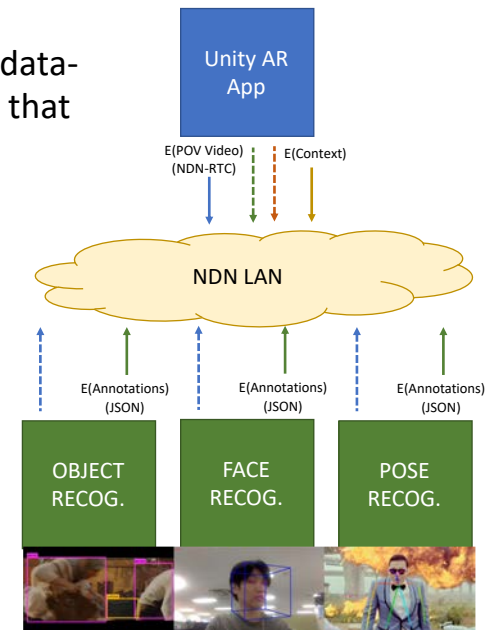
# Embrace Multicast

- Though TCP/IP supports multicast in theory, in practice, build around unicast
  - Because of connection-level security
  - Limited infrastructure support
  - Handling in a host-based way is cumbersome
- NDN provides intrinsically
  - In networks, via stateful forwarding
  - In local context, e.g., by leveraging broadcast nature of many media
- Follows easily from host independence

## AR Browser

Clearest example of multicast benefit:

- A variety of edge services all offering different real-time analysis of the same stream, without increasing the load on the mobile device.
- Have to design data-centric security that allows for this.



# Enable (and Use) Storage Everywhere

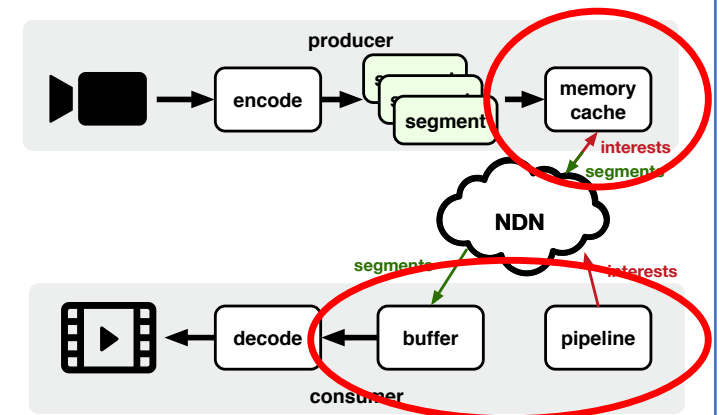
- No difference at network layer between accessing stored and on-demand data
- Every node has some storage
- Publish-and-forget for high level applications; using in-memory and persistent storage
- Challenges:
  - Retention policies
  - Data for namespaces you don't control
  - Relationship to data access patterns.

## AR Browser

Wide use of publish-and-forget approach on the producer side.

On the consumer side, our current buffers only target the kind of content that can be displayed, but this will soon give way to prefetching at a variety levels, and the *content candidates*.

- SVC enhancement layers
- AR content that *may* be selected through navigation or interaction.



Work in progress: How can nodes share information they have that's not in their own prefix?

# Communicate Assertively & Opportunistically



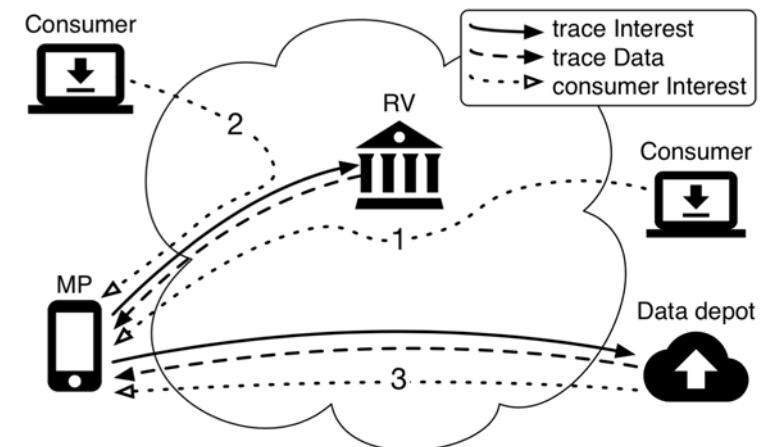
- Best-effort delivery is an important part of Internet's success
- Extend best-effort to *attempts to communicate*
- In many cases, data-centric designs and deployments should aim for async communications, whenever they can
- Offers resilience
- Considered originally for tactical applications and other IoT, but applicable even just for one's laptop or desktop

## AR Browser

Not only must the local networking stack seek out and use all ways to move bits.

This is generally below the browser itself. The application must have an async design that handles intermittent connectivity well.

In NDN, the network itself can be opportunistic and assertive in providing paths for Interest/Data exchange. This will be discussed in the publisher mobility section.



# Share Namespace Updates, Not Connections

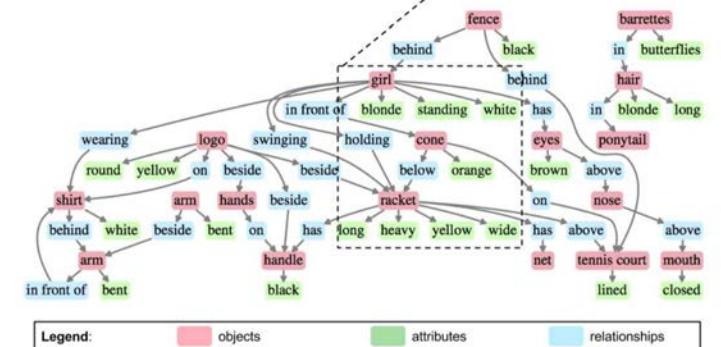
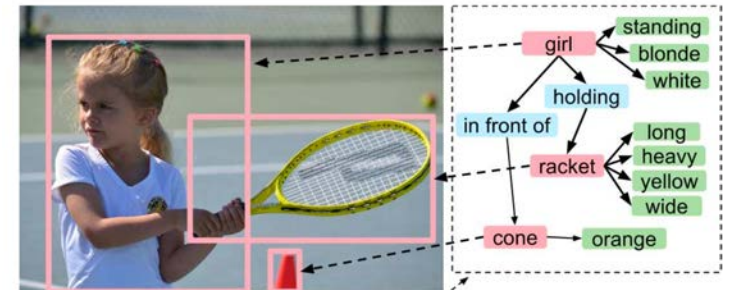
- Namespaces as (selectively) shared memory between different ecosystem components.
- TCP can be seen as two-party synchronization, with simple chunk numbering (for each end) as the namespace
- NDN can generalize this to multi-party sync using app namespace
- Leverage efficient set reconciliation techniques
- *Not covered due to time limits. See previous tutorials and NDN Tech Report NDN-0053.*

## AR Browser

Not only many consumers for one publisher....

Also many publishers for one (to many) consumers, all potentially filling the same namespace.

No need for notion of a connection.



Name\* ALL THE  
THINGS!



\* - and sign, and optionally encrypt.



## Next presentations

- **Real-time communication over NDN.**

AR builds on real-time media. We explain our data-centric design, the NDN-RTC library, and introduce a protocol approach to fetch real-time data in NDN networks.

- **Library updates and explorations.**

We discuss the current library suite, recent updates, and new API explorations being used in the AR browser application.

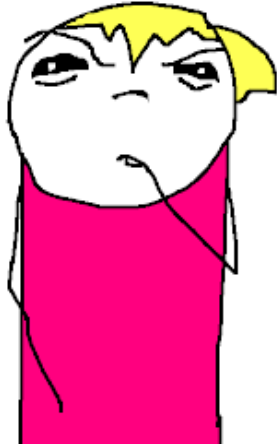
- **Publisher mobility.**

AR as conceived here is inherently mobile. We introduce KITE and its use in this application, which will be expanded on in the corresponding conference paper.

- Come back after lunch to explore **data-centric security.**

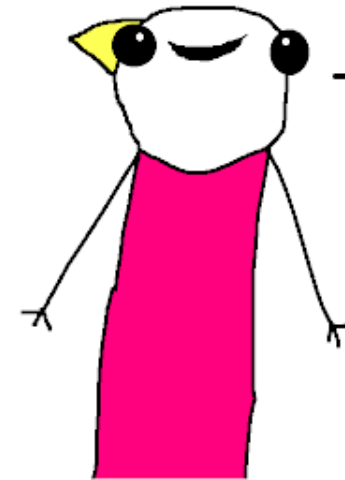


work?



No...

INTERNET!



FOREVER...

For all the memes, and the story of any translational research:  
<http://hyperboleandahalf.blogspot.com/2010/06/this-is-why-ill-never-be-adult.html>

## How to Learn More (Recent Papers)

- NDN Architecture
  - A. Afanasyev et al. “A Brief Introduction to Named Data Networking”, *MILCOM 2018*.
  - <http://named-data.net/>
- Design patterns
  - J. Burke et al., “NDN Impact on Tactical Application Development,” *MILCOM 2018*.
- Application examples
  - W. Shang et al. “Breaking out of the cloud: local trust management and rendezvous in named data networking of things,” *IOTDI 2017*.
  - J. Burke “Browsing an Augmented Reality with Named Data Networking,” *ICCCN 2017*.
  - <http://ice-ar.named-data.net/>