Mobile Data Repositories at the Edge

Ioannis Psaras, Onur Ascigil, Sergi Reñé , George Pavlou

UCL, London, UK

Alex Afanasyev

Florida International University, Miami, USA

Lixia Zhang

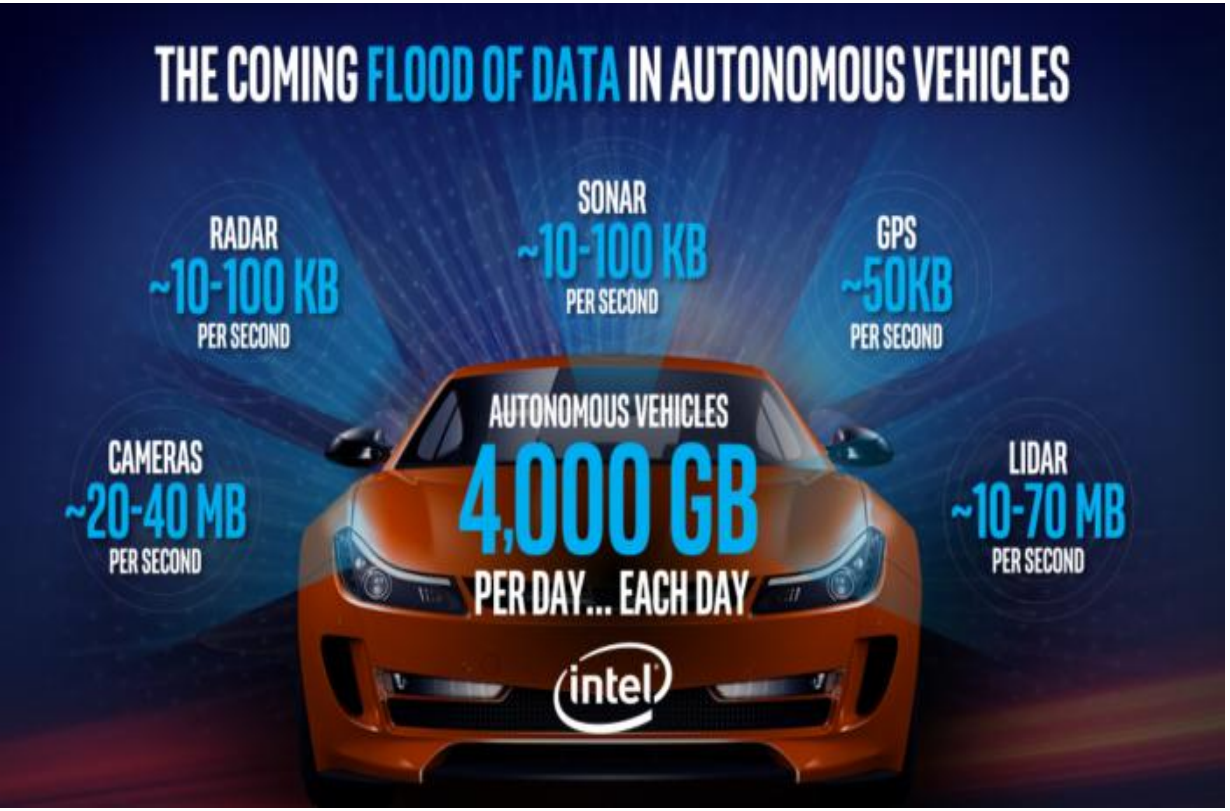
UCLA, Los Angeles, USA

usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

USENIX Workshop on Hot Topics in Edge Computing (HotEdge '18)
Boston, MA, July 10, 2018

Disruptive Edge Application Deployment

- Humongous data production at the edges
  - Global edge data production to exceed 1.6 zettabytes by 2020 [Cisco].
  - Traffic flow is reversing – Intel proposed a reverse CDN



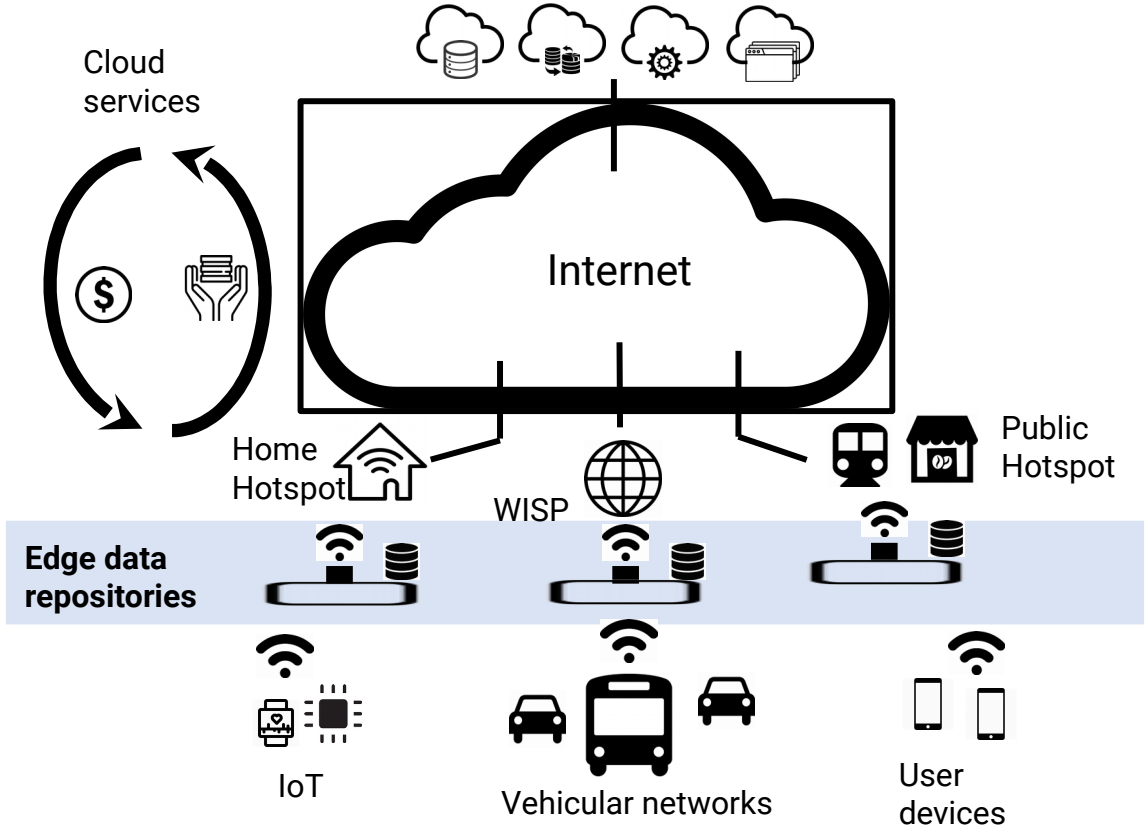
- Problems
  - Edge networks are not engineered for such large reverse flows
  - Mobile backhaul network capacity is limited
  - Upgrades to infrastructure are costly

Edge-Data Production Examples

- Biker's helmet camera or car's dash camera
  - A system to report accident-related videos to insurance companies
  - Only accident/collision related data is useful
  - Detect collision and send, filter out the rest
- Amber Alert system
  - A system to detect missing children
  - Uses public camera feeds and image processing
  - Requires support for real-time storage/processing ideally near where data originates
- Mobile Content Generating Apps
  - Stream videos from phones

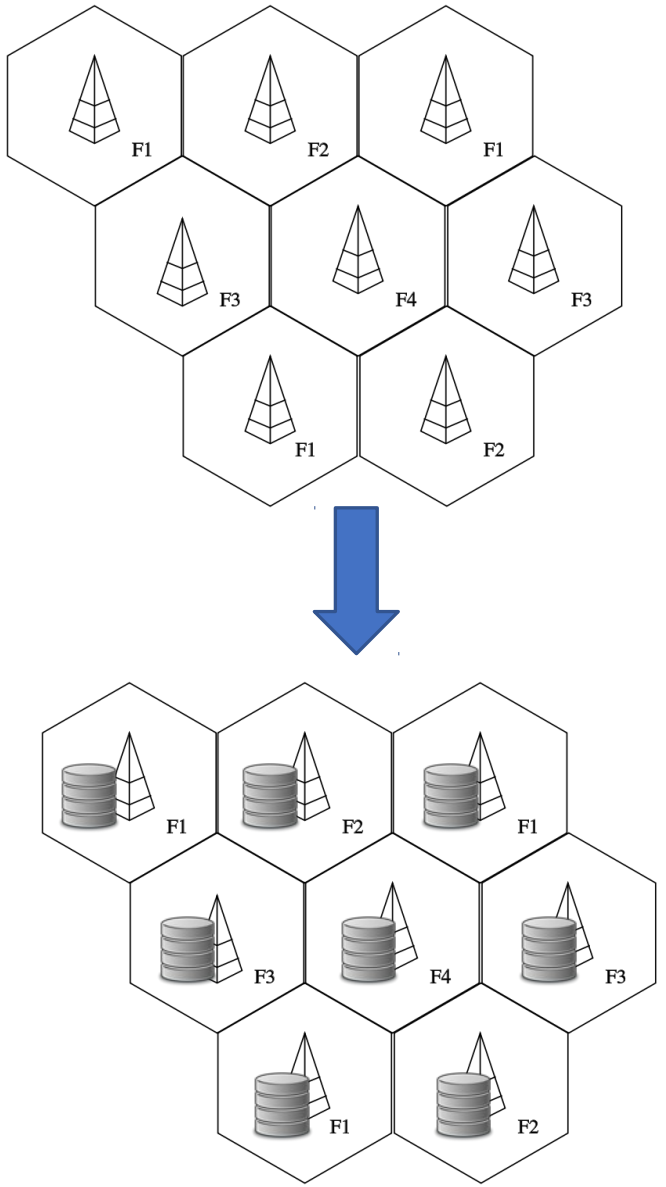
Plain Connectivity is NOT the Answer

- Data Upload
  - Limited bandwidth and storage resources
- Data Sharing
  - Some data is useful locally
- Data Processing
  - Limited battery life



Edge Data Repositories

- Our proposal
  - Complement network wire/cell connectivity with local edge data repositories
  - Virtual user storage allowance
  - Possibly as a service from the MNO
  - A data-centric communication approach that secures data not the channel

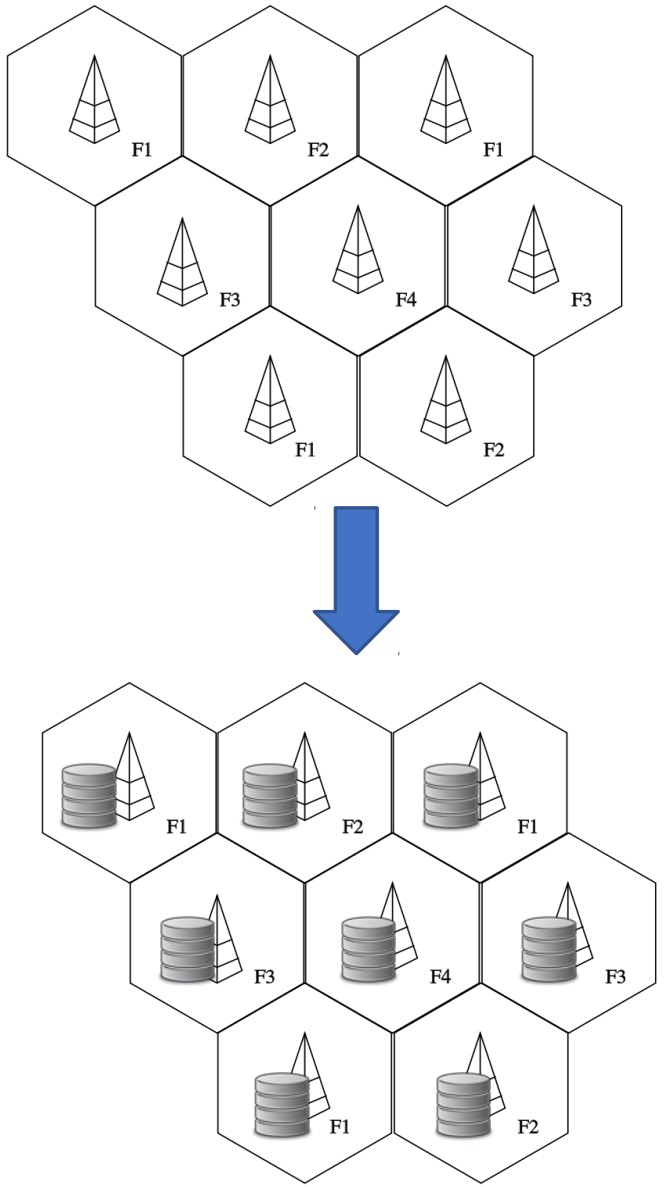


Edge Data Repositories

Store – Process – Send

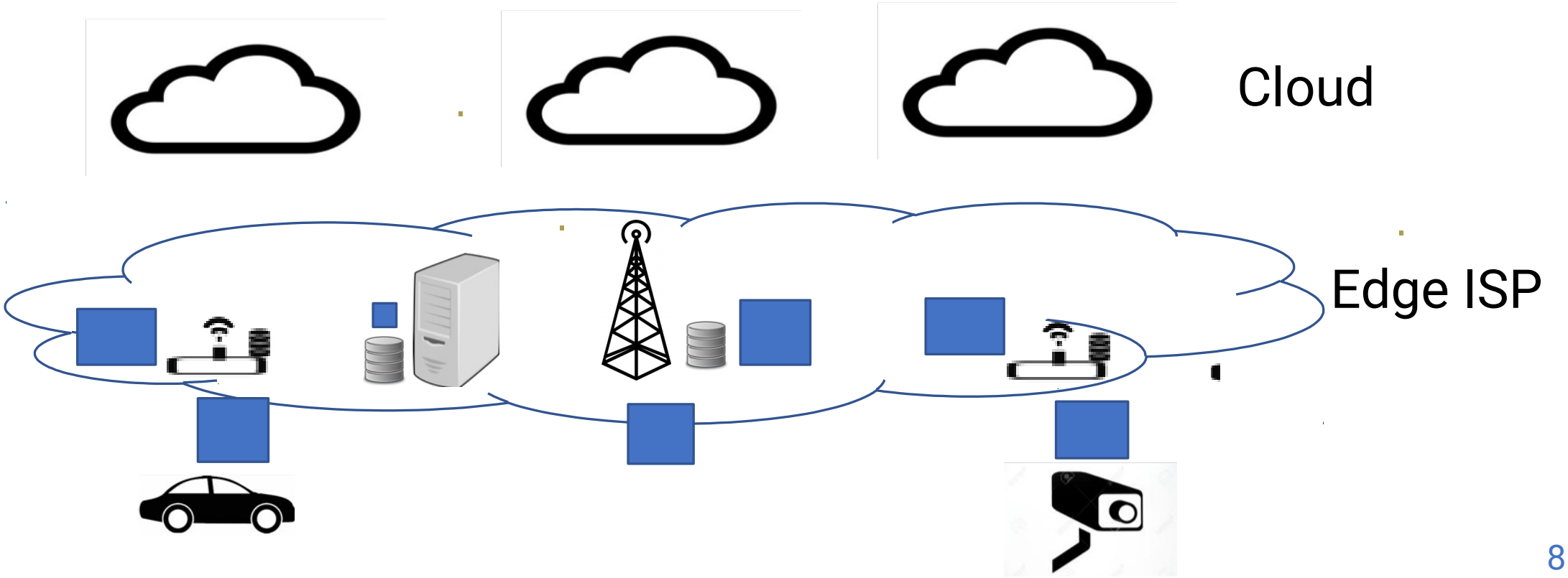
Edge Data Repositories

- Benefits
  - Support for asynchronous data collection
  - Support for local processing of data
  - Use storage as buffering to reduce peak data sending rates
  - Provides inherent support for producer mobility
  - Reduce Costs for ISP and MNO



Store-Process-Send Model

- Local processing of data
  - Bandwidth (i.e., cost) reduction
  - Increased performance (low latency) for users



IP Challenges For Edge Data Production

- IP = Point-to-point, “wire” connectivity between end-points
- “Synchronous” communication is the only service offered
  - IP forces applications to synchronize data immediately with cloud
    - Uploading data to Dropbox does not actually require synchronous communication
- Store-process-send model is not natively supported
  - Local processing before synchronizing with cloud
    - Can significantly reduce the information sent upstream
- “Access to information” is not in the forefront
  - Focuses on the location of information storage points
    - Secures channels not the information itself
- Mobility is problematic in IP
  - Can break session-based communications

Named Data Networking Communication Model

https://named-data.net/publications/tutorials/

Interest packets

Name

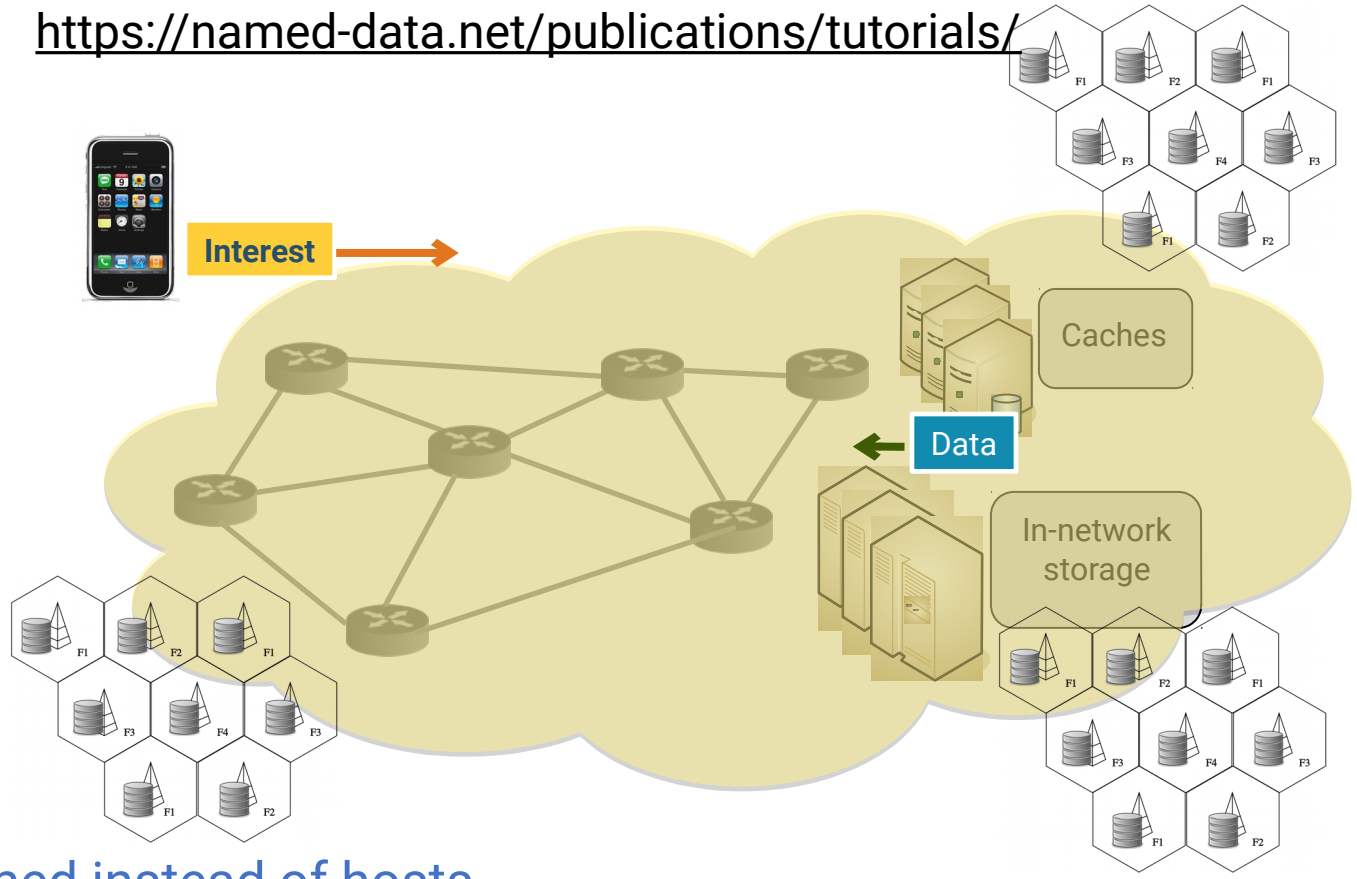
Optional fields

Data packets

Name

Content

Signature



Data is explicitly named instead of hosts

- The smallest unit of data (fits in a single packet) is a chunk

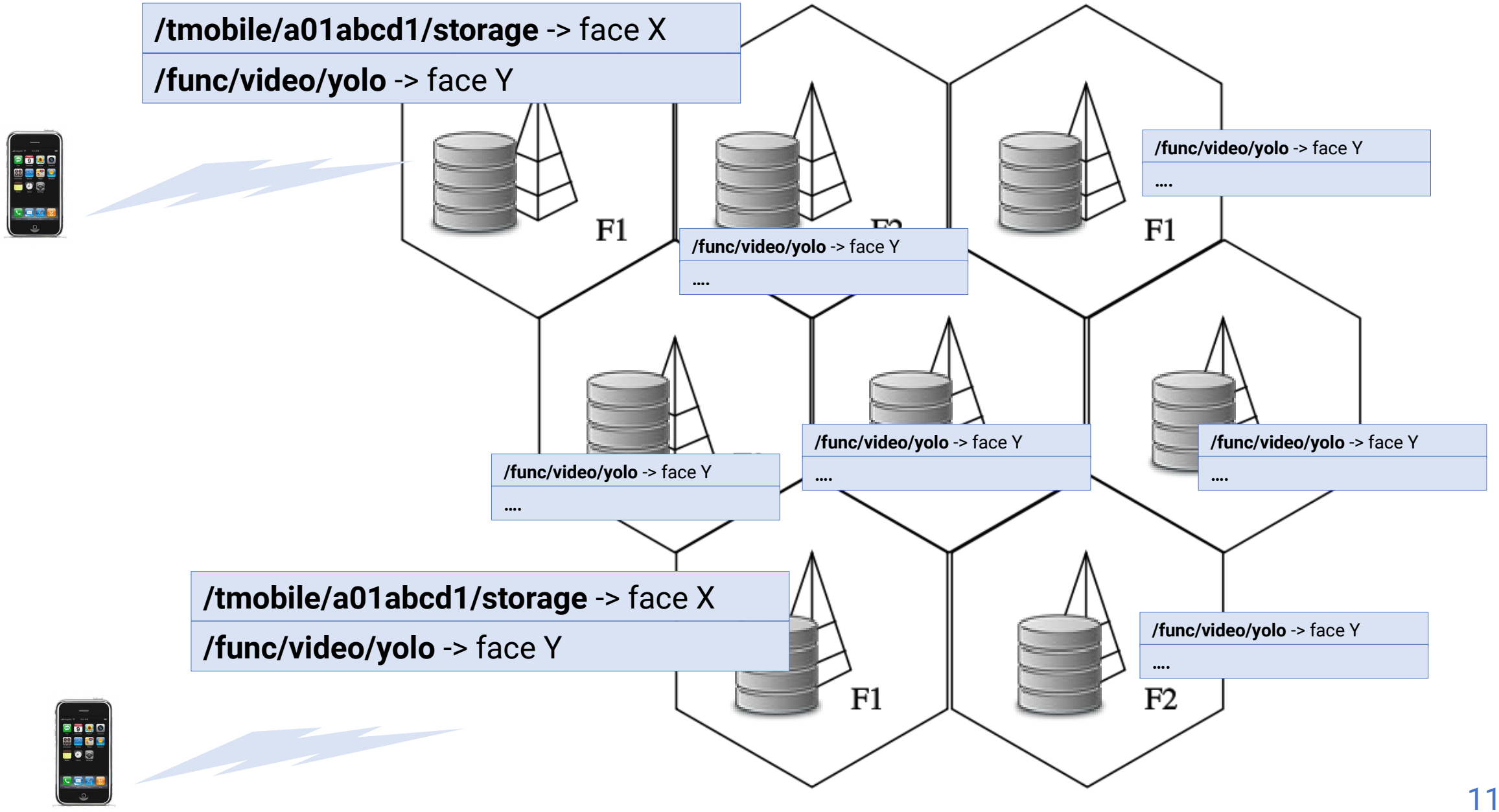
Naming data chunks enables the network to route based on names

- A persistent name that does not change with mobility

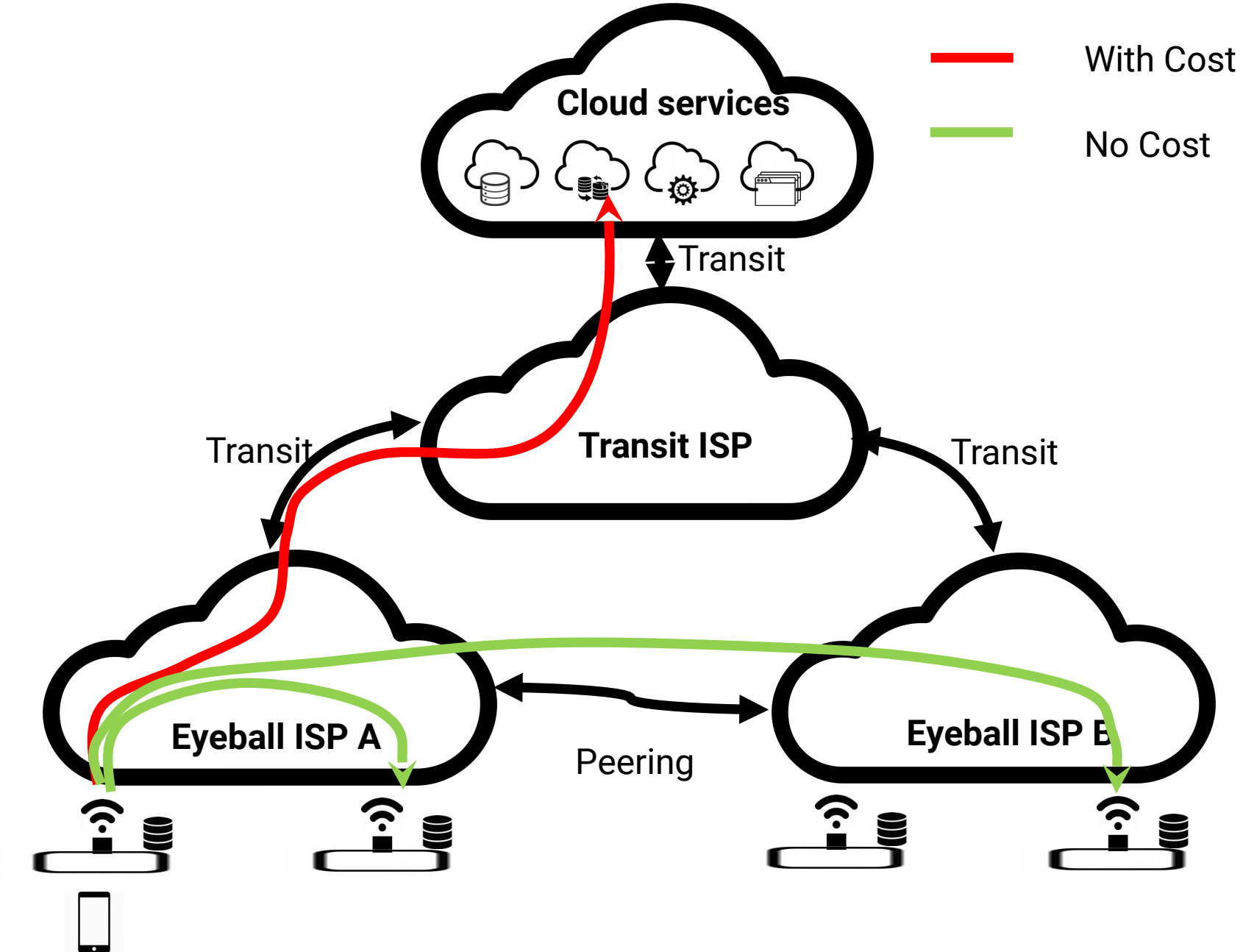
Data contains additional meta-data for authenticating data directly

- Signature: computed over the data
- Key verification info.: name or location of a certificate to verify signing keys

Name-Based Forwarding in NDN



Cost Savings with Edge Processors and Repositories



Technical Challenges

- Name resolution & producer mobility
  - Steering requests for named data to the right storage location
    - Challenging especially for near real-time applications
  - Possible Solution: “DNS”-like system
    - APs informs an authoritative NDNS server of a “forwarding hint”
- Pulling application data at the APs
  - Instantiate lightweight versions of applications inside the edge repositories.

Future Directions: How to Manage Edge Data?

- How long and where to store data locally at the edges?
- When to process data locally?
- When to send data to cloud?

- Metadata includes “hints” on how to manage data
  - Data access Scope: local to the edge ISP, global or a mix of both
  - Shelf-life: data is irrelevant after expiration
  - Processing requirements: what kind of processing and where
  - Deadlines: processing images to extract information within a certain time
- Data management at the edges
  - Pro-active: send all the incoming data to the cloud
  - Re-active: inform the cloud of the arrival of data and wait for a request to send data
  - Hybrid: use the meta-data to decide based on access scope, urgency, etc.

Future Directions: Deployment considerations

- How much storage/processing resources are needed at the edge?
  - Applications may require keeping data around (e.g., for forensics)
- Who owns and manages the storage and processing infrastructure?
  - Edge ISP, third-party (e.g., CDN model), cloud service provider, etc.
- Design of the edge repository architecture
  - Service model and API
- Deployment of data-centric communication at the edges
  - Compatibility with IP
  - Overlay vs underlay

Conclusions

- Edge Data Repositories to mitigate challenges of massive data production
  - Buffer data when channel capacity is not sufficient and send it later
- Store-process-send data for bandwidth savings and low-latency
- Next step: developing mechanisms to manage data in a distributed system with heterogeneous (federated) control