RUTGERS

# Delivering Internet-of-Things (IoT) Services in MobilityFirst Future Internet Architecture

Jun Li, Y. Shvartzshnaider, J. Francisco, R. Martin, K. Nagaraja and D. Raychaudhuri

WINLAB, Rutgers University

October 24-26th, 2012



WINLAB WIRELESS INFORMATION NETWORK LABORATORY

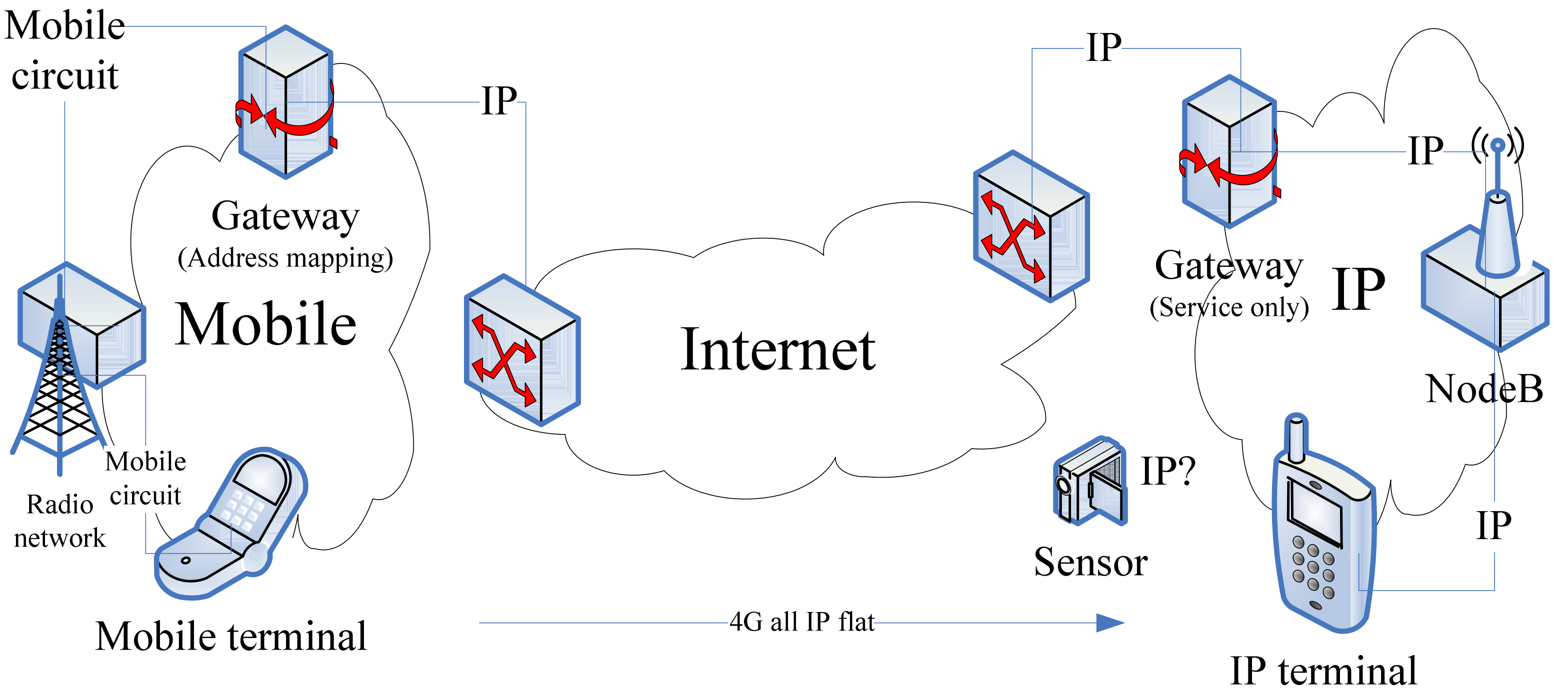
# A Big Question

- Does Internet of Things (IoT) need a new FIA design?
  - No, it is an overlay
    - IoT is just another name of Web of Things (WoT)
    - IoT is just a different expression of M2M, CPS (Cyber physical system) applications
  - Yes, it requires new in-network features
    - IoT is a network connecting to physical world objects same as Internet to computers now – for example, everything is addressable with an IPv6 address / identity
    - IoT is a pervasive / ubiquitous computing platform
- MobilityFirst – yes, IoT is a part of FIA
  - Things have Identities at MobilityFirst core network
  - Data from/to Things are distributed, processed and accessible at MobilityFirst core network

# The Core Challenges of IoT

- Universal identity
  - EPCglobal, IPv6 enough? Security is the key
- Data and middleware API standards
  - The main reason that causes isolated information islands, IoT ≠ M2M Apps
- New business model
  - Mobile operator monopoly vs. open Internet service

# Mobile networks – all IP flat networks




Sensors are IP nodes? All Things are IP nodes?

# Problems of IPv6 ID?

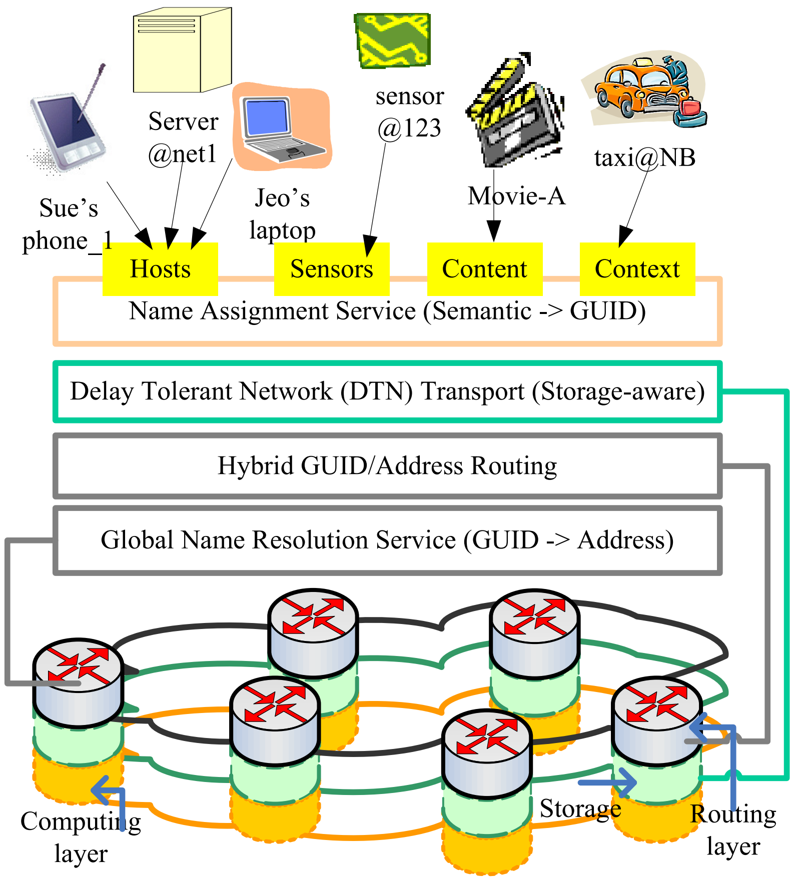
- IPv6 (address as) Identity is not secure
  - DoS attack – address can be spoofed
  - In-network pay service not possible – extra layer, end-to-end session required
- When a Thing assigned to an IP identity
  - It may not run TCP/IP, in many cases, not need to do so
  - It is tied to a network resource associated to a network operator, inflexible for Things with multi-homing, dynamic-homing or no homing

# MobilityFirst – GUID

- Global unique identification (GUID), separated from network location / operator:
  - For any networked objects: hosts, sensors, content or services
- Fundamentally secure
  - Anti-spoofing – DoS avoidance
  - Self-certifying – in-network pay service possible
- Transport requires no end-to-end session (TCP/IP)
  - Routing, transport are identity (GUID) based for hop-by-hop data blocks
  - Easily support mobility (disruptive service), in-network multicasting and in general any in-network service

| GUID: | Public Key of Owner | Optional Suffix |
|---|---|---|

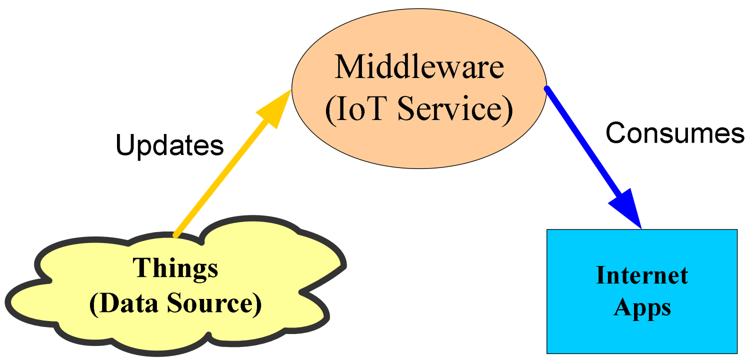
# MobilityFirst Future Internet Architecture

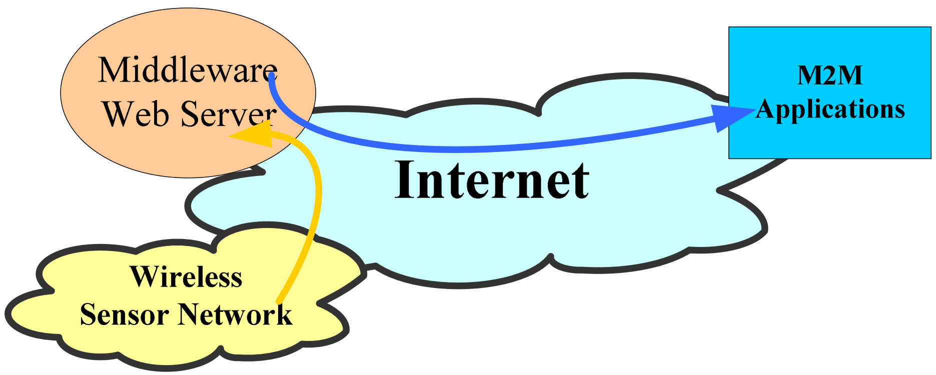



- Key Functions
  - Fast name resolution (GNRS): GUID to address mapping at 50-100ms time scale
  - Routing of GUID objects
  - Delay tolerant network (DTN): Transport without end-to-end,
- Key Features
  - Self certifying, Multi-homing, In-network multicasting
  - In-network caching and computing layers

# Things in Future Internet

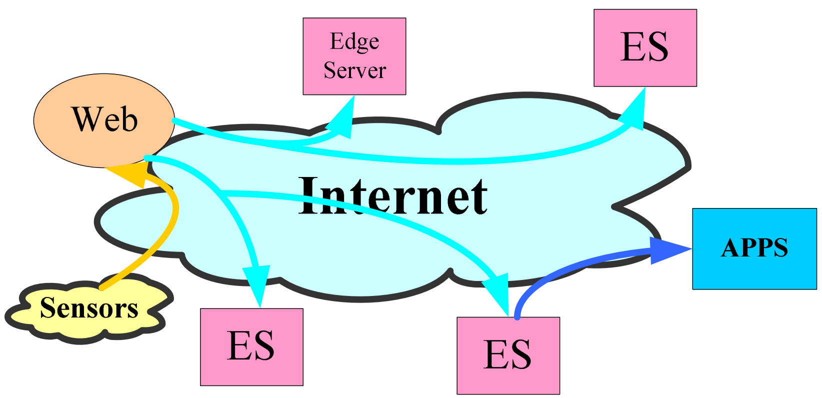
- Things are source of dynamic data of interest to Internet applications
- Raw data are usually processed by IoT service (middleware)
- Challenges of traditional application layer approach:
  - Isolated information islands – no unified platform
  - High latency and traffic load over Internet

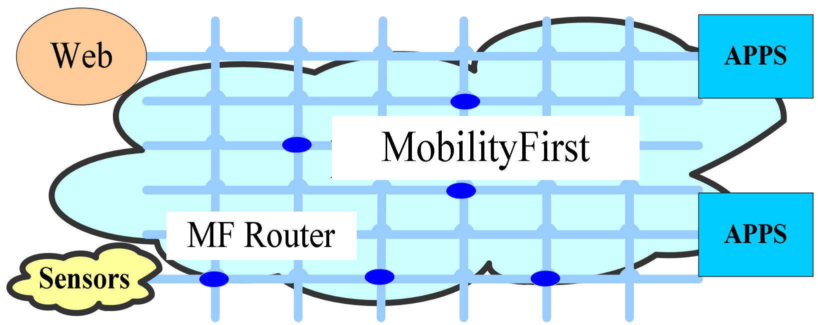


# Overlay vs. In-network Distribution

- CDN (Content Distribution Network) solution
  - Overlay network with edge servers (ES) to reduce latency and traffic load
  - Services are accessed by URLs cached at ES
- MobilityFirst – in-network distribution
  - MF routers directly route, cache, compute GUID identified data and middleware (servicelet), enabled by in-network caching and computing layers



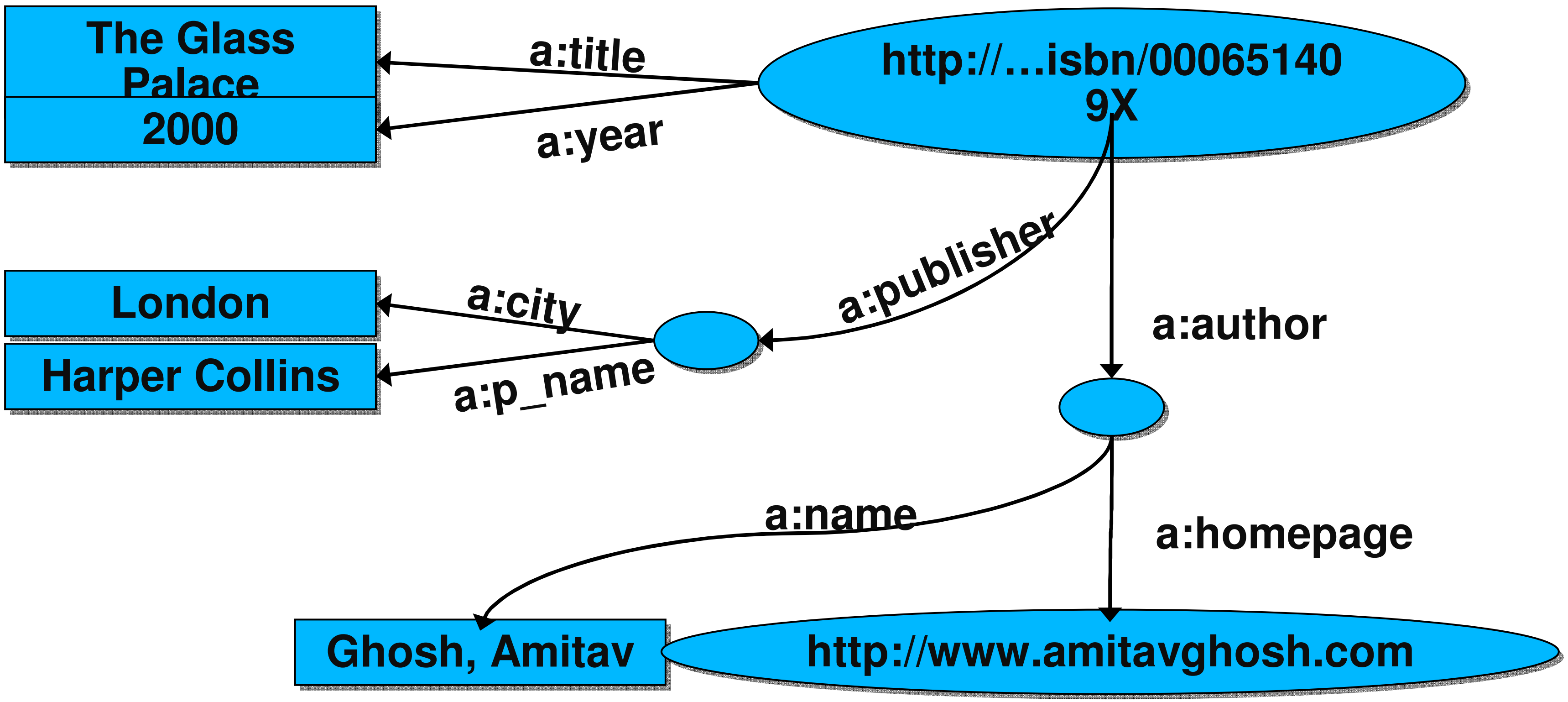
# Challenges on Middleware Distribution

- GUID solves identity problem, but more challenges on middleware, which are
  - Lack of standards, complex, app-specific (Mobiiscape, UBIWARE, HYDRA etc.)
  - The main reason prevents the convergence of data (from Things). IoT remains difference from M2M apps.
- Linked-Data Space, the semantic web approach, could be the future of middleware for IoT
  - Things are data in Linked-Data Space
  - Middleware are database operations to Linked-Data Space

# Semantic Web Technology

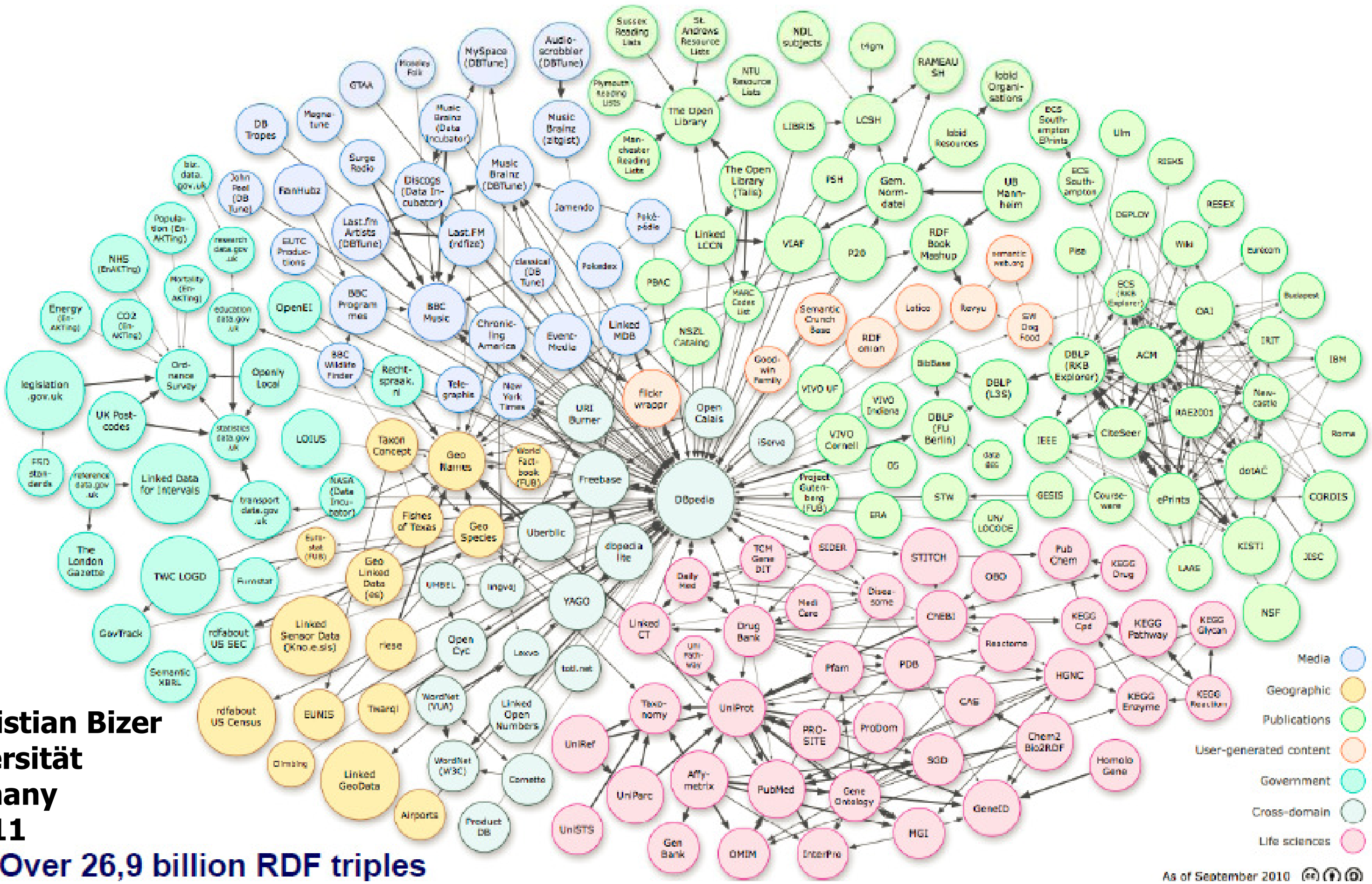
- Building up the relationships between data
  - Store web data with semantic links
  - Discover data from semantic query
- Basics
  - The relationship of data is represented in RDF (resource description framework) triples and graphs
  - The data source with semantic attributes can be query by SPARQL (an RDF query language)
- Linked Data
  - A huge collection of semantic databases over web
  - Sensors can also be linked data, live streaming data

# An RDF graph sample




- Source: Ivan Herman W3C, Oct. 2011

# Linked Data (Sept. 2010)




Source: **Christian Bizer**
**Freie Universität**
**BerlinGermany**
**BNCOD'2011**

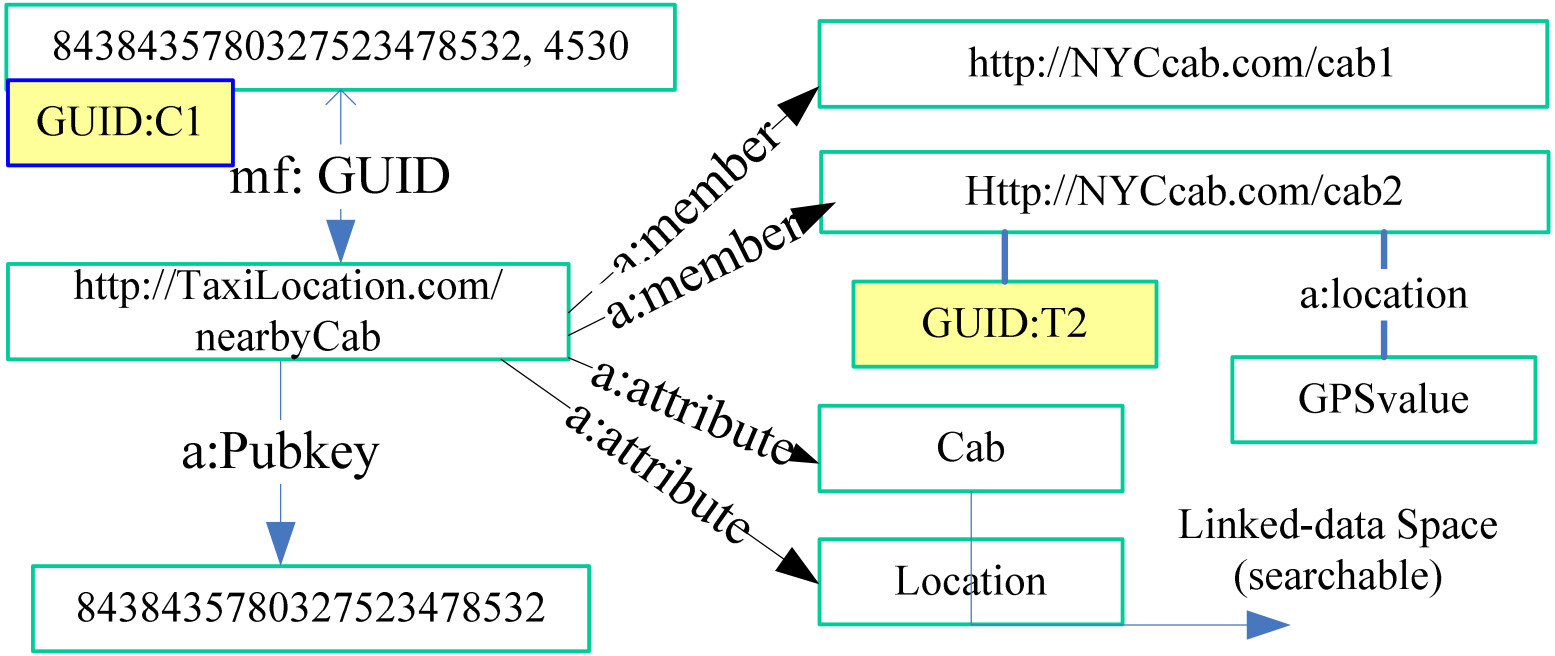
- Over 26,9 billion RDF triples

# Example: A context-aware IoT Service

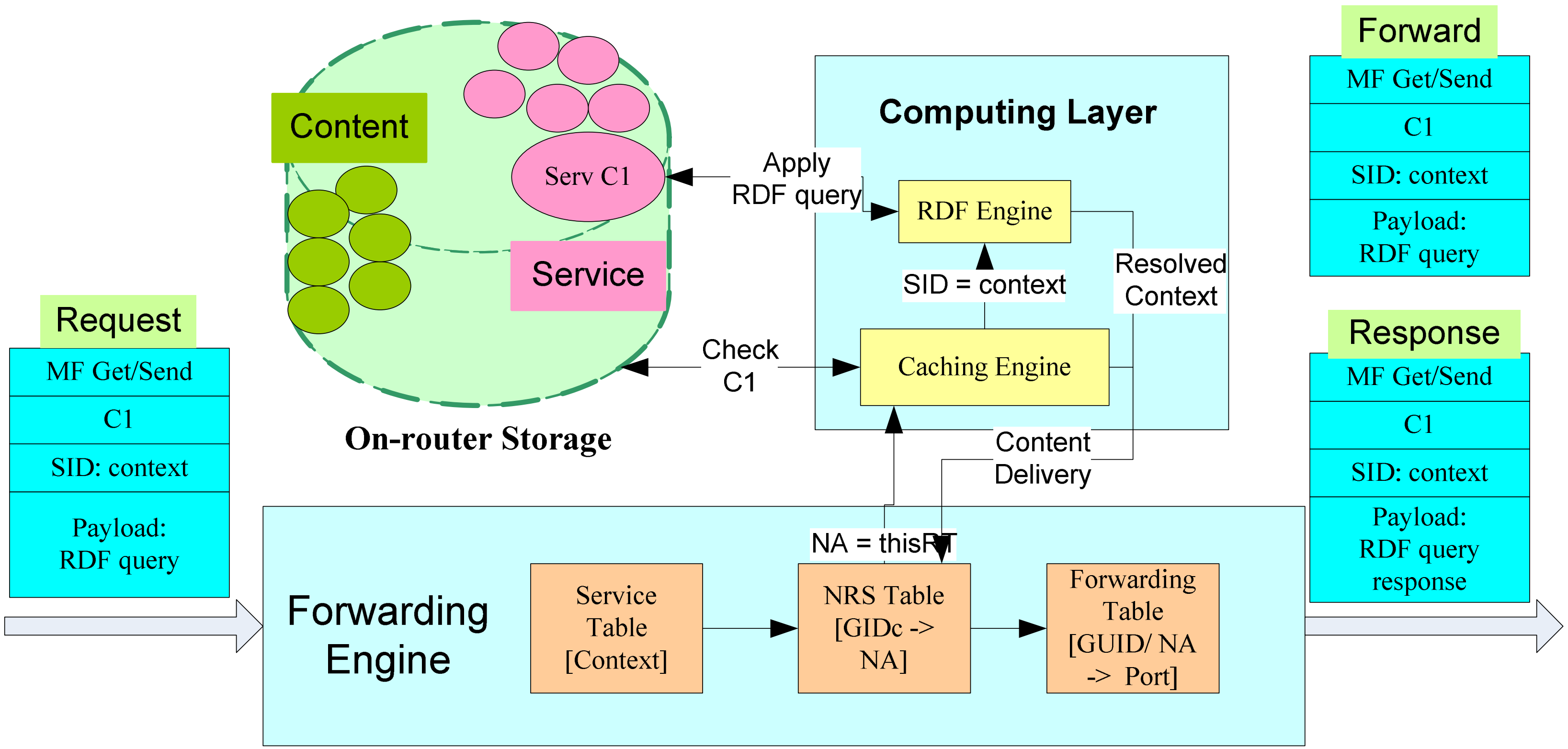
- UbiCab, defined as
  - *"James, walking on NYC streets, makes a call to a CONTEXT "Nearby Cab"* – A phone call from James is automatically routed to a nearby taxi driver.
- Things: James and cabs, connected to network through their phones
- Data: GPS locations on their phones
- Middleware: an IoT service redirect a call from James to a "nearby cab"
- Overlay server: a web service runs at Taxilocation.com
- How in-network service is enabled in MobilityFirst

# RDF Graph – as a Universal Service Description

- The IoT service is described in RDF (resource description framework) graph
- Service GUID: C1, Cab2 GUID: T2
- T2 subscribe/update to C1 are database operation over the RDF graph



# MF Router: an Edge Server for IoT Service



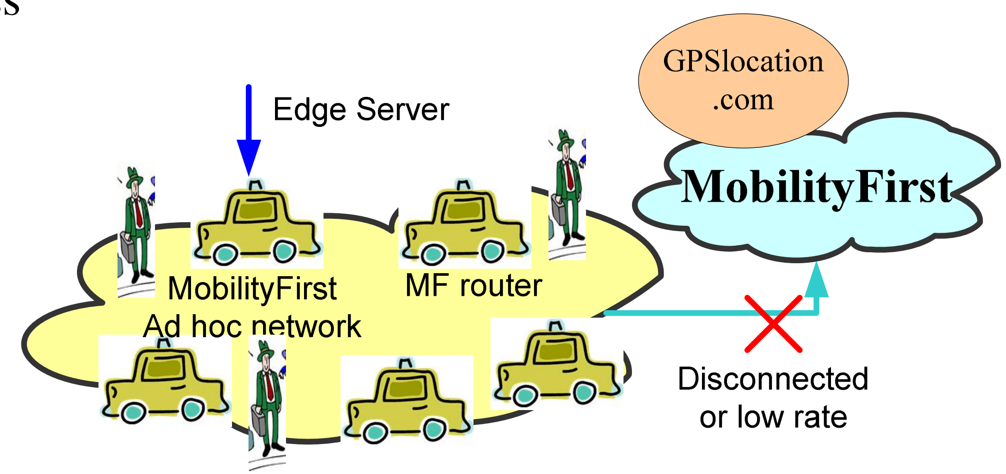
# Choice of Edge Servers

- GNRS server overloading
  - C1 maps to T2 based on dynamic computation (James Loc as input) on GNRS server for C1
  - Pros: simple, statelesss, Cons: location of GNRS not near
- Nearby MF router caching
  - James' request to C1 is computed at a nearby MF router E1 where the IoT service (RDF graph) is cached
  - Pros: location-aware, Cons: caching consistency

# Typical IoT Services

- Key features of IoT services
  - Limited processing, sensitive to delay
  - Dependent on context (time, location & more)
- In-network service distribution is more beneficial and feasible
  - Fast response, traffic load balancing based on location information
  - Light-weight process

- A V2V ad hoc net:
  - Disconnected / low rate to back haul
  - Traffic only locally significant
  - Fast response, light-weight process



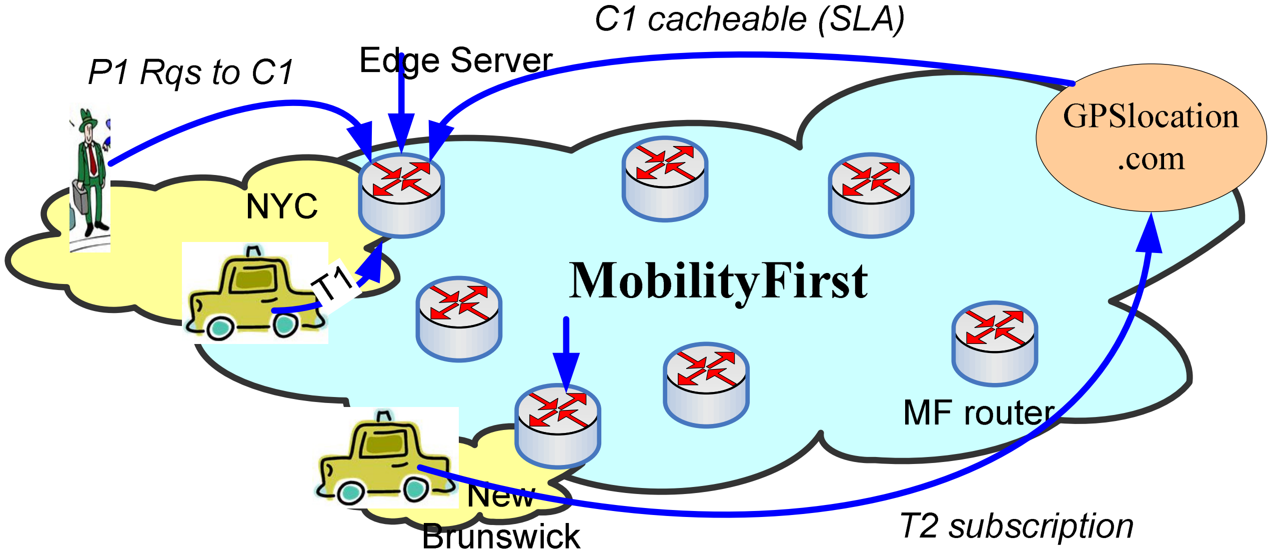

Not for apps requiring heavy duty middleware

# New Business Model: GUID based charging

- Internet, CDN and Cloud computing
  - Accounting based on access control and secure channels required
  - Authentication and Authorization via account management
- MobilityFirst – pervasive computing
  - Authentication and authorization via GUID certificate
  - Accounting based on GUID signature verification
  - Can implement charging to access GUID (flat rate), service GUID (800#) and user GUID (pay-per-view)
  - No access control and/or secure channel are needed

# Charging on GUID

- C1 agrees to pay for MobilityFirst in-network service caching
- T1, T2 agree to pay service provider of C1 at subscription
- T1, P1 requests to C1 are accounted by in-network service and charged to service provider GPSlocation.com



# Conclusions

MobilityFirst routers and protocol stack enable efficient IoT service distribution

– Universal identity (GUID) and middleware service description (RDF)
– MF routers offer in-network processing of GUID identified / RDF described IoT service
– GUID identity based business models are feasible between MF and IoT service (operators), IoT services and subscribers, IoT services and consumers.