

Named Data Networking for Data Spaces

Y. Thomas, N. Fotiou, I. Pittaras and G. Xylomenos

Mobile Multimedia Laboratory, Department of Informatics

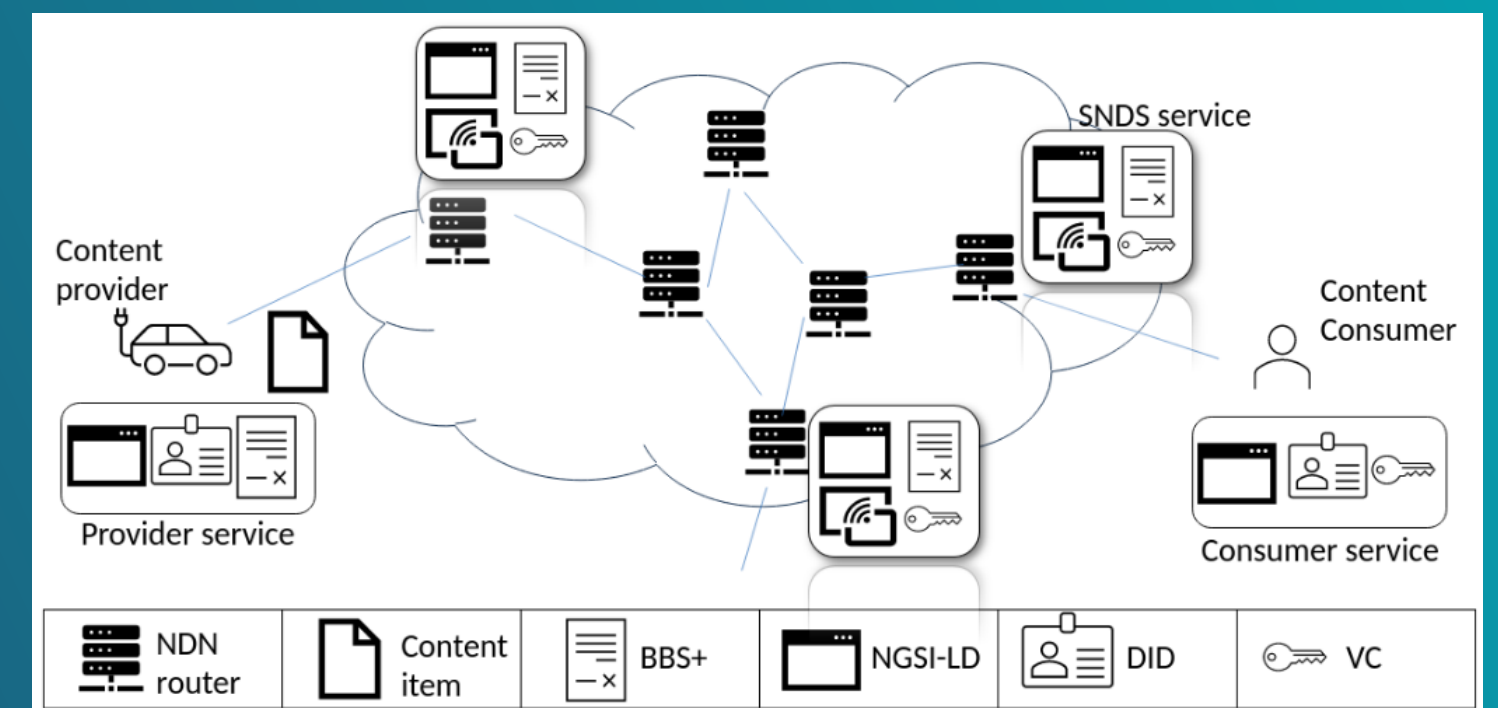
Athens University of Economics and Business, Greece

{thomasi, fotiou, pittaras, xgeorge}@aueb.gr

Motivation

- ETSI Data Spaces
 - Standard way to look at data
 - Broker between data provider and consumer
- Named Data Networking (NDN)
 - Information-centric architecture
 - Caching, native multicast, multisource provision
- Secure Named Data Sharing (SNDS)
 - Offers NGSI-LD data spaces over NDN
 - HTTP on the outside, NDN on the inside

SNDS Design

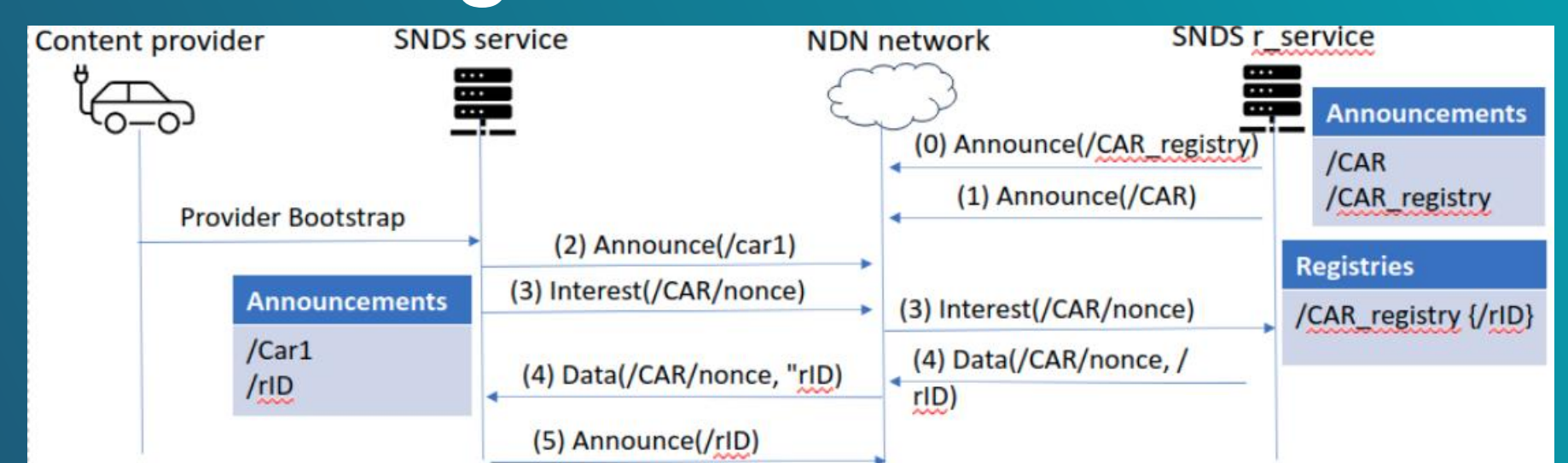


- SNDS Service
 - Receives NGSI-LD queries (ID, TYPE, Attributes)
 - Translates them to NDN Interest and Data messages

NGSI-LD to NDN

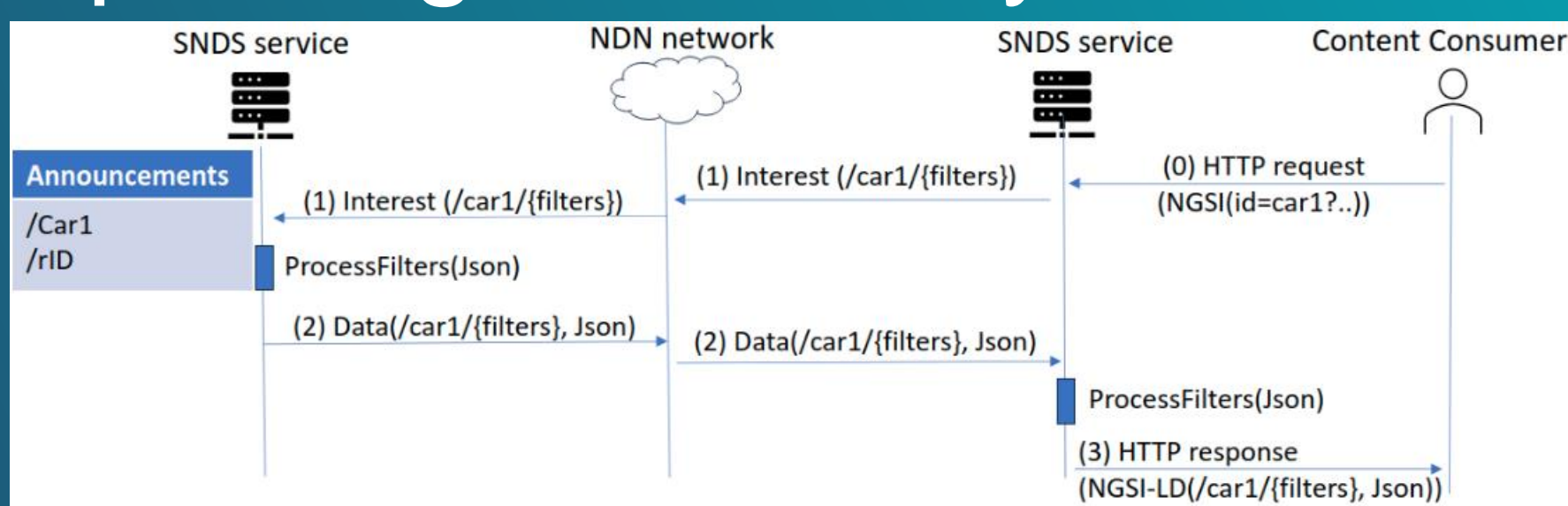
- Queries by ID are simple
 - The ID is used as the unique NDN name
- Queries by TYPE are hard!
 - They match many content items
- We propose using a Registry
 - Each SNDS node handles some types
 - Two special names per type
 - /TYPE: discovers new providers for TYPE
 - /TYPE_REGISTRY: lists all providers for TYPE

Announcing Content



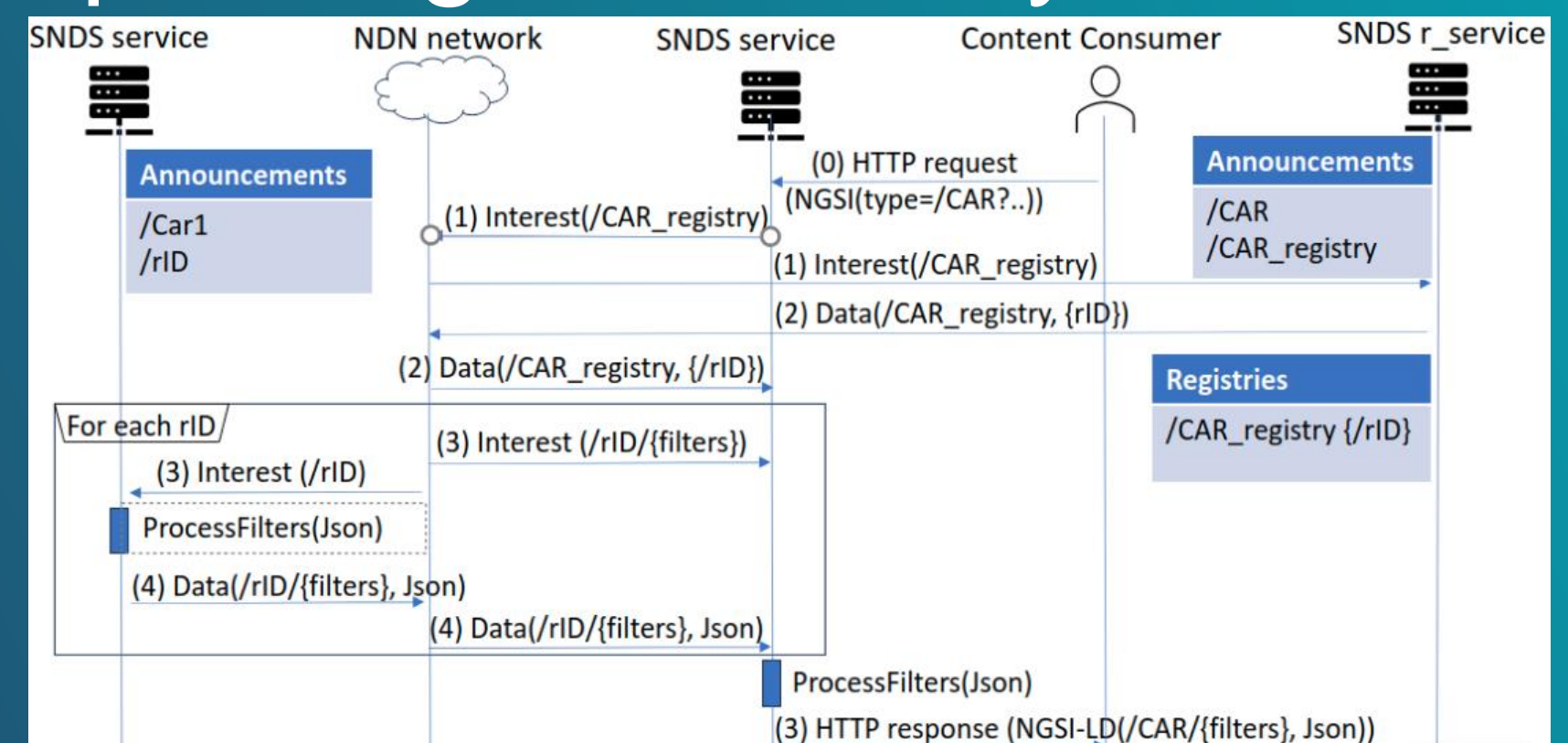
- Example: TYPE=CAR
 - Registry announces /CAR and /CAR_registry
 - Provider announces /car1, Interest for /CAR/nonce
 - Registry returns /rID and adds it to CAR_registry
 - Provider announces /rID (alias of /car1)

Requesting Content by ID



- Example: ID=car1
 - SNDS service receives NGSI-LD query inside HTTP
 - Sends Interest and gets Data for /car1
 - Returns NGSI-LD response inside HTTP

Requesting Content by TYPE



- SNDS service sends Interest for /CAR_registry
 - Then it sends Interests for each rID in CAR_registry

Implementation and Evaluation

- Prototype built on top of NDN implementation
 - HTTP proxy for NGSI-LD
 - SNDS service and registry
- Some initial results
 - Star pattern: one hub and four spokes
 - Links: 100 ms delay
 - Response latency in ms

Operation	Collocated	Separated
Announce	452	1122
Request by ID	548	585
Request by TYPE	860	1297